COMPUTER SIMULATIONS OF ELASTIC LIGHT SCATTERING

IMPLEMENTATION AND APPLICATIONS

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam,
op gezag van de Rector Magnificus
prof. dr. P.W.M. de Meijer,
ten overstaan van een door het college van dekanen ingestelde
commissie in het openbaar te verdedigen in de Aula der Universiteit
(Oude Lutherse Kerk, ingang Singel 411, hoek Spui)
op 11 februari 1994 te 12:00 uur

door

Antoon Georgius Hoekstra

geboren te Irnsum

Promotor:	prof. dr. L.O. Hertzberger
Co-promotor:	dr. P.M.A. Sloot
Fakulteit:	Wiskunde en Informatica

The research presented in this thesis was financially supported by the Netherlands Organisation for Scientific Research (NWO), via a matched funding from "FOM", "SION", and "foundation for Biophysics", grant number NWO 810-410-04 1.

I gratefully acknowledge technical support from Parsytec Computer GmbH, Aachen, Germany.

©1994 Alfons Hoekstra. All rights reserved.

Printed by CopyPrint 2000, Enschede, the Netherlands.

TABLE OF CONTENTS

VOORWOORD[1]

Hoewel het in mijn vijfde studiejaar voor mij al duidelijk was dat ik promotie-onderzoek ambieerde, kon ik toen, als student technische natuurkunde en zich specialiserend in de oppervlaktefysika, niet voorzien dat mijn onderzoek en voorliggend proefschrift zich zouden wagen op een interdisciplinair onderzoeksterrein, waar high performance computing, informatica, fysica van elastische lichtverstrooiing en experimentele biofysica elkaar treffen. Dat het toch zo gelopen is, heb ik te danken aan Peter Sloot.

Mijn eerste kontakt met Peter is tekenend voor hem en karakteristiek voor wat er sinds oktober 1987 is gebeurd. Naar aanleiding van een open sollicitatie naar een mogelijke plaats als vervangend dienstplichtige, was ik in maart 1987 te gast bij de afdeling radiotherapie van het Antoni van Leeuwenhoekhuis (het Nederlands Kanker Instituut, NKI). Daar werd ik tijdens de lunch door mijn gastheer aan Peter voorgesteld, die toen als promovendus bij het NKI was aangesteld. Voor ik het wist zat ik bij Peter op de kamer te vertellen over mijn afstudeerwerk, greep Peter zijn kontakten aan om "iets" te regelen en kon ik in oktober 1987 bij Peter beginnen. Dit was het begin van een turbulente tijd, die hopelijk nog vele jaren zal voortduren.

Na Peters promotie in november 1988, werden alle zeilen bijgezet en schreef Peter in korte tijd een hele reeks projecten. Hier komt Bob Hertzberger, Peters, en ook mijn promotor, in beeld. Hij haalde Peter naar de UvA, en gaf de voorzet om aan parallel rekenen te gaan denken (de 64-node Meiko-machine voor de UvA was toen net een feit geworden). Enfin, één project werd ingediend bij het stimuleringgebied Fysische Informatica van het FOM. Over de behandeling van deze aanvraag is een zeer spannend verhaal te schrijven (Peter, we wachten hier nog steeds op), maar uiteindelijk werd het toch gehonoreerd en kon ik in september 1990 beginnen met mijn promotie-onderzoek. Dat ik in de periode tussen de vervangende dienstplicht en de aanvang van het promotie-onderzoek niet op een houtje hoefde te bijten, is ook weer te danken aan Peter, en aan Carl Figdor, die voor de nodige financiën zorgde.

Toen volgden de "Atenlab blues", waar het fundament werd gelegd voor de huidige Parallel Wetenschappelijk Rekenen en Simulatie Groep. Hier zetten we onze eerste schreden op het glibberige pad van parallel rekenen. Vier jaar en twee verhuizingen later zitten we op de tweede verdieping van de nieuwe WCW-vleugel, hebben Peter en Bob het voorzichtige begin uitgebouwd tot een grote onderszoeksgroep, is en passant het IC^{3}A opgericht (waar toch één van de grootste parallelle computers in Europa is ondergebracht) en is binnen de fakulteit Wiskunde en Informatica van de UvA een nieuw curriculum op het gebied van parallel wetenschappelijk rekenen opgezet.

Het is fascinerend om als promovendus dit alles mee te kunnen maken. Niet alleen op wetenschappelijk gebied heb ik zeer veel van Peter geleerd, maar ook bij alles wat er tegenwoordig komt kijken om wetenschap te kunnen bedrijven. Daarnaast kijk ik, van binnen weer schaterend van het lachen, terug op de vele borrels en feesten bij het NKI ("..mag het licht uit..") en de UvA ("..in het midden..") die Peter en ik bezocht hebben. En een aantal zeer geslaagde

---

1 Preface, in Dutch

conferentiebezoeken (een opera in de Grand Canyon, spelletjes hints in een klooster aan het meer van Genève en een voetbalvandaal en bruine bieren in Praag) moeten ook genoemd worden. Woorden schieten tekort, Peter, hartstikke bedankt !

Dit proefschift kon natuurlijk niet tot stand komen zonder de (morele) hulp van velen. Ten eerste wil ik Carl Figdor en Bob Hertzberger nogmaals noemen. Carl, dank voor de steun die jij ten alle tijd hebt gegeven aan het (vervolg) onderzoek van het CACE projekt. Dit gaf mij de mogelijkheid om de onzekere tijd tussen vervangende dienstplicht en promotie-onderzoek door te komen. En Bob, mijn gewaardeerde promotor, dank voor de mogelijkheid die jij bood om onder jouw beschermende vleugels het onderzoek gestalte te geven.

Tijdens de NKI-tijd zijn we begonnen met het meten van de verstrooiingsmatrix van witte bloedcellen in een flowcytometer. In samenwerking met de groep van Bart de Grooth zijn deze experimenten voortgezet aan de Universiteit Twente. Dit had voor mij de prettige bijkomstigheid dat ik regelmatig "back to the roots" kon gaan. Bart de Grooth, Richard Doornbos, Herke-Jan Noordmans en Kirsten Deurloo, Lissajous-figuren en zigzaglijnen in scatterplots, wie had dat gedacht! Ik hoop dat we onze goede samenwerking kunnen blijven voortzetten en dat we in de toekomst $\mathbf{S}_{34}$-elementen van (getrapte) Basophilen kunnen meten, en doorrekenen.

Turners, praktizerend of in ruste, zijn interessante mensen. Tijdens een reunie van de turnvereniging "Linea Recta" in Enschede kwam ik in gesprek met Geert Streekstra. Dit mondde uit in een uitgebreide samenwerking, waarin we ons stortten op het probleem van near-forward scattering van rode bloedcellen in een Couette-flow. Resultaat: twee artikelen en een aantal conferentiebijdragen. En voor mij een aanleiding om eens met de T-matrix-methode te gaan stoeien. Het resultaat is hoofdstuk 7 van dit proefschrift. Geert, bedankt voor de goede samenwerking, de vele prettige avonden en de zeer geslaagde trip naar Japan.

Wetenschap is voor 1% inspiratie en voor 99% transpiratie: wie zei dat ook al weer? Menig transpiratieprocentje is opgebracht door Maroen de Haan en Stephan Meijns, die met veel doorzettingsvermogen de oorsponkelijke Occam-programmatuur van de parallelle CG-methode schreven. Mannen, bedankt ! Ook moeten Michiel van Muijswinkel, Joep Vesseur, Frank van der Linden en Peter Trenning worden genoemd, die de gekoppelde dipool code hebben gepoort naar I-Server-Occam, Express en PVM, en vervolgens de performance-metingen hebben uitgevoerd. Nogmaals, mannen, bedankt.

Arthur Rob en Robbert Belleman, ook door Peter Sloot van het NKI naar de UvA gehaald, bedankt voor de geweldige tijd als collega's. Herinner je je nog dat onsmakelijke eendje op het Alkmaarder meer?

Dames en heren van het secretariaat, hopelijk zijn jullie enigzins bijgekomen van de shock van PWR&S op de gang. Bedankt voor alle hulp die ik altijd van jullie heb gekregen. Alle andere kollega's van zowel het NKI als de UvA, teveel om bij naam te noemen, hartstikke bedankt voor de gezellige tijd, de produktieve samenwerking en de (wilde) borrels en feesten.

Harry Hodzelmans, jouw wil ik speciaal bedanken voor de moeite die jij hebt genomen om alle Nederlandse stukken in dit proefschrift enigzins leesbaar te krijgen. In jouw eigen woorden: "zo pik ik, na mijn eigen natuurkunde-"experiment" toch nog een resultaat mee." Bedankt voor dit resultaat!

Lieve vrienden en kennissen, sorry dat de contacten de afgelopen maanden nogal minimaal waren. Met name de SSPA-groep en het Olympia-bestuur heb ik de afgelopen tijd nauwelijks met mijn aanwezigheid kunnen vereren. Bedankt voor de interesse, peptalk en afleiding die ik altijd bij jullie kon vinden. Hopelijk kan ik met het promotiefeest weer eens wat terugdoen!

Pa en ma, bij momenten als deze realiseer ik me weer welk een stimulans jullie voor mij zijn geweest, hoeveel steun en begrip ik altijd weer bij jullie vond en hoeveel jullie voor mij betekenen. Bedankt voor alles!

Lieve, lieve Gerda, we leerden elkaar pas kennen in de laatste stukje van mijn promotie-onderzoek. Dat had veel eerder moeten gebeuren. Ik ben je ontzettend dankbaar voor alles wat je het afgelopen jaar voor mij hebt gedaan, voor je vetrouwen en je liefde.

I did it my way...

Alfons Hoekstra

Chapter 1

GENERAL INTRODUCTION

GENERAL INTRODUCTION

1.1 PREFACE

The study of Elastic Light Scattering (ELS) from human white bloodcells (leukocytes) is the guiding research theme of this thesis. This problem will be addressed from a computational and an experimental point of view. Such an interdisciplinary approach tries to combine the very different fields of experimental biophysics, physics of ELS, and computational science. The main part of the thesis consists of the development of a computer simulation of ELS and subsequent applications. In that respect the computational study of ELS serves as a very rich example application for modern computational science, where feasibility of Massively Parallel Processing (MPP) to solve "real" problems is a major research item.

Experiments, be it computational or "real", and validation of these experiments are the most important ingredients of the research presented in this thesis. The computational experiment to study Elastic Light Scattering from human white bloodcells is developed from scratch. Computational methods and techniques necessary to construct such an experiment will be discussed in some depth, with emphasis on MPP techniques. The "real" experiments build upon a decade of previous research [1], and are part of a larger collaboration to measure the scattering of leukocytes in flowcytometers. A short historical sketch will be drawn to put the experimental studies in their right perspective, and to provide essential arguments for the development of the computational experiment.

1.2 PARALLEL COMPUTING

1.2.1 General introduction; some history

The 1990's are very exciting for computer science. Old concepts, which emerged during the early developments in electronic computers, or even before that time, are now actually realised in hardware. Decrease in switching speed of digital circuits, VLSI technique allowing to integrate a micro computer on a single chip, fast memories and I/O devices, but especially the introduction of parallelism on every level of digital computing has dramatically increased computational speeds.

In only half a century the computational speed has moved up from 100 floating point operations (flop) per second (the EDSAC, summation of a large array of real numbers [2]) to 60 Gflop/s (October 1993; a 1024 node CM-5 running

the linpack benchmark [3]). If we compare this increase of almost 12 orders of magnitude with the increase in clock speeds of 2 μs on the EDSAC1 [4] to approximately 1 ns on top of the bill vector super computers, and account for the fact that the arithmetic itself is much more accurate (36 bit fixed point arithmetic compared to 64 bit floating point arithmetic), it is obvious that the largest gain in computational speed has to be attributed to architectural - and (system) software innovations.

Low level parallelism has played a major role in these innovations. We should mention bit-parallel arithmetic operations, a number of functional units for e.g. addition, multiplication, or index calculations operating in parallel, and pipelining inside functional units, allowing overlapping operations on several data items. Introduction of banked main memories and a fast cache between main memory and the arithmetic registers dramatically increased the transfer rate of data and instructions between memory and registers. Finally, the important innovation of vector registers and vector operations, introduced in commercial machines by Seymour Cray, opened the way to come close to the theoretical peak performance of the inverse of the clock cycle time (of course on well tuned problems).

To go even beyond these performance figures, the next logical step is to let more than one such powerful vector processor work on a problem in concert. Now make a clear distinction between the *inter CPU parallelism* described in the previous paragraph, which is immersed inside one complete general purpose processor unit, and parallelism which is introduced by means of a replication of complete processor units. By *Parallel Computing* we exclusively mean computing using hardware consisting of replicated processor units. These processor units may contain every level of sophisticated inter CPU parallelism, but may also consist of very simple bit-serial processors.

Probably the first mention of parallel computing is by L. F. Manebrea, in his 1842 publication "*Sketch of the Analytical Engine Invented by Charles Babbage*", where he writes (we take this quotation from reference 5, page 8): "*...Likewise, when a long series of identical computations is to be performed, such as those required for the formation of numerical tables, the machine can be brought into play so as to give several results at the same time, which will greatly abridge the whole amount of the processes...*".

A very early description of a, in modern terminology, massively parallel, distributed memory computer, operating in loosely synchronous mode is described by L. F. Richardson in 1922 (this example is taken from reference [6], the original is reference [7]). In his book *Weather Prediction by Numerical Process* we find in chapter 11.2: "*...If the time-step were 3 hours, then 32 individuals could just compute two points so as to keep pace with the weather, if we allow nothing for the very great gain in speed which is invariably noticed when a complicated operation is divided up into simpler parts, upon which individuals specialise. If the co-ordinate chequer were 200 km square in plan, there would be 3200 columns on the complete map of the globe. In the tropics the weather is often foreknown, so that we may say 2000 active columns. So that 32×2000=64,000 computers would be needed to race the weather for the whole globe. That is a staggering figure. Perhaps in some years time it may be possible to report a simplification of the process. But in any case, the organisation indicated is a central forecast-factory for the whole globe, or for portions extending to boundaries where the weather is steady, with individual*

computers specialising on the separate equations. Let us hope for their sake that they are moved from time to time to new operations....". Of course, Richardson's computers were *people*.

An example of an early, actually working computational pipeline is presented by R. Feynman in his book *Surely You're Joking Mr. Feynman*. He talks about how he spent his time in Los Alamos, during the second world war, and how his group would carry out numerical calculations (Feynman headed the so-called IBM group). They had figured out that they needed a number of IBM machines (mechanical tabulators, multipliers etc.), and they arranged them in a loop, thus 'programming' their mechanical computer. The idea was to put a card in, do a calculation, and put the resulting card in the next machine. Feynman's problem was that this was not a very fast way to work. They needed nine months to figure out three problems. From this point I will cite from Feynman's book, because he describes their trick to speed things up (see part 5, Los Alamos from below): "*But one of the secret ways we did our problems was this. The problems consisted of a bunch of cards that had to go through a cycle. First add, then multiply-and so it went through the cycle of machines in this room, slowly, as it went around and around. So we figured a way to put a different coloured set of cards through a cycle too, but out of phase. We'd do two or three problems at a time.*" This trick however presented Feynman with another problem. At some time he had to come up with the answer of a specific problem within a month; "*...So Bob Christy came down and said, "We would like the results for how this thing is going to work in one month"-or some very short time, like three weeks. I said, "It's impossible". He said, "Look, you're putting out nearly two problems a month. It takes only two weeks per problem, or three weeks per problem." I said, "I know. It really takes much longer to do the problem, but we're doing them in parallel. As they go through, it takes a long time and there's no way to make it go around faster.*" In terms of Hockney's ($n_{1/2}$, r_∞) description [5], Bob Christy was impressed by r_∞ , without knowing anything about $n_{1/2}$. By the way, Feynman's group did work out a way to do the problem in this very short time.

The first electronic digital computer, the ENIAC, contained much parallelism (e.g. 20 accumulators) and could in principle be programmed in a multi program way (different micro programs for each accumulator). However, this design, which can be described as loosely coupled MIMD (see section 1.2.3), was too ambitious for its time, and the first stored-program computers, the EDVAC, the EDSAC and the UNIVAC1, which are true von Neumann type sequential computers, were much more powerful. These sequential architectures slowly evaluated into today's vector super computers containing a small number of parallel processors with a shared memory.

During the first years of electronic computing in the 1950's, much research has also been devoted to parallel computing. Unfortunately, these ideas could not be realised into efficient machines until the 1980's. Based on a theoretical paper by von Neumann in 1952 [8], Unger proposed a practical design for a two dimensional array of processors [9]. This line of development resulted in the ILLIAC IV and the ICL DAP [see e.g. 10, 11], which were arrays of processors, each with their own memory, operating in SIMD mode (see section 1.2.3).

The idea to assemble a large number of processors, each executing their own program and working on their own data, into one parallel computer can

be traced back to a paper in 1959 by Holland [12]. This paper influenced later work by Pease, who introduced the concept of the hypercube architecture [13], which resulted in the very successful cosmic cube [14]. A detailed historical account of parallel computing can be found in reference [5], chapter 1.

The second half of the 1980's shows a true explosion of parallel computing, with the introduction of many successful massively parallel systems. Examples are the CM-2 of Thinking Machines, the Intel hypercubes (iPSC 1 and 2), the Meiko Computing Surface, and the Parsytec Super Cluster. The last two are based on the transputer, a microprocessor which is specifically designed for parallel computing. We may view these systems as the first generation parallel computers. The experience gained with these first generation systems has proven that massively parallel systems can be build and reliably operated, and that parallel programs can run very efficiently on hundreds or thousands of processors. Most of these systems however could not compete with the vector supercomputers. In the last three years some new massively parallel systems have entered the market. These second generation systems, such as the CM-5, or the Intel Paragon have comparable, or even better performance figures as e.g. the Cray C90. Furthermore, these MPP systems open the way to sustained Tflop/s performance.

We have not mentioned software yet. Without disregarding the many important breakthroughs in e.g. system software, high level programming languages, and programming environments, poor programmability of parallel systems has been, and still is, a serious drawback of parallel computing. Although the parallel programming environments of the second generation parallel systems alleviate much of the problems encountered in programming parallel computers, it is generally acknowledged that programming in parallel, and porting large sequential, or vectorised codes to parallel systems is not a trivial task. One paradigm of parallel computing, the Single Program Multiple Data paradigm of parallel computing [15], emerged as a very useful way of parallel programming for large scientific and engineering applications, and most modern parallel systems support this mode of programming.

1.2.2 Why parallel computing ?

The chase to ever increasing computational speeds is fuelled by a strong application pull. Researchers and engineers constantly need more computational power in order to carry out larger simulations (e.g. from two to three dimensions), more accurate simulations (e.g. better resolution of computational grids), or to run their simulations continually faster (e.g. weather forecasting). The goal of the Tflop/s is justified in the USA High Performance Computing and Communications program (HPCC) by identifying a large number of so-called grand challenges, strategic and highly relevant applications for the nation's well-being, which require Tflop/s performance. Examples include computer aided drug design, climate modelling, semiconductor modelling, or simulation of fluid turbulence. Comparable initiatives in Japan (the Real World Computing program) and in Europe (High Performance Computing and Networking) also identify this strong application pull to reach for the Tflop/s. All these programs presume that massively parallel computing is the only way to arrive at such performance figures.

Let us try to assess fundamental limitations on the computational speed of one single processor. As was argued in the section 1.2.1, well tuned architectures are now capable to deliver one floating point operation during every clock cycle on well tuned problems. Of course, this is a theoretical upper limit on the computational speed, but it suits our goal. Define R_{max} as the maximum computational speed, and δ the clock cycle time. We can immediately write

$$R_{max} = \delta^{-1} . \qquad [1]$$

In the past several arguments, based on the finite speed of light and Heisenberg's uncertainty principle, were proposed to find theoretical upper limits for R_{max} (see e.g. reference 16). We will now investigate if these two physical constraints really pose a theoretical upper limit to the maximum calculation speed.

According to Heisenberg the accuracy of a time measurement Δt and an energy measurement ΔE are related by $\Delta E \, \Delta t > h/2\pi$, h is Planck's constant. If we identify δ with Δt, we find $\delta > h/(2\pi \, \Delta E)$ and $R_{max} < 2\pi \, \Delta E/h$. Let the energy difference between a 0 bit and a 1 bit be equal to ΔE. Strictly speaking the uncertainty principle does not impose an upper limit to R_{max}, because ΔE can be increased. This would imply an increasing dissipation in the processor, which in turn will be the limiting factor. However, because h is very small, small values for ΔE already give a very large upper bound. For instance, if we put ΔE = 2 eV (bandgap in a semiconductor, energy of a visible photon) we already find R_{max} < 3 Pflop/s. We can conclude that the uncertainty principle does not impose any limits on attainable computational speed.

The finite speed of light however will present a true constraint on computational speed. Computation requires transportation of information from memory to the processing unit and back. Assume that for every floating point operation information must travel a total distance d. The cycle time is now limited by the finite speed of light c according to $\delta > d/c$, and consequently $R_{max} < c/d$. If we for instance assume that d equals 3 cm, which is not unreasonable, we find δ > 0.1 ns and R_{max} < 10 Gflop/s. We have to reduce d to 300 μm to have a R_{max} < 1 Tflop/s. Even if memory (e.g. a large cache) and processing unit are integrated on a single chip, such a small value of d is probably not possible. Therefore we can conclude that the finite speed of light presents an upper limit to processing speed of say 10 to 100 Gflop/s. So the grand challenge problems cannot be simulated on these “ultimate” serial computers, and parallelism must be introduced.

Despite the previous arguments, the real limitation at this moment is technology. Switching times are in the order of 1 ns, and it is very difficult to reduce this in case of CMOS technology. A possible candidate to reduce switching speeds is superconducting logic based on Josephson junctions [17]. The first attempts, made in the 1970's, were not very successful. Although IBM demonstrated a small signal processor with a cycle time of 665 ps, they terminated their efforts for a superconducting computer in 1983. In Japan, research in superconducting chips, as part of the project 'High Speed Computing Systems for Scientific and Technological Use' [18], continued during the 1980's. In 1990 a Fujitsu group demonstrated a working chip containing 23,000 Josephson junctions, capable of performing a multiplication

of two 8 bit numbers in 240 ps and addition of two 13 bit numbers in 410 ps [19].

Currently, we see a "third wave" of research interest in superconducting logic [20], based on the so-called rapid single flux quantum (RSFQ) logic [21]. Simple systems, based on RFSQ logic, with switching times of 20 ps have been demonstrated. However, to scale these systems to complete processors, and to build (superconducting) memories with fast enough access times remains a huge technological challenge [17, 20].

Finally we should mention the revolutionary concepts of complete optical computers [see e.g. 22] or true quantum computers [see e.g. 23]. It is difficult to foresee whether these ideas will result in working devices which can compete with existing technology.

We can conclude that massively parallelism is currently the only way to satisfy the computational requirements of the grand challenge applications. Of course, not everybody needs the enormous performance of top of the bill computers. Much computational research relies on powerful workstations, or uses e.g. a Cray Y-MP in single processor mode. In this segment we observe a strong technology push, where cost effectiveness plays an important role. Parallel systems, with computational speeds comparable to a Cray, can be built for a fraction of the cost of a Cray. Therefore, many institutes acquire such a "poor man's supercomputer", and in this way turn to parallel computing. Furthermore, the trend of cluster computing, i.e. turning a network of workstations into a loosely coupled parallel system, also provides a relatively easy and cheap first step into parallel computing.

1.2.3 Classification of (parallel) computers

The large number of different architectures, and organisation of processor networks and memories in parallel computers calls for a unifying taxonomy of the various systems. Such a taxonomy should not describe every detail of the architecture, as in the algebraic-style structural notation of Hockney and Jesshope [5, chapter 1.2.4], but has to provide just enough details for the users to distinguish the main characteristics of the system.

A well known, and very useful taxonomy is due to Flynn [24]. This taxonomy distinguishes computers by their processing of data. A computer can perform single or multiple instructions, which can work on single or multiple data items. The Single Instruction Single Data class (SISD) are the serial, von Neumann, computers. The Multiple Instruction Single Data class (MISD) is void. The Single Instruction Multiple Data class of computers (SIMD) issue a single stream of instructions, which operate on multiple data items. These multiple data items can be vectors in vector processors, or data items distributed among the processors of an array processor. Therefore, the SIMD class contains both the single processor vector computers (e.g. the Cray1), and the array processors which operate in single instruction mode, like the ICL DAP or the CM-2. The last class, the Multiple Instruction Multiple Data type (MIMD) contains all multicomputers. Therefore, this class contains both the multiprocessor Cray Y-MP, and massively parallel systems like the Parsytec GCel or the CM-5, but also clusters of workstations.

Flynn's taxonomy is very useful, but too broad to distinguish quite different architectures like the Cray1 and the CM-2. Another often made division is between shared memory and distributed memory computers. Raine

[25] has used this as the starting point for a taxonomy which is based on the physical location of data in the computer, and how the data is presented to the programmer in "address space". The physical memory in a (parallel) computer can be shared or distributed, and the logical address space can be shared or disjoint. The Shared Address space Shared Memory class (SASM) contains the serial von Neumann computers, but also the single - and multi vector processors like the Cray1 and the Cray Y-MP. The Disjoint Address space Shared Memory class (DASM) is void. The class of Disjoint Address space Distributed Memory computers (DADM) contains most massively parallel systems like the CM-2, the CM-5, or the Parsytec GCel. Finally, the Shared Address space Distributed Memory class (SADM), also known as Virtual Shared Memory systems, is very interesting because it combines the potential of massively parallelism with the ease of programming (virtual) shared memory multiprocessors. An example of a commercial machine in this class is the Kendall Square research KSR1.

Let us now combine Raine's and Flynn's taxonomy. Table 1 shows the results of this combination.

	SISD	SIMD	MIMD
SASM	serial	Cray1	Cray-YMP
DADM	void	CM-2 ICL-DAP	CM-5 Parsytec GCel Workstation Clusters
SADM	void	?	KSR1

Table 1: Combination of Raine's and Flynn's taxonomy, examples of different parallel computers are included.

This combination of taxonomies provides a reasonable distinction between very different kind of architectures like the KSR1, CM-5, CM-2 and Cray machines. If we would exclude the inter CPU parallelism (as defined in section 1.2.1) from the taxonomy, the Cray1 would be of SISD type, and the Flynn-Raine taxonomy gives a very consistent view of many different architectures. An additional taxonomy should be provided to describe the inter CPU parallelism of the processing elements.

1.2.4 Performance of parallel programs

1.2.4.1 The basic assumption of massively parallel computing

Parallel computers offer the potential of ever increasing computational speed, by merely increasing the number of processors in the parallel system. Let R be the computational speed of one processor (expressed in e.g. Mflop/s) for a specific computational task. The basic assumption of massively parallel computing is that a network of p processors is able to run the same problem at a computational speed which is in the order of $p \times R$. In the sequel we will investigate to what extent this basic assumption can be fulfilled.

Define T(1) as the execution time of a program on 1 processor, and T(p) the

execution time of the parallel version of the program on p processors. The speedup of the parallel program is defined as

$$S_p = T(1) / T(p) . \qquad [2]$$

In the ideal case $T(p) = T(1) / p$ and $S_p = p$. However, as will become clear in the following sections, this ideal parallel speedup will be degraded by many sources and $1 \leq S_p \leq p$. If we divide the speedup by p, we find the efficiency:

$$E_p = S_p / p . \qquad [3]$$

The efficiency measures how well a parallel program utilises the computational power which is offered by a parallel computer. The efficiency is bounded by $1/p \leq E_p \leq 1$. The "fair" speedup and efficiency is obtained if the fastest known sequential implementation is used as the reference (i.e. T(1)). However, in the sequel we will always refer to the "relative" speedup and efficiency, where T(1) = T(p=1); i.e. T(1) is the execution time of the parallel program running on 1 processor.

We will assess the maximum speedup which can be obtained on an ideal parallel system, and investigate this speedup under three different work load constraints. Secondly, speedup properties under finite resources will be investigated. Many of the following material is adapted from reference 26, chapter 3 and from reference 27.

1.2.4.2 Average parallelism

Consider a hypothetical parallel computer with an infinite number of equal processors and negligible communication latency between the processors. This parallel computer runs in MIMD mode, but because of the zero communication latency, the clocks of all processor can be synchronised. This allows us to establish how many processors are used during the execution of a parallel program at a certain time period.

Define the *Degree of Parallelism* (DOP) as the number of processors in use to execute a parallel program during a certain time period. The DOP as a function of time is called the *parallelism profile* of a program. As an example, consider the parallel addition of 16 numbers on our ideal computer. As is shown in figure 1, this is established by pair-wise additions in a tree structure. After 4 time slices the addition is completed. The corresponding parallelism profile is drawn in figure 2.

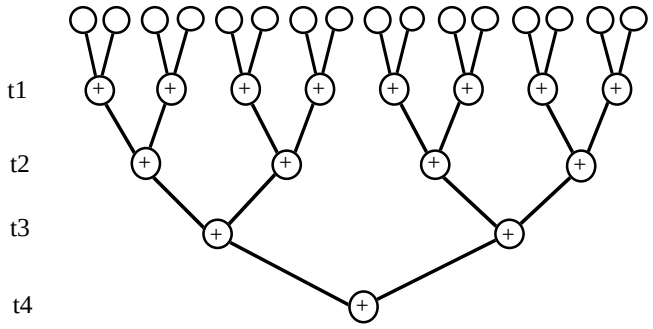


figure 1: parallel addition of 16 numbers by pair-wise addition

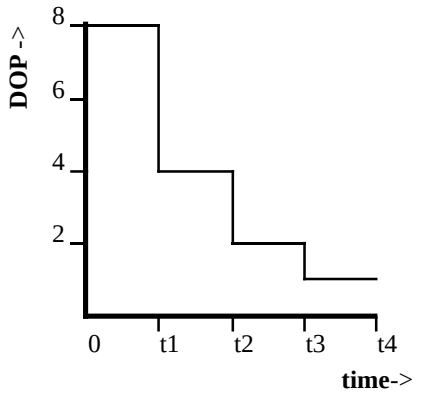


figure 2: Parallelism profile of the parallel addition of 16 numbers

Calculate the average parallelism A of a parallelism profile:

$$A = \frac{1}{t}\int_0^t DOP(t')dt' \,, \qquad [4]$$

with t the total run time of the parallel program on the ideal parallel computer. In discrete form we can write

$$A = \sum_{i=1}^{m} i \cdot t_i \Big/ \sum_{i=1}^{m} t_i \,, \qquad [5]$$

where t_i is the amount of time during which DOP = i, and m is the maximum DOP in the parallelism profile. The term $\sum_{i=1}^{m} i \cdot t_i$ can be identified with the run time of the parallel program on 1 processor. Therefore, the average parallelism A is equal to the speedup on the ideal parallel computer, and is an upper bound to the speedup which can be reached on real machines.

Now assume that a parallel program consists of a sequential part and a parallel part. If we normalise the total run time on 1 processor to 1, the sequential part is α and the parallel part is $1-\alpha$. The parallelism profile is drawn in figure 3, where we assume that the parallel part is running on p processors of the ideal machine. Off course execution of the sequential and parallel part can be interleaved, but in figure 3 both parts are lumped together to clarify the picture.

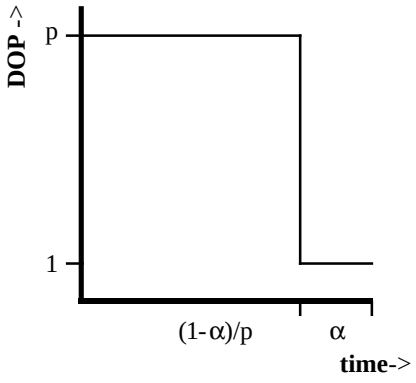


Figure 3: The parallelism profile for the fixed workload (Amdahl)

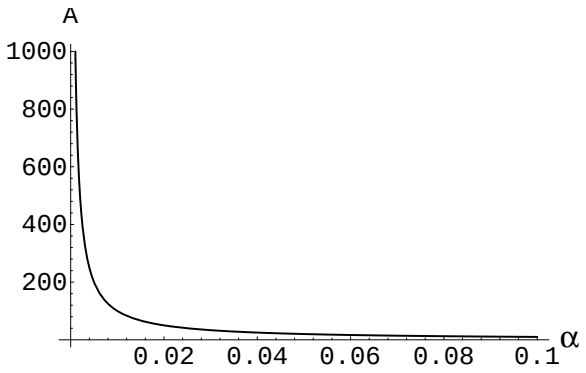


Figure 4: A_∞ as a function of α.

In this case the average parallelism is equal to

$$A = \frac{p}{1+(p-1)\alpha}\ , \qquad [6]$$

which is exactly Amdahl's law [28] applied to parallel computing. If we take the limit of p to infinity we find $A_\infty = 1/\alpha$. The maximum speedup is limited by the sequential part of the program. In figure 4 A_∞ is drawn as a function of α. Already very small values of α reduce the maximum attainable speedup to very small figures; for instance, $\alpha = 0.01$ results in $A_\infty = 100$. Amdahl's argument raised much questions regarding the validity of the basic assumption of parallel computing.

In Amdahl's law the amount of computational work is kept constant as the number of processors is increased. This fixed-load leads to Amdahl's sequential bottle-neck and prevents to reach very high speedups. Although the fixed-load constraint is essential in some application areas (e.g. real time control applications) the situation in many engineering and research applications is very different.

Gustafson [29] has formulated a fixed-time concept, which results in scaled speedup models. If the available computational power is increased, one usually increases the computational load too. This is done for instance by performing simulations on finer and/or larger grids, with more particles, or in more dimensions. Instead of trying to solve the same problem faster, one tries to solve a larger problem in approximately the same amount of time. Assume, if the problem size is increased, that the sequential part of the program remains constant and that the parallel part grows. In a first approximation this behaviour is observed in many applications. The serial part consists of I/O, initialisation, and the like, which are independent of problem size. Now assume that the problem size is increased in such a way that the total execution time of the parallel part, running on p processors, also remains constant. Figure 5 shows the parallelism profile for this fixed-time situation.

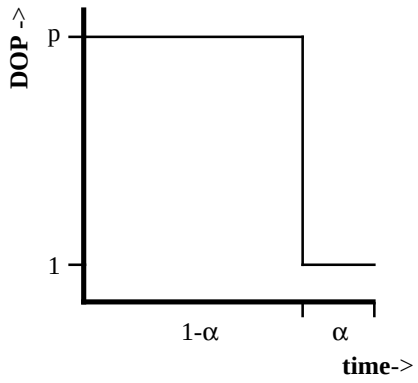


Figure 5: The parallelism profile for the fixed-time situation (Gustafson)

Figure 6: Comparison of the average parallelism for the fixed-load situation (solid line) and the fixed-time situation (dotted line) for 1024 processors.

The average parallelism in the fixed-time constraint is

$$A = p - \alpha(p - 1). \qquad [7]$$

By proportional scaling of the parallel workload with the number of processors, the speedup does not suffer from the sequential bottle-neck, and large speedups are possible. In figure 6 the average parallelism for the fixed-load constraint and the fixed-time constraint are drawn for p = 1024. We can conclude that in the fixed-time constraint situation large speedups are possible. Gustafson [29] has reported measurements of scaled speedups of 1016 to 1021 on three applications actually running on a 1024 node hypercube architecture.

1.2.4.3 Finite resources, memory bounded speedup

In this section we will derive expressions for the speedup if the number of processors is limited, especially the situation in which $p < m$ (m is the maximum DOP of a program) will be examined. The results of Amdahl and Gustafson will be generalised, and a third scaled speedup model will be discussed; the memory-bounded speedup model of Sun and Ni [27].

Define Δ as the computing capacity of a processor (expressed in e.g. Mflop/s). The amount of work executed while running a part of the program with DOP = i is $W_i = \Delta\, i\, t_i$, and the total amount of work is

$$W = \sum_{i=1}^{m} W_i \ .$$

If W_i is executed on p processors, the execution time of W_i is

$$t_i(p) = \frac{W_i}{i\Delta}\left\lceil \frac{i}{p} \right\rceil , \qquad [8]$$

with $\lceil x \rceil$ the ceiling function of x. The workloads with $i < p$ model the load imbalance in the parallel program.

The total execution time T(p) equals

$$T(p) = \sum_{i=1}^{m} t_i(p) = \sum_{i=1}^{m} \frac{W_i}{i\Delta} \left\lceil \frac{i}{p} \right\rceil . \qquad [9]$$

We can now formulate the generalisation of Amdahl's law, the fixed-load speedup factor, as

$$S_p = \frac{T(1)}{T(p)} = \frac{\sum_{i=1}^{m} W_i}{\sum_{i=1}^{m} \frac{W_i}{i} \left\lceil \frac{i}{p} \right\rceil} . \qquad [10]$$

Here we have ignored communication latencies and other overheads. If Q(W, p) is the total elapsed time which is due to overheads, and put Q(W, 1) = 0, the fixed-load speedup becomes

$$S_p = \frac{T(1)}{T(p) + Q(p)} = \frac{\sum_{i=1}^{m} W_i}{\sum_{i=1}^{m} \frac{W_i}{i} \left\lceil \frac{i}{p} \right\rceil + Q(W, p)} . \qquad [11]$$

If $p > m$ and $Q(W, p) = 0$, we find that $S_p = A$ (see equation 5). In general $S_p \leq A$.

In the sequel we will restrict ourselves to the situation where $W_i = 0$ if $i \neq 1$ and $i \neq p$ and $Q(W, p) = 0$. If we insert this in equation 11 we reproduce Amdahl's law (equation 6):

$$S_p = \frac{W_1 + W_p}{W_1 + W_p / p} .$$

We will now generalise the scaled speedup models, by considering the only constraint in scaling the workload: available memory. The memory of each node of a real parallel computer is limited. Therefore, scaled speedup models have to consider memory limitations. Sun and Ni [27] proposed a memory-bounded speedup model. Their idea is to scale the problem to its maximum amount, thus fully utilising both memory capacities and computational power of a parallel computer.

The computational work of the problem on a single node is $W_1 + W_p$, and for the scaled problem $W_1^* + W_p^*$. Using equations 9 and 10 for the scaled workload, the memory bounded speedup is

$$S_p^* = \frac{W_1^* + W_p^*}{W_1^* + W_p^* / p + Q(W^*, p)} . \qquad [12]$$

The workload of the sequential part is assumed to be independent of both problem size and system size:

$$W_1 = W_1^* .$$

The scaled parallel workload needs some more consideration. The workload W and the memory requirement m for this workload are related by W = g(m). If we assume that g(x) is a semihomomorphism[2] and that the total memory capacity M of one processor is available for the workload W_p, we can show that on the ideal parallel computer

$$W_p^* = \bar{g}(p) W_p . \qquad [13]$$

On the single processor, $W_p = g(M)$. On p processors the memory increases with a factor p to pM. On the ideal parallel system, without communication latencies, remote data access is equivalent to local data access and therefore replication of data is not necessary. In fact, for any Shared Address space system replication of data is not strictly necessary. The total available memory can now be used for the parallel portion of the scaled problem, i.e.

$$W_p^* = g(pM) = \bar{g}(p) W_p .$$

The function $\bar{g}(p)$ describes the increase in parallel workload after increasing the total amount of memory in the system with a factor p. The resulting memory-bounded scaled speedup is

$$S_p^* = \frac{W_1 + \bar{g}(p) W_p}{W_1 + \bar{g}(p) W_p / p} . \qquad [14]$$

Let us investigate three special cases.

- 1 $\bar{g}(p) = 1$. This corresponds to the fixed-problem size and equation 14 reduces to Amdahl's law.
- 2 $\bar{g}(p) = p$. The workload increases linearly with the available memory, keeping the total execution time fixed. This corresponds to Gustafson's law.
- 3 $\bar{g}(p) > p$. Here, the workload increases faster than the memory requirements of the parallel program, and the resulting speedup is larger than the fixed-time scaled speedup.

In order to find expressions for $\bar{g}(p)$ we usually have to perform an order of magnitude analysis, where we only keep the highest order terms. We will investigate three examples. The first is a parallel matrix matrix product,

---

2 A function g(x) is a semihomomorphism if $g(cx) = \bar{g}(c)g(x)$. For instance, the function $g(x) = ax^b$ is a semihomomorphism with $\bar{g}(x) = x^b$.

which has been the subject of detailed research by Sun and Ni [27]. Secondly we will investigate a parallel matrix vector product, where in one case the matrix is kept in main memory, and in the other case the matrix elements are not kept in memory, but are calculated as they are needed. This last version of the matrix vector product is actually implemented in the parallel Coupled Dipole method, as described in chapters 4 and 5 of this thesis.

The matrix matrix product **A B** = **C** requires to store three n×n matrices, therefore the memory requirement is $M = 3n^2$. The total work (assuming that it can be done in parallel) is $W_p = n^2(2n - 1) \sim 2n^3$ for large n. Therefore $W_p = 3^{-3/2} \times 2\, M^{3/2}$, and we immediately find $\bar{g}(x) = x^{3/2}$ and

$$W_p^* = p^{3/2} W_p \ . \qquad [15]$$

This result is easily derived by putting $M^* = pM$. From this we find for n^*, which is the size of the scaled matrix, $n^* = p^{1/2}n$. Therefore,

$$W_p^* = 3(n^*)^2 = 2\, p^{3/2} n^3 = p^{3/2} W_p \ .$$

This is an example where $\bar{g}(p) > p$, and the scaled speedup is even better than for fixed-time speedup.

Secondly consider the parallel matrix vector product with the n×n matrix stored in main memory. The memory requirement is $M = n^2 + 2n$. The first term is the memory of the matrix, the second term is for the argument and result vector. The work is $W_p = n(2n - 1)$. If we assume that n is very large, $W_p = 2M$, and $\bar{g}(x) = x$. Therefore

$$W_p^* = pW_p \ ; \qquad [16]$$

memory bounded speedup and fixed time speedup are equivalent in this case. Again, this result is easily derived by putting $M^* = pM$.

Finally, consider the case of the parallel matrix vector product, where the matrix is not kept in memory. This is a very relevant example for the parallel Coupled Dipole method, which is described in chapters 4 and 5 of this thesis. In actual production runs we do not keep the matrix in core. Now assume that we can calculate the matrix elements, and that the amount of work to calculate one element equals e. In that case, the total amount of parallel work in the matrix vector product equals $W_p = n(2n - 1) + en^2$. Now we only have to store the argument and result vector, and therefore $M = 2n$. Assuming large n we find

$$W_p = \frac{2 + e}{4} M^2 \ ,$$

and $\bar{g}(x) = x^2$. In this special case the memory bounded work increases as the square of the number of processors,

$$W_p^* = p^2 W_p \ . \qquad [17]$$

We will postpone numerical calculations of the resulting scaled speedups until the next chapter, where the influence of data replication is investigated.

1.2.4.4 data replication and memory efficiency

We will now focus our attention to Distributed Memory architectures. Here, as was pointed out by Sun and Ni [27], data in parallel programs usually has to be replicated. This is due to the fact that in many parallel calculations some data items are needed in all processors. Replication of this data in memory of all processors is more efficient than to keep it stored in memory of 1 processor and communicate it to other processors. However, due to this replication of data, the relation between the scaled and original workload in the memory bounded speedup model, as expressed in equation 13, no longer holds.

Sun and Ni circumvented this problem by defining

$$G(p) = \frac{W_p^*}{W_p} . \qquad [18]$$

With this definition the memory bounded speedup is

$$S_p^* = \frac{W_1 + G(p)W_p}{W_1 + G(p)W_p/p + Q(W^*, p)} , \qquad [19]$$

where we also included the overhead function Q. We will introduce the concept of memory efficiency of a parallel program, and will find a straightforward expression for the scaling function G(p). With this we will analyse the three examples of the previous section, and finally show some numerical results for speedup in the three speedup models.

Define m(p) as the memory requirement per processor of a parallel program running on p processors. Equivalent to the definition of the efficiency of a parallel program (equation 3) we can define the memory efficiency ε_m of a parallel program as

$$\varepsilon_m = \frac{m(1)}{pm(p)} . \qquad [20]$$

The memory efficiency is a measure of the scalability of a parallel program in terms of usage of the distributed memory of a parallel computer. In Shared Address space computers $\varepsilon_m = 1$. However, in Disjoint Address space computers the memory efficiency can take any value between 1/p and 1.

Let M be the total memory requirement of a program on 1 processor, and m_R is the size of the memory per node. If this program is to be executed on p processors, the following expressions must hold:

$$M < \varepsilon_m \, p \, m_R . \qquad [21]$$

This implies that the total workload of a parallel program is limited by

$$W < g(M) = \bar{g}(\varepsilon_m\, p)\, w_{max}, \qquad [22]$$

with w_{max} the maximum attainable workload of a program running on 1 processor. This limit on the workload of a program gives an upper bound to the scaling of workloads in scaled speedup models. In fact, we will show that the function G(p), as defined by equation 18, equals $\bar{g}(\varepsilon_m\, p)$.

The memory requirement per node of the scaled workload in the memory bounded speedup model equals the memory requirement of the original workload:

$$m^*(p) = M.$$

Furthermore, the total memory requirement of the scaled workload equals the memory requirement per node of the scaled workload for p = 1:

$$m^*(1) = M^*.$$

If we substitute these two relations in the definition of the memory efficiency (equation 20), we find

$$M^* = \varepsilon_m^*\, pM\ . \qquad [23]$$

A memory efficiency of 1 reproduces the scaling law for the Shared Address space case. Using equations 18 and 23 we can now derive an expression for G(p):

$$G(p) = \frac{W_p^*}{W_p} = \frac{g(M^*)}{g(M)} = \bar{g}(\varepsilon_m^*\, p)\ , \qquad [24]$$

and a final resulting expression for the memory bounded speedup

$$S_p^* = \frac{W_1 + \bar{g}(\varepsilon_m^*\, p)\, W_p}{W_1 + \bar{g}(\varepsilon_m^*\, p)\, W_p / p + Q(\bar{g}(\varepsilon_m^*\, p)\, W, p)}\ . \qquad [25]$$

Note that this memory bounded speedup is not necessarily the optimal scaled speedup, because the overhead function Q depends strongly on the details of the parallel program and the underlying parallel hardware. However, if we neglect the overhead Q, but still include the effects of data replication via ε_m, equation 25 results in a realistic upper bound of speedup. As will become clear, in many situations this upper bound is between the fixed-load upper bound of Amdahl and the fixed-time upper bound of Gustafson.

Let us now consider once more the examples of the matrix matrix product and the matrix vector product. In order to find expressions for the memory efficiency, we have to specify how the parallel workload W_p is actually carried out. Figure 7 shows how the example problems can be executed in parallel.

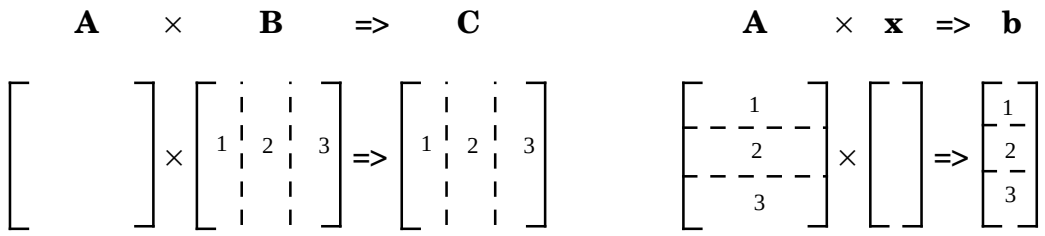


Figure 7: The parallel matrix matrix product shown left, and the parallel matrix vector product shown right.

Matrix **A** in the matrix matrix product is replicated in every processor, matrix **B** and **C** are column-block decomposed, such that each processor has n/p columns of **B** and **C** in memory. Obviously this is not the most optimal way to perform a matrix matrix product in parallel (for this see e.g. [30]) but it is a good illustration of the memory efficiency concept, and Sun and Ni also consider this example to find an expression for the function G(p).

The parallel matrix vector product is carried out by a row-block decomposition of the matrix. The argument vector has to be replicated completely in memory of each processor. This parallel matrix vector product is the kernel of the parallel Coupled Dipole method (see chapter 4 and 5).

As was shown in the previous section, for the matrix matrix product we have $\underline{g}(x) = x^{3/2}$. From figure 7 we can derive that for this parallelization of the matrix matrix product the memory requirement per node is

$$m(p) = n^2 + 2n^2 / p,$$

and therefore the memory efficiency equals

$$\varepsilon_m = \frac{3}{2+p}.$$

We can now immediately find the expression for G(p):

$$G(p) = \left(\frac{3p}{2+p} \right)^{3/2}, \qquad [26]$$

which is equal to the result of Sun and Ni [27]. For large p, $G(p) = 3^{3/2}$, which is larger than the fixed-load speedup (G = 1), but much smaller than the fixed-time speedup (G = p). Due to data replication the memory capacity requirements increase much faster than the computational requirements.

The memory requirement of the matrix vector product, with the matrix stored in memory, is

$$m(p) = n^2 / p + n + n / p,$$

and the memory efficiency is

$$\varepsilon_m = \left(1 + \frac{p-1}{n+2}\right)^{-1} .$$

If we assume that n >> p, we find $\varepsilon_m = 1$ and $\bar{g}(x) = x$ (see previous section). Therefore, for the in-core matrix vector product G(p) equals

$$G(p) = p, \qquad [27]$$

which is exactly the fixed time case.

Finally consider the matrix vector product without storing the matrix. In this case the memory requirement per node is

$$m(p) = n + n / p,$$

and the memory efficiency is

$$\varepsilon_m = \frac{2}{1+p} .$$

Remembering that in this case $\bar{g}(x) = x^2$, the memory bounded scaling function for the out-of-core matrix vector product is

$$G(p) = \left(\frac{2p}{1+p}\right)^2 . \qquad [28]$$

Again, due to the replication of data, the memory capacity requirements grow faster than the computational requirements, and G(p) ~ 4 for large p. Memory bounded speedup will be slightly better than fixed-load speedup, but will not come close to the fixed-time speedup of Gustafson.

Figures 8 and 9 show the resulting scaled speedup for the out-of-core matrix vector product as a function of α (the sequential fraction of the program, see section 1.2.4.2) for p = 512, and as a function of p for $\alpha = 0.01$ respectively. The speedup was calculated for the fixed-load case, the fixed-time case and the memory-bounded case, with - and without data replication. Note that in figure 9 the memory bounded speedup without data replication is indistinguishable from the fixed time case, and therefore not drawn.

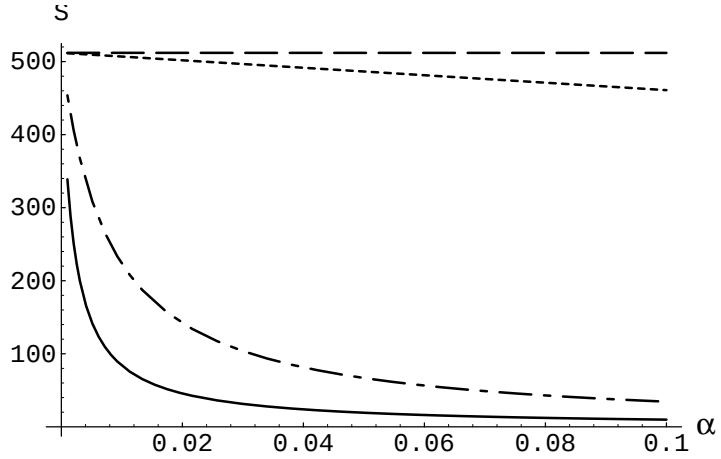


Figure 8: Scaled speedup S of the out-of-core matrix vector product as a function of the sequential portion of the program α, for 512 processors; the solid line is the fixed-load case (Amdahl), the dotted line is the fixed-time case (Gustafson), the dashed line is the memory bounded speedup without data replication, and the dashed dotted line is the memory bounded speedup with data replication.

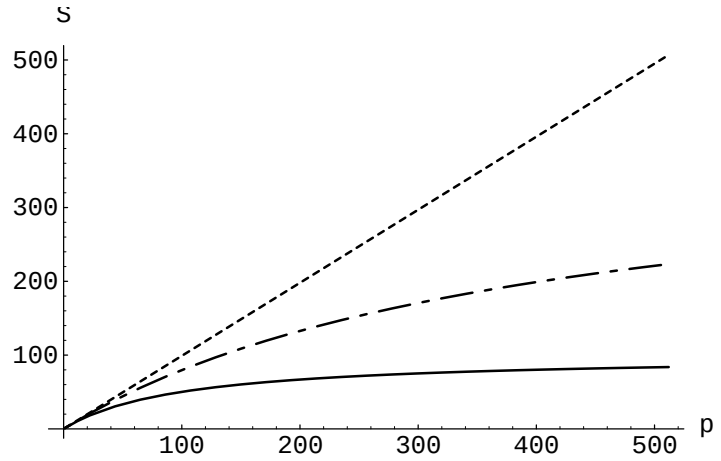


Figure 9: Scaled speedup S of the out-of-core matrix vector product as a function of the number of processors, for $\alpha = 0.01$; the solid line is the fixed-load case (Amdahl), the dotted line is the fixed-time case (Gustafson), and the dashed dotted line is the memory bounded speedup with data replication. The memory bounded speedup without data replication is indistinguishable from the fixed time case, and therefore not drawn.

From figures 8 and 9 we can draw two important conclusions. Scaled speedup models are *the* solution to Amdahl's sequential bottle-neck, and memory bounded speedup models with $\varepsilon_m = 1$ will give almost ideal upper bounds to speedup. However, if the implementation of the parallel program is such that the memory efficiency is inversely proportional to p, e.g. due to the large data buffers in the out-of-core matrix vector product, memory bounded speedup will only result in a modest improvement compared to Amdahl's law.

1.2.4.5 How can scaled speedups be interpreted ?

The scaled speedup concept was an argument against Amdahl's sequential bottle-neck. Amdahl's law applied to parallel computing is straightforward: if a program contains just 0.1% sequential code, the speedup

is limited to 1000. Therefore, many researchers did not believe that speedups of larger than 1000 could ever be reached, because "typical" programs will contain much more sequential code.

Gustafson introduced the idea of scaled speedup. He showed that by increasing the parallel workload, which will happen in the common practice of parallel computing, large speedups are definitely possible for typical values of α (see figure 6). However, we should realise that, by using the scaling law, Gustafson is actually solving *another*, larger problem with a smaller sequential part! Gustafson measured the execution time on p processors and estimates the execution time of a "hypothetical" run on 1 processor [29]. Subsequently he describes this situation using the original problem (equation 7). This formulation gives rise to conceptual problems.

The curves in e.g. figure 9, where S_p is plotted as a function of p, for a fixed α of the original problem, are in fact speedups for different sized problems (except the curve for Amdahl's fixed-load situation), and every point on a scaled speedup curve also represents a different sized problem. Actually we should draw a third axes in these figures representing the total workload W. The Amdahl curve will be in a (S_p, p) plane with a constant workload. The scaled speedup curves will all intersect with the Amdahl curve for p = 1, and according to their scaling law, will lie somewhere on the $S_p(p,W)$ plane.

Let us now revert the argumentation. Suppose we have a large problem running on a parallel computer. Which scaling law should we use to find an expression for the speedup. This question is a non-issue. We can use any scaling law to end up with the large parallel problem, but now the starting points of all scaling laws will be different. The problems will all have different values for the sequential part α. Actually we should not use any scaling law at all, and try to estimate the run time on 1 processor and from this calculate the speedup. In fact we are back in the fixed-load (Amdahl) regime.

The bottom line has to be that for many large scientific and engineering applications α can actually be very small and although Amdahl's bottle-neck still exists, it gives upper-bounds which are larger than the number of processors present in typical massively parallel systems. We may wonder if those who used Amdahl's argument have actually estimated the size of the sequential part of very large research and engineering applications, which are the typical programs running on (massively parallel) supercomputers. The important conclusion of the scaled speedup laws is that if the number of processors is increased one should also increase the problem size to stay ahead of Amdahl's sequential bottle-neck. The several scaling models, which were discussed in section 1.2.4.3 and 1.2.4.4 merely show if - and how one can scale the workload W_p.

1.2.5 Concluding remarks

Parallel computing will become the computing paradigm of the next decade. The first generation of parallel systems has proven the feasibility of parallel computing. The original scepticism toward the basic assumption of massively parallel computing (section 1.2.4.1), based on Amdahl's sequential bottle-neck (section 1.2.4.2), vanished after the introduction of scaled speedup models and actual measurements of very large scaled speedups.

The basic notion is that the parallel workload has to be kept large enough

compared to the sequential work, but also compared to the communication overheads Q(W,p). The efficiency in the fixed-load speedup model is (see equation 3 and 11)

$$E_p = \left(1 + \frac{(p-1)W_1}{W_1 + W_p} + \frac{pQ}{W_1 + W_p} \right)^{-1}, \qquad [29]$$

where we assumed that the load imbalance terms (W_i for $i \neq 1$ and $i \neq p$) are not present.

As was shown in the section 1.2.4 and 1.2.5, scaling of the parallel workload can remove the sequential bottle-neck (the second term in equation 29) and in many cases will be negligible compared to the third term. This third term in equation 29 is the quotient of p times the total time spent in overhead (especially communication) and the total workload. The inverse of this quotient is sometimes referred to as *grainsize* of the parallel program. The grainsize of a parallel program therefore is the amount of useful work that can be performed per processor per second of overhead time. If the grainsize, defined in this way, can be kept large enough as a function of p, the efficiency can remain at a (constant) value close to 1, and the parallel program has a good scalability.

In order to analyse the scalability of a parallel program one has to find an expression for Q. The overhead function will depend strongly on the method of parallelization, but also on the parallel hardware, system software and programming language. A thorough scalability analysis therefore has to take all these variables into account. In chapter 4 of this thesis an example of such a scalability analysis is presented.

Experience has learned that in many real scientific and engineering applications very efficient parallel programs with good scalability properties can be developed and implemented on real parallel computers [see e.g. reference 31 and the research presented in this thesis). However, the same experience has also shown that programmability of parallel computers, portability of parallel programs, parallelization of large sequential (commercial) production codes, and standardisation of parallel programming paradigms has by no means reached the stage of maturity that is required to change the status of parallel computing from an experimental research tool to a standard production tool. Future research in parallel computing will therefore have to concentrate on these items.

1.3 ELASTIC LIGHT SCATTERING

1.3.1 General

Elastic Light Scattering phenomena have always intrigued mankind. The blue of the sky, the rainbow, or ice-crystal halos are well-known examples of light scattering in the open air. Maybe less well-known, but as characteristic as

the light scattering perceptions visible to the naked eye, are the polarisation properties of the scattered light. Since Maxwell formulated his famous equations in 1864 [32], a firm understanding, based on electromagnetic theory, of many of these Elastic Light Scattering (ELS) observations has been established. Lord Rayleigh explained, in 1871, the blue of the sky as scattering of sunlight from small molecules in the atmosphere, by treating the molecules as radiating dipoles [33]. The exact solution of scattering of light by a sphere, as formulated by Lorenz in 1890 [34], and independent by Mie in 1908 [35], provides the theoretical framework to understand the rainbow in terms of light scattering of sunlight by water droplets (for recent work see [36]).

Besides its natural, day-to-day occurrence, researchers and engineers are utilising ELS under controlled laboratory conditions as a tool to assess crucial information from their systems under study (for many examples, see e.g. [37]). The development of sensitive, reliable and easy-to-use detectors (mostly semiconductor devices), intense mono-chromatic light sources (lasers), completely automated data-acquisition equipment, and sophisticated optics offer unprecedented possibilities to measure, in an ever increasing detail, the complete scattering matrix of particle suspensions, and even of single particles. A striking example is offered by the rapid developments in flowcytometry [see e.g. 1].

The development of the proficiencies to study ELS experimentally are paralleled by a continuing deepening of theoretical insights. Exact solutions of the Maxwell equations are formulated for the single scattering of light from spheres [34, 35], concentric spheres [38], multi-layered spheres [39]; ellipsoids [40], and infinitely long cylinders [41]. Furthermore, many approximate theories, valid for limiting values of particle size and/or refractive index, have been developed [see e.g. 37 or 42]. We have, for instance, successfully applied the Rayleigh-Debey-Gans theory to describe ELS from osmotically active human lymphocytes [43, 44], and anomalous diffraction theory to calculate the near-forward scattering from deformed, oriented, human red bloodcells [45, 46].

The last decade has shown an increasing interest in ELS from arbitrary shaped particles [47]. Both experiments and theoretical predictions demonstrate that even small deviations from the “simple” particle forms, as mentioned in the previous paragraph, have pronounced effects on the scattering matrix (see e.g. [48, 49, 50]). Today's powerful computers allow modelling and simulation of ELS from very complicated particles, such as fluffy particles (e.g. interplanetary dust, see [51]).

The information contained in the polarisation properties of the scattered light has received more and more attention as well. Hunt et al. [52] developed a fast and accurate modulation technique to measure the complete scattering matrix $\mathbf{S}$ of a suspension of particles. This principle is used by many research groups [see e.g. 53]. Sloot et al. demonstrated the possibility to measure the scattering matrix of particles in a flowcytometer, using suitable combinations of polarizers and analyzers [54, 55]. The extreme sensitivity of particularly the $\mathbf{S}_{34}$ element on small morphological changes was shown dramatically by Bickel et al. in their experiments of ELS from several types of Bacteria [56]. The sign of the $\mathbf{S}_{14}$ element of helical particles reflects the handedness of these particles, as was shown by Bustamante et al. [57] in their analysis of Circular Intensity Differential Scattering from DNA molecules.

Our ELS research reflects these modern developments. Computational

studies shift from approximate theories applied to symmetrical models [43] to simulations of ELS from realistic particles [58, 59]. On the experimental level we started efforts to measure the complete scattering matrix of white bloodcells [54], thus utilising all the information present in the polarisation properties of the scattered light. The sequel of this section will introduce the basics of ELS theory, present a survey of ELS by human white bloodcells, emphasising our previous efforts, and introduce the Coupled Dipole model which we use to simulate ELS from arbitrary particles.

1.3.2 Elastic Light Scattering from small particles

1.3.2.1 Introduction

Consider a particle in an external electromagnetic field. The applied field induces an internal field in the particle and a field scattered from the particle. The intensity of the scattered field in the full solid angle around the particle can be measured. This ELS pattern can be viewed as a fingerprint of the particle and is extremely sensitive to morphological and optical properties of the particle [37]. Therefore it is possible to distinguish different particles by means of ELS. This non-destructive remote sensing of particles is a very important application of ELS. The question arises whether it is possible to fully describe a particle solely on the basis of its complete ELS pattern (the inverse scattering problem). In principle this is impossible without knowledge of the internal field in the particle [60]. Fortunately, in most applications we have knowledge of the particles in question and in that case it is usually possible to solve the inverse problem within a desired accuracy (see e.g. reference [61]). In many cases measurement of just a very small part of the ELS pattern suffices to be able to identify particles. A good example is offered by flowcytometry analysis of human white bloodcells, where measurement of forward - and sideward scattered light suffices to identify Lymphocytes, Monocytes, and Granulocytes [62]. Another, more recent example is offered by Massoli et al. [63], who determine the radius, refractive index, and associated temperature of transparent droplets by measuring horizontally polarised cross sections in the forward and rainbow regions.

1.3.2.2 Basic theory of ELS

We will introduce some basic definitions and notations to describe ELS. The full details can be found in text books of e.g. Bohren and Huffman [37] or van de Hulst [42].

Figure 10 gives the basic scattering geometry. A particle is situated in the origin of a Cartesian coordinate system, and is illuminated by an incident beam travelling in the positive z-direction. A detector at $\mathbf{r}$ measures the intensity of the scattered light. The distance $|\mathbf{r}|$ is very large compared to the size of the particle. The far field scattered intensity is measured. The far field is only dependant on the angles θ and ϕ (and a trivial $1/|\mathbf{r}|$ dependence due to the spherical wave nature of the far field) [64]. The plane through $\mathbf{r}$ and the wave vector of the incident beam (in this case the z-axes) is called the scattering plane. The angle θ between the incident wave vector and $\mathbf{r}$ is the scattering angle.

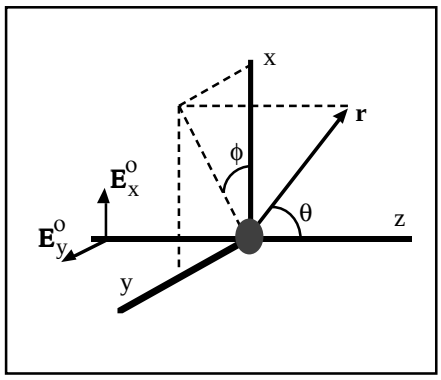


Figure 10: Scattering geometry

As a simplification we assume $\phi = \pi/2$, the yz plane is the scattering plane. The incident and scattered electric field are resolved in components perpendicular (subscript $\perp$) and parallel (subscript $||$) to the scattering plane. In this case $(\mathbf{E}^0)_{||} = (\mathbf{E}^0)_y$ and $(\mathbf{E}^0)_\perp = (\mathbf{E}^0)_x$, where the superscript 0 denotes the incident light. The formal relationship between the incident electric field and the scattered electric field (superscript s), written as Jones vectors, is

$$\begin{pmatrix} \mathbf{E}_{||}^s \\ \mathbf{E}_\perp^s \end{pmatrix} = \frac{e^{ik(r-z)}}{-ikr} \begin{pmatrix} S_2 S_3 \\ S_4 S_1 \end{pmatrix} \begin{pmatrix} \mathbf{E}_{||}^0 \\ \mathbf{E}_\perp^0 \end{pmatrix}. \qquad [30]$$

The matrix elements S_j (j = 1,2,3,4) are the complex amplitude scattering functions.

The scattering functions S_j depend on the space angles θ and ϕ, the shape, structure, and optical properties of the particle, and the size parameter α, defined as

$$\alpha = \frac{2\pi r}{\lambda}, \qquad [31]$$

with λ the wavelength of the incident light, and r a "radius" of the particle. Analytical expressions exist only for homogeneous (concentric or layered) spheres, ellipsoids and infinite cylinders (see section 1.3.1). Furthermore, a large number of approximations for limiting values of the scattering parameter α and the relative refractive index m exist (see [42], paragraph 10.1).

Another convenient way to describe ELS is in the framework of Stokes vectors and Mueller matrices [37, chapter 3.3]. The incident and scattered light is represented by a four-vector with real elements (the Stokes vector), and the transition from incident to scattered light is described by the 4×4 real scattering matrix **S** (Mueller matrix). The elements of the scattering matrix are easily derived from the amplitude scattering functions S_i [37, page 65].

The 16 elements of the scattering matrix are not independent. As can be seen from equation 30, 7 functions suffice[3] to describe the scattered field as a function of the incident field. Therefore, the scattering matrix contains a

3 An overall phase factor can be omitted

maximum of 7 independent elements, and one can derive 9 relations between these elements (see e.g. [65, 66, 67]. Furthermore, by merely considering symmetry properties, such as mirror symmetry and reciprocity of the particles, more simple and symmetric forms of the scattering matrix can be derived [42, chapter 5]. Using these arguments it can be shown that a suspension of randomly oriented particles with e.g. mirror symmetry has a scattering matrix

$$\mathbf{S} = \begin{bmatrix} S_{11} & S_{12} & 0 & 0 \\ S_{12} & S_{22} & 0 & 0 \\ 0 & 0 & S_{33} & S_{34} \\ 0 & 0 & -S_{34} & S_{44} \end{bmatrix}, \qquad [32]$$

and that a (concentric) sphere has a scattering matrix

$$\mathbf{S} = \begin{bmatrix} S_{11} & S_{12} & 0 & 0 \\ S_{12} & S_{11} & 0 & 0 \\ 0 & 0 & S_{33} & S_{34} \\ 0 & 0 & -S_{34} & S_{33} \end{bmatrix}. \qquad [33]$$

For completeness, we also present the scattering matrix of Rayleigh-Debye-Gans theory (see e.g. reference 54):

$$\mathbf{S} = \begin{bmatrix} S_{11} & S_{12} & 0 & 0 \\ S_{12} & S_{11} & 0 & 0 \\ 0 & 0 & S_{33} & 0 \\ 0 & 0 & 0 & S_{33} \end{bmatrix}. \qquad [34]$$

The Stokes vectors and scattering matrix formalism is very convenient to describe ELS experiments with polarised incident - and scattered light [see e.g. 37, page 415]. Throughout the thesis this formalism will be used.

Observables, such as the differential cross sections, extinction -, absorption -, and scattering cross section, asymmetry factor, or albedo are readily obtained from the S_i or $\mathbf{S}_{ij}$ elements.

1.3.3 Elastic Light Scattering from Biological Cells

1.3.3.1 A mini-survey of theories

Physiological and morphological information of living cells can be retrieved using ELS. Differences between cell populations, which are difficult to observe using microscopic methods can be determined with ELS, or a (physiological) response to e.g. drug treatment, a change in rate of phosphorilation, or a change of the osmolarity of the suspending medium, can be demonstrated with ELS.

This important notion has long been recognised by many researchers. For instance, the effect of penicillin on bacteria is seen within minutes in the

scattered light [68], changes in heat-treated bacteria can be observed [69], bacteria [70, 71], and human white bloodcells can be identified [62], changes in the shape of platelets can be assessed [72], perpendicular and forward scattered light can be used to determine nuclear and cellular morphology [73], or cell activation might be probed [74]. Many more fascinating examples can be found throughout the literature.

Such enormous amount of empirical data of ELS from biological cells calls for a theoretical support, because

- "a theoretical framework to interpret these changes in terms of scattering theory is needed" [75];
- to help the experimenter, quoting Brunsting [76]: "..what is the optimal solid angle for the light-scattering detector to subtend so that a certain cell parameter may be maximally monitored...".

As will be clear from the sequel, these quotations still are very relevant today.

The basic question now is to find an appropriate scattering theory [see e.g. 1, chapter 1]. For this we should realise that most biological cells have two characteristics in common. First, the refractive index of the cells, or cell compartments, has a very limited range, and is very close to the refractive index of the suspending medium (for wavelengths in the visible region, where absorption of cellular molecules is absent). For instance, Brunsting and Mullaney measured the refractive index of the cytoplasm and the nucleus of Chinese Hamster Ovary cells [77]. They found $n_{cytoplasm} = 1.3703$ and $n_{nucleus} = 1.392$. Relative to water ($n_{water} = 1.333$), we find 1.03 and 1.04 respectively. The relative refractive index of human white bloodcells is in the range $1.01 < m < 1.08$ [78]. In general, most biological cells can be viewed as near index matching particles, i.e. the relative refractive index m is very close to 1. This observation reduces the possible scattering theories to (excluding variational - and numerical techniques to solve the Maxwell equations):

- approximate theories for which m must be close to 1 (see reference 42, paragraph 10.1); i.e. Rayleigh-Debye-Gans (RDG) scattering, and Anomalous Diffraction (AD);
- approximate theories for which m is not restricted, such as the geometrical optics (ray tracing) method (see e.g. reference 37, chapter 7), or Rayleigh scattering (provided that the particle is not in the resonance region, see reference 42, paragraph 6.4).
- exact analytical solutions of the Maxwell equation, such as Lorenz-Mie scattering.

The second common feature of biological cells is their size. If we concentrate on (mammalian) cells, their radius is approximately $2\ \mu m \leq r \leq 8\ \mu m$ (see e.g. table I of reference 75, or chapter 1 of reference 1). Taking the wavelength $\lambda = 500$ nm we find for the scattering parameter $25 < \alpha < 100$. This range immediately excludes Rayleigh scattering and the geometrical optics approach.

The size - and refractive index range of cells reduces the theories to describe ELS of these cells to RDG scattering, AD theory, and exact theories, especially Lorenz-Mie theory [1, chapter 1]. All these theories have been used, with varying success. For instance, Mullaney and Dean [79] modelled cells as homogeneous spheres with $m = 1.05$, and investigated the near forward scattering using Mie theory. They show that in this model, the logarithm of the

$\theta = 0.5^0$ scattered intensity is proportional to the volume of the cell, for $10 < \alpha < 100$. However they realise that "The structure of a cell is certainly more complicated than assumed here; the concept of a mean refractive index for all cells in a population may be an oversimplification".

The next logical step is to model cells as two concentric spheres, where the inner sphere models the nucleus and the coating models the cytoplasm. Brunsting and Mullaney [80] propose this model and investigate the Aden-Kerker solution of scattering by two concentric spheres [38] in the (α, m) range of biological cells. They conclude that "the influence of the core on scattering is not significant in the forward direction", but that "the main influence of the cell nucleus occurs at angles outside the main scattering lobe". In an experimental study on the scattering of CHO cells in M and G_1 phase, which can be modelled by homogenous - and concentric spheres respectively, these theoretical conclusion where confirmed [77]. However, Meyer and Brunsting [81] have studied the concentric sphere model in some more detail, and they point out that the small angle scattering is highly dependent on the nucleus to cell diameter ratio. Furthermore, they observe that "the scattering patterns of nucleated cells are found to have a fine lobe (high frequency) structure dependent on whole cell size, and an envelope lobe (low-frequency) structure dependent on relative nucleus size.

Sloot and Figdor [82] propose a modification of the RDG theory for concentric spheres, called mRDG theory, which has a larger domain of applicability. They show that the resulting model, in the (α,m) range for biological cells, agrees very well with the Aden-Kerker theory. We applied this model very successfully to explain the anomalous Forward Scattering of osmotically stressed human Lymphocytes [43, 44, see also chapter 2 of this thesis].

Examples of the use of AD theory in the context of biological cells are provided by Stramski et al. [83], who study ELS from phytoplanktonic cells, and by Streekstra et al. [45, 46] who examined forward scattering from deformed red blood cells.

Although successful, these theories all have their limitations, which are most obvious if one takes the polarisation properties of the incident and scattered fields into account. Consider an experiment with crossed polarizers, i.e. the incident light is polarised perpendicular to the scattering plane, and the component of the scattered light parallel to the scattering plane is measured. In terms of Bohren and Huffman [37, table 13.1], this is a $P_{\perp}A_{//}$[4] experiment, and the measured intensity is proportional to

$$1/4\ (\mathbf{S}_{11} - \mathbf{S}_{12} + \mathbf{S}_{21} - \mathbf{S}_{22}) \qquad [35]$$

Both RDG scatterers and (concentric) spheres produce a zero intensity, as is obvious from Equations 33, 34, and 35. However, de Grooth et al. [84] measured

---

4 P denotes a polarizing element in the incident beam, A is the analyzer, a polarizing element just before the detector, U denotes unpolarized light, i.e. no polarizing element present. The subscripts denote which polarizing element is used: $\perp$ is a perpendicular linear polarizer; // is a parallel linear polarizer; + is a linear polarizer wich makes an angle of $+45^0$ with the scattering plane; - is a linear polarizer wich makes an angle of -45^0 with the scattering plane; R produces right handed circularly polarized light; L produces left handed circularly polarized light.

considerable signals for all types of human white bloodcells[5], and deviations of the $\mathbf{S}_{22}$ element from the $\mathbf{S}_{11}$ element have been reported for e.g. different kinds of pollen [56], and marine organisms [85, 86]. We may conclude that we cannot describe this situation with (m)RDG scattering, nor with models which view a cell as a (concentric) sphere, with equivalent volume and refractive index.

Another scattering matrix element which has received much attention is the $\mathbf{S}_{34}$ element. For many biological cells $\mathbf{S}_{34}$ signals have been reported. Once more we refer to Bickel et al. [56], Loftus et al. [86] and Voss et al. [85], and to van de Merwe et al. [87]. All these results show an extreme sensitivity of the $\mathbf{S}_{34}$ element for very small morphological changes. Furthermore, using an experimental flowcytometer which is described in reference 54, we have found indications that human white bloodcells give rise to $\mathbf{S}_{34}$ signals (data not shown).

RDG scatterers have $\mathbf{S}_{34} = 0$. The $\mathbf{S}_{34}$ is very sensitive to small morphological changes. For instance, in reference 50 (see also chapter 7) we showed that a small surface roughness on a sphere has a pronounced effect on the $\mathbf{S}_{34}$ element. Therefore, we conclude that RDG scattering and equivalent sphere models are also not suited for $\mathbf{S}_{34}$ calculations of biological cells.

Finally we refer to Circular Intensity Differential Scattering (CIDS) [57, 88, 89, 90, 91]. In a CIDS experiment the difference between scattered intensities for left - and right circularly polarised incident light is measured. In terms of Bohren and Huffman,

$$\mathrm{CIDS} = \frac{\mathrm{P_L}\,\mathrm{U} - \mathrm{P_R}\,\mathrm{U}}{\mathrm{P_L}\,\mathrm{U} + \mathrm{P_R}\,\mathrm{U}} = -\mathbf{S}_{14} \ . \qquad [36]$$

RDG scatterers, with an isotropic polarizability, have a zero CIDS. Anisotropic polarizabilities are required in RDG theory to produce CIDS signals. However, so-called form-CIDS, induced by e.g. the helical structure of a particle, is not present in RDG theory. Yet this type of CIDS is demonstrated in e.g. the helical sperm head of the octopus *Eledone cirrhosa* [92]. This technique is also very relevant in the study of secondary - and ternary structures of macromolecules, such as e.g. the supercoiling of DNA [93].

In conclusion, approximate theories such as (m)RDG scattering, and exact theories such as Lorenz-Mie theory are very successful to describe certain scattering properties of biological cells. Especially near-forward scattering and total cross sections are treated adequately by these theories. However, if large angle scattering, back scattering, or polarisation properties of the scattered light are of interest, more realistic models are required.

1.3.3.2 ELS from healthy - and malignant human leukocytes

Leukocytes, or white bloodcells, make up our immune system. In peripheral blood they consist of 3 classes, the Lymphocytes, Monocytes, and Granulocytes [94] (see table 2). A further subdivision can be made on morphological and physiological grounds, and some subclasses are also shown

---

5 Even if one takes the so-called aperture depolarization into account, which explains for non-zero signals of spheres in an actual experiment (due to finite detector apertures), the cells still have significantly larger signals.

in table 2. Much more detail can be introduced by studying the biochemistry of the cell surface or of the cell compartments, but that will not concern us here. In table 2 the relative occurrence of (sub) classes is also shown. These numbers are for a typical healthy donor, and are subject to large inter-donor variations (see e.g. [95]).

<table>
<tr><td rowspan="5">Lymphocytes
(30%)</td><td colspan="2">B Lymphocytes</td></tr>
<tr><td rowspan="3">T Lymphocytes</td><td>Helper cells</td></tr>
<tr><td>Supressor cells</td></tr>
<tr><td>Cytotoxic cells</td></tr>
<tr><td colspan="2">Large Granular
Lymphocytes</td></tr>
<tr><td colspan="2">Natural Killer
cells</td></tr>
<tr><td colspan="3">Monocytes
(5%)</td></tr>
<tr><td rowspan="3">Granulocytes</td><td colspan="2">Neutrophiles
(60%)</td></tr>
<tr><td colspan="2">Eosinophiles
(4%)</td></tr>
<tr><td colspan="2">Basophiles
(1%)</td></tr>
</table>

Table 2: Classes and subclasses of human white bloodcells, the numbers in brackets denote relative occurrence for a typical healthy donor.

The three classes of leukocytes have a clear different morphology, as is easily seen under a light microscope [94]. Furthermore, morphological differences between subclasses exist. Most notably are the Neutrophilic - and Eosinophilic Granulocytes, the main difference being the size and concentration of granules in the cytoplasm of the cells and the shape of the nucleus. More difficult to determine however are morphological differences between Lymphocyte subclasses. Still, examples are known: using a computer assisted microscopic technique, Bartels et al. [96] revealed morphological differences between (fixed) B - and T - lymphocytes, and van Es et al. [97] reported morphological differences between T4 - and T8 positive lymphocytes. Since the scattering matrix is very sensitive to (variations of) morphological features, we expect that it is possible to identify subclasses of leukocytes by measuring ELS from those cells. Furthermore, physiological responses of leukocytes often involve a morphological change, and pathological stages of leukocytes (such as leukemias) usually show clear morphological signatures. Therefore, ELS can be a very useful technique for fast and non-destructive monitoring of responses or pathologies.

Without the intention of being complete, a small overview of scattering from leukocytes will be presented. Up till now we have not encountered examples of measurements of $\mathbf{S}_{ij}$ elements as a function of the scattering angle

θ. Most experimental results stem from flowcytometric experiments.

A flowcytometer consists of a flow-cell, in which cells are forced to flow into a long linear sequence, using a technique called hydro-focusing. A laser beam is focused on the cells, such that cells enter the focused beam one-by-one. Each cell produces a pulse of scattered light, and this scattered light is usually measured in the two principal directions: the Forward Scattered (FS; usually $1^0 < \theta < 3^0$) - and the Sideward Scattered direction (SS; usually $65^0 < \theta < 115^0$). The FS and SS signals consist of scattered light integrated over a solid angle defined by the aperture of the detectors. Usually the cells are stained with fluorescent probes which bind to specific molecules on the cell surface or in the cell, and several fluorescence signals are measured, in combination with the light scattering signals. In this way N independent signals are measured for every cell in the sample, and the sample is represented as a N-dimensional histogram. After identifying separate clusters in the histogram, by visual inspection or fully automated, using sophisticated statistical techniques [see e.g. 98], different subsets in the original cell sample can be assessed. Many more details of this important technique, extensively used in analytical cytology, can be found in reference 99.

In classical flowcytometry only two light scattering (FS and SS) signals are measured. Furthermore, the incident light is provided by a laser, which usually emits linear polarised light. If we assume perpendicular polarised incident light, the classical light scattering measurements in flowcytometry are of $P_{\perp}U$ type, and therefore

$$\mathrm{FF,SS} = C \iint_{\substack{\text{detector} \\ \text{surface}}} (\mathbf{S}_{11} - \mathbf{S}_{12}) dA \ , \qquad [37]$$

with C a constant. Only a very small portion of the information which is potentially present in the scattered light is exploited. Yet, this suffices to identify the three classes of leukocytes, as was first demonstrated by Salzman et al. [62]. Figure 11 schematically shows the FS-SS histogram which was measured by Salzman.

Lymphocytes and Monocytes can be identified on the basis of their FS signal, and by measuring the SS signal at the same time, the Granulocytes can also be discriminated. The clusters in the histogram are Gaussian distributions (see reference 98), and the width of the histograms is probably due to biological variability of the cells. However, the influence of the so-called Lissajous loops (see reference 100, and chapter 3) is still under investigation.

A popular, quantitative interpretation of the histogram of figure 11 is that FS (small angle scattering) is sensitive to cell size and that SS (perpendicular scattering) is sensitive to cell structure or cell granularity. These arguments are based on the early work of e.g. Brunsting, Mullany, and Latimer (see previous section). Although these arguments help to understand some features of FS-SS histograms, one should be careful using them. Light scattering is much more subtle, and every experiment should be considered in its own right. Anyhow, since Monocytes are larger than Lymphocytes, and Granulocytes have more internal structure (granules, polymorphological nucleus) than both Lymphocytes and Monocytes, these arguments allow us to appreciate the data

of figure 11.

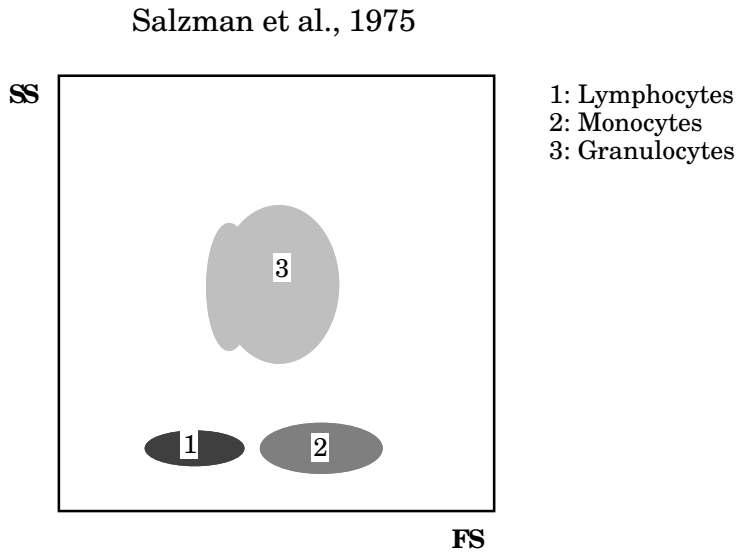


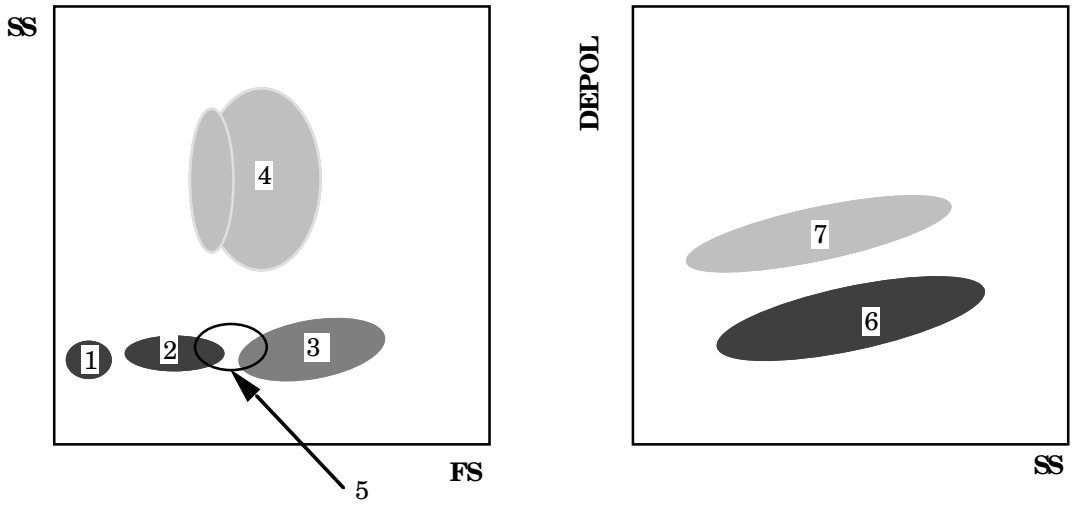
Figure 11: A schematic drawing of the FS - SS histogram of human leukocytes, as measured by Salzman et al. [62].

Detection and data analysis techniques in flowcytometry are more and more refined, allowing better and more accurate identification of the (sub) classes of leukocytes. Furthermore, de Grooth et al. [84] introduced the crossed polarizers experiment, (see previous section) into flowcytometry of human leukocytes. This allowed them to separate Eosinophiles from Neutrophiles. In figure 12, taken from reference 101, the resulting FS-SS histogram and SS-Depol histogram (Depol is the acronym for the crossed polarizers experiment with the analyzer in the SS direction) is drawn.

The FS-SS histogram contains two new distributions. First, at a low FS a cluster containing debris, platelets and red cells is included. Furthermore, a small cluster containing Basophiles (number 5) was located between, or overlapping the Lymphocyte and Monocyte clusters. This identification was possible by immunofluorescence measurements. Furthermore, the large Granulocyte cluster in the FS-SS histogram could be separated into a Neutrophil and Eosinophil cluster by the crossed polarizers experiment. Therefore, by measuring only three light scattering signals of each cell a clear distinction between Lymphocytes, Monocytes, Neutrophiles and Eosinophiles can be made.

Improvements are still being made, and a state-of-the-art measurement is drawn in figure 13. These data were taken from reference 102. The large Granulocytes cluster consists of two overlapping distributions containing the Eosinophiles and Neutrophiles. Furthermore, it was shown that the Basophiles overlap with the lymphocyte fraction, and that a subset of Lymphocytes, the Natural Killer (NK) cells, can be found at larger FS and SS signals in the lymphocyte fraction, and that some NK cells can be demonstrated at low SS signals in the Monocyte cluster. These overlapping clusters of cells were found using three immunofluoresence signals.

Terstappen et al., 1988

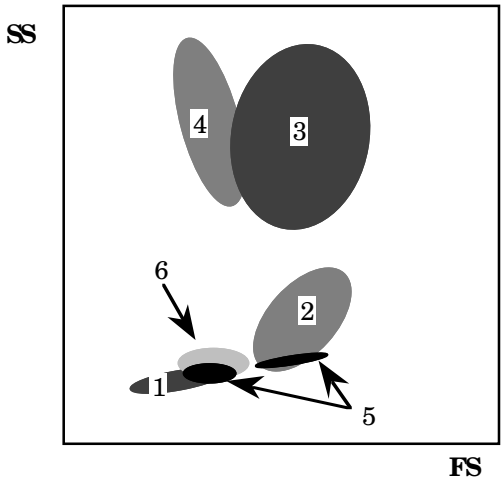


1: platelets, debris, red cell ghosts
2: Lymphocytes
3: Monocytes
4: Granulocytes
5: Basophils, Lymphocytes and Monocytes

6: Neutrophils
7: Eosinophils

Figure 12: A schematic drawing of the FS - SS histogram, and the SS - Depol histogram of human leukocytes, as measured by Terstappen et al. [101].

Terstappen et al., 1990



1: Lymphocytes
2: Monocytes
3: Neutrophils
4: Eosinophils
5: Natural Killer Cells
6: Basophils

Figure 13: A schematic drawing of the FS - SS histogram of human leukocytes, as measured by Terstappen et al. [102].

Furthermore, Terstappen et al. have studied in great detail the SS of human Lymphocytes. They discovered that the SS histogram contains two overlapping distributions, as drawn schematically in figure 14. This effect was first described for T8 positive Lymphocytes [103]. It was suggested that the population with small SS consists of Supressor cells and the population with large SS of Cytotoxic cells. Next, Terstappen et al. demonstrated this effect for the complete Lymphocytes class [104]. It was shown that B Lymphocytes and

Helper / Supressor cells have a low SS, and that "the presence of cytotoxic determinants on Lymphocytes is correlated with a large SS". Furthermore, strong indications exist that all Lymphocytes with large SS are in fact Cytotoxic Lymphocytes.

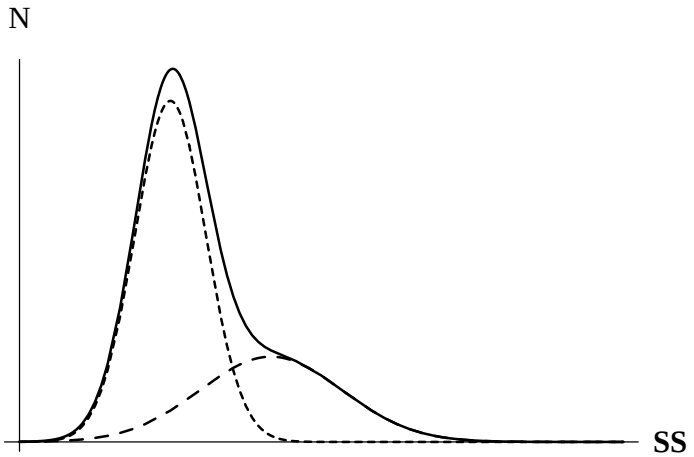


Figure 14: A schematic drawing of the measured SS histogram of Lymphocytes (solid line) and the two overlapping (Gaussian) distributions which produce the total histogram (dashed lines); N is the number of cells.

This body of experimental data shows that all subsets in table 2 can probably be distinguished using ELS. Now we must recall the quotation of Brunsting: "..what is the optimal solid angle for the light-scattering detector to subtend so that a certain cell parameter may be maximally monitored...". Several groups have gone beyond the classical FS-SS measurements. The importance of the crossed polarizers experiment of de Grooth et al. has been mentioned. They have also investigated the possibilities of measuring the scattered light in the range $3^0 < \theta < 11^0$, in combination with FS. This experiment revealed the same information as the FS-SS histogram of figure 12, but now cluster 1 (platelets etc.) is much better separated from clusters 2, 3, and 5 [101].

Stewart et al. [105] measured Axial Light Loss (ALL), the decrease of laser intensity upon entrance of a particle into the beam. ALL seems to result in an even better separation of dead cells, Lymphocytes and Monocytes than the FS signal.

Using their mRDG theory, Sloot and Figdor extensively studied differential cross sections of cells and concluded that measurement of back scattered light (BS; $160^0 < \theta < 174^0$), combined with FS and SS, provides additional and independent morphological information of the cells [82]. Based on this theoretical work, and in the spirit of Brunsting, Sloot has included BS measurements into flowcytometry [1]. Many experiments were carried out and the data suggest that BS does provide valuable information (see also section 1.2.3.3).

The use of ELS to study pathologies of leukocytes is not widespread, but some examples are known to us. We will restrict ourselves to Chronic Lymphocyte Leukemia (CLL) [106], the most common form of leukemia in Europe and the United States. In more than 95% of the cases the CLL develops from a malignant transformation of B Lympocytes or B Lymphocyte precursors. Several forms of this leukemia are described with clear

morphological features and correlated immunological properties [107]. Examples are

- true B Chronic Lymphocytic Leukemia (B-CLL): small cells, hardly any cytoplasm;
- Prolymphocytic Leukemia (PLL): larger cells, abundant cytoplasm, nucleus with very prominent central nucleolus;
- Hairy cell leukemia: cells of moderate size, eccentric, oval nuclei, and "hairy" cytoplasmic projections.

Once more, these morphological differences suggest that ELS can be used to identify the cells, thus allowing e.g. an initial, fast screening of cell samples. Flowcytometric experiments, which measured the SS histograms of these cells, only lift a corner of the veil, and much more theoretical and experimental study should be devoted to this application of ELS.

Van Bockstaele et al. [108] showed that Hairy cells have a three to four fold larger SS than lymphocytes. In combination with ALL they were able to count the concentration of Hairy cells, and follow the effects of treatment with alpha-interferon. Terstappen et al. studied the SS of B-CLL cells. In this case the T8 positive cell histograms resembled the SS of healthy donors (see figure 14), but with the small SS cluster missing [109], and in the SS histogram of all Lymphocytes the large SS cluster is missing [110]. Furthermore, the SS histogram of the Lymphocytes of B-CLL is clearly different than those from other CLL such as Leukemic Follicular Non Hodgkin Lymphoma and a prolymphocytoid transformation of B-CLL [110]. Finally, the effect of splenic irradiation of a B-CLL patient could be demonstrated in the SS histogram [95].

1.3.3.3 Our previous contributions to ELS from leukocytes

Our interest in ELS from leukocytes stems from an application in immunology. Here, the Biologists need large amounts of purified leukocytes, preferably not contaminated with chemicals which are used in many separation techniques (e.g. flowcytometric sorting using immunofluorescence). Therefore, a very powerful physical separation and purification technique, so-called centrifugal elutriation, was developed to separate leukocytes into samples of (sub) classes of leukocytes [111]. Sloot integrated a flowcytometer with the centrifugal elutriator, thus enabling a real time monitoring (and in the future also real time control) of the cell separation process [1, 112, 113]. Obviously, in this situation only ELS from the cells can be used to identify them. Therefore, we are looking for flowcytometric ELS experiments able to identify all subclasses of leukocytes. Furthermore, malignant leukocytes and bonemarrow cells can be included. In this effort experimental - and theoretical investigations have to go hand in hand.

The mRDG theory of Sloot and Figdor has been mentioned in section 1.3.3.2. We used this theory to explain the anomalous Forward Scattering (FS) of osmotically stressed human Lymphocytes [43]. Based upon mRDG calculations of the FS of osmotically stressed Lymphocytes we even predicted a totally unexpected biological phenomenon, the change of nuclear volume of Lymphocytes in aniosmotic conditions. Recently we have proven this prediction to be correct [44, see also chapter 2 of this thesis].

Furthermore, as said in section 1.3.3.2, mRDG calculations laid the foundation of BS measurements in flowcytometry. A large body of experimental

data indicates that combined measurement of FS, SS, and BS allows identification of Lymphocytes, Monocytes, Eosinophiles and Neutrophiles (unpublished data). Currently our group is finalising data reduction techniques to analyse the three dimensional histograms in real time [114]. With this the potential of BS measurements will be analysed in detail.

We also started to investigate the scattering matrix of leukocytes, because we expect that only this can provide the necessary information. First we showed that the scattering matrix can be measured quantitatively in a flowcytometer [54, 55]. Based on a symmetry argument and using the equality of ensemble - and time averaging, we postulated that the scattering matrix of cells, measured in a flowcytometer, has the form of equation 32. Recent experiments carried out in collaboration with de Grooth of the University of Twente, the Netherlands, indicate that this hypothesis is correct, although more conclusive data must be obtained. Furthermore, these experiments, and experiments reported in [54] and [55], showed that the scattering matrix of most human leukocytes deviate from the (m)RDG approximation; i.e. non-zero $\mathbf{S}_{34}$, $\mathbf{S}_{22} \neq \mathbf{S}_{11}$, and $\mathbf{S}_{33} \neq \mathbf{S}_{44}$.

The research which is presented in this thesis continues our experimental developments. Furthermore, the main part of the thesis consists of the development of a simulation environment for Elastic Light Scattering from biological particles. In this way we try to fill the computational gap which was identified in the previous sections. Finally, first applications of the computational experiment will be reported.

1.3.4 The Coupled Dipole model

1.3.4.1 Why use the Coupled Dipole model ?

Our aim is to calculate the complete scattering matrix of Leukocytes. This provides the possibility to define optimal light scattering experiments, able to distinguish (sub) classes of Leukocytes and bone marrow. Approximate theories and exact solutions for highly symmetric particles are not suited for these calculations, as was shown in the previous section. A well known, though computational intensive method to calculate the ELS from non-spherical particles is the T-matrix method [115]. Unfortunately this method is currently only practically applicable to homogeneous particles with relatively smooth surfaces, which makes this method unfit to solve our problem. The conclusion must be that we have to consider other, more rigorous approaches to calculate the ELS pattern of arbitrary particles.

We must rely on numerical methods to solve the scattering problem. Basically one can distinguish two methods; the first concentrates on numerically integrating the macroscopic Maxwell equation in the time or frequency domain. The second approach, the Coupled Dipole method, is a physically more intuitive approach, although it can formally be derived from the Maxwell equations. We intend to calculate the ELS pattern by means of the Coupled Dipole method.

1.3.4.2 Description of the Coupled Dipole method

The Coupled Dipole (CD) method of ELS was first introduced in 1973 by

Purcell and Pennypacker [116]. The concept of coupled radiating dipoles to study the interaction between light and matter on a microscopic level is however much older (see e.g. [117]). The main difference is that the dipoles in the CD method of ELS are not single atoms, as in the original formulations, but consist of large numbers of atoms, confined to a sub volume that is small compared to the wavelength of the incident light.

Consider an arbitrary particle, illuminated by a monochromatic electromagnetic field $\mathbf{E}^0(\mathbf{r})$, as drawn in figure 10. 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 **S** of the particle can be calculated [37].

The CD method divides the particle in N equal sub volumes. The size of a sub volume 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 $\lambda/20 < d < \lambda/10$, with d the size of a sub volume [50, see also chapter 7].

The field at $\mathbf{r}_i$ radiated by a dipole located at $\mathbf{r}_j$ is [64]

$$\mathbf{E}(\mathbf{r}_i) = \frac{1}{4\pi\varepsilon_0}\left(k^2(\mathbf{n}\times\mathbf{p})\times\mathbf{n}\frac{e^{ikr}}{r} + [3\mathbf{n}(\mathbf{n}\cdot\mathbf{p})-\mathbf{p}]\left(\frac{1}{r^3}-\frac{ik}{r^2}\right)e^{ikr}\right); \qquad [38]$$

k is the wave number, defined by $k = 2\pi/\lambda$; **p** is the induced dipole moment; **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:

$$\mathbf{p} = \boldsymbol{\alpha}\mathbf{E}, \qquad [39]$$

$\boldsymbol{\alpha}$ is the polarizability tensor. The field at $\mathbf{r}_i$ radiated by a dipole located at $\mathbf{r}_j$ can now be written as

$$\mathbf{E}(\mathbf{r}_i) = \mathbf{F}_{ij}\boldsymbol{\alpha}_j\mathbf{E}_j \; , \qquad [40]$$

with the functional $\mathbf{F}_{ij}$ defined by equations 38 and 39.

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

$$\mathbf{E}(\mathbf{r}_i) = \mathbf{E}^0(\mathbf{r}_i) + \sum_{j\neq i}^{N}\mathbf{F}_{ij}\boldsymbol{\alpha}_j\mathbf{E}_j \; , 1 \le i \le N. \qquad [41]$$

The summation in equation 41 runs over all dipoles, except dipole i. This term, the so-called eigenterm, is neglected in the original CD method (see however

section 1.3.4.4).

Equation 41 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

$$\mathbf{x} = \begin{pmatrix} \mathbf{E}(\mathbf{r}_1) \\ \vdots \\ \mathbf{E}(\mathbf{r}_N) \end{pmatrix}, \mathbf{b} = \begin{pmatrix} \mathbf{E}^0(\mathbf{r}_1) \\ \vdots \\ \mathbf{E}^0(\mathbf{r}_N) \end{pmatrix} \text{and } \mathbf{A} = \begin{pmatrix} \mathbf{a}_{1,1} & \cdots & \mathbf{a}_{1,N} \\ \vdots & \ddots & \vdots \\ \mathbf{a}_{N,1} & \cdots & \mathbf{a}_{N,N} \end{pmatrix}, \quad [42]$$

where $\mathbf{a}_{i,i} = \mathbf{I}$, the 3×3 identity matrix, and $\mathbf{a}_{i,j} = -\mathbf{F}_{i,j}\boldsymbol{\alpha}_j$ if $i \neq j$. The matrix $\mathbf{A}$ is the 3N×3N interaction matrix. 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 $\mathbf{r}_{obs}$

$$\mathbf{E}^S(\mathbf{r}_{obs}) = \sum_{i=1}^{N} \mathbf{F}_{obs,i}\boldsymbol{\alpha}_i\mathbf{E}(\mathbf{r}_i) \; . \quad [43]$$

1.3.4.3 The interaction matrix

The interaction matrix $\mathbf{A}$ is a complex, 3N×3N matrix (the 3 arising from the 3 spatial components of the electric field vectors). In practical applications the number N must be very large (e.g. ~10^5). To investigate the properties of $\mathbf{A}$, we must first note that (see equation [40])

$$\mathbf{F}_{ij} = \mathbf{F}_{ji}, \text{ and} \quad [44.a]$$
$$\mathbf{F}_{ij} = (\mathbf{F}_{ij})^T \; . \quad [44.b]$$

where the subscript T denotes the transpose of a matrix. At this point we are forced to make assumptions about the response properties of the dipoles, that is, the tensor $\boldsymbol{\alpha}_i$.

First we will consider the very important class of homogeneous, isotropic particles, or in terms of the coupled dipole formalism,

$$\boldsymbol{\alpha}_i = \alpha\mathbf{I}, \qquad i = 1..N, \quad [45]$$

were α is a scalar polarizability. From equation 42, 44, and 45 it is obvious that the interaction matrix is symmetric:

$$\mathbf{A}^T = \mathbf{A}, \quad [46]$$

and that $\mathbf{a}_{ij} = -\alpha\mathbf{F}_{ij}$ for $i \neq j$.

The class of homogeneous, non-isotropic particles is characterised by

$$\boldsymbol{\alpha}_i = \boldsymbol{\alpha} \qquad i = 1..N. \quad [47]$$

Without loss of generality we can assume that $\boldsymbol{\alpha}$ is symmetric. We can now rewrite the matrix equation to

$\mathbf{B}\,\boldsymbol{\alpha}\,\mathbf{E} = \mathbf{E}^0$, or

$$\mathbf{B}\,\mathbf{P} = \mathbf{E}^0, \qquad [48]$$

where

$$\mathbf{P} = \begin{pmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_N \end{pmatrix} \text{ and } \mathbf{B} = \begin{pmatrix} \mathbf{b}_{1,1} & \cdots & \mathbf{b}_{1,N} \\ \vdots & \ddots & \vdots \\ \mathbf{b}_{N,1} & \cdots & \mathbf{b}_{N,N} \end{pmatrix},$$

$\mathbf{b}_{ii} = (\boldsymbol{\alpha})^{-1}$ and $\mathbf{b}_{ij} = -\mathbf{F}_{ij}$ if $i \neq j$. Because $\boldsymbol{\alpha}$ is symmetric, $(\boldsymbol{\alpha})^{-1}$ also is symmetric and therefore the modified interaction matrix **B** is symmetric. The elements of the vector **P** are the induced dipole moments; $\mathbf{p}_i = \boldsymbol{\alpha}\,\mathbf{E}(\mathbf{r}_i) = \mathbf{p}(\mathbf{r}_i)$.

Finally, consider non-homogeneous particles. Now the polarizabilities $\boldsymbol{\alpha}_i$ can differ from each other, although we still assume symmetric polarizability tensors. The matrix equation can be rewritten as equation 48, but now $\mathbf{b}_{ii} = (\boldsymbol{\alpha}_i)^{-1}$, $\mathbf{b}_{ij} = -\mathbf{F}_{ij}$ if $i \neq j$, and $\mathbf{p}_i = \boldsymbol{\alpha}_i\,\mathbf{E}(\mathbf{r}_i) = \mathbf{p}(\mathbf{r}_i)$. In this case the modified interaction matrix also is symmetric.

In the sequel of this thesis we will be dealing with isotropic, homogeneous particles, unless stated otherwise.

1.3.4.4 Calculation of the polarizability , the Optical Theorem, and derivation from the Maxwell equations

Correct calculation of the scalar polarizability α is an important ingredient of the CD method. In the original paper of Purcell and Pennypacker [116] the dipoles are placed on a cubic grid with grid spacing d, and α was calculated using the Clausius-Mosotti relation:

$$\alpha = \frac{4\pi\varepsilon_0}{3n}\frac{m^2-1}{m^2+2}, \qquad [49]$$

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

$n = d^{-3}$.

Although the Clausius-Mosotti relation is only rigorous for point-like dipoles on infinite grids, this connection between the polarizability of the dipoles with the relative refractive index of the scattering particle is very satisfactory. Comparison of CD simulations, using equation 49, of differential scattering cross sections from a homogeneous sphere with Mie theory show almost perfect agreement (see e.g. chapter 7 and 8).

An important limitation of the use of the Clausius-Mosotti relation lies in the fact that the optical theorem is violated. This is a well-known problem and is treated in many textbooks in the context of e.g. Rayleigh scattering. Non-absorbing particles with real m yield dipoles with real α according to equation

49. However, the single dipoles in the simulation do scatter light and their polarizabilities should have an imaginary part, according to the optical theorem. As was noted by several authors [118, 119], this restriction does not introduce severe errors in the simulation of the differential cross sections. Nevertheless, it remains an important point of principle. The CD simulation, in combination with the Clausius-Mosotti relation, violates the optical theorem. As a consequence, the optical theorem cannot be used to calculate total scattering cross sections of particles.

Several solution to this problem have been suggested. First, Draine [118] introduced an imaginary part into the polarizability by considering a radiative reaction field on the dipoles. Dungey and Bohren [120] introduce an exact formulation of the electric dipole polarizability of small spheres in the CD formalism. This formulation basically is the first term in the Mie series, and includes the radiative reaction term of Draine as the third order term of an expansion in the size parameter of the spherical dipole. Furthermore, Dungey and Bohren use the Maxwell-Garnett relation to find the dielectric permeability of the dipoles as a function of the electric permeability of the bulk material. They show that their approach yields the most reliable simulations of the extinction coefficient with the CD method.

Another way to proceed is to include eigenterms into the CD formalism. In equation 41 the summation runs over all dipoles, except dipole i. Intuitive one expects that this term should also be present, due to the finite size of the dipoles in the simulation. In standard derivations of dipole radiation due to a distribution of current densities, resulting in equation 38, one assumes that the currents are confined to a very small volume, and that the observation point is outside this volume (see e.g. reference 64, page 392 and further). One may ask what happens if the observation point is *inside* the source volume, and identify this contribution with the eigenterm. Such an analysis, in the context of the CD method, remains to be done.

If one solves the Maxwell equation in the frequency domain, using a Green's function formalism, and discretizes the resulting integro-differential equation, using the method of moments, a comparable eigenterm naturally occurs (see e.g. [119, 121]. Furthermore, the resulting equations bear a strong resemblance with the CD equations. Lakhtakia has shown that this resemblance has a fundamental significance, by deriving the CD method directly from the Maxwell equations [122, 123, 124]. He shows that the CD method, using the Clausius-Mosotti relation, and the above mentioned integration and discretization of the Maxwell equations are in fact equivalent. Furthermore, he gives a sound theoretical basis for the CD method and an analysis to include eigenterms into the CD simulations. Up till now numerical tests of Lakhtakia's expressions have not been published. However, experiments of Hage [121] and Goedecke and O' Brien [119] show that the eigenterms also resolve the problems with the optical theorem.

1.3.4.5 Some results obtained with the CD method

The Coupled Dipole method can be viewed as an instance from the field of Computational Electromagnetics. Applications in Computational Electromagnetics, like those from e.g. Computational Fluid Dynamics or Structural Mechanics, are generally very compute intensive. In chapter 5 it is

shown that in our application this is also the case. As a consequence, the CD method is not (yet) widely used in the light scattering community. Despite this fact, some important and fascinating results have been obtained with the CD method.

The Coupled Dipole method contains several parameters. The size, form, and position of the dipoles and the value of the polarizability tensor, as a function of the morphological and optical properties of the scattering particle, have to be specified. The original formulation of Purcell and Pennypacker [116] assumes spherical dipoles on a cubic grid. They suggest that the grid size d should satisfy $2\pi d/\lambda < 0.7$. Singham [125] extended the original CD method to intrinsic optical active particles. The constitutive relations appropriate for optical active materials induce a coupling between the electric and magnetic field components. Therefore the resulting matrix equation is of order $6N \times 6N$, with N the number of dipoles. Here we will not consider optical active particles.

Purcell and Pennypacker tested the CD method by comparing it with known analytical solutions of the scattering problem. At that time (1973) only Mie's solution of scattering of plane waves by a sphere was available. The CD method is compared with Mie results. Two tests were performed. Differential scattering cross sections of a sphere composed of N = 136 dipoles, $\alpha = 1.5$, and bulk refractive index n = 1.33 were calculated for perpendicular and parallel polarised incident plane waves. Furthermore, the extinction-, scattering- and absorption efficiencies (Q_{ext}, Q_{scat}, and Q_{abs} respectively) for a 136 dipole sphere with n = 1.7 - 0.1i where computed, as a function of α. In both cases the agreement between the CD results and the Mie results was very good.

These first results prompted more research to the applicability of the CD method. Yung [126] notes that "further progress requires a better numerical approach". Using a variational principle he connects the minimum energy theorem with scattering cross sections, and from this derives an algorithm which can be identified as the original Conjugate Gradient method of Hesteness and Stiefel [127]. This approach allows him to analyse particles with as much as 15600 dipoles. This result is rather surprising, because his numerical algorithm can only be applied to Hermitian matrices [128]. As was shown in section 1.3.4.3, the interaction matrix **A** is symmetric, but definitely not Hermitian.

Druger et al. [129] demonstrate the usefulness of the method for inhomogeneous particles. The single polarizability α is replaced by a position dependent α_i. The light scattering from a concentric sphere, and from a sphere embedded in a larger sphere with the centre of the inner sphere displaced halfway to the surface of the outer sphere, was calculated with the Extended Boundary Condition Method (ECBM, [115]) and compared with the CD method. The polarizabilities of the inner and outer region where calculated with the Clausius Mosotti relation. The results are good, however not as good as for homogenous spheres. The sensitivity of the Back Scattering signals for the displacement of the inner sphere is clearly present in the CD calculations, although the effect is weaker than in the ECBM results. The deviation from the exact model is, according to Druger et al., mainly due to "the graininess of the model", and the agreement would be better with a larger number of dipoles (see also chapter 7, where we quantify Druger's notion of graininess).

Singham and Salzman [130] were the first to calculate the complete 4×4 scattering matrix with the CD method. This allowed them to test the CD

method qualitatively by checking the form of the calculated matrix with general forms of scattering matrices as predicted by symmetry relations (see section 1.3.2.1). For a single sphere they find that "the form of the calculated matrix is precisely as predicted provided that the dipoles are arranged symmetrically about all three coordinate axes". CD calculations on single linear and helical particles show the predicted form. Furthermore, orientational averaged linear and helical particles result in the predicted matrix form. The number of orientations for proper averaging varied from 3000 to 15000 for particles with dimensions in the order of the wavelength of the incident light. Even more orientations are probably necessary for larger particles. A quantitative test of the CD method was performed by comparing calculated scattering matrices of spheres and coated spheres with equivalent sphere Mie calculations. The results for the spheres show that the results for the $\mathbf{S}_{11}$, $\mathbf{S}_{12}$, and $\mathbf{S}_{33}$ are very good. The $\mathbf{S}_{34}$ element, the only term which is totally due to dipole-dipole interactions, shows the general behaviour of the Mie calculations, but a quantitative agreement is not achieved. A second test was performed on a hollow sphere. The polarizability is obtained from the Clausius-Mossoti relation. The results are good , though not as good as for the homogeneous case (see also [129]). Singham and Salzman impute this to imperfections of the discretization of the hollow sphere, but they also note that "the Clausius-Mossoti relation is rigorous only for points on a cubic lattice". Finally, Singham and Salzman investigate the possibility of ellipsoidal dipoles. In this case the polarizability becomes a diagonal 3×3 tensor. The elements of the polarizability tensor are calculated by a Clausius-Mossoti-like formula. The dielectric constant of the ellipsoid, occurring in this formula, is calculated by means of Maxwell-Garnett theory. The scattering of a hollow sphere, simulated with ellipsoidal dipoles was calculated. The results are as good as with spherical dipoles. Even in the limiting case for very flat oblate ellipsoids, simulating a very thin coating, the agreement with the Mie theory is good, except for the $\mathbf{S}_{34}$ element.

Singham and Salzman solve the system of equations with a direct method (LU decomposition), which has the disadvantage that the number of dipoles cannot be to large ($O(10^3)$). However, the advantage of a direct method is that randomising the orientation of the particle can be accomplished by explicitly solving the matrix equation just once for one orientation and applying rotation matrices to calculate the solution for other orientations [131].

The CD method simulates the scattering matrix of small spheres very well, except for the $\mathbf{S}_{34}$ element. Therefore Singham has, in more detail, investigated the applicability of the CD method to simulate the $\mathbf{S}_{34}$ term [132]. Comparison with Mie calculations on equivalent spheres show that good agreement can be obtained if $d < \lambda/45$, in contrast to the other matrix elements which are already in excellent agreement with the Mie results for $d < \lambda/20$. In chapter 7 (see also reference 50) we show that this conclusion is too pessimistic, and that the problems with simulation of the $\mathbf{S}_{34}$ element are due to the graininess of Singham's discretization of the sphere.

A number of papers [133, 134, 135] concentrated around the Jacobi iterative method [136] to solve the Coupled Dipole equations. Although it is known that convergence of the Jacobi iteration is very slow and that the iteration does not converge if the condition number of the matrix is too large, every step of the Jacobi iteration can be interpreted as a multiple scattering event between the

dipoles (see e.g. [133, 134]). Therefore, the CD method, in combination with the Jacobi iteration, can provide a better physical understanding of various light scattering phenomena. For instance "the results show that the diagonal scattering matrix elements and $\mathbf{S}_{12}$ are determined largely by the incident field and single scattering from neighbouring dipoles, while the 2×2 off-block-diagonal elements and $\mathbf{S}_{34}$ require longer range and higher-order internal scattering", or "this demonstrates that backscattering is much more sensitive than forward scattering to particle structure and that internal scattering processes are particularly important in calculating $\mathbf{S}_{11}$ near the backward directions" [134].

Draine [118] solves the set of linear equations by means of a Conjugate Gradient method, which can be identified with the CGNR method in Ashby's taxonomy [128]. Draine shows that the number of iterations required for convergence is relatively small compared to the number of dipoles. For instance, the set of equations for a sphere modelled with 2320 dipoles, $\alpha = 5$, and $m = 1.7 + 0.1i$, is solved in 17 iterations. For rectangular solids Flautau et al. exploit the block-Toeplitz structure of the resulting matrix to find a direct solution of the matrix equation, in execution times comparable with the iterative CG method [137].

The CD method has been used, among other, to calculate scattering from dielectric grains [116], magnetite dust [138], polynucleosome superstructures [93], fractal aggregates [139], dielectric helices [140, 141], and particles on surfaces [142, 143]. Many more examples can be found in the literature. Furthermore, Flatau et al. [144] simulated scattering by two spheres in contact and compared the results with modal analysis. The agreement is very good, which illustrates the applicability of the CD method to simulate irregular particles.

1.4 OUTLINE OF THE THESIS

This thesis contains three related clusters. The first cluster, chapter 2 and 3, is a continuation of our experimental studies of the ELS from human white blood cells. In this cluster modelling and subsequent computer simulation of the models, combined with a feedback of the simulation results to the ELS experiments, is the foundation of the research.

Chapter 2 elaborates on the hypothesis, which was formulated by Sloot et al. in reference 43, that the nuclear volume of osmotically stressed lymphocytes has to change in order to explain the observed ELS data. It is shown by means of slit scan experiments that this hypothesis is correct. Furthermore, we propose a possible mechanism of nuclear growth, a mechanical link between the cell membrane and the nuclear envelope, and model this system by means of a straightforward one-dimensional visco-elastic system.

As was argued in section 1.3.3 the polarisation properties of the scattered light contain a wealth of information, and are absolutely necessary to distinguish all subsets of human white blood cells and bonemarrow. Therefore we have continued the effort to measure the complete scattering matrix of

particles in flowcytometers. In co-operation with Dr. B.G. de Grooth of the University of Twente, the Netherlands, we extended the first experiments which are described in reference 54. During calibration of the new equipment we encountered an exiting phenomenon: Lissajous loops in scatterplots of spheres. In chapter 3 these loops are investigated by a computer simulation of the experiment. We show that these loops originate from the extreme dependency of the intensity of the sideward scattered light on the radius of the spheres.

The second cluster, chapters 4, 5 and 6, describes the development and implementation of a computer experiment to simulate ELS from arbitrary particles. In section 1.3.3 we showed that analytical and approximate theories of ELS cannot model all ELS properties of biological cells. The Coupled Dipole method can in principle simulate the complete scattering matrix of arbitrary particles (section 1.3.4). However, the Coupled Dipole applied to human white blood cells needs huge computational power, as will be shown in chapter 5. Parallel computers can offer this computational power if the parallel program has good scalability properties (see section 1.2.5). In chapter 4 we investigate the scalability properties of the kernel of the Coupled Dipole method; the calculation of the fields on the dipoles (equation 41). The parallel Coupled Dipole method is described in chapter 5, and issues of programmability, portability, and performance of parallel programs, with the parallel Coupled Dipole method as a case study, are addressed in chapter 6.

Finally, in cluster three, chapter 7 and 8, the parallel Coupled Dipole method is applied to simulate the scattering of spheres. These simulations serve as a further calibration of the computer experiment. In chapter 7 we investigate the influence of the size of the dipoles on the quality of Coupled Dipole simulations of the scattering matrix elements. In chapter 8 we introduce a new element into Coupled Dipole simulations, scattering of focused beams from arbitrary particles. Scattering by focused beams is highly relevant for the simulation of ELS in flowcytometers. By comparing the simulation results with generalised Lorenz Mie calculations, we will show that the Coupled Dipole method can predict scattering from focused beams.

Chapter 9 contains concluding remarks, and a status report. We will critically discuss the suitability of the current computer experiment to model ELS from human white blood cells, and provide pointers to further improvements of this simulation environment.

1.5 REFERENCES

1] P.M.A. Sloot, *Elastic Light Scattering in the development of computer assisted cell separation*, Ph.D. dissertation, University of Amsterdam, the Netherlands, 1988.

2] M.V. Wilkes and D.J. Wheeler, *The Preparation of Programs for an Electronic Digital Computer* (Addison-Wesley, Cambridge, MA, 1951).

3] J.J. Dongarra, H.W. Meuer, and E. Strohmaier, *TOP500 Supercomputers*, published electronically, for more information send electronic mail to top500@rz.uni-mannheim.de.

4] M.V. Wilkes and W. Renwick, "The EDSAC, an electronic calculating machine," J. Sci. Instrum. **26**, 385-391 (1949).

5] R.W. Hockney and C.R. Jesshope, *Parallel Computers 2* (Adam Hilger, Bristol and Philadelphia, 1988).
6] D. Wallace, "Algorithms and Architectures for Grand Challenges in Physics," In *Very Large Scale Computation in the 21st Century* (Capital City Press, J.P. Mesirov Ed., Ch. 1, pp. 1-22, Montpelier, Vermont, 1991).
7] L.G. Richardson, *Weather Prediction by Numerical Process* (Cambridge University Press, 1922).
8] J. von Neumann, "A system of 29 states with a general transition rule," In *Theory of Self Reproducing Automata* (Illinois University Press, A.W. Burks Ed., Urbana, Illinois, paper was first published in 1952, 1966).
9] S.H. Unger, "A computer oriented towards spatial problems," Proc. Inst. Radio Eng. **46**, 1744-1750 (1958).
10] D.L. Slotnick, "Unconventional systems," in *AFIPS Conference Proceedings*, 1967, pp. 477 - 481.
11] D.L. Slotnick, "The fastest computer," Sci. Am. **224**, 76-87 (1971).
12] J.H. Holland, "A universal computer capable of executing an arbitrary number of subprograms simultaneously," in *Proc. East. Joint Comput. Conf.*, 1959, pp. 108 - 113.
13] M.C. Pease, "The indirect binary n-cube microprocessor array," IEEE Trans. Comput. **C-26**, 458 - 473 (1977).
14] G. Fox, M. Johnson, G. Lyzenga, S. Otto, J. Salmon, and D. Walker, *Solving Problems on Concurrent Processors 1: General techniques and Regular Problems* (Prentice-Hall, Englewood Cliffs, New Jersey, 1988).
15] F. Darema, D.A. George, V.A. Norton, and G.F. Pfister, "A single-program-multiple-data computational model for EPEX/FORTRAN," Parallel Computing **7**, 11-24 (1988).
16] I. Danesh, "Physical Limitations of a Computer," Computer Architecture News **21**, 40-45 (1993).
17] U. Ghoshal and T. Van Duzer, "Superconductivity researchers seek to remove computational bottle-necks," Computers in Physics **6**, 585 - 592 (1992).
18] H. Kashiwagi, "Japanese super-speed computer project," In *High Speed Computing* (Springer Verlag, J.S. Kowalik Ed., pp. 117 - 125, Berlin, 1984).
19] S. Kotani, A. Inoue, T. Imamura, and S. Hassuo, Dig. Tech. Papers, 148 - 149 (1990).
20] G. Taube, "Is the Third Time a Charm for A Superconducting Computer?," Science **261**, 1670 - 1671 (1993).
21] K.K. Likharev and V.K. Semenov, IEEE Trans. Appl. Supercond. **1**, 3 - 28 (1991).
22] A.D. McAulay, "Researchers look to optics to move computer technology forward," Computers in Physics **6**, 594 - 602 (1992).
23] S. Lloyd, "A Potentially Realizable Quantum Computer," Science **261**, 1569 - 1571 (1993).
24] M.J. Flynn, "Some computer organisations and their effectiveness," IEEE Trans. Comput. **C-21**, 948 - 960 (1972).
25] S. Raine, *Virtual Shared Memory: A Survey of Techniques and Systems,* Tech. Rept. CSTR-92-36, University of Bristol, Computer Science Department, 1992.
26] K. Hwang, *Advanced Computer Architecture. Parallelism, Scalability, Programmability* (McGraw-Hill , New York, 1993).
27] X.H. Sun and L.M. Ni, "Scalable Problems and Memory-Bounded Speedup," J. Parallel Distrib. Comput. **19**, 27-37 (1993).
28] G. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities," in *Proc. AFIPS Conference*, 1967, pp. 483 - 485.
29] J.L. Gustafson, "Re-evaluating Amdahl's Law," Communications of the ACM **31**, 532 - 533 (1988).
30] G.C. Fox, S.W. Otto, and A.J.G. Hey, "Matrix algorithms on a hypercube I: Matrix multiplication," Parallel Computing **4**, 17 - 31 (1987).
31] A. Reinefeld, *The ZEUS Consortium, Massively Parallel Computing 1992/93,* Tech. Rept. PC2/TR-007-93, Paderborn Centre for Parallel Computing, Paderborn, Germany, 1993.
32] J.C. Maxwell, *Treatise on Electricity and Magnetism* (reprint by Dover, New York, 1954).
33] L. Rayleigh, "On the light from the sky, its polarisation and colour," Philos. Mag. **41**, 107 -120 (1871).
34] L.V. Lorenz, "Upon the light refracted by a transparent sphere," Vidensk. Selsk. Shrifter **6**, 1-62 (1890).
35] G. Mie, "Considerations on the optics of turbid media, especially colloidal metal sols," Ann. d. Physik **25**, 377-442 (1908).

36] R.T. Wang and H.C. van de Hulst, "Rainbows: Mie computations and the Airy approximation," Applied Optics **30**, 106-117 (1991).
37] C.F. Bohren and D.R. Huffman, *Absorption and Scattering of Light by Small Particles* (John Wiley & Sons, 1983).
38] A.L. Aden and M. Kerker, "Scattering of Electromagnetic Waves from Two Concentric Spheres," Journal of Applied Physics **22**, 1242-1246 (1951).
39] R. Bhandari, "Scattering coefficients for a multilayered sphere: analytic expressions and algorithms," Applied Optics **24**, 1960-1967 (1985).
40] S. Asano and G. Yamamoto, "Light Scattering by a Speriodal Particle," Applied Optics **14**, 29-49 (1975).
41] J.R. Wait, "Scattering of a Plane Wave from a Circular Dielectric Cylinder at Oblique Incidence," Can. J. Phys. **33**, 189-195 (1955).
42] H.C. van de Hulst, *Light Scattering by Small Particles* (Dover Publications, Inc., second edition, 1981).
43] P.M.A. Sloot, A.G. Hoekstra, and C.G. Figdor, "Osmotic Response of Lymphocytes Measured by Means of Forward Light Scattering: Theoretical Considerations," Cytometry **9**, 636-641 (1988).
44] A.G. Hoekstra, J.A. Aten, and P.M.A. Sloot, "Effect of aniosmotic media on the T-lymphocyte nucleus," Biophysical Journal **59**, 765-774 (1991).
45] G.J. Streekstra, A.G. Hoekstra, E.J. Nijhof, and R.M. Heethaar, "Light scattering by red blood cell in ektacytometry: Fraunhofer versus anomalous diffraction," Applied Optics **32**, 2266-2272 (1993).
46] G.J. Streekstra, A.G. Hoekstra, and R.M. Heethaar, "Anomalous diffraction as a tool for the characterisation of red blood cells in ektacytometry," in *proceedings of the 3rd International Congress on Optical Particle Sizing '93 - Yokohama*, Ed. M. Maeda, 1993, pp. 37-41.
47] D.W. Shuerman, *Light Scattering by Irregularly Shaped Particles* (Plenum Press, New York and London, 1980).
48] W.J. Wiscombe and A. Mugnai, "Scattering from nonspherical Chebyshev particles. 2: Means of angular scattering patterns," Applied Optics **27**, 2405-2421 (1988).
49] A. Mugnai and W.J. Wiscombe, "Scattering from nonspherical Chebyshev particles. 3: Variability in angular scattering patterns," Applied Optics **28**, 3061-3073 (1989).
50] A.G. Hoekstra and P.M.A. Sloot, "Dipolar Unit Size in Coupled-Dipole Calculations of the Scattering Matrix Elements," Optics Letters **18**, 1211-1213 (1993).
51] J.I. Hage and J.M. Greenberg, "A model for the optical properties of porous grains," Astrophysical Journal **361**, 251 (1990).
52] A.J. Hunt and D.R. Huffman, "A new polarisation-modulated light scattering instrument," Rev. Sci. Instrum. **44**, 1753-1762 (1973).
53] F. Kuik, P. Stammes, and J.W. Hovenier, "Experimental Determinations of Scattering Matrices of Water Droplets and Quartz Crystals," Applied Optics **30**, 4872-4881 (1991).
54] P.M.A. Sloot, A.G. Hoekstra, H. van der Liet, and C.G. Figdor, "Scattering matrix elements of biological particles measured in a flow through system: theory and practice," Applied Optics **28**, 1752-1762 (1989).
55] P.M.A. Sloot and A.G. Hoekstra, "Arbitrarily-shaped particles measured in Flow-through systems," in *proceedings of the 2nd international congress on optical particle sizing*, Ed. E.D. Hirleman, 1990, pp. 605-611.
56] W.S. Bickel and M.E. Stafford, "Biological particles as irregularly shaped scatterers," In *Light Scattering by Irregularly Shaped Particles* (Plenum Press, D.W. Shuerman Ed., 1980).
57] C. Bustamante, M.F. Maestre, and I. Tinoco, Jr, "Circular intensity differential scattering of light by helical structures. II. Applications," J. Chem. Phys. **73**, 6046-6055 (1980).
58] A.G. Hoekstra and P.M.A. Sloot, "New computational techniques to simulate Light Scattering from arbitrary particles," in *proceeding of the Third International Congress on Optical Particle Sizing*, 1993.
59] A.G. Hoekstra and P.M.A. Sloot, "Implementation of a parallel conjugate gradient method for simulation of elastic light scattering," In *Physics Computing '92* (World Scientific, J. Nadrchal and Groot, R. Ed., pp. 345-346, Singapore, 1993).
60] R.W. Hart and E.P. Gray , "Determination of particle structure from light scattering," J. Appl. Phys. **35**, 1408-1415 (1964).
61] O. Glater and M. Hofer, "Solution of the inverse scattering problem for small monodisperse nonspherical and inhomogenous particles," in *proceedings of the 3rd*

International Congress on Optical Particle Sizing '93 - Yokohama, Ed. M. Maeda, 1993, pp. 203-204.

62] G.C. Salzman, J.M. Crowell, J.C. Martin, T.T. Trujillo, A. Romero, P.F. Mullaney, and P.M. LaBauve, "Cell Classification by Laser Light Scattering: Identification and Separation of Unstained Leukocytes," Acta Cytologica **19**, 374-377 (1975).

63] P. Massoli, F. Beretta, A. D'Alessio, and L. M., "Temperature and size of single transparent droplets by light scattering in the forward and rainbow region," Applied Optics **32**, 3295-3301 (1993).

64] J.D. Jackson, *Classical Electrodynamics, second edition* (John Wiley & Sons, New York, Chichester, Brisbane, Toronto, Singapore, 1975).

65] E.S. Fry and G.W. Kattawar, "Relationships between elements of the Stokes matrix," Applied Optics **20**, 2811-2814 (1981).

66] J.W. Hovenier, H.C. van de Hulst, and C.V.M. van der Mee, "Conditions for the elements of the scattering matrix," Astron. Astrophys. **157**, 301-310 (1986).

67] J.W. Hovenier and C.V.M. van der Mee, "Simple characteristics of scattering matrices," in *proceedings of the 3rd International Congress on Optical Particle Sizing '93 - Yokohama*, Ed. M. Maeda, 1993, pp. 179-182.

68] R.M. Berkman, P.J. Wyatt, and D.T. Philips, "Rapid detection of penicillin sensitivity in Staphylococus aureus," Nature **228**, 458 (1970).

69] R.M. Berkman and P.J. Wyatt, "Differential light scattering measurements of heat-treated bacteria," Appl. Microbiol. **20**, 510 (1970).

70] P.J. Wyatt, "Differential light scattering: a physical method for identifying living bacterial cells," Applied Optics **7**, 1879 (1968).

71] P.J. Wyatt, "Identification of bacteria by differential light scattering," Nature **221**, 1257 (1969).

72] F. Michal and G.V.R. Born, "Effect of the Rapid Shape Change of Platelets on the Transmission and Scattering of Light through Plasma," Nature New Biology **231**, 220-222 (1971).

73] M.C. Benson, D.C. McDougal, and D.S. Coffey, "The Application of Perpendicular and Forward Light Scatter to Access Nuclear and Cellular Morphology," Cytometry **5**, 515-522 (1984).

74] E. Kraus and R. Niederman, "Changes in neutrophil right-angle light scatter can occur independent of alterations in cytoskeletal actin," Cytometry **11**, 272-282 (1990).

75] P. Latimer, D.M. Moore, and F. Dudley Bryant, "Changes in Total Scattering and Absorption Caused by Changes in Particle Conformation," J. Theoret. Biol. **21**, 348-367 (1968).

76] A. Brunsting, "Can light-scattering techniques be applied to flow-through cell analysis?," J. Histochem. Cytochem. **22**, 607-615 (1974).

77] A. Brunsting and P.F. Mullaney, "Differential light scattering from spherical mammalian cells," Biophysical Journal **14**, 439-453 (1974).

78] K.W. Keohane and W.K. Metcalf, "The cytoplasmic refractive index of lymphocytes, its significance and its changes during active immunisation," Quart. J. Exp. Physiol. Cognate Med. Sci. **44**, 343 (1959).

79] P.F. Mullaney and P.N. Dean, "The small angle light scattering of biological cells," Biophysical Journal **10**, 764-772 (1970).

80] A. Brunsting and P.F. Mullaney, "Light Scattering from Coated Spheres: Model for Biological Cells," Applied Optics **11**, 675-680 (1972).

81] R.A. Meyer and A. Brunsting, "Light scattering from nucleated biological cells," Biophysical Journal **15**, 191-203 (1975).

82] P.M.A. Sloot and C.G. Figdor, "Elastic light scattering from nucleated blood cells: rapid numerical analysis," Applied Optics **25**, 3559-3565 (1986).

83] D. Stramski, A. Morel, and A. Bricaud, "Modelling the light attenuation and scattering by spherical phytoplanktonic cells: a retrieval of the bulk refractive index," Applied Optics **27**, 3954-3956 (1988).

84] B.G. de Grooth, L.W.M.M. Terstappen, G.J. Puppels, and J. Greve, "Light-Scattering Polarisation Measurements as a New Parameter in Flow Cytometry," Cytometry **8**, 539-544 (1987).

85] K.J. Voss and E.S. Fry, "Measurement of the Mueller matrix for ocean waters," Applied Optics **23**, 4427-4439 (1984).

86] K.D. Loftus, M.S. Quinby-Hunt, A.J. Hunt, F. Livolant, and M. Meastre, "Light scattering by Prorocentrum micans: a new method and results," Applied Optics **31**, 2924-2931 (1992).

87] W.P. van de Merwe, D.R. Huffman, and B.V. Vonk, "Reproducibility and sensitivity

of polarised light scattering for identifying bacterial suspensions," Applied Optics **28**, 5052-5057 (1989).
88] C. Bustamante, M.F. Maestre, and I. Tinoco, Jr, "Circular intensity differential scattering of light by helical structures I. Theory," J. Chem. Phys. **73**, 4273-4281 (1980).
89] C. Bustamante, I. Tinoco, Jr, and M.F. Maestre, "Circular intensity differential scattering of light by helical structures. III. A general polarizability tensor and anomalous scattering," J. Chem. Phys. **74**, 4839-4850 (1981).
90] C. Bustamante, I. Tinoco, Jr, and M.F. Maestre, "Circular intensity differential scattering of light. IV. Randomly oriented species," J. Chem. Phys. **76**, 3440-3446 (1982).
91] C. Bustamante, M.F. Maestre, D. Keller, and I. Tinoco, Jr, "Differential scattering (CIDS) of circularly polarised light by dense particles," J. Chem. Phys. **80**, 4817-4823 (1984).
92] M.F. Maestre, C. Bustamante, T.L. Hayes, J.A. Subirana, and I. Tinoco, Jr, "Differential scattering of circularly polarised light by the helical sperm head from the octopus Eledone Cirrhosa," Nature **298**, 773-774 (1982).
93] S. Zietz, A. Belmont, and C. Nicolini, "Differential scattering of circularly polarised light as a unique probe of polynucleosome superstructures," Cell Biophysics **5**, 163-187 (1983).
94] H. Begemann and J. Rastetter, *Atlas of clinical hematology* (Springer Verlag, New York, 1979).
95] L.W.M.M. Terstappen, B.G. de Grooth, W. van Berkel, C.H.H. ten Napel, and J. Greve, "Application of Orthogonal Light Scattering for Routine Screening of Lymphocyte Samples," Cytometry **9**, 220-225 (1988).
96] P.H. Bartels, G.B. Olson, J.M. Layton, R.E. Anderson, and G.L. Wied, "Computer Discrimination of T and B Lymphocytes," Acta Cytologica **19**, 53-57 (1975).
97] A. van Es, C.J.L.M. Meyer, P.J. Oljans, and H.J. Tanke, "Mononuclear cells in renal allografts: correlation with peripheral blood T-Lymphocyte subpopulations and graft prognosis," Transplantation **37**, 134-139 (1984).
98] P.M.A. Sloot, E.H.M. van der Donk, and C.G. Figdor, "Computer assisted centrifugal elutriation II: Multiparametric statistical analysis," Comp. Meth. Prog. Biomed. **27**, 37 (1988).
99] M.A. van Dilla, P.N. Dean, O.D. Laerum, and M.R. Melamed, *Flow Cytometry: Instrumentation and Data Analysis* (Academic Press, Analytical Cytology, 1985).
100] A.G. Hoekstra, R.M.P. Doornbos, K.E.I. Deurloo, H.J. Noordmans, B.G. de Grooth, and P.M.A. Sloot, "Another face of Lorenz-Mie scattering: monodisperse distributions of spheres produce lissajous-like patterns," Applied Optics **33**, 494-500 (1994).
101] L.W.M.M. Terstappen, B.G. de Grooth, K. Visscher, F.A. van Kouterik, and J. Greve, "Four-Parameter White Blood Cell Differential Counting Based on Light Scattering Measurements," Cytometry **9**, 39-43 (1988).
102] L.W.M.M. Terstappen, R.A. Mickaels, R. Dost, and M.R. Loken, "Increased Light Scattering Resolution Facilitates Multidimensional Flow Cytometric Analysis," Cytometry **11**, 506-512 (1990).
103] L.W.M.M. Terstappen, B.G. de Grooth, G.M.J. Nolten, C.H.H. ten Napel, W. van Berkel, and J. Greve, "Physical Discrimination Between Human T-Lymphocyte Subpopulations by Means of Light Scattering, Revealing Two Populations of T8-Positive Cells," Cytometry **7**, 178-183 (1986).
104] L.W.M.M. Terstappen, B.G. De Grooth, C.H.H. Ten Napel, W. Van Berkel, and J. Greve, "Discrimination of human cytoxic lymphocytes from regulatory and B-lymphocytes by orthogonal light scattering," Journal of Immunological Methods **95**, 211-216 (1986).
105] C.C. Stewart, S.J. Stewart, and R.C. Habbersett, "Resolving leukocytes using axial light loss," Cytometry **10**, 426-432 (1989).
106] R.P. Gale and K.A. Foon, "Chronic Lymphocytic Leukemia, Recent Advances in Biology and Treatment," Annals of Internal Medicine **103**, 101-120 (1985).
107] G.J. den Ottolander, H.R.E. Schuit, J.L.M. Waayer, L. Huibregtsen, W. Hijmans, and J. Jansen, "Chronic B-Cell Leukemias: Relation between Morphological and Immunological Features," Clinical Immunology and Immunopathology **35**, 92-102 (1985).
108] D.R. van Bockstaele, Z.N. Berneman, and M.E. Peetermans, "Flow Cytometric Analysis of Hairy Cell Leukemia Using Right-Angle Light Scatter," Cytometry **7**, 217-220 (1986).
109] L.W.M.M. Terstappen, B.G. de Grooth, W. van Berkel, C.H.H. ten Napel, and J. Greve, "Abnormal distribution of CD8 subpopulation in B-Chronic lymphocytic leukemia

identified by flow cytometry," Leuk. Res. **12**, 551-557 (1988).
110] L.W.M.M. Terstappen, B.G. de Grooth, W. van Berkel, C.H.H. ten Napel, and J. Greve, "Flow cytometric characterisation of chronic lymphocytic leukemias using orthogonal light scattering and quantitative immunofluorescence," Blut **56**, 201-208 (1988).
111] C.G. Figdor, *Separation of human leukocytes by physical methods*, Ph.D. dissertation, University of Amsterdam, Amsterdam, the Netherlands, 1982.
112] P.M.A. Sloot, M.J. Carels, P. Tensen, and C.G. Figdor, "Computer-assisted centrifugal elutriation. I. Detection system and data acquisition equipment," Comp. Meth. Prog. Biomed. **24**, 179-188 (1987).
113] P.M.A. Sloot, M.J. Carels, and A.G. Hoekstra, "Computer Assisted Centrifugal Elutriation," in *Computing Science in the Netherlands* , 1989, pp. 441-462.
114] R.G. Belleman, P.M.A. Sloot, and L.O. Hertzberger, "Multivariate data processing system: Transputer based data aquisition analysis and presentation," in *Parallel Computing and Transputer Applications*, Ed. M. Valero, E. Onate, M. Jane, J.L. Larriba, and B. Suarez, IOS press, 1992.
115] P.W. Barber and C. Yeh, "Scattering of electromagnetic waves by arbitrarily shaped dielectric bodies," Applied Optics **14**, 2864 (1975).
116] E.M. Purcell and C.R. Pennypacker, "Scattering and absorption of light by nonspherical dielectric grains," The Astrophysical Journal **186**, 705-714 (1973).
117] M. Born and E. Wolf, *Principles of Optics* (Pergamon Press, 3 rd ed., 1965).
118] B.T. Draine, "The discrete dipole approximation and its application to interstellar graphite grains," Astrophys. J. **333**, 848-872 (1988).
119] G.H. Goedecke and S. O'Brien, "Scattering by irregular inhomogeneous particles via the digitised Green's function algorithm," Applied Optics **27**, 2431-2438 (1988).
120] C.E. Dungey and C.F. Bohren, "Light scattering by nonspherical particles: a refinement to the coupled-dipole method," J. Opt. Soc. Am. **8**, 81-87 (1991).
121] J.I. Hage, M. Greenberg, and R.T. Wang, "Scattering from arbitrarily shaped particles: theory and experiment," Applied Optics **30**, 1141-1152 (1991).
122] A. Lakhtakia, "Macroscopic theory of the coupled dipole approximation method," Opt. Comm. **79**, 1-5 (1990).
123] A. Lakhtakia, "Strong and weak forms of the method of moments and the coupled dipole method for scattering of time-harmonic electromagnetic fields," Int. J. Mod. Phys. C **3**, 583-603 (1992).
124] A. Lakhtakia, "General theory of the Purcell-Pennypacker scattering approach and its extension to bianisotropic scatterers," Astrophysical Journal **394**, 494-499 (1992).
125] S.B. Singham, "Intrinsic optical activity in light scattering from an arbitrary particle," Chem. Phys. Letters **130**, 139-144 (1986).
126] Y.L. Yung, "Variational principle for scattering of light by dielectric particles," Applied Optics **17**, 3707-3709 (1978).
127] M.R. Hestenes and E. Stiefel, "Methods of Conjugate Gradients for solving linear systems," Nat. Bur. Standards J. Res. **49**, 409-436 (1952).
128] S.F. Ashby, T.A. Manteuffel, and P.E. Saylor, "A taxonomy for conjugate gradient methods," SIAM J. Numer. Anal. **27**, 1542-1568 (1990).
129] S.D. Druger, M. Kerker, D.S. Wang, and D.D. Cooke, "Light scattering by inhomogeneous particles," Applied Optics **18**, 3888-3889 (1979).
130] S.B. Singham and G.C. Salzman, "Evaluation of the scattering matrix of an arbitrary particle using the coupled dipole approximation," J. Chem. Phys. **84**, 2658-2667 (1986).
131] M.K. Singham, S.B. Singham, and G.C. Salzman, "The scattering matrix for randomly oriented particles," J. Chem. Phys. **85**, 3807-3815 (1986).
132] S.B. Singham, "Theoretical factors in modelling polarised light scattering by arbitrary particles," Applied Optics **28**, 5058-5064 (1898).
133] P. Chiappetta, "Multiple scattering approach to light scattering by arbitrarily shaped particles," J. Phys. A: Math. Gen. **13**, 2101-2108 (1980).
134] S.B. Singham and C.F. Bohren, "Light scattering by an arbitrary particle: a physical reformulation of the coupled dipole method," Optics Letters **12**, 10-12 (1987).
135] S.B. Singham and C.F. Bohren, "Light scattering by an arbitrary particle: the scattering-order formulation of the coupled-dipole method," J. Opt. Soc. Am. A **5**, 1867-1872 (1988).
136] J. Stoer and R. Bulirish, *Introduction to Numerical Analysis* (Springer-Verlag, New York, Heidelberg, Berlin, 1980).
137] P.J. Flatau, G.L. Stephens, and B.T. Draine, "Light scattering by rectangular solids in the discrete-dipole approximation: a new algorithm exploiting the Block-Toeplitz structure," J. Opt. Soc. Am. A **7**, 593-600 (1990).

138] P.R. Shapiro, "Interstellar polarisation: magnite dust," Astroph. J. **201**, 151-164 (1975).
139] J. Frey, J.J. Pinvidic, R. Botet, and R. Jullien, "Light Scattering by fractal aggregates: a numerical investigation," J. Phys. France **49**, 1969-1976 (1988).
140] P. Chiappetta and B. Torresani, "Electromagnetic scattering from a dielectric helix," Applied Optics **27**, 4856-4860 (1988).
141] R. Haracz, L.D. Cohen, A.R.W. Presley, and A. Cohen, "Scattering of linearly polarised microwave radiation from a dielectric target including the interaction between target elements," Applied Optics **28**, 1338-1344 (1989).
142] M.A. Taubenblatt, "Light scattering form cylindrical structures on surfaces," Optics Letters **15**, 255-257 (1990).
143] M.A. Taubenblatt and T.K. Tran, "Calculation of light scattering from particles and structures on a surface by the Coupled Dipole method," J. Opt. Soc. Am. A **10**, 912-919 (1993).
144] P.J. Flatau, K.A. Fuller, and D.W. Mackowski, "Scattering by two spheres in contact: comparisons between discrete-dipole approximation and modal analysis," Applied Optics **32**, 3302-3305 (1993).

Chapter 2

THE EFFECT OF ANIOSMOTIC MEDIA ON THE VOLUME OF THE T-LYMPHOCYTE NUCLEUS

Alfons G. Hoekstra, Jacob A. Aten and Peter M.A. Sloot,

Biophysical Journal **59**, 1991, p. 765-774.

The effect of aniosmotic media on the volume of the T-lymphocyte nucleus

ABSTRACT

Time resolved measurements of the nuclear volume response of human peripheral T-lymphocytes, under aniosmotic conditions, are presented. In the experiments slit scanning flowcytometry methods were used. We propose an extension to the standard solid viscoelastic model to interpret the observed dynamical behaviour of the nucleus. It is shown that experimental and theoretical evidence indicates a passive nuclear response merely induced by a mechanical link (i.e. the cytoskeleton) between the cell membrane and the nuclear envelope. Implications of this work to the field of cellular mechanics and cytoskeletal rheology are surveyed.

2.1 INTRODUCTION

Recently it was shown experimentally that the light scattered by lymphocytes in the forward directions in aniosmotic media, varies inversely with cell volume [1]. These results appear to be inconsistent with the observation that low-angle light scattering is proportional to particle size [2]. We solved this paradox by including changes in the optical density of the cell, induced by variations in the osmolarity of the solution, in a light scattering model [3]. A crucial point in this model is the assumption of a nuclear volume response directly related to the cytoplasmic volume response. It is well known that *isolated* nuclei do not respond to aniosmotic conditions [4]. On the other hand it is not clear whether nuclei of *intact* cells can respond to these conditions. Furthermore, if the nuclei of intact cells respond to aniosmotic conditions, as predicted by the light scattering theory, then the mechanism of this response in not clear. We therefore hypothesised a physical link between the cytoplasmic membrane and the nuclear envelope (the cytoskeletal network) resulting in an induced nuclear volume response.

The response of lymphocytes in aniosmotic media has been studied extensively. It was noted recently that the behaviour of human B- and T-lymphocytes in hypoosmotic media is completely different. After suspension in a hypoosmotic medium the total cell volume response of T-lymphocytes consists of two steps. First, a fast initial volume increase and next, for cells suspended in a NaCl solution, a slow decrease in volume back to the initial isoosmotic value [5]. In KCl solutions, the initial fast increase in cell volume is followed by a much slower increase in cell volume. The first step is interpreted as an

osmotic flow of H_2O through the membrane into the cell. The increase in volume is well described by the Boyle van 't Hoff relation [1]. The second, volume regulatory step, is explained by the activation of potassium and chloride pathways [5, 6]. Human B-lymphocytes in hypoosmotic NaCl solutions show the initial (fast) volume increase, but in contrast to T-lymphocytes remain in a swollen state [6, 7].

Under hyperosmotic conditions, (T- and B-) lymphocytes decrease in volume but are not capable of any volume regulation [8, 9]. This volume decrease also follows the Boyle van 't Hoff relation [1, 8].

Apart from a total cell volume response, analysis of Transmission Electron Microscopy recordings of lymphocytes also revealed a *nuclear* response [10]. However, no attention was paid to the sometimes complex total cell volume response of lymphocytes as a function of time, which might affect the nuclear response. The question why the nucleus responds to aniosmotic conditions was also not addressed.

In this study we introduce time resolved measurements of the nuclear volume response of human peripheral T-lymphocytes, after suspension in both hypoosmotic- and hyperosmotic media (with dominant ion K^+ or Na^+).

A mechanistic interpretation of the nuclear response is presented. We modelled the response of the nuclei as a passive phenomenon, induced by a mechanical link between cell membrane and nuclear envelope. The data are fitted to an extended standard solid viscoelastic model. The nature of the mechanical link is discussed in terms of the cytoskeleton.

2.2 MATERIALS AND METHODS

2.2.1 Cell handling

2.2.1.1 Isolation of cells

Large amounts of human peripheral T-lymphocytes were obtained by Computer Assisted Centrifugal Elutriation (CACE), as described in detail by Sloot et al. and Figdor et al. [11, 12].

2.2.1.2 Preparation of suspending media

The Phosphate-Buffered Saline (PBS, in this paper referred to as Na^+ medium) was prepared according to Dulbecco and Vogt [13]. Phosphate-Buffered potassium medium (PBK, this paper: K^+ medium) is identical to Na^+ medium with the sodium and potassium interchanged, and the calcium concentration halved (see also [5]).

A stock solution of 10x isoosmotic K^+ - and Na^+ medium was prepared. This solution was diluted with distilled water to obtain media with the desired osmolarity. The isoosmotic osmolarity Π_0 equals 285 mOsm/l. The aniosmotic media are characterised by Π_0/Π, with Π the osmolarity of the aniosmotic medium. If Π_0/Π is > 1, the medium is hypoosmotic; for $\Pi_0/\Pi < 1$ the medium is

hyperosmotic.

The osmolarity of the media was measured by means of a freezing point depression method (Knauer Halbmikro-Osmometer).

2.2.1.3 Stock suspension

Freshly isolated T-lymphocytes were resuspended in isoosmotic Na^+ medium at a concentration of 10^6 cells/ml. We refer to this as the stock cell suspension.

2.2.1.4 Staining

For the measurement of the nuclear diameter in the slit scan flowcytometer (see next section), the nucleus of the cells was stained with the fluorescent dye Hoechst 33342 (Sigma, St Louis, Mo, USA). Hoechst 33342 (Hö) was added to 200 µl stock cell suspension to a final concentration of 2 µg/ml Hö, and incubated at room temperature for 20 minutes.

As a control experiment the total cell volume response in an aniosmotic medium was measured (with the coulter counter, see next section) in the absence and presence of Hö. Both responses were equal.

In one experiment the total cell volume was also measured in the flowcytometer. The cells were stained with carboxyfluoresencediacetate (cFDA [14]). A stock solution of 20 mg/ml cFDA in acetone was prepared; 5 µl of this stock solution was added to 2.5 ml isotonic Na^+ medium, with 250 µl stock cell suspension and incubated for 15 minutes at 37^0 C. After incubation the cells were washed twice with isotonic Na^+ medium and resuspended in 50 µl isotonic Na^+ medium. Under the fluorescence microscope, the cells showed a bright, homogeneous fluorescence. For times extending beyond about 15 minutes after staining however, the cells developed protuberances and showed bleaching of the fluorescence.

2.2.1.5 Resuspension of cells in aniosmotic media

The cells were resuspended in the aniosmotic media in two different ways, depending on the method of analysis used (see next section). In case of the coulter measurements (total cell volume measurement), 100 µl of stock cell suspension was added to 10 ml of aniosmotic medium.

The procedure for the slit scan flowcytometry experiments was as follows: after staining, the cells were centrifuged at 200 g for 3 minutes, the supernatant was removed and the cells were resuspended in 250 µl aniosmotic medium, with 2 µg/ml Hö added to maintain a constant level of staining (this was not necessary for cFDA staining). Addition of low concentrations of Hö did not affect the osmolarity of the media. Immediately after resuspension (within 10-20 seconds), the measurements on the cell suspensions were started.

2.2.2 Measuring the total cell volume

The total cell volume was obtained by electronic sizing with a coulter counter model ZF connected to a coulter channelyzer model C 1000. Immediately after the resuspension of cells in the aniosmotic media, the cells were transported to the coulter counter and after ± 1 minute the first measurement was performed. Typically about 20,000 cells were sized in approximately 20 seconds. The time between two measurements was 2 minutes.

After sizing of the cells, a hardcopy of the volume distribution was produced and digitised with a Nummonics 1220 digitiser into memory of a Hewlett Packard HP 9816 computer. The mean value of the volume distribution V_m was calculated. As lymphocytes show a homogeneous response to an osmotic shock, V_m is a good parameter to represent the volume of the cell distribution [5].

Since we measure volumes in aniosmotic media (V_{cell}) relative to the isoosmotic volume $V(0)_{cell}$ by

$$\left[\frac{V(t)}{V(0)}\right]_{cell} = \frac{V_m(t)}{V_m(0)} \quad , \qquad [1]$$

with t the time after the osmotic shock and $V_m(0)$ the mean volume of the isoosmotic volume distribution, an absolute calibration of the apparatus is not necessary [15, 16].* Here it is assumed that the electronic shape factor is constant for lymphocytes as a function of the osmolarity of the suspending medium [15].

2.2.3 Measuring the nuclear volume

Two techniques were applied to determine the nuclear volume: fluorescence microscopy and flowcytometry.

2.2.3.1 Fluorescence microscopy

Following the resuspension of the cells, a drop of stained cells in ani- or isoosmotic medium was placed on an objective glass and immediately covered to prevent water evaporation and subsequent osmolarity changes. The cells were examined under a Zeiss microscope with 40x objective -, 10x ocular magnification and UV illumination. The cells were photographed and the negatives were further analysed on the Nummonics 1220 digitiser. Absolute calibration of the measured nuclear diameters was not necessary since the nuclear diameters in aniosmotic media are measured relative to the isoosmotic

---

* The measured voltage pulse height Δv as a function of volume V of a particle in the Coulter Counter is $\Delta v = K\rho i V$, with i the constant current through the orifice, r the electric resistivity of the medium and K the electronic shapefactor dependent on the particle shape and orifice geometry. The electric resistivity of the media is a function of the osmolarity, therefore particles of the same volume in different media produce different Δv. This source of error was investigated by measuring 10 μm polystyrene spheres (Duke Scientific) in media with Π_0/Π = 0.5 and 1.5. The measured $V_m/V(0)_m$ was 0.98 and 1.00 for Π_0/Π = 0.5 and 1.5 respectively. This small systematic error is considered insignificant.

values.

2.2.3.2 Flowcytometry

The stained cell suspensions were analysed using a cytofluorograph 30 FlowCytometer (Becton Dickinson, Mountain View, CA, USA) with a Spectra Physics (Mountain View, CA, USA) 2000 Argon ion laser tuned to the UV. The measuring principle is drawn schematically in figure 1. Special purpose optics, using crossed cylindrical lenses, was applied to focus the laser beam at the centre of the quartz flow channel. The laser beam in the focal plane is line shaped to produce a 'slit' of laser illumination through which the stained cells flow (see e.g. [17], chapter 2). After hydrofocusing [17] the cells pass the focal plane one by one. The dye is excited by the intense UV beam and the resulting fluorescence signal is measured, as a function of time. The width of the fluorescence pulse is a measure of the nuclear diameter.

The fluorescence signal of the illuminated cell nuclei was detected through a 490 nm band pass filter by a photomultiplier. The fluorescence pulse shape was obtained by digitising the signals from the photomultiplier at a rate of 10 Mhz and storing into the memory of a dedicated computer system for slit scanning analysis. The values of the total (integrated) fluorescence intensity and the width of the fluorescence pulse (defined as the full width at half maximum; fwhm) from individual particles are stored in a 2-dimensional histogram [18]. These modes of operation are shown schematically in figure 2.

Approximately 500 cells per second are analysed. After 15 seconds the system has sampled enough cells to allow reliable off line statistical analysis of the 2-dimensional histogram. The histogram is stored and the system starts filling a new histogram. Typically 15 subsequent histograms were measured.

From the 2-dimensional histogram a (1-dimensional) pulse width histogram is calculated by integration over the total intensity. An average pulse width is obtained by calculating the first moment of the pulse width histogram. This value is used to calculate the nuclear diameter.

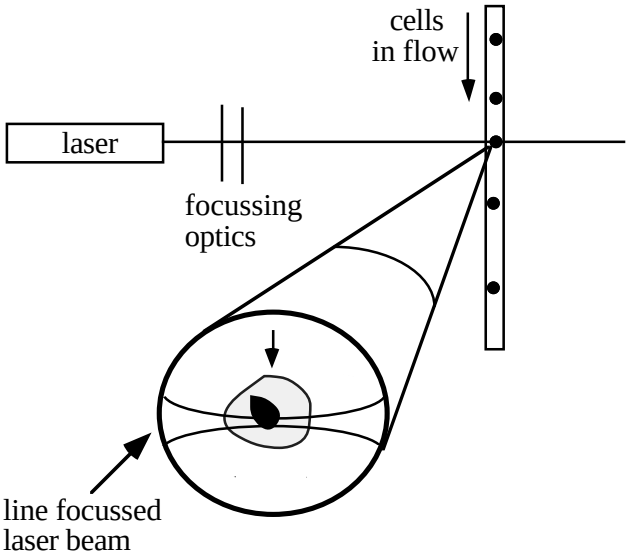


Figure 1: A schematic drawing of the slit scan principle. Hydrofocused cells in a flow pass

the line focused laser beam one by one. The stained nucleus produces a fluorescence signal. The width of the fluorescence pulse is a measure of the nuclear diameter.

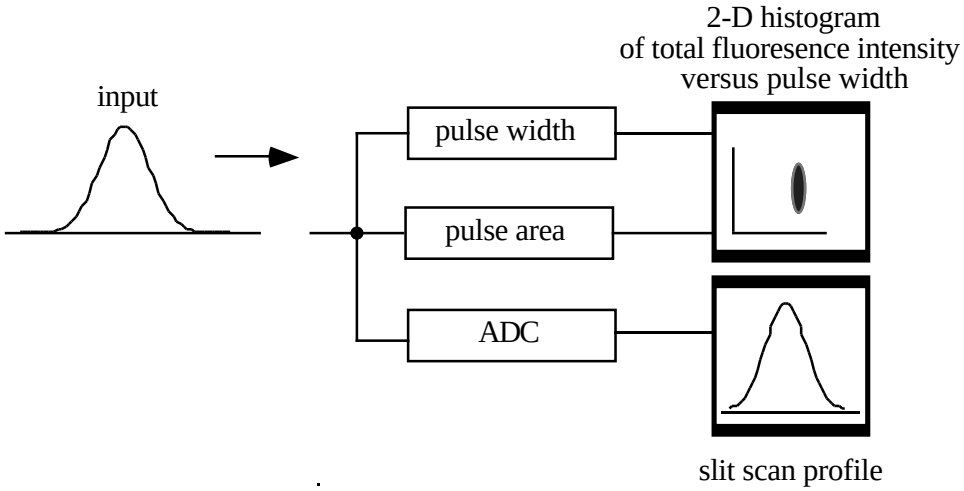


Figure 2: Modes of operation of the data aquisition equipment. In the experiments described in this paper the 2-dimensional histogram is measured and further analysed to obtain an average pulse width. The pulse area box measures the total (integrated) fluorescence intensity; the ADC box measures pulse profiles and the pulse width box measures the width of the fluorescence pulse.

The measured fluorescence intensity I(t) as a function of time t (i.e. the pulse profile) is a convolution of the nuclear fluorescence emission distribution c(x,y,z) with the laser irradiance profile H(z) in the focal plane [17]:

$$I(vt) = \int c(x, y, vt - \xi)\, H(\xi)\, d\xi dxdy\ , \qquad [2]$$

where v is the velocity of cell flow along the z axis. After deconvolution of I(t) with H(z), the nuclear profile C(z) is found, where

$$C(z) = \int c(x, y, z)\, dxdy\ . \qquad [3]$$

Assume that the nucleus is a sphere and that c(x,y,z) is homogeneous. It can be shown that C(z) is, to third order in z, a Gaussian function with width $1/2\sqrt{2}D$, where D is the diameter of the nucleus. The laser intensity H(z) is also a Gaussian function. Convolution of two Gaussian functions yields a new Gaussian function. The variance of this new function is the sum of the variances of the original functions. Therefore, the width of C(z) is:

$$\text{width}(C(z)) = \sqrt{(\text{width}(I(vt)))^2 - (\text{width}(H(z)))^2}\ . \qquad [4]$$

In our system the nuclear diameter of the T-lymphocytes was approximately a factor of 4 larger than the width of the focused laser beam. Therefore the measured pulse width is, within a few per cent, equal to $1/2\sqrt{2}D$.

The volume of the nucleus $V_N(t)$ at time t, relative to the isotonic value $V_N(0)$ can now be calculated from

$$\frac{V_N(t)}{V_N(0)} = \left[\frac{w_m(t)}{w_m(0)}\right]^3, \qquad [5]$$

where w_m is the mean of the pulse width distribution and the nucleus of the lymphocyte is assumed to be a sphere [10]. The linearity of w_m with particle diameter was verified by introducing test spheres of different diameter in the system. The measured w_m was proportional to the test spheres diameter (data not shown).

2.3 RESULTS

In this section the measurements of the nuclear volume response of human peripheral T-lymphocytes in aniosmotic media are presented. This response was measured in both hypo- and hyperosmotic conditions for two different media (Na^+ and K^+ media), as a function of time ($0 < t < 6$ minutes). The total cell volume response depends on these parameters. Since the volume regulatory phase in hypoosmotic medium is not observed in hyperosmotic medium and the dominant ion in the hypoosmotic medium determines the nature of the volume regulation, we expect the nuclear volume response to depend also on these parameters.

The first step was to assess if, under various aniosmotic conditions, a nuclear volume response could be observed at all. Furthermore, in this pilot experiment, the results of the fluorescence microscopy and the flowcytometry have been compared.

We prepared two aniosmotic solutions (hyperosmotic Na^+ medium with $\Pi_0/\Pi = 0.8$ and hypoosmotic Na^+ medium with $\Pi_0/\Pi = 1.4$) and isoosmotic Na^+ medium ($\Pi_0 = 285$ mOsm/l). Freshly isolated lymphocytes were stained as described in the previous section, and resuspended in the isoosmotic- and the two aniosmotic media. After 15 minutes, cells were introduced into the flowcytometer and the nuclear diameter was measured. This procedure was repeated 4 times for all media. The nuclear diameter in isoosmotic medium remains constant over a long period of time. The results are shown in table 1.

		Π_0/Π	
		0.8	1.4
V/V_0	FCM	0.77 (0.02)	1.07 (0.02)
	FM	0.7 (0.1)	1.3 (0.3)

Table 1: The nuclear volume response after resuspension of T-lymphocytes in a

hyperosmotic - and hypoosmotic Na^+ medium, measured by fluorescence microscopy (FM) and flowcytometry (FCM). The FCM data are the average of 4 independent measurements per medium. The FM data are based on the measurement of 80 nuclei per medium. The numbers in brackets are standard deviations.

The fluorescence microscopy data are also tabulated in table 1; 5 minutes after resuspension the cells were placed under the microscope and photographed. The nuclei appeared to be round. In all cases the diameters of 80 nuclei were measured.

A nuclear volume response is unambiguously observed, both by pulse width flowcytometry analysis and by fluorescence microscopy. Since flowcytometry allows analysis of much more cells than fluorescence microscopy the standard deviation on flowcytometry results is much smaller than those for the fluorescence microscopy. Therefore, in the following only flowcytometry has been applied to obtain the nuclear volume.

Next, the total cell volume response and the nuclear volume response were measured simultaneously as a function of time. Four aniosmotic solutions were prepared; two K^+ - and two Na^+ media, both with $\Pi_0/\Pi = 0.5$ and $\Pi_0/\Pi = 1.5$ respectively, and isoosmotic K^+ and Na^+ media. A freshly isolated pool of lymphocytes was divided in two parts. The first part was used to measure the total cell volume response, whereas the second part was used to measure the nuclear volume response (by means of coulter counter and flowcytometry respectively). Both experiments were repeated four times for all the four aniosmotic media and the two isoosmotic media. The averaged results are shown in figure 3a-d.

First we note that the behaviour of the total cell volume response of the T-lymphocytes is as expected. In hyperosmotic medium the volume decreases fast to an end value. After this fast response no further volume changes occur (coulter counter measurements at t = 30 minutes also reveal no further volume change, data not shown). The total cell volume response in hypoosmotic medium clearly shows the two steps. A fast increase in volume is followed by the volume regulatory phase. The decrease in volume in Na^+ medium is somewhat slow compared to experiments reported in literature [5], but eventually (after ± 50 minutes) the volume returns to the initial isoosmotic value (data not shown).

The measured nuclear response, under the four different aniosmotic conditions, shows resemblance to the total cell volume response. In both K^+ - and Na^+ hyperosmotic medium the nuclear volume reaches a plateau after ±1 minute but the relative volume decrease of the nucleus is much less than the total cell volume decrease (Na^+ medium: $[V/V_0]_{nucleus} \sim 0.85$ versus $[V/V_0]_{cell} \sim 0.6$; K^+ medium: $[V/V_0]_{nucleus} \sim 0.9$ versus $[V/V_0]_{cell} \sim 0.5$). Furthermore it can be seen from figure 3b that in hyperosmotic K^+ medium the nucleus shows a tendency to increase in volume for $t > 3$ minutes.

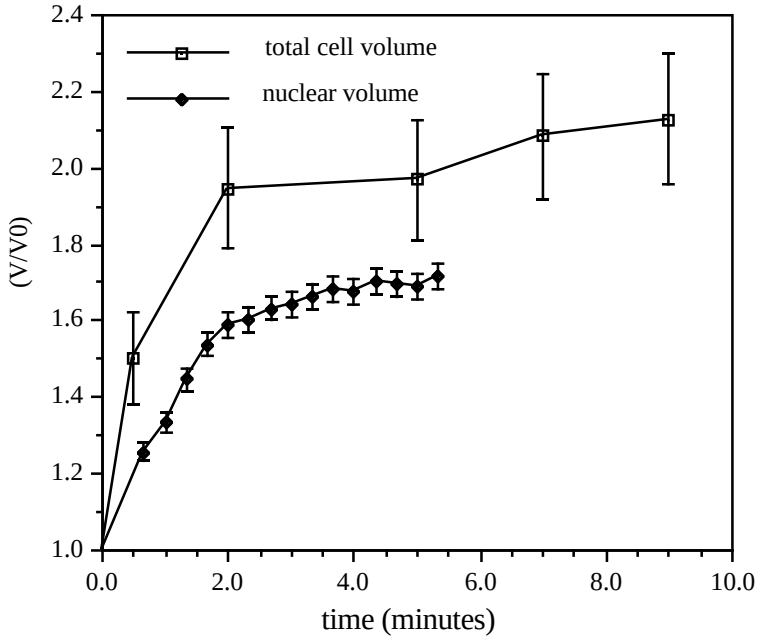


Figure 3a: The total cell volume response and the nuclear volume response as a function of time, after resuspension of cells at t = 0 minutes; K^+ medium, Π_0/Π = 1.5. The error bars indicate the standard deviation of 4 independent measurements

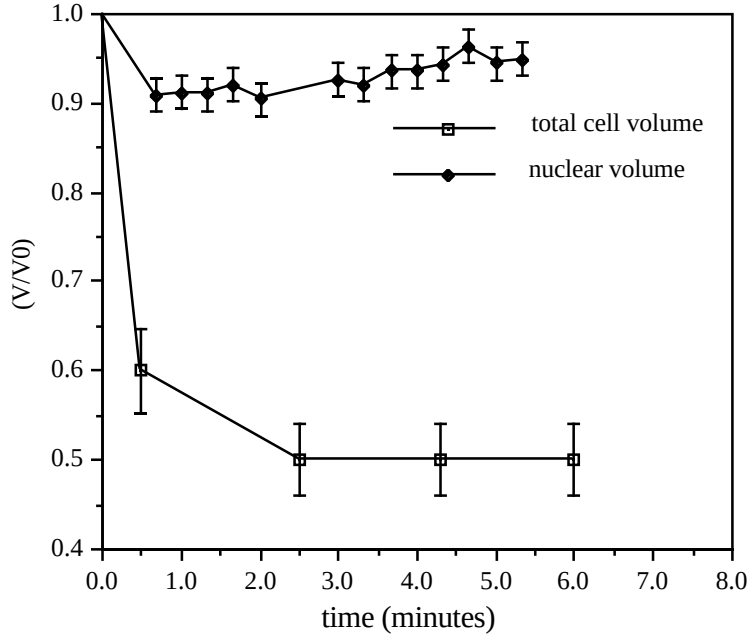


Figure 3b: The total cell volume response and the nuclear volume response as a function of time, after resuspension of cells at t = 0 minutes; K^+ medium, Π_0/Π = 0.5. The error bars indicate the standard deviation of 4 independent measurements

The nuclear volume response under hypoosmotic conditions is much more pronounced. The fast increase in volume, followed by a volume regulatory phase, is observed in both hypoosmotic media. The first part of the increase in nuclear volume ($0 < t < 2$ minutes) is much slower (order of minutes) with respect to the total cell volume (order of seconds). After ±2 minutes in K^+ medium the rate of volume increase of the total cell volume and the nuclear

volume are approximately the same. For longer times the nuclear volume seems to reach a plateau value, whereas the total cell volume continues to increase.

On the other hand, in hypoosmotic Na^+ medium the increase in nuclear volume is followed by a very slow (that is, slower then the total cell volume) decrease in volume.

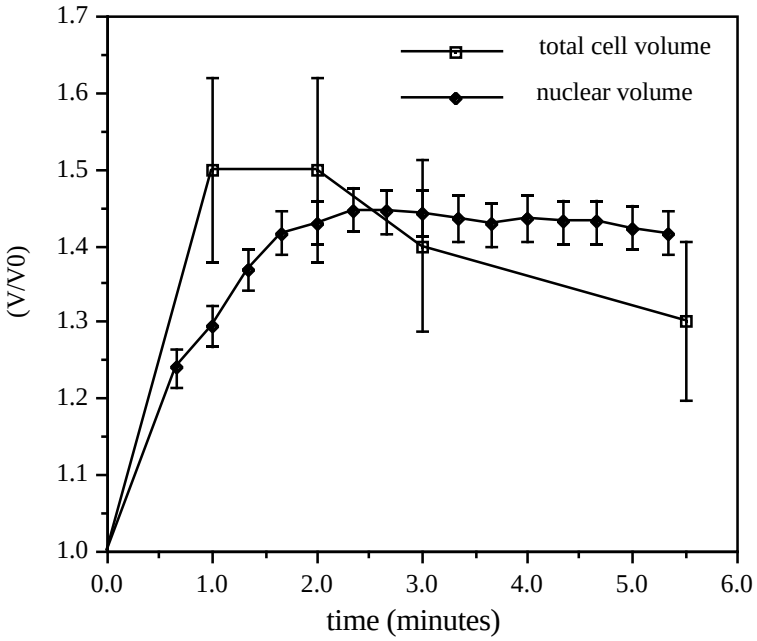


Figure 3c: The total cell volume response and the nuclear volume response as a function of time, after resuspension of cells at t = 0 minutes; Na^+ medium, Π_0/Π = 1.5. The error bars indicate the standard deviation of 4 independent measurements

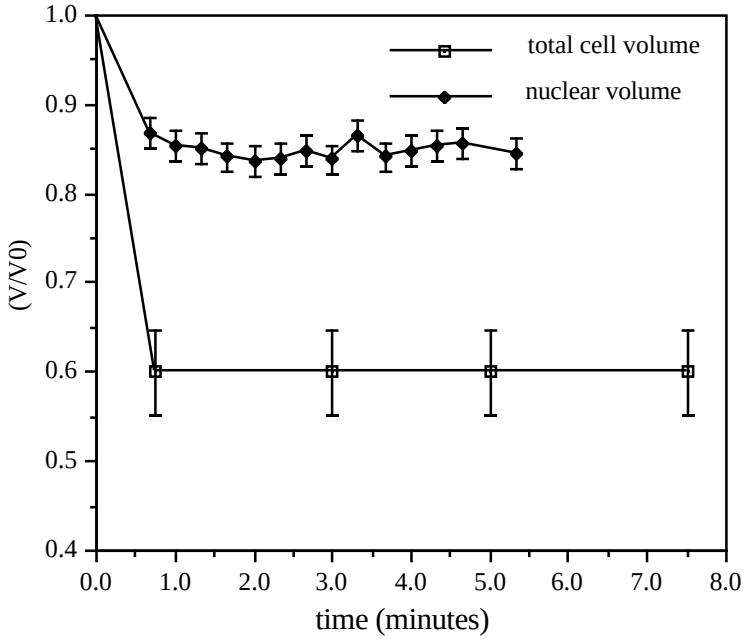


Figure 3d: The total cell volume response and the nuclear volume response as a function of time, after resuspension of cells at t = 0 minutes; Na^+ medium, Π_0/Π = 0.5. The error bars indicate the standard deviation of 4 independent measurements

2.4 DISCUSSION

The response of lymphocytes after suspension in aniosmotic media has received much attention over the past years, especially the details of the volume regulatory phase and its potential as a diagnostic tool are extensively studied [5, 6]. The response of intracellular structures, such as the nucleus, has barely been investigated.

Relying on a physical (light scattering) model, we predicted a nuclear response [3] although it is known that isolated nuclei, due to large pores in the nuclear membrane, do not behave as osmometers [4]. We have demonstrated in Table 1 and figure 3 that the nuclei of intact lymphocytes *do* respond to several aniosmotic conditions.

In this section the mechanism of the nuclear volume response will be discussed. It will be shown that the response is a passive viscoelastic response, and the role of the cytoskeleton in this response will be discussed. Finally, a possible link between this work and the field of cellular mechanics and cytoskeletal rheology is surveyed.

Across the cell membrane a chemical potential gradient is the driving force of the fast total cell volume response. The nucleus of intact eukariotic cells has large pores to allow the active transport of macromolecules [19]. As a consequence, a chemical potential gradient across the nuclear membrane can not exist. Another possible mechanism for a nuclear response is a conformational change of chromatin in the nucleus due to variations in ion concentration. The volume of isolated nuclei of rat liver cells varies with Mg^{++} concentrations. However, this effect is only observed for divalent cations [20, 21]. The nucleus of intact cells, like isolated nuclei, cannot respond in an active way to variations in osmotic pressure, induced by K^+ or Na^+ concentration changes.

We suggested in earlier work that the fibrillar networks inside the cell might regulate the changes in size of the nucleus [3] i.e. that the nuclear response is induced via a mechanical link between the cell membrane and the nuclear envelope, and that the cytoskeleton provides this mechanical link.

The main components of the cytoskeleton (microfilaments, microtubules and intermediate filaments) occur in the cytoplasm of lymphocytes [22]. Dense microfilamentous networks are observed underneath the plasma membrane [22, 23, 24, 25] as well as in regions close to the nuclear membrane [22]. The microtubules are usually separated from the plasma membrane by microfilaments [25]. Microfilaments, microtubules and intermediate filaments form a highly interconnected structural network; the cytoskeleton [26, 27]. The microfilaments are joined to the plasma membrane [28, 29, 30] and it is shown that the nuclear matrix is physically associated with the cytoskeleton [28, 31]. These observations suggest that a mechanical link between the cell membrane and the nuclear membrane exists.

Next we consider the nuclear response in more detail by assuming that the response is a linear viscoelastic response. Leukocytes are viscoelastic [32]. The relaxation of leukocytes after small deformations can be modelled by the standard solid viscoelastic model [33, 34], where the leukocyte is assumed to be

an isotropic homogeneous spherical body. We have extended the standard solid viscoelastic model with a Voigt element to simulate the retardation of the nucleus. In appendix A this model is presented and the measurements are fitted to this model. The result of this fit for Na^+ hypoosmotic medium is shown in figure A4. For $t < 4$ minutes the nuclear response can be fitted to this model with one retardation time. This clearly indicates that the nuclear response may be modelled by a passive viscoelastic response.

In phase I (see appendix A) the fast initial total cell volume response was considered as a step function. It is known that this initial response is very fast; within 20 seconds an end value is reached (see e.g. [5]). On the time scale of the nuclear response (which is order of minutes) this fast initial response may be regarded as a step function.

It is important to note that in figure A4 the nuclear response was fitted to the model for $t < t_1$ (= 1.8 minutes) yielding the model parameter τ. With known values for τ, τ_c and t_1, the response for $t > t_1$ is calculated. For $t_1 < t < 4$ minutes the model calculations agree with the measurements.

For $t > 4$ minutes the calculations decrease too fast. A possible explanation for this discrepancy can be found in the suggested behaviour of actin filament solutions. Actin filament solutions appear to be viscoelastic solids on one time scale, but show liquid like behaviour over longer periods of time [32]. If actin filaments play an important role in the mechanics of a cell, similar behaviour might occur. In the study of leukocyte deformation and recovery two models emerged [32]. On the one hand the above-mentioned solid like behaviour for relative small deformations over a short time interval ($t < \pm 200$ seconds) and on the other hand a viscous like behaviour for large deformations and long periods of time. This difference might be explained by the behaviour of actin filament solutions, as suggested by Elson [32]. The deviations from the proposed solid like model, as observed in figure A4, may be explained by a change of the cytoskeleton from a solid like to a liquid like behaviour. Obviously, more detailed studies to unravel these data are necessary.

In appendix A we noticed that the time steps in the measurement of the total cell volume response are too long to allow for an accurate evaluation of the response integral (equation A6). For the Na^+ hypoosmotic medium this response could be approximated by simple functions, allowing an analytical solution of equation A6. This is not the case for the K^+ hypoosmotic medium. Therefore, in a preliminary experiment, we measured in the flowcytometer the total cell volume response with the same time resolution as the nuclear response. The cells were stained with cFDA (see Materials and Methods). From these measurements it was concluded that the total cell volume response can be measured accurately in the flowcytometer, after staining of the cells with cFDA (data not shown). In the future we will focus on dual-staining techniques to measure simultaneously the total cell volume response and the nuclear response of each individual cell. These improvements will allow more accurate analysis in terms of the proposed extended standard solid viscoelastic model.

The hyperosmotic data cannot be fitted by the proposed viscoelastic model because one expects the end value $[V/V_0]_{cell}$ and $[V/V_0]_{nucleus}$ to be equal. Obviously the nucleus resists further shrinkage beyond $[V/V_0]_{nucleus} < 0.9$ (see figure 3). This nonlinear behaviour is probably caused by the increase in density in the nucleus during the shrinking. In table 1 a hyperosmotic response of 0.77 was measured for $\Pi_0/\Pi = 0.8$, whereas in figure 3d a response

of 0.85 was obtained for $\Pi_0/\Pi = 0.5$. These measurements were performed on T-lymphocytes from different donors. The difference in the results is probably due to donor variability.

A complete three dimensional 'concentric sphere Lamb model' in combination with high resolution measurements, should permit detailed analysis in terms of macroscopic rheological properties. This will result in finding k_i and μ_i for both the cytoplasm and the nucleus. It is our opinion that these macroscopic properties can then be combined with microscopic theories, such as Nossal's study of the elasticity of cytoskeletal networks [35]. Experiments with cytochalasin D suggest that the deformability of lymphocytes is mainly determined by the nucleus [32]. In our experiment however, cytochalasin D or other cytochalasins, would have a drastic effect on the nuclear response since cytochalasins completely disrupt the cytoskeleton [36, 37].

From in vitro studies much is known about the polymerisation of actin, the rheological properties of actin gels and the influence of numerous chemical agents on these properties (e.g.[38, 39, 40]. The translation of this knowledge to the observed in vivo effects of e.g. cytochalasin is still not clear [40]. The type of experiments and data analysis as described in this work may serve as a suitable test system to assemble the data on isolated cytoskeletal components and the observed in vivo effects.

2.5 CONCLUSIONS

In this paper we present both theory and experimental techniques to study the dynamical behaviour of osmotically active lymphocytes.

In contrast to the general notion that the nuclei of lymphocytes do not respond to aniosmotic conditions, we predicted a definite response of the nuclei. The data presented here unambiguously support this view. We have shown that the nuclear response is a complex function of time and is very sensitive to the aniosmotic environment.

A linear viscoelastic model is presented to interpret our measurements. This extension to the standard solid viscoelastic model regards the nuclear response as a passive response which is induced by a mechanical link between the cellular- and the nuclear membrane. We suggest that the cytoskeleton provides this mechanical link. Our model is well suited to describe the observed response.

We believe that our measurements can provide knowledge on rheological properties of the nucleus and the cytoskeleton in vivo that is of crucial importance to understand cellular mechanics.

2.6 ACKNOWLEDGEMENTS

We wish to thank J. Klomp, F van Buitenen and R. Houben of the Immunology department of the Netherlands Cancer Institute for operating the CACE equipment, Dr. J. Borst of the Immunology department of the Netherlands Cancer Institute for providing the cFDA, and Prof. van der Wallen Mijnlieff of the Applied Physics department of the University of Twente, the Netherlands, for helpful discussions.

2.7 REFERENCES

1] L.E. McGann, M.L. Walterson, and L.M. Hogg, "Light scattering and cell volumes in osmotically stressed frozen-thawed cells," Cytometry **9**, 33-38 (1988).

2] P.M.A. Sloot and C.G. Figdor, "Elastic light scattering from leukocytes: rapid numerical analysis," Applied Optics **25**, 3559 (1986).

3] P.M.A. Sloot, A.G. Hoekstra, and C.G. Figdor, "Osmotic response of Lymphocytes measured by means of forward light scattering: theoretical considerations," Cytometry **9**, 636-641 (1988).

4] D.B. Roodyn, "Some methods for the isolation of nuclei from mammalian cells," In *Subcellular components preparation and fractionation* (Butterworths, G.B. Birnie and S.M. Fox Ed., pp. 15-42, London, 1969).

5] S. Ben-Sasson, R. Shaviv, S. Bentwich, and F. Doljanski, "Osmotic behaviour of normal and leukemic lymphocytes," Blood **46**, 891-899 (1975).

6] E.W. Gelfand, R.K.K. Cheung, K. Ha, and S. Grinstein, "Volume regulation in lymphoid leukemia cells and assignment of cell lineage," N. Eng. J. Med. **311**, 939-944 (1984).

7] K.K. Cheung, Roy, S. Grinstein, and E.W. Gelfand, "Volume regulation by human lymphocytes: identification of differences between the two major lymphocyte subpopulations," J. Clin. Invest. **70**, 632-638 (1982).

8] H.G. Hempling, "Permeability of the human leukocyte and leukemic cell to water," South. Med. J. **67**, 951-958 (1974).

9] H.G. Hempling, "The permeability of the lymphocyte membrane: applying a particle size analyzer and a hybrid computer to measure rapid changes in cell volume," Acta Cytologica **21**, 96-100 (1977).

10] G.W. Schmid-Schönbein, Y.Y. Shih, and S. Chien, "Morphometry of human leukocytes," Blood **56**, 866-875 (1980).

11] P.M.A. Sloot, M.J. Carels, P. Tensen, and C.G. Figdor, "Computer assisted centrifugal elutriation I. detection system and data acquisition equipment," Comp. Meth. Prog. Biomed. **24**, 179-188 (1987).

12] C.G. Figdor, W.S. Bont, J.E. de Vries, and W.L. van Es, "Isolation of large numbers of highly purified lymphocytes and monocytes with a modified centrifugal elutriation technique," J. Immunol. Meth. **40**, 275-288 (1981).

13] R. Dulbecco and M. Vogt, "Plaque formation and isolation of pure lines with poliomyelitis viruses," J. Exp. Med. **99**, 167-182 (1954).

14] J.W. Bruning, M.J. Kardol, and R. Arentzen, "Carboxyfluorescein fluorochromasia assays. I. Non-radioactively labelled cell mediated lympholysis," J. Immunol. Meth. **33**, 33-40 (1980).

15] G.B. Segel, G.R. Cokelet, and M.A. Lichtman, "The measurement of lymphocyte volume: importance of reference particle deformability and counting solution tonicity," Blood **57**, 894-899 (1981).

16] N.B. Grover, J. Naaman, S. Ben-Sasson, and F. Doljanski, "Electrical sizing of particles in suspensions I. theory," Biophys. J. **9**, 1398-1414 (1969).

17] M.A. van Dilla, P.N. Dean, O.D. Learum, and M.R. Melamed, *Flow Cytometry: Instrumentation and Data Analysis* (Academic Press, 1985).

18] J.A. Aten, *The data flow-2 multi dimensional multi-channel analyzer with built-in slit scan storage mode,* Tech. Rept. B.E.M. sheet 73, Butech Electronics, Vinkeveen, the Netherlands, 1984.

19] J.W. Newport and D.J. Forbes, "The nucleus: structure, function, and dynamics," Ann. Rev. Biochem. **56**, 535-565 (1987).
20] M. Grattarola, P. Carlo, G. Giannetti, R. Finollo, R. Viviani, and A. Chiabrera, "Laser flow measurement of scattering and fluorescence from cell nuclei in the presence of increasing Mg++ concentrations," Biophys. J. **47**, 461-468 (1985).
21] G. Manning, "The molecular theory of polyelectrolyte solutions with applications to the electrostatic properties of polynucleotides," Quart. Rev. Biophys. **11**, 179-246 (1978).
22] I. Yahara and G.M. Edelman, "Electron microscopic analysis of the modulation of lymphocyte receptor mobility," Exp. Cell Res. **91**, 125-142 (1975).
23] K.G. Sundqvist, P. Otteskog, L. Wanger, R. Thorstensson, and G. Utter, "Morphology and microfilament organisation in human blood lymphocytes," Exp. Cell Res. **130**, 327-337 (1980).
24] A. Fagraeus, K. Lidman, and G. Biberfeld, "Reaction of human smooth muscle antibodies with human blood lymphocytes and lymphoid cell lines," Nature **252**, 246-247 (1974).
25] S. de Petris, "Concavalin A receptors, immunoglobulins, and theta antigen of the lymphocyte surface. Interactions with Concavalin A and with cytoplasmic structures," J. Cell Biol. **65**, 123-146 (1975).
26] M. Schliwa and J. van Blerkom, "Structural interaction of cytoskeletal components," J. Cell Biol. **90**, 222-235 (1981).
27] L.M. Griffith and T.D. Pollard, "Evidence for actin filaments-microtubule interaction mediated by microtubule-associated proteins," J. Cell Biol. **78**, 958-965 (1978).
28] J. Braun and E.R. Unanue, "The lymphocyte cytoskeleton and its control of surface receptor functions," Sem. Hemat. **20**, 322-333 (1983).
29] B.H. Barber and M.J. Crumpton, "Actin associated with purified lymphocyte plasma membrane," FEBS let. **66**, 215-220 (1976).
30] D. Hoessli, E. Rungger-Brändle, B.M. Jockusch, and G. Gabiani, "Lymphocyte alfa-actinin, relationship to cell membrane and co-capping with surface receptors," J. Cell Biol. **84**, 305-314 (1980).
31] D.G. Capco, K.M. Wan, and S. Penman, "The nuclear matrix, 3D architecture and protein complement," Cell **29**, 847-858 (1982).
32] E.L. Elson, "Cellular mechanics as an indicator of cytoskeletal structure and function," Ann. Rev. Biophys. Biophys. Chem. **17**, 397-430 (1988).
33] G.W. Schmid-Schönbein, P.K.L. Sung, H. Tözeren, R. Skalak, and S. Chien, "Passive mechanical properties of human leukocytes," Biophys. J. **36**, 234-256 (1981).
34] K.L.P. Sung, C. Dong, G.W. Schmid-Schönbein, S. Chien, and R. Skalak, "Leukocyte relaxation properties," Biophys. J. **54**, 331-336 (1988).
35] R. Nossal, "On the elasticy of cytoskeletal networks," Biophys. J. **53**, 349-359 (1988).
36] M. Schliwa, "Action of cytochalasin D on cytoskeletal networks," J. Cell Biol. **92**, 79-91 (1982).
37] M. Britch and T.D. Allen, "The effects of cytochalasin B on the cytoplasmic contractile network revealed by whole-cell transmission electron microscopy," Exp. Cell Res. **131**, 161-172 (1981).
38] D.W. Godette and C. Frieden, "Actin Polymerisation, the mechanism of action of cytochalasin D," J. Biol. Chem. **261**, 15974-15980 (1986).
39] I. Tadanao, K.S. Zaner, and T.P. Stossel, "Nonideality of volume flows and phase transitions of F-actin solutions in response to osmotic stress," Biophys. J. **51**, 745-753 (1987).
40] J.A. Cooper, "Effects of cytochalasins and phalloidin on actin," J. Cell Biol. **105**, 1473-1478 (1987).
41] H. Lamb, *Hydrodynamics* (Dover Press, London, 6th ed., 1965).
42] Y.C. Fung, *Foundations of solid mechanics* (Prentice-Hall inc., Engelwood Cliffs, New Jersey, 1965).

Appendix A
Fitting of the data to an extended standard solid viscoelastic model

The relaxation of leukocytes after small deformations can be modelled by

the standard solid viscoelastic model (see figure A1). The exact 3-dimensional solution of this model for isotropic, homogeneous spheres is given by Lamb [41]. This 'Lamb model' was applied to describe the retardation of leukocytes after a specific load history [33, 34].

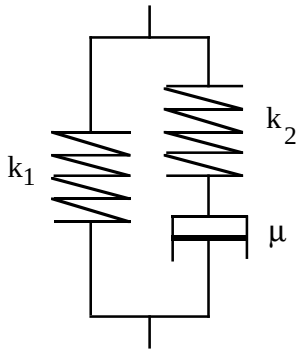


Figure A1: The standard solid viscoelastic model; k_1 and k_2 are resistive elements and μ is a capacitative element.

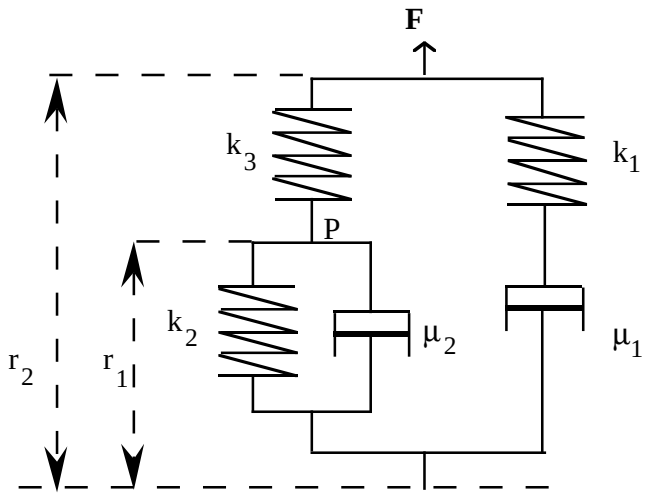


Figure A2: The extended standard solid viscoelastic model; r_1 models the nuclear volume and r_2 the total cell volume; k_i are resistive elements and μ_i are capacitative elements.

We have extended this model with a Voigt element to describe the retardation of the nucleus, see figure A2. Systems with resistive and capacitative elements in them may be fitted by a variety of models. Our model is a direct extension of the standard model, that allows for a retardation of the nuclear volume and preserves the total volume relaxation and retardation properties of the standard model.

Here, r_1 mimics the nuclear volume and r_2 the total cell volume, F is a force (the osmotic pressure) to which r_1 and r_2 respond. The response in r_1 and r_2, due to a change in F, can be calculated from the model in figure A2. We shall take a different approach, compatible with the idea that the nuclear volume response is induced by the total cell volume response. At time t = 0 F changes instantaneously to another constant value (step function in F). This results in a change of r_2 which then induces a change in r_1. We shall calculate

the response $r_1(t)$ as a function of the stimulus $r_2(t)$.

Considering equilibrium of forces at point P in figure A2:

$$k_3 (r_2 - r_1) = k_2 r_1 + \mu_2 \frac{dr_1}{dt}, \qquad [A1]$$

and taking the Laplace transform of the resulting differential equation, one finds for $R_1(p)$ and $R_2(p)$, the Laplace transforms of $r_1(t)$ and $r_2(t)$,

$$\frac{R_1(p)}{R_2(p)} = K\frac{1}{1 + p\tau}, \qquad [A2]$$

$$\text{with } K = \frac{k_3}{k_2 + k_3} \text{ and } \tau = \frac{\mu_2}{k_2 + k_3}.$$

J(p), the Laplace transform of the step response function J(t), is [42]:

$$J(p) = \frac{R_1(p)}{pR_2(p)}. \qquad [A3]$$

Inverse Laplace transformation gives

$$J(t) = K(1 - e^{-t/\tau})H(t), \qquad [A4]$$

with H(t) the Heaviside step function. The corresponding response $r_1(t)$ is

$$r_1(t) = \int_{-\infty}^{\infty} r_2(t')\dot{J}(t-t')dt' \ , \qquad [A5]$$

$$\text{with } \dot{J}(t) = \frac{dJ(t)}{dt} \ .$$

The response of a lymphocyte to aniosmotic conditions is isotropic, therefore it is assumed that a 1-dimensional model is sufficient to describe the deformation (i.e. the volume changes). In analogy with figure A2 and equations A2 - A5, the nuclear volume response is described with one retardation time and it is assumed that K = 1:

$$\left[\left(\frac{V(t)}{V(0)}\right)_N - 1\right] = \int_{-\infty}^{\infty} \left[\left(\frac{V(t')}{V(0)}\right)_{cell} - 1\right]\dot{J}_N(t-t')dt' \qquad [A6]$$

and

$$J_N(t) = (1 - e^{-t/\tau}) \ . \qquad [A7]$$

Since K = 1, V_N/V_{cell} is constant for t -> ∞, independent of the osmolarity of the

solution, in agreement with Sloot et al. [3]. This volume ratio is approximately 0.5 [2, 32, 10].

If $[V(t)/V(0)]_{cell}$ is accurately known, $[V(t)/V(0)]_N$ can be calculated from equation A6 and fitted to the measured data. However, from figure 3 it is obvious that the time step between two subsequent measurements of $[V(t)/V(0)]_{cell}$ is too large to allow for accurate numeric evaluation of equation A6. Therefore the total cell volume response is approximated by an analytical expression and the integral A6 can then be evaluated. Only the response in hypoosmotic Na^+ medium is further analysed.

Dividing the total cell volume response in two phases we obtain (see figure A3):

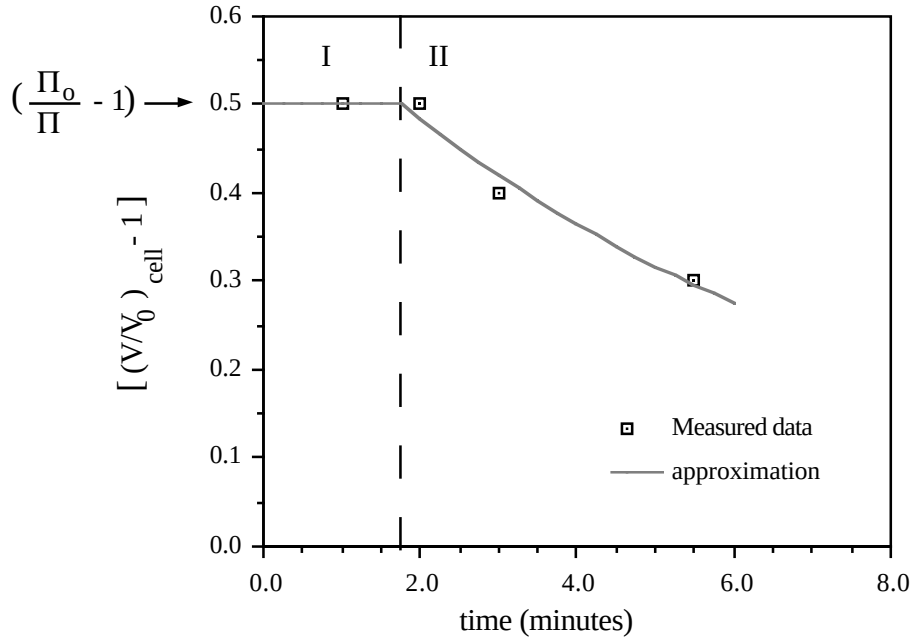


Figure A3: The total cell volume response for the hypoosmotic Na^+ medium and the approximation of this response. Phase I is a Heaviside step function; phase II is an exponential decay.

phase I: The fast initial response is approximated by $(\Pi_0/\Pi - 1)H(t)$, where it is assumed that the fast increase in volume is described by the Boyle van 't Hoff relation, with the volume of the "dry mass" (the b value) set to zero [3].

phase II: The volume regulatory phase is approximated by an exponential decay:

$$(\frac{\Pi_0}{\Pi} - 1)e^{-(t-t_1)/\tau_c}.H(t-t_1),$$

with t_1 = 1.8 minutes and τ_c = 7.1 minutes. Substituting this approximation of $[V(t)/V(0)]_{cell}$ in equation A6 gives:

$$(\frac{\Pi_0}{\Pi} - 1)(1 - e^{-t/\tau}) \quad \text{for } 0 < t < t_1 \text{, and} \qquad \text{[A8.1]}$$

$$[(\frac{V(t)}{V(0)})_N - 1] =$$

$$(\frac{\Pi_0}{\Pi} - 1)[\, e^{-(t-t_1)/\tau} - e^{-t/\tau} + \frac{\tau_c}{\tau_c - \tau}(e^{-(t-t_1)/\tau_c} - e^{-(t-t_1)/\tau})] \,, \qquad \text{[A8.2]}$$

for $t > t1$.

The measured $[(V(t)/V(0))_N - 1]$ data for $0 < t < t_1$ (see figure 3c) are fitted with a least squares method to equation A8.1, yielding τ = 1.0 minute. With known values for τ, τ_c and t_1, the response for $t > t_1$ can be calculated from equation A8.2. The result of the fit for $t < t_1$ and the calculations for $t > t_1$ are depicted in figure A4.

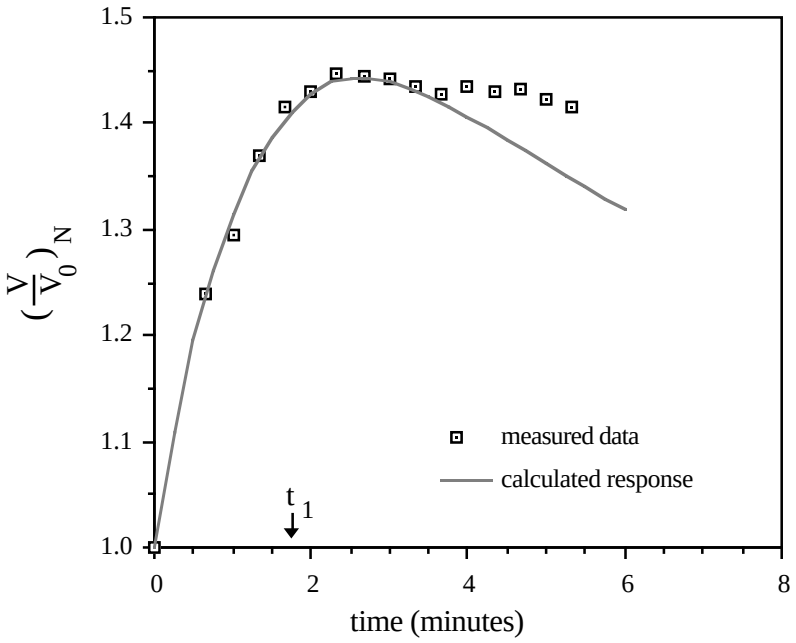


Figure A4: The measured nuclear volume response in hypoosmotic Na^+ medium and the fitted response; t_1=1.75 minute, τ_c=7.1 minute and τ=1.0 minute.

Chapter 3

ANOTHER FACE OF LORENZ-MIE SCATTERING: MONODISPERSE DISTRIBUTIONS OF SPHERES PRODUCE LISSAJOUS-LIKE PATTERNS

Alfons G. Hoekstra, Richard M.P. Doornbos, Kirsten E.I. Deurloo, Herke Jan Noordmans, Bart G. de Grooth, and Peter M.A. Sloot

Applied Optics **33**, 1994, p. 494-500

Another face of Lorenz-Mie scattering: monodisperse distributions of spheres produce Lissajous like patterns

ABSTRACT

The complete scattering matrix **S** of spheres was measured in a flowcytometer. The experimental equipment allows simultaneous detection of two scattering matrix elements for every sphere in the distribution. Two parameter scatterplots with x- and y- co-ordinates determined by the $\mathbf{S}_{11} + \mathbf{S}_{ij}$ and $\mathbf{S}_{11} - \mathbf{S}_{ij}$ values are measured. Samples of spheres with very narrow size distributions were analysed with a flowcytometer and produced unexpected two parameter scatterplots. Instead of compact distributions we observed Lissajous-like loops. Simulation of the scatterplots, using Lorenz-Mie theory, shows that these loops are not due to experimental errors, but due to true Lorenz-Mie scattering. We show that the loops originate from the sensitivity of the scattered field on the radius of the spheres. This work demonstrates that the interpretation of rare events and hidden features in flowcytometry needs reconsideration.

3.1 INTRODUCTION

The problem of scattering of electromagnetic plane waves by an isotropic, homogeneous sphere of arbitrary size and refractive index was solved in 1890 by Lorenz [1]. Eighteen years later Mie, independent of Lorenz, arrived at the same, exact, analytical solution [2]. An excellent historical account of these important results can be found in the proceedings of the Ludvig V. Lorenz session of the Optical Particle Sizing conference 1990 [3]. The mathematical richness of the formula is amazing and inspired many researchers to probe in still more detail the (differential) cross sections as a function of the radius or the refractive index of the sphere, or as a function of the wavelength of the incident light. The advent of modern computers and the development of efficient algorithms to calculate the complex functions appearing in the Lorenz-Mie scattering formula stimulated these efforts even more.

The Lorenz-Mie scattering formula posses some remarkable properties, most of which were demonstrated in scattering experiments. Well-known examples are the interference and ripple structure of the extinction cross section (e.g. [4]), glare points (e.g. [5]), or rainbows and glories (e.g. [6,7]). Despite the fact that the Lorenz-Mie solution is known for over a century, active research to the wealth of physically intriguing phenomena contained in Lorenz-Mie scattering continues. For instance, the internal electric field in the sphere receives more and more attention (e.g. [8]).

Lorenz-Mie scattering comes in many disguises. Thurn and Kiefer measured Raman spectra from optically levitated glass- and liquid spheres and observed a ripple structure superimposed on the bulk Raman spectrum [9]. The ripples proved to be due to structural resonances of the internal electric field, as could be demonstrated with the Lorenz-Mie theory. These structural resonance features could also be detected in stimulated Raman scattering from individual liquid droplets [10]. Tzeng et al. observed laser emission from small droplets at Lorenz-Mie resonance wavelengths [11]. These three effects are all due to the enhancement of the internal field intensity at a structural resonance.

This paper reports on yet another face of Lorenz-Mie scattering, which is based on the extreme sensitivity of the scattered field on the radius of the sphere. We measure the total scattering matrix of spheres with a narrow size distribution ($\Delta r/r \sim 1$ %, with r the radius of the sphere) in a dedicated flowcytometer (FCM). This experimental equipment allows us to measure $\mathbf{S}_{11}+\mathbf{S}_{ij}$ and $\mathbf{S}_{11}-\mathbf{S}_{ij}$ (ij = 12, 33, 34, $\mathbf{S}$ is the 4×4 scattering matrix) simultaneously for every single sphere in the distribution. Flowcytometry data is usually analysed by generating a N-dimensional histogram (N being the number of observables per particle, here N = 2) from the experimental data and trying to identify different data clusters in the histogram with different particles in the sample [12]. At first sight a distribution of homogeneous spheres with a very narrow Gaussian size distribution is expected to produce a single, narrow, Gaussian-like 2-dimensional histogram. However, it turned out that the measured histograms are all but Gaussian.

Here we will explain that this effect is not an experimental error, but due to true Lorenz-Mie scattering. We calculate the 2-dimensional histograms and obtain agreement between theory and experiment. In section 3.2 the theoretical and experimental background is shortly addressed, the results are presented in section 3.3 and discussed in section 3.4. Emphasis will be on the agreement between theory and experiment, but the consequences of this work for routine FCM experiments are also shortly pointed out. In section 3.5 conclusions are drawn.

3.2 MATERIALS AND METHODS

3.2.1 Flowcytometry

Flowcytometry [12] is an important technique in the biological sciences to identify and separate various populations of e.g. white bloodcells. Hydrodynamic focusing forces the cells to flow through a focused laser beam one by one. Usually the cells are stained with fluorescent probes and the fluorescence of a cell in the laser beam is measured. Furthermore, the forward - and sideward scattered light is used as an important additional parameter for the analysis. In this way we can measure several fluorescence and elastic light scattering (ELS) signals for each cell in the sample.

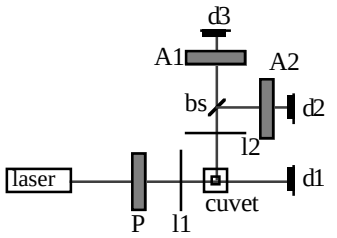


Figure 1: Schematic drawing of the optical system of the flowcytometer. The laser beam is focused by lens **l1** on a cell, flowing through the cuvet. The incident beam is polarised by means of the polariser **P**. The intensity of the forward scattered light is measured by detector **d1**, the sideward scattered light is focused by lens **l2** on detectors **d2** and **d3**, **bs** is a beamsplitter. The side scattering signals are passed through two different analyzers **A1** and **A2**.

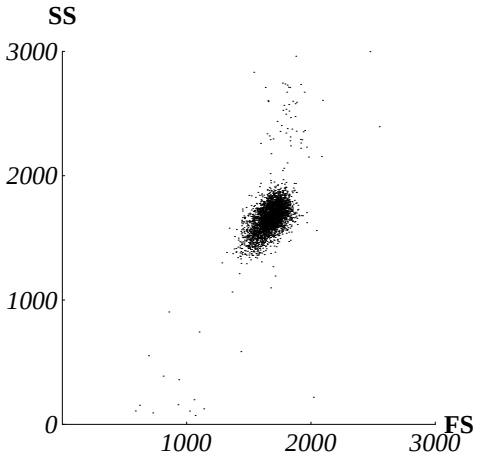


Figure 2: An example of a two parameter scatterplot for a large number of polystyrene spheres with a mean diameter of 1.98 μm, measured with a flowcytometer. The wavelength was 0.6328 μm. Every dot represents a single sphere, the value of the x-coordinate is the intensity of the forward scattered light, the value of the y-coordinate is the intensity of the sideward scattered light. The intensities are in arbitrary units.

Sloot et al. argue that in many research- and clinical applications staining of cells is undesirable [13]. As a consequence a complete characterisation of the sample must be obtained solely on the basis of ELS measurements. We expect that this is only possible by measuring suitable combinations of scattering matrix elements in the three principal FCM directions (forward-, sideward-, and backward scattered light [13]). The depolarisation experiments of de Grooth et al. [14] to distinguish between neutrophilic- and eosinophilic granulocytes are a good illustration of this point.

We developed optics to measure the total scattering matrix in a flowcytometer and showed that quantitative determination of the scattering matrix elements of particles in flow is possible [13,15]. The measuring principle was straightforward: a polarizer P was situated in the incident beam and an analyzer A just before a detector in one of the principal directions. The matrix elements are obtained by measuring scattered intensities for various P-A combinations. We have extended the optics to allow simultaneous measurement of two P-A combinations in the side scattering direction, see figure 1.

A laser beam (λ = 0.6328 μm) is focused by the circular lens l1 on the cells flowing through the cuvet, and polarised by the polarizer P. The beam waist radius in the focal point is 12.5 μm. After P the incident beam is either linearly of circularly polarised. The forward scattered light is detected by a silicon detector d1. Lens l2, a microscope objective (20 $\times$, NA = 0.4), collects the scattered light for $72.5^\circ < \theta < 107.5^\circ$, with θ the usual scattering angle. The sideward scattered light is divided in two beams by beamsplitter bs. The beams are directed onto different analyzers (A1 and A2) and the intensity after the analyzers is measured by the photomultipliers d2 and d3. The measuring principle is the same as described by Sloot et al. [13], however here it is possible to measure a P-A1 and P-A2 combination for every single cell. This allows a direct measurement of single elements of the scattering matrix. Details of this equipment, and its application to measure the scattering matrix of white bloodcells will be published elsewhere.

3.2.2 Data handling

Our equipment measures three parameters for each cell: the forward scattered light and two P-A intensities in the sideward direction. The analog signals are digitised by 12 bit A/D converters and stored in memory of the controlling computer. In every run 4096 particles are measured, the results are stored on harddisk for off-line analysis.

The data are plotted in two parameter scatterplots in which each cell is represented by a dot in a x-y plot. The x- and y- co-ordinates are determined by one of the three measured parameters. Figure 2 gives an example of a scatterplot for polystyrene spheres with a mean diameter of 1.98 μm. The forward scattering is drawn along the x-axis and one side scattering signal is drawn along the y-axis. The spheres appear as a cloud of points in the scatterplot.

The shape of the data cloud in the scatterplots and the physical interpretation are the main items of this paper.

3.2.3 Polystyrene spheres

The experiments were performed with polystyrene microspheres from Duke Scientific.* The diameters of the spheres are 1.98±0.05 μm, and 7.04±0.05 μm.

In addition to the diameter of the sphere, Lorenz-Mie calculations require the relative refractive index of the sphere. The refractive index of distilled water, in which the spheres are suspended, and polystyrene can be calculated from [16]:

$$n_{water} = n_0 + n_2/\lambda^2 + n_4/\lambda^4, \qquad [1]$$

with λ in micrometers, and $n_0 = 1.3236$, $n_2 = 3.35\times10^{-3}$, and $n_4 = -3.45\times10^{-5}$ for water, and $n_0 = 1.5711$, $n_2 = 4.82\times10^{-3}$, and $n_4 = 6.78\times10^{-4}$ for polystyrene. In our

* Duke Scientific Polystyrene microspheres 1135D. San Antonio Palo Alto CA 94303

case λ = 0.6328 μm, which gives n_{water} = 1.3318 and $n_{polystyrene}$ = 1.5874. In the calculations we will use

nrelative = npolystyrene / nwater = 1.192,

and diameters as reported above.

3.2.4 Simulation of the scatterplots

To simulate the two parameter scatterplots the intensities measured by the sideward detectors must be calculated. For spheres, using the P-A combinations as described by Sloot et al., the intensity of the scattered light after analyzer A1 and A2 is [13]

$$I_A = I_0\, C\, (\mathbf{S}_{11} \pm \mathbf{S}_{ij}),\ ij = 12,\ 33,\ \text{or}\ 34, \qquad [2]$$

with I_0 the intensity of the laser beam, C an apparatus constant, and **S** the 4×4 scattering matrix of the sphere. The total intensity on the detector is obtained by integrating over the full solid angle dΩ defined by the field of view of the objective,

$$I_{det} = \int_{d\Omega} I_A\, d\omega = I_0\, C \left[\int_{d\Omega} \mathbf{S}_{11}\, d\omega \pm \int_{d\Omega} \mathbf{S}_{ij}\, d\omega \right] \equiv I_0\, C \left[\mathbf{s}_{11} \pm \mathbf{s}_{ij} \right] \quad , \qquad [3]$$

with $\mathbf{s}_{ij}$ an integrated matrix element.

To calculate the scattering matrix of a sphere in a focused laser beam, the traditional Lorenz-Mie theory cannot be applied. Here we must rely on the generalised Lorenz-Mie theory which describes the scattering of a sphere in a Gaussian beam [17]. The $\mathbf{g}_n$ coefficients appearing in this theory are calculated using the localised interpretation [18,19]. We use the same programs as described in reference 13. The beamwaist radius of the Gaussian beam is 12.5 μm (see section 3.2.1). The particles are located in the focal point of the beam.

The procedure to generate a simulated scatterplot is as follows. First we calculate the scattering matrix $\mathbf{S}(\theta)$ as a function of the scattering angle θ (resolution $d\theta = 0.1^\circ$) for a sphere with relative refractive index 1.192, diameter d, wavelength of the incident light 0.6328 μm, and the appropriate beamwaist diameter. Then the integrated scattering elements $\mathbf{s}_{ij}$ are calculated, as described in reference 13. The $\mathbf{s}_{ij}$ are calculated for 500 different values of d in the range

$$d_{mean} - 4\sigma_d \le d \le d_{mean} + 4\sigma_d\ , \qquad [4]$$

d_{mean} is the mean diameter in the distribution and σ_d the standard deviation (assuming a normal distribution in diameter). This results in arrays of $\mathbf{s}_{ij}(d_i)$, with discrete values d_i as defined above. For every d_i the $\mathbf{s}_{ij}(d_i)$ determine co-ordinates of a dot in the two parameter scatterplot, and the total scatterplot is generated by drawing dots for every value of d_i. To simulate the relative occurrence of spheres with diameter d_i in the distribution, the radius of the dots in the scatterplot is weighted with a Gaussian function:

$$r_{dot} = r_{max}\,\mathrm{Exp}\left[-\frac{(d_i - d_m)^2}{2\sigma_d^2}\right], \qquad [5]$$

r_{dot} is the radius of the dot for a sphere with diameter d_i, r_{max} is the maximum radius of dots in the simulated scatterplots.

3.2.5 Comparison between theory and experiment

The result of the calculations is a set $(\mathbf{s}_{11}, \mathbf{s}_{12}, \mathbf{s}_{33}, \mathbf{s}_{34})_i$, i runs over all the values of d. The measurements give arrays of detector signals, as in formula 3. We need an independent scaling for both theory and experiment in order to compare the two. The experimental results are scaled such that the apparatus constants I_0 and C are removed and the experimental scatterplots are entirely described in terms of scattering matrix elements. The scaling factor for the experimental results is

$$\mathrm{scale}_{exp} = \frac{1}{2p}\sum_{i=1}^{p}(I^i_{A1} + I^i_{A2}) = \frac{1}{p} I_0 C \sum_{i=1}^{p} \mathbf{s}^i_{11}, \qquad [6]$$

with p the total number of measured spheres, and I_{A1} and I_{A2} the intensities of the scattered light after analyzer A1 and A2. Dividing the measurements by this factor results in two parameters for each measured sphere, independent of I_0 and C:

$$\left(\frac{1}{p}\sum_{i=1}^{p} \mathbf{s}^i_{11}\right)^{-1} (\mathbf{s}_{11} \pm \mathbf{s}_{ij}). \qquad [7]$$

These normalised experimental parameters are used as (x,y) co-ordinates in the scatterplots.

The term

$$\frac{1}{p}\sum_{i=1}^{p} \mathbf{s}^i_{11}$$

is the weighted mean $\mathbf{s}_{11}$ of the distribution of spheres. Assuming a normal distribution in diameter, this term is easily calculated from theory:

$$\overline{\mathbf{s}_{11}} = \int_0^{\infty} \frac{1}{\sigma_d\sqrt{2\pi}}\,\mathrm{Exp}\left[-\frac{(d - d_{mean})^2}{2\sigma_d^2}\right] \mathbf{s}_{11}(d)\,\delta d\,. \qquad [8]$$

This integral is approximated by numerical evaluation for $d_{mean} - 4\sigma_d \le d \le d_{mean} + 4\sigma_d$, using Simpson's rule. The calculated integrated scattering matrix elements are divided by the value of this integral. After scaling, both theory and experiment can be compared. In the sequel of this paper the

scatterplots of experimental and theoretical data are always scaled accordingly.

3.3 RESULTS

This section presents results of measurements and calculations of two parameter scatterplots of spheres. The normalised experimental and theoretical results are drawn in one figure. The x-axis always gives the $\mathbf{s}_{11}+\mathbf{s}_{ij}$ signal, and the y-axis the $\mathbf{s}_{11}-\mathbf{s}_{ij}$ signal, with ij as before. The dots in the theoretical curves are drawn according to equation 5. The small inset in the figures shows the theoretical curve once more, without scaling of the dot diameters.

Figure 3 shows the ($\mathbf{s}_{11}+\mathbf{s}_{12}$, $\mathbf{s}_{11}-\mathbf{s}_{12}$) scatterplot for the 7.04 μm spheres; the ($\mathbf{s}_{11}+\mathbf{s}_{33}$, $\mathbf{s}_{11}-\mathbf{s}_{33}$) and ($\mathbf{s}_{11}+\mathbf{s}_{34}$, $\mathbf{s}_{11}-\mathbf{s}_{34}$) scatterplot for this sample are drawn in figure 4 and 5 respectively. Finally the ($\mathbf{s}_{11}+\mathbf{s}_{12}$, $\mathbf{s}_{11}-\mathbf{s}_{12}$) scatterplot for the 1.98 μm spheres is drawn in figure 6.

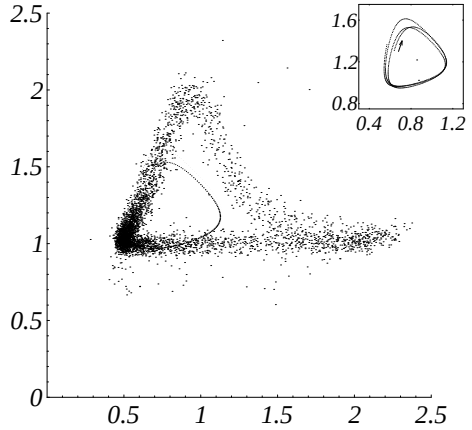


Figure 3: The normalised experimental and theoretical ($\mathbf{s}_{11}+\mathbf{s}_{12}$, $\mathbf{s}_{11}-\mathbf{s}_{12}$) scatterplot for polystyrene spheres with a mean diameter of 7.04 μm. The horizontal axes is the $\mathbf{s}_{11}+\mathbf{s}_{12}$ signal in the sideward direction, the vertical axes is the $\mathbf{s}_{11}-\mathbf{s}_{12}$ signal in the sideward direction. The inset shows the theoretical curve only, without scaling of the dot diameter. The arrow represents the starting point ($d_i = d_{mean} - 4\sigma_d$) and the loop direction, as d_i grows, of the theoretical curve.

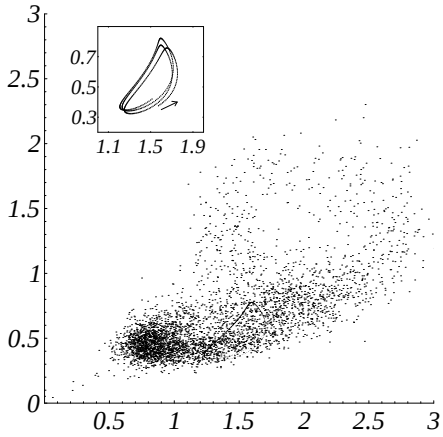


Figure 4: Same as figure 3, but now for the ($\mathbf{s}_{11}$+$\mathbf{s}_{33}$, $\mathbf{s}_{11}$-$\mathbf{s}_{33}$) scatterplot.

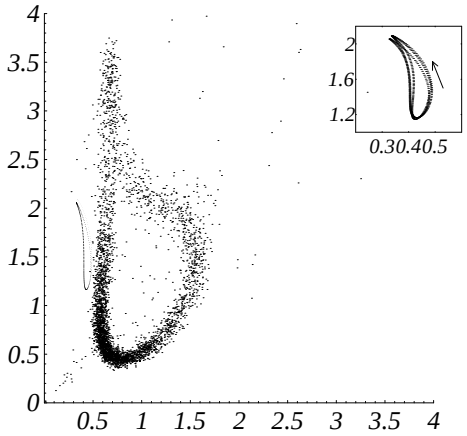


Figure 5: Same as figure 3, but now for the ($\mathbf{s}_{11}$+$\mathbf{s}_{34}$, $\mathbf{s}_{11}$-$\mathbf{s}_{34}$) scatterplot.

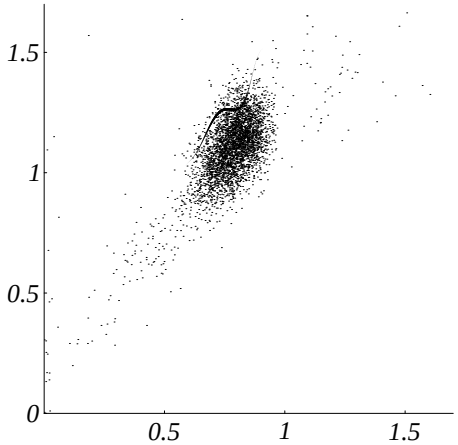


Figure 6: The normalised experimental and theoretical ($\mathbf{s}_{11}$+$\mathbf{s}_{12}$, $\mathbf{s}_{11}$-$\mathbf{s}_{12}$) scatterplot for polystyrene spheres with a mean diameter of 1.98 μm. The horizontal axes is the $\mathbf{s}_{11}$+$\mathbf{s}_{12}$ signal in the sideward direction, the vertical axes is the $\mathbf{s}_{11}$-$\mathbf{s}_{12}$ signal in the sideward direction.

3.4 DISCUSSION

Figures 3, 4, and 5 show the three normalised experimental and theoretical scatterplots for the 7.04 μm spheres. All three experimental scatterplots are loops, most obvious for the ($\mathbf{s}_{11}$+$\mathbf{s}_{12}$, $\mathbf{s}_{11}$-$\mathbf{s}_{12}$) and the ($\mathbf{s}_{11}$+$\mathbf{s}_{34}$, $\mathbf{s}_{11}$-$\mathbf{s}_{34}$) scatterplot.

The occurrence of loops in the scatterplots depends on the diameter of the spheres. Figure 6 is the ($\mathbf{s}_{11}$+$\mathbf{s}_{12}$, $\mathbf{s}_{11}$-$\mathbf{s}_{12}$) scatterplot for the 1.98 μm sphere. The experimental scatterplot is a dense distribution of points, in agreement with the theoretical results. The same holds for the other two scatterplots (data not shown).

The form and position of the normalised theoretical scatterplots for the 7.04 μm spheres compare very well with the experimental results. The agreement between theory and experiment for the ($\mathbf{s}_{11}$+$\mathbf{s}_{12}$, $\mathbf{s}_{11}$-$\mathbf{s}_{12}$) scatterplot is very good. The other two scatterplots only show a quantitative agreement between the theoretical and experimental results. Nevertheless, this demonstrates that the observed loops in the experimental plots are due to Lorenz-Mie scattering, and cannot be attributed to optical misalignments or other experimental errors.

Failure of a quantitative agreement between theory and experiment for the ($\mathbf{s}_{11}$+$\mathbf{s}_{34}$, $\mathbf{s}_{11}$-$\mathbf{s}_{34}$) scatterplot is probably due to the quality of the circular analyzers which are used to measure the $\mathbf{s}_{34}$ terms. Without going into details here, imperfect circular analyzers will cause a mixing of the $\mathbf{s}_{33}$ and $\mathbf{s}_{34}$ terms. This effect is still under investigation.

Theory and experiment can be compared on still another aspect. In the theory a normal distribution of spherical diameters was assumed, and this was simulated by weighting the radius of the dots in the scatterplot with a Gaussian function (see equation 5). The theoretical curves nicely show the distribution of the spherical diameter. However, as is obvious from the scatterplots for the 7.04 μm sphere, this distribution is in error with the experimental results. The experimental scatterplots have most points in the lower corner of the loops, whereas the theoretical results show that most points should show up along a long side of the loops. A closer look at the theoretical curves shows that this error between theory and experiment can be explained by assuming that the actual mean diameter of the spheres is somewhat larger than 7.04 μm, the value provided by the supplier of the spheres. The small inset in figures 3, 4, and 5 shows the theoretical curves, without scaling of the dot radius, and the starting point and loop direction of the curves. The starting point is for $d = d_{mean} - 4\sigma_d$, and for increasing diameter the curve loops in the direction of the arrow. The ($\mathbf{s}_{11}$+$\mathbf{s}_{12}$, $\mathbf{s}_{11}$-$\mathbf{s}_{12}$) scatterplot loops in clockwise direction, whereas the other two curves loop in counter clockwise direction. If the mean diameter of the sphere is increased, the position of the dots with large diameter in the theoretical curves shifts in the loop direction. Therefore, if the mean diameter of the sphere is increased to 7.08 μm, all three theoretical scatterplots reproduce the measured distribution correctly. Furthermore, we measured the scatterplots for a slightly larger wavelength of the incident light (λ = 0.647 μm). In that case one expects that the same experimental loops occur, but with a small shift of the distribution of the data points in the loop. In all

three scatterplots we observed this shift of the data points. The direction and magnitude of the shift are in agreement with calculated values (data not shown).

The origin of the loops in the scatterplots can be understood by examining the integrated matrix elements as a function of the diameter of the sphere. Figure 7 plots $\mathbf{s}_{11}$, $\mathbf{s}_{12}$, $\mathbf{s}_{33}$, and $\mathbf{s}_{34}$ as a function of the diameter of the sphere, for d as in equation 4, and d_{mean} is 7.04 μm. The matrix elements posses an extreme sensitivity on the diameter of the sphere. For 6.84 μm < d < 7.24 μm, $\mathbf{s}_{11}$ goes through three minima and maxima. The amplitude of the oscillations is approximately 20% of the mean value of $\mathbf{s}_{11}$. The other integrated scattering matrix elements have the same properties, although the oscillations are not in phase with the $\mathbf{s}_{11}$ oscillations. However, for spheres with 1.78 μm < d < 2.18 μm the $\mathbf{s}_{ij}$ elements increase monotonously with increasing d (data not shown).

In the diameter range of figure 7 the integrated scattering functions are almost periodic. The $\mathbf{s}_{11}$ strongly resembles a sine function. The other (quasi) periodic scattering matrix elements can be viewed as a Fourier series of sine and cosine functions. Fourier transformation of the data in figure 7 supports this view. For all four scattering matrix elements the absolute value of the Fourier transform peaks around the same ground frequency ν_0, and around higher harmonics $k\nu_0$ (with k an integer > 1). The amplitude of the third and higher harmonics are negligible compared to the amplitude of the ground frequency (data not shown).

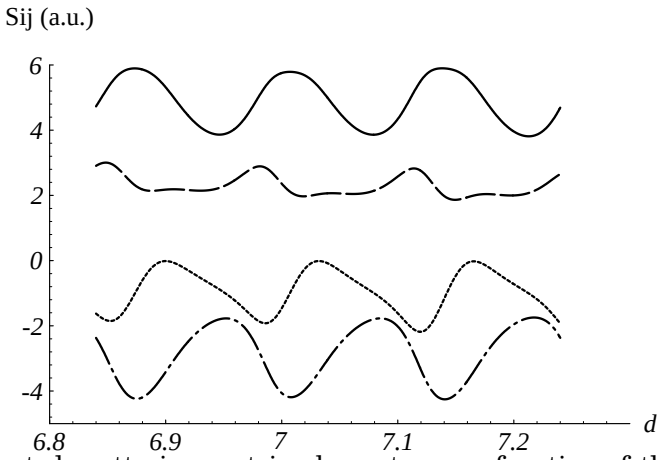


Figure 7: The integrated scattering matrix elements, as a function of the diameter d of the sphere; d is in micrometer, the $\mathbf{s}_{ij}$ are in arbitrary units; the solid line is $\mathbf{s}_{11}$; the dotted line is $\mathbf{s}_{12}$; the dashed line is $\mathbf{s}_{33}$; the dashed-dotted line is $\mathbf{s}_{34}$.

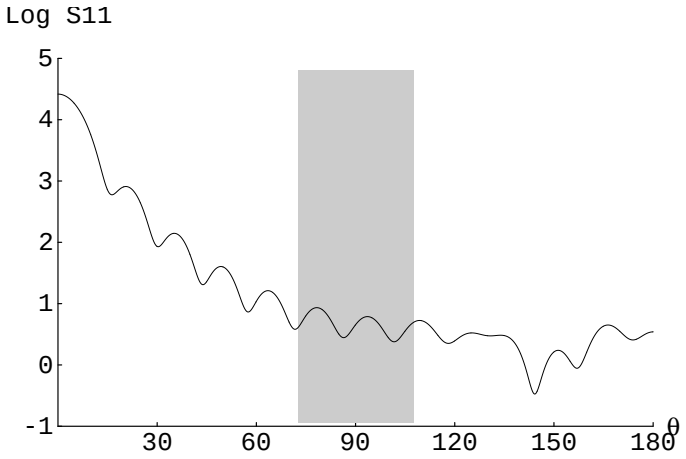


Figure 8: The $\mathbf{S}_{11}$ element as a function of the scattering angle θ, for d = 1.98 μm. The grey area denotes the field of view of the side scattering detectors.

When constructing the theoretical scatterplot, we actually draw a parametric plot, with the diameter of the sphere d as the only parameter. The functions on the x - and y-axes of the scatterplot are approximately combinations of sines and cosines of some ground frequency and higher harmonics. Therefore the scatterplots can be viewed as Lissajous plots. The oscillations in the integrated matrix elements, which are not in phase with each other, give rise to the Lissajous loops in the two parameter scatterplots. Note that in principle it is possible to measure more complicated scatterplots, with e.g. double loops (a ground frequency oscillation in one direction and a first harmonic oscillation in the other direction). Actually, in one instance we measured scatterplots with such double loops. We are still working on the interpretation of these experiments.

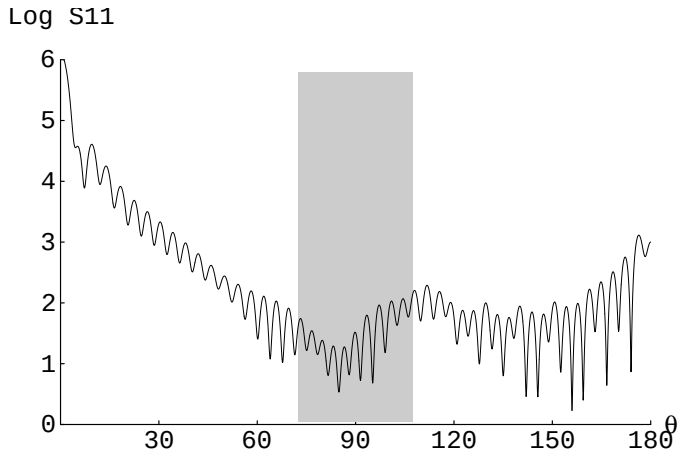


Figure 9: The $\mathbf{S}_{11}$ element as a function of the scattering angle θ, for d = 7.04 μm. The grey area denotes the field of view of the side scattering detectors.

Figures 8 and 9 show $\mathbf{S}_{11}$ as a function of the scattering angle θ, for d is 1.98 μm and 7.04 μm respectively. The grey area denotes the field of view of the side scattering detectors. If the diameter of the sphere is increased, the minima and maxima in $\mathbf{S}_{11}(\theta)$ gradually shift to the forward directions, and new

minima and maxima seem to originate from $\theta = 180^{\circ}$ [4]. Slightly increasing the size of the 7.04 μm sphere results in entrance and exit of local minima and maxima in the field of view of the detectors, and changes in the amplitude of the minima and maxima, giving rise to the oscillations in the curves of the integrated matrix elements as a function of the diameter of the sphere (figure 7). The distance between the local minima and maxima in the $\mathbf{S}_{11}(\theta)$ curve for the 1.98 μm sphere is too large to induce strong oscillations in the integrated matrix elements after small changes of the diameter of the sphere.

Measurement of polarised light scattering in FCM is by no means a routine procedure yet, nevertheless our results contain an important warning. It is common practice in FCM to measure a side scattering signal. Since the incident light is always linearly polarised (most lasers emit light which is linearly polarised, perpendicular to the scattering plane), the intensity on the side scattering detectors is always a combination of $\mathbf{s}_{11}$ and $\mathbf{s}_{12}$. Therefore a narrow monodisperse distribution of spheres can produce bimodal histograms in the side scattering channels (this can be seen in figure 3 for the $\mathbf{s}_{11} + \mathbf{s}_{12}$ signal). Especially if spheres are applied to calibrate the instrument, extra care should be taken, and small spheres should be used to avoid the above mentioned problems. Furthermore, interpretation of rare events and hidden distributions in scatterplots requires careful analysis in view of the above mentioned effect. We are currently investigating to which extent the Lissajous loops can be expected in scatterplots from biological particles.

3.5 CONCLUSIONS

This report shows yet another face of Lorenz-Mie scattering; unexpected, Lissajous-like loops in two parameter scatterplots of spheres, as obtained by flowcytometry. The complete scattering matrix of spheres, with a very narrow distribution in size, was measured with a dedicated flowcytometer. The measured two parameter scatterplots can contain unexpected, Lissajous-like loops. By simulating these scatterplots, it was shown that the experimental results are true Lorenz-Mie scattering phenomena, and not due to experimental errors. The occurrence of loops in the scatterplots is dependent on the diameter of the spheres. It was shown that oscillations in the integrated matrix elements, as a function of the diameter of the sphere, form the basis for the Lissajous loops. These oscillations in their turn originate from the interference structure in the differential scattering cross sections.

A consequence of this behaviour is the origin of bimodal histograms in the side scattering channels, due to monodisperse samples. This will hamper the interpretation of rare events and hidden distributions in the scatterplots.

3.6 REFERENCES

1] L.V. Lorenz, "Upon the light reflected and refracted by a transparent sphere," Vidensk. Selsk. Shrifter **6**, 1-62 (1890).
2] G. Mie, "Considerations on the optics of turbid media, especially colloidal metal sols," Ann. d. Physik **25**, 377-442 (1908).
3] E.D. Hirleman (Ed). *Proceedings of the 2nd international congress on optical particle sizing*. Arizona State University Printing Services, March 1990 (part of the proceedings are published in the 20 November 1990 issue of Applied Optics) .
4] C.F. Bohren and D.R. Huffman, *Absorption and Scattering of Light by Small Particles* (John Wiley & Sons, 1983).
5] H.C. van de Hulst and R. Wang, "Glare Points," Applied Optics **30**, 4755-6763 (1991).
6] R.T. Wang and H.C. van de Hulst, "Rainbows: Mie computations and the Airy approximation," Applied Optics **30**, 106-117 (1991).
7] H.M. Nussenzveig, "Complex angular momentum theory of the rainbow and the glory," J. Opt. Soc. Am. **69**, 1068-1079 (1979).
8] J.A. Lock, "Interference enhancement of the internal fields at structural scattering resonances of a coated sphere," Applied Optics **29**, 3180-3187 (1990).
9] R. Thurn and W. Kiefer, "Structural resonances observed in the Raman spectra of optically levitated liquid droplets," Applied Optics **24**, 1515-1519 (1985).
10] J.B. Snow, S.X. Qian, and R.K. Chang, "Stimulated Raman scattering from individual water and ethanol droplets at morphology-dependent resonances," Optics Letters **10**, 37-39 (1985).
11] H.M. Tzeng, K.F. Wall, M.B. Long, and R.K. Chang, "Laser emission from individual droplets at wavelengths corresponding to morphology-dependent resonances," Optics Letters **9**, 499-501 (1984).
12] M.A. van Dilla, P.N. Dean, O.D. Laerum, and M.R. Melamed, *Flow Cytometry: Instrumentation and Data Analysis* (Academic Press, Analytical Cytology, 1985).
13] P.M.A. Sloot, A.G. Hoekstra, H. van der Liet, and C.G. Figdor, "Scattering matrix elements of biological particles measured in a flow through system: theory and practice," Applied Optics **28**, 1752-1762 (1989).
14] B.G. Grooth, L.W.M.M. Terstappen, G.J. Puppels, and J. Greve, "Light-Scattering Polarisation Measurements as a New Parameter in Flow Cytometry," Cytometry **8**, 539-544 (1987).
15] P.M.A. Sloot and A.G. Hoekstra, "Arbitrarily-shaped particles measured in flow through systems," in *Proceedings of the 2nd international congress on optical particle sizing*, Ed. E.D. Hirleman, 1990.
16] E. Gulari, "Latex particle size distributions from multiwavelength turbidity spectra," Part. Charact. **4**, 96 (1987).
17] G. Gouesbet, B. Maheu, and G. Grehan, "Light scattering from a sphere arbitrarily located in a Gaussian beam, using a Bromwich formulation," J. Opt. Soc. Am. A **5**, 1427-1443 (1988).
18] G. Grehan, B. Maheu, and G. Gouesbet, "Scattering of laser beams by Mie scatter centres: numerical results using a localised approximation," Applied Optics **25**, 3539-3548 (1986).
19] G. Gouesbet, G. Grehan, and B. Maheu, "Localised interpretation to compute all the coefficients g(n,m) in the generalised Lorenz-Mie theory," J. Opt. Soc. Am. A **7**, 998-1007 (1990).

Chapter 4

A PARALLEL CONJUGATE GRADIENT METHOD APPLIED TO ELASTIC LIGHT SCATTERING SIMULATIONS

A.G. Hoekstra, P.M.A. Sloot, W. Hoffmann, and L.O. Hertzberger

submitted to the Journal of Computer & Software Engineering

A parallel Conjugate Gradient method applied to elastic light scattering simulations

ABSTRACT

We describe parallelization of a preconditioned Conjugate Gradient method for distributed memory computer systems. The method is used to solve linear systems of equations emerging from Elastic Light Scattering simulations. Based on a time complexity analysis, the Conjugate Gradient method was implemented with a rowblock data decomposition on a ring of transputers. The measured - and theoretically calculated execution times agree within 5%. Finally convergence properties of the algorithm are investigated and the suitability of a polynomial preconditioner is examined.

4.1 INTRODUCTION

Elastic Light Scattering (ELS) is a powerful non-destructive particle detection and recognition technique, with important applications in diverse fields such as astrophysics, biophysics, and environmental studies. Our goal is to calculate ELS from small biological objects, specifically human white bloodcells.

Many exact and approximate theories to calculate ELS from particles are known [1, 2]. Nevertheless, important classes of particles fall outside the range of these theories. This prompted much research in the field of light scattering by arbitrary shaped particles [3]. The coupled dipole method, due to Purcell and Pennypacker [4], is one method that in principle allows calculation of ELS from any particle.

The computational most demanding part of the coupled dipole method is a large set of linear equations that must be solved. Human white bloodcells [2] give rise to matrices with dimensions of $O(10^4)$ to $O(10^6)$. To keep calculation times within acceptable limits, a very efficient solver, implemented on a powerful computer is required. We apply a Conjugate Gradient (CG) method, implemented on a transputer network, to solve the system of equations.

This paper concentrates on parallelization of a CG method suited to our application. After a brief description of the coupled dipole method of ELS in section 4.2, section 4.3 gives a theoretical time complexity analysis of the CG method for different parallelization strategies. Based on the results of section 4.3 the CG method was implemented on a bi-directional ring of transputers, with a rowblock decomposition of the system matrix. Section 4.4 describes this implementation, and section 4.5 presents performance measurements and

convergence behaviour of the method. The results are discussed in section 4.6 and some conclusions are presented in section 4.7.

4.2 THE APPLICATION

4.2.1 The coupled dipole method

In the coupled dipole method of ELS a particle is divided into N small subvolumes called dipoles. Dipole i (i = 1,..,N) is located at position $\mathbf{r}_i$ ($\mathbf{r} = (x,y,z)^T$). An externally applied electric field $\mathbf{E}^0(\mathbf{r})$ is incident on the particle. An internal electric field $\mathbf{E}(\mathbf{r}_i)$ at the dipole sites, due to the external field and the induced dipole fields is generated. The internal field is

$$\mathbf{E}(\mathbf{r}_i) = \mathbf{E}^0(\mathbf{r}_i) + \sum_{j \neq i}^{N} \mathbf{F}_{i,j}\mathbf{E}(\mathbf{r}_j) \ , \qquad [1]$$

where $\mathbf{F}_{i,j}$ is a known 3×3 matrix of complex numbers describing the electric field at $\mathbf{r}_i$, radiated by a dipole located at $\mathbf{r}_j$. The field radiated by dipole j is calculated by multiplying $\mathbf{F}_{i,j}$ with the electric field $\mathbf{E}(\mathbf{r}_j)$ at dipole j. $\mathbf{F}_{i,j}$ depends on the relative refractive index of the particle, the wavelength of the incident light and the geometry of the dipole positions. As soon as the internal field $\mathbf{E}(\mathbf{r}_i)$ is known, the scattered field is easily calculated [4].

Equation 1 results in a system of N coupled linear equations for N unknown fields, and can be formulated as a matrix equation

$$\mathbf{AE} = \mathbf{E}^0 \ , \qquad [2]$$

where

$$\mathbf{E} = \begin{pmatrix} \mathbf{E}(\mathbf{r}_1) \\ \vdots \\ \mathbf{E}(\mathbf{r}_N) \end{pmatrix} \text{ and } \mathbf{A} = \begin{pmatrix} \mathbf{a}_{1,1} & \cdots & \mathbf{a}_{1,N} \\ \vdots & \ddots & \vdots \\ \mathbf{a}_{N,1} & \cdots & \mathbf{a}_{N,N} \end{pmatrix}, \qquad [3]$$

with $\mathbf{a}_{i,i} = \mathbf{I}$ (the 3×3 identity matrix) and $\mathbf{a}_{i,j} = -\mathbf{F}_{i,j}$ if i ≠ j. The vector $\mathbf{E}^0$ has the same structure as $\mathbf{E}$. The complex matrix $\mathbf{A}$ is referred to as the interaction matrix. Equation 2 is a set of 3N equations of 3N unknowns (the 3 arising from the 3 spatial dimensions). All numbers in equation 2 are complex. The 3N×3N interaction matrix is a dense, diagonally dominant, symmetric matrix. For human white bloodcells the number of dipoles N to model the particle lies in the range $O(10^4)$ to $O(10^6)$ (see chapter 5 of this thesis). Calculation of the internal electric fields at the dipole sites, that is to solve the system of equations 2 is the computational most demanding part of the coupled dipole method. In the sequel we will address this problem in detail.

4.2.2 The Conjugate Gradient method

From a numerical point of view, the coupled dipole method boils down to solving a very large system of linear equations $\mathbf{Ax} = \mathbf{b}$, with $\mathbf{A}$ a n×n complex symmetric matrix, $\mathbf{b}$ a known complex vector and $\mathbf{x}$ the unknown complex vector.

Solely the size of the system matrix forces the use of iterative methods. A very powerful iterative method is the Conjugate Gradient (CG) method [5]. Usually this method is applied to systems with large banded matrices, like the ones arising from discretisations of partial differential equations. Draine [6] however showed that the CG method is also very well suited to solve linear systems arising from the coupled dipole method. For instance, for a typical small particle with 2320 dipoles (n = 6960) the CG method only needs 17 iterations to converge.

The original CG method of Hestenes and Stiefel [7] (the CGHS method in Ashby's taxonomy [8], to which we will confirm ourselves) is only valid for Hermitian positive definite matrices. Since the interaction matrix of the Coupled dipole method is not Hermitian (but symmetric), CGHS cannot be employed. We will use the PCGNR method, in the Orthomin implementation (see reference [8] for details). The PCGNR method is suitable for any system matrix [8]. The algorithm is shown below:

The PCGNR algorithm

Initialise: Choose a start vector $\mathbf{x}_0$ and put

$k = 0$

$\mathbf{r}_0 = (\mathbf{b} - \mathbf{A}\mathbf{x}_0)$ calculate the residual vector

$\mathbf{s}_0 = \mathbf{M}^{-1}\mathbf{M}^{-H}\mathbf{A}^H\mathbf{r}_0$ preconditioning

$\mathbf{p}_0 = \mathbf{s}_0$ the first direction vector

Iterate: **while** $|\mathbf{r}_k| \geq \varepsilon\,|\mathbf{b}|$ iterate until the norm of the residual vector is small enough

$$\alpha_k = \frac{(\mathbf{A}^H\mathbf{r}_k)^H\mathbf{s}_k}{(\mathbf{A}\mathbf{p}_k)^H(\mathbf{A}\mathbf{p}_k)}$$

$\mathbf{x}_{k+1} = \mathbf{x}_k + a_k\mathbf{p}_k$ calculate new iterate

$\mathbf{r}_{k+1} = \mathbf{r}_k - a_k(\mathbf{A}\mathbf{p}_k)$ update residual vector

$\mathbf{s}_{k+1} = \mathbf{M}^{-1}\mathbf{M}^{-H}\mathbf{A}^H\mathbf{r}_{k+1}$ preconditioning

$$\beta_k = \frac{(\mathbf{A}^H\mathbf{r}_{k+1})^H\mathbf{s}_{k+1}}{(\mathbf{A}^H\mathbf{r}_k)^H\mathbf{s}_k}$$

$\mathbf{p}_{k+1} = \mathbf{s}_{K+1} + b_k\mathbf{p}_k$ calculate new direction vector

$k = k + 1$

stop $\mathbf{x}_k$ is the solution of $\mathbf{Ax} = \mathbf{b}$

The number ε is the stopping criterion, and is set to the square root of the machine precision. Since all implementations are in double precision (i.e. a 52

bit fraction) ε is set to 10^{-8}. The vector $\mathbf{r}_k$ is the residual vector of the k-th iteration, the vector $\mathbf{p}_k$ is the direction vector, and **M** is the preconditioning matrix. The iterations stops if the norm of $\mathbf{r}_k$ is smaller than the norm of **b** multiplied by ε.

The purpose of preconditioning is to transform ill-conditioned system matrices to a well-conditioned form, thus increasing the convergence rate of the conjugate gradient method. The preconditioning matrix **M** must approximate the system matrix **A** as closely as possible but still allow a relative easy calculation of the vector $\mathbf{s}_k$. A good preconditioner decreases the total execution time of the conjugate gradient process. This means that a good *parallel* preconditioner not only decreases the total number of floating-point operations, but also possesses a high degree of parallelism. A good preconditioner depends both on the system matrix *and* the parallel computer. For instance, the incomplete Cholesky factorisation preconditioner [9] is very successful on sequential computers, but performs not as good on vector- and parallel computers.

Polynomial preconditioners [10] are very well suited for parallel computers [11, 12], and experiments have shown that, implemented on a distributed memory computer, they can be much more effective than incomplete factorisation preconditioners [see e.g. 13]. Therefore we adopt the concept of polynomial preconditioning and put

$$\mathbf{M}^{-1} = \sum_{i=0}^{m} \gamma_i \mathbf{A}^i \ . \qquad [4]$$

The choice of m and γ_i is topic of active research [e.g. 11,14], but is beyond the scope of this paper. Here we concentrate on parallelization of the PCGNR method. We take the von Neumann series as the polynomial preconditioners.

$$\mathbf{M}^{-1} = \sum_{i=0}^{m} \mathbf{N}^i , \qquad [5]$$

where **N = I - A**.

4.3 TIME COMPLEXITY ANALYSIS

4.3.1 Introduction

The time complexity of several versions of the parallel PCGNR method is derived and compared. The algorithm is parallelised by a data decomposition of the system matrix. A comparison is made, based on the estimated execution time of the parallel program T_{par} as a function of the number of processors p and the matrix dimension n, and the parameters τ_{calc} , $\tau_{startup}$, and τ_{comm}, the time to perform one floating point operation, to startup a point-to-point

communication, and to send one byte between two processors after the communication is started, respectively. In the following a concise summary of the time complexity analysis will be presented. For details we refer to [15].

4.3.2 Decomposition

One iteration of the PCGNR algorithm[6] contains 3 vector updates ($\mathbf{x}_{k+1}$, $\mathbf{r}_{k+1}$, $\mathbf{p}_{k+1}$), 3 vector inner products ($[(\mathbf{A}^H\mathbf{r}_k)^H\ \mathbf{s}_k]$, $[(\mathbf{A}\mathbf{p}_k)^H\ (\mathbf{A}\mathbf{p}_k)]$, $[\mathbf{r}_k\ \mathbf{r}_k]$), and 2 + 2m matrix vector products ($\mathbf{A}^H\mathbf{r}_k$, $\mathbf{A}\mathbf{p}_k$, and 2m for the polynomial preconditioning). Both the vector operations and the matrix vector product will be executed in parallel.

The parallel vector routines are straightforward. The vectors are divided in equal parts and assigned to the processors. The parallel vector update can be performed completely in parallel. The parallel inner product requires a global summation of the partial inner products which are calculated in parallel.

We have analysed three different decompositions of the matrix; the rowblock -, grid -, and columnblock decomposition. [15] Here we will only summarise the results for the first two decompositions.

The rowblock decomposition is achieved by dividing **A** in blocks of rows, with every block containing $\lceil N/p \rceil$ or $(\lceil N/p \rceil - 1)$ consecutive rows, and assigning one block to every processing element ($\lceil\ \rceil$ is the ceiling function). Note that in our application **A** is symmetric so that $\mathbf{A}^H$ is also decomposed in row-block. In the grid decomposition the (square) matrix **A** is decomposed in p square blocks, which are distributed among the processing elements.

The parallel matrix vector product ($\mathbf{A} \times$ vector *and* $\mathbf{A}^H \times$ vector) for the rowblock decomposition is shown schematically in figure 1.

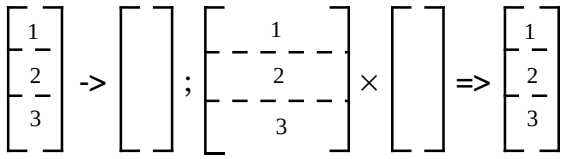


Figure 1: the parallel matrix vector for a rowblock decomposed matrix, a "->" denotes a communication, and a "=>" a calculation.

The argument vector must reside in memory of every processing element. Therefore, before calculating the matrix vector product, every processing element must gather the argument vector (a vector gather operation). The result is already decomposed in the correct way for further calculations (inner products, vector updates, or matrix vector products).

The parallel matrix vector product for the grid decomposition is drawn in figure 2.

---

6 From here on we will calculate the time complexity of one iteration step of the PCGNR algorithm. The initialization time is comparable to the time needed for one iteration (due to the same number of matrix vector products). Therefore it is not necessary to include it in the analysis. Furthermore, if the number of iterations becomes large, the initialization time can be neglected compared to the total iteration time.

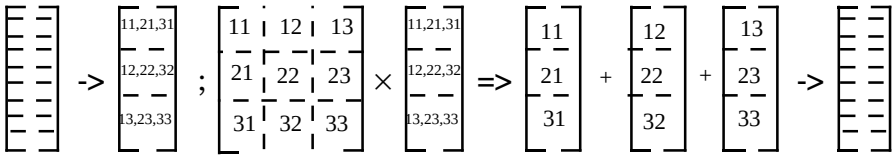


Figure 2: the total parallel matrix vector product for a grid decomposed matrix, a "->" denotes a communication, and a "=>" a calculation.

Now two communication routines are needed. The first communication routine is a partial vector gather operation, the second is a partial vector accumulate, followed by a scatter operation.

4.3.3 Topology

As a consequence of the data decomposition (parts of) the vectors and scalars must be communicated between processors. In this section we investigate two network topologies: a bidirectional ring and a square mesh with wrap-around in one direction (a cylinder) with bidirectional links. The rowblock decomposition is analysed on the ring, the grid decomposition is considered in conjunction with the cylinder.

The parallel PCGNR with rowblock decomposition of the matrix contains a vector gather operation and a complex accumulate operation (parallel inner product). During the vector gather operation (see figure 1) every processor receives from all other processors in the network a part of the vector, which is subsequently stored. On the bidirectional ring this is achieved as follows:[7]

1) In the first step every processor sends its local part of the vector to the left and the right processor and, at the same time, receives from the left and the right processor their local part.

2) In the following steps, the parts received in a previous step, are passed on from left to right and vice versa, and in parallel, parts coming from left and right are received and stored. The complex accumulate operation is executed similar, but now a complex number is sent.

The parallel PCGNR with grid decomposition of the matrix contains a partial vector gather operation, a partial vector accumulate operation, and a complex accumulate operation. The partial vector gather operation can be performed by communicating the data vertically through the cylinder. The accumulate operations are performed in two stages: first communication in horizontal direction, followed by communication in vertical direction through the cylinder. Details of these operations and derivations of execution times of the two parallel versions of the PCGNR method can be found in [15].

Using these expressions, we can find approximate formula for the efficiency ε of the parallel PCGNR for both the rowblock-ring - and the grid-cylinder combination [15]:

$$\varepsilon = \frac{T_{seq}}{pT_{par}} \approx \left(1 + O\left(\frac{p}{n}\right) + O\left(\frac{p}{n}\right)\frac{\tau_{comm}}{\tau_{calc}} + O\left(\frac{p^2}{n^2}\right)\frac{\tau_{startup}}{\tau_{calc}}\right)^{-1}. \qquad [6]$$

---

7 We assume that all bidirectional links can operate fully parallel, e.g. like transputer links.

Equation 6 shows that deviations from $\varepsilon = 1$ occur due to three sources: non-parallel computations, communications, and communication startups. Let us first concentrate on the term involving τ_{comm}. This term can be identified with the communication overhead f_c defined by Fox et al.[16, section 3-5]. For a problem dimension d_p and a dimension of the complex computer d_c Fox et al. show that in general (in our notation)

$$f_c = \begin{cases} \dfrac{constant}{\left(n^2/p\right)^{1/d_p}} \dfrac{\tau_{comm}}{\tau_{calc}} & \text{if} \quad d_c \geq d_p \\ p^{(1/d_c - 1/d_p)} \dfrac{constant}{\left(n^2/p\right)^{1/d_p}} \dfrac{\tau_{comm}}{\tau_{calc}} & \text{if} \quad d_c < d_p \end{cases} . \qquad [7]$$

Thus, for the row-block-ring combination ($d_p = 2$ and $d_c = 1$) f_c is

$$f_c = constant \frac{p}{n} \frac{\tau_{comm}}{\tau_{calc}} , \qquad [8]$$

which is in agreement with equation 6. However, for the grid-cylinder combination ($d_c = 2$) one would expect

$$f_c = constant \frac{\sqrt{p}}{n} \frac{\tau_{comm}}{\tau_{calc}} , \qquad [9]$$

which disagrees with our result. The reason for this is the implementation of the partial vector accumulate. The first step of this operation cannot exploit the full dimensionality of the network.

The second term in equation 6 is due to non-parallel computation (the grid cylinder combination) or due to load imbalance (the rowblock-ring combination) and has the same order of magnitude as the communication overhead. The last term in the denominator describes the efficiency reduction due to communication startup times. This term however is an order of magnitude smaller than the previous two.

As long as τ_{calc}, τ_{comm}, and $\tau_{startup}$ have the same order of magnitude, and $n >> p$, the efficiency of the parallel PCGNR algorithm can be very close to unity. In the next paragraph we will investigate this in more detail.

4.3.4 The hardware parameters

Our parallel computing system is a Meiko computing surface, consisting of 64 T800 transputers, each with 4 Mb RAM, hosted by a Sun sparc workstation. The configuration is described in more detail by Hoffmann and Potma [17].

The communication times were measured by sending different sized packets over the bidirectional links and measuring the total sending time. Fitting of the measurements to a straight line resulted in $\tau_{startup} = 13.3$ μs, and

τ_{comm} = 0.99 μs/byte.

The parameter τ_{calc} should not just incorporate the raw computing power of the floating point unit of the transputer, but also the overheads due to indexing and memory access. As a consequence it is virtually unachievable to define one unique value for τ_{calc}, since different operations, such as a single addition, or a vector update, give rise to different overheads. Fortunately most floating-point operations in the PCGNR algorithm take place in the matrix vector products. Therefore we timed the complex matrix vector product, including all overheads, and used these results to derive τ_{calc}. This results in τ_{calc} = 1.62 μs/(floating-point operation) for double precision numbers.

With the experimental values for the system parameters a numerical calculation of T_{par} is possible. Comparison of T_{par} of the rowblock-ring and grid-cylinder combination revealed that both parallelization strategies are equivalent [15]. Depending on the exact values of n and p, one is faster than the other. However, the relative difference in total execution time is very small. This is true as long as n>>p, which will be the normal situation.

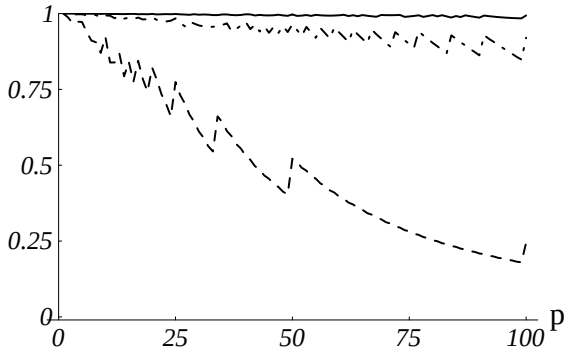


Figure 3: The theoretical efficiency for parallel PCGNR without preconditioning (m=0) with a rowblock decomposition of the matrix, implemented on a bidirectional ring of transputers. The dashed line is for n=100, the dot-dashed line is for n=1000 and the full line is for n=10.000; p is the number of transputers

Since the execution time T_{par} cannot justify a choice between the rowblock-ring - and the grid-cylinder combination, other criteria enter the discussion. From an implementation point of view the rowblock-ring combination is preferable. The rowblock decomposition introduces just one communication routine: the gather operation. Parallel PCGNR with a grid decomposition on the other hand contains more, and more complex communication routines. Furthermore, from a user point of view the rowblock-ring combination has one important advantage. The ring can have any number of processors, and therefore the maximum number of free processors can always be used during production runs. This is important in parallel computing environments where the users can request any number of processors. These considerations are in favour of the rowblock-ring combination. Therefore it was decided to implement parallel PCGNR with a rowblock decomposition of the matrix, on a bi-directional ring of transputers.

Figure 3 show the theoretical efficiency of this parallel PCGNR. If the number of rows per processor is large enough the efficiency will be very close to unity. Therefore parallel PCGNR with rowblock decomposition, implemented on a bidirectional ring is very well suited for coarse grain distributed memory

computers.

4.4 IMPLEMENTATION

The parallel PCGNR was implemented in Occam 2 [18] on the Meiko Computing Surface. The transputers all run the same two processes: a router and a calculator. Router processes on neighbouring transputers are connected to each other via channels. These channels are associated with hardware transputer links. The router process calls communication routines from a communication library. These routines, such as e.g. the vector gather operation, take data from local memory and send it to other routers, and process data that is received during the communication routine.

The calculator process performs the work on the decomposed data. If data needs to be communicated, the calculator sends a command, in the form of a single character, via an internal channel to the router process. The router process receives this character, interprets it and issues the desired communication routine. During this communication step the calculator process is idle. After finishing the communication, the router process sends a 'ready' signal to the calculator process, which then proceeds.

In principle the communication hardware and the CPU and FPU of the transputer can work in parallel, thus allowing to hide the communication behind calculations. We decided not to use this feature since total communication time is very small compared to calculation times.

The present implementation keeps the matrix in memory. The maximum matrix that fits in the local memory of one transputer is n=495. On the full 63 transputer ring, the maximum matrix size is n=3885. For realistic, larger problems (n is $O(10^4)$ to $O(10^6)$), the matrix cannot be kept in memory. The matrix elements will then be calculated as soon as they are needed, using the definition in equation 3.

The main advantage of our implementation is the complete uncoupling of the details of the communication and the "useful" work in the parallel program. The calculator code closely resembles the sequential code, with just some extra statements issuing commands to the router. The algorithm is easily adapted by making changes to the calculator process. This is important for e.g. testing the influence of the parameter m in the preconditioner.

4.5 RESULTS

This section presents the results of two experiments. First we measured the performance of the implementation of the parallel PCGNR for m=0 (no preconditioning), by measuring the execution time of one iteration of the PCGNR algorithm, as a function of p and n. These results are compared with

the theoretical expressions. Secondly, we tested the convergence of the PCGNR algorithm, for typical matrices of the coupled dipole method, for m=0 and m=1.

4.5.1 Performance measurements

We have measured the execution time of one iteration of the parallel PCGNR, with m=0, for n = 60, 219, and 495 as a function of p, where $1 \leq p \leq 63$. The error in the time measurement is one clock tick, which is 1.0 μs for the high priority transputer clock. Figures 4, 5, and 6 show the measured and calculated efficiencies of one iteration of the parallel PCGNR.

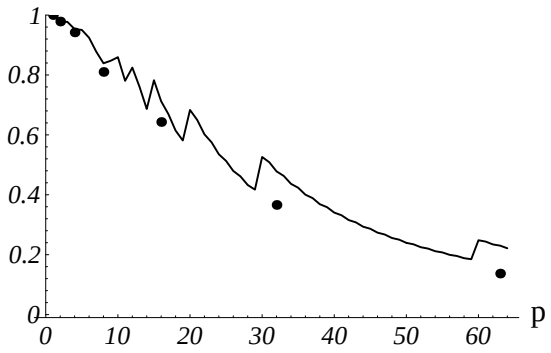


Figure 4: The efficiency on one iteration of the parallel PCGNR for n = 60, as a function of the number of processors. The black dots are the experimental results, the solid line is the theoretical efficiency.

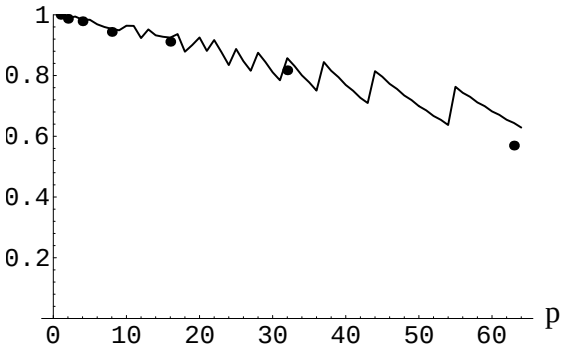


Figure 5: The efficiency on one iteration of the parallel PCGNR for n = 219, as a function of the number of processors. The black dots are the experimental results, the solid line is the theoretical efficiency.

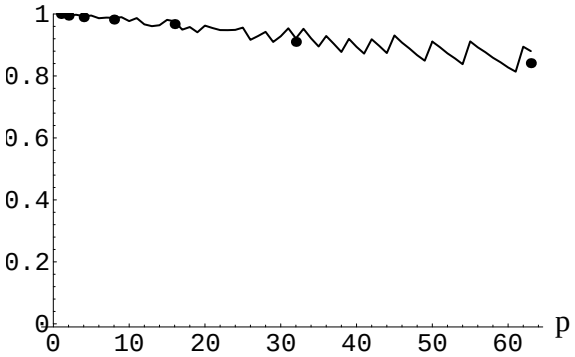


Figure 6: The efficiency on one iteration of the parallel PCGNR for n = 495, as a function of the number of processors. The black dots are the experimental results, the solid line is the theoretical efficiency.

Table 1 gives the difference, in terms of percentage, between the theoretical and measured T_{par} as a function of p and n. Finally we measured T_{par} for the maximum problem on 63 transputers, i.e. n=3885. The difference between theory and experiment was 3.4 %. The theoretical efficiency in this case is 0.98.

	p						
	1	2	4	8	16	32	63
n 60	2.2	2.2	1.1	1.3	8.3	27.7	64.4
219	0.5	0.8	1.0	1.5	2.0	5.4	13.6
495	0.7	0.8	1.1	1.3	1.7	2.0	5.4

Table 1: The difference, in terms of percentage, between the experimentally measured and theoretically calculated execution time of one iteration of the PCGNR, as a function of p and n.

4.5.2 Convergence behaviour

The matrix **A**, as defined by equation 3 depends on the relative refractive index n_{rel} of the particle of interest, the wavelength of the incident light, and the position and size of the dipoles. To test the convergence behaviour of PCGNR for these matrices, the norm of the residual vector, as function of the iteration number k was measured, for some typical values of the parameters. Here we will show results for the currently largest possible matrix (n=3885, i.e. 1295 dipoles). The results for smaller matrices are comparable.

The wavelength of the incident light was set to λ = 488.0 nm (blue light), and the diameter of the dipoles to $\lambda/20$. The scattering particle was a sphere, the dipoles were put on a cubic grid with spacing $\lambda/20$. The relative refractive index was chosen to depict some representative material compounds of interest: n_{rel} = 1.05 and 1.5 to give the range of indices of biological cells in suspension; n_{rel} = 1.33 + 0.05i (dirty ice); n_{rel} = 1.7 + 0.1i (silicates); and n_{rel} = 2.5 + 1.4i (graphite).

Figures 7 and 8 show the logarithm of the norm of the residual vector divided by the stopping criterion (ε times the norm of **b**) for n_{rel} = 1.05 and for graphite respectively. If the measured function is smaller than zero, the iteration has converged. We tested for m=0 (no preconditioning) and for m=1 (first order von Neumann preconditioning). The convergence behaviour for the other refractive indices is comparable [15].

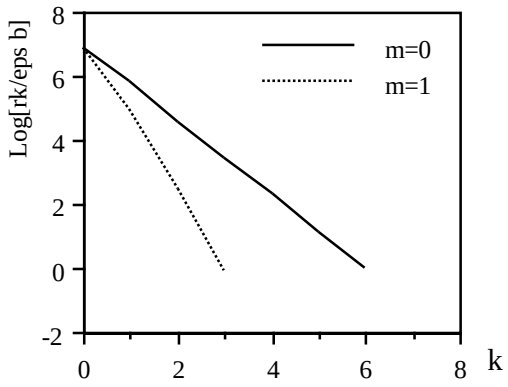


Figure 7: The norm of the logarithm of the residual vector $\mathbf{r}_k$ divided by the norm of **b** times ε, as a function of the iteration number, for n_{rel} = 1.05.

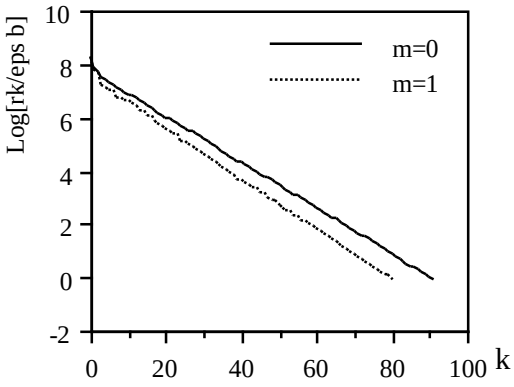


Figure 8: The norm of the logarithm of the residual vector $\mathbf{r}_k$ divided by the norm of **b** times ε, as a function of the iteration number, for n_{rel} = 2.5 + 1.4i (graphite).

4.6 SUMMARY AND DISCUSSION

Our aim is to simulate the scattering of (visible) light by biological cells, specifically human white bloodcells. For this we exploit the Coupled Dipole method (see section 4.2). This model gives rise to a very large system of linear equations. The size of this system forces us to use an iterative method to solve the equations. Moreover, acceptable run times can only be achieved if the iterative solver converges fast, and if the calculations can be performed at a very high speed. The first demand led to the choice of the CG method, the second one to a parallel computer.

The CG method is almost always applied to banded (sparse) matrices, coming from e.g. finite element discretisations. For these type of problems the CG method is successfully parallelised (see e.g.[16], chapter 8), also together

with (polynomial) preconditioners (see e.g. [19]). Application of the CG method for full matrices is less common. However, systems of equations with full, but diagonally dominant matrices as coming from the Coupled Dipole method, or from other computational electromagnetics techniques (see e.g. [20]) can be solved efficiently with a (preconditioned) CG method.

We analysed the rowblock decomposition in combination with a ring topology and the grid decomposition in combination with a cylinder topology. If $\tau_{startup}$ and τ_{comm} have the same order of magnitude, the startup time for communications is negligible compared to the pure sending time. In that case the communication overhead for the cylinder, and the less rich ring topology have the same order of magnitude. This shows that the rowblock-ring combination and the grid-cylinder combination will have comparable execution times. As long as the time constants have the same order of magnitude, and n>>p, both parallel implementations will have an efficiency close to 1 (see equation 6).

The time constants were measured and introduced in the expression for T_{par}. The rowblock-ring - and the grid-cylinder combination are almost indistinguishable if one looks at their execution time. This shows that even on a low dimensional network as a ring it is possible to implement a parallel CG method with a very high efficiency (see figure 3), comparable with implementations on a cylinder topology. Keeping this in mind we implemented the rowblock-ring combination, based on two practical considerations; programming effort and system resources.

Figures 4 to 6, and table 1 show that the agreement between the theoretical and the measured values of T_{par} are within 5 % of each other, provided that n/p is not too small. For small n and large p the difference between theory and experiment is very high. In this situation every processor has only a few rows in memory and in that case the single parameter τ_{calc}, used to describe the calculation time in a single processor, is not very accurate. However, this is not a real problem, since we are mainly interested in the situation n/p >> 1, where the difference between theory and experiment is very small.

The second experiment concerned the convergence properties of the CG algorithm, and the influence of one preconditioner on the convergence speed. As was noted in section 4.2.3 a good preconditioner for a parallel PCGNR implementation must not only decrease the total number of floating point operations needed to find a solution, but furthermore the preconditioning steps must allow efficient parallelization. The polynomial preconditioner of equation 4 can be implemented using matrix vector products only (assuming that the coefficients γ_i are known at forehand). From section 4.3 it is obvious that the matrix vector product can be parallelised very efficiently in the rowblock-ring combination. Therefore the polynomial preconditioner is an ideal candidate for parallel implementations. Our implementation of the first order von Neumann preconditioner (m=1, see equation 5) supports this point. Measurements of the execution times for this implementation show that the efficiency is very close to 1 (data not shown). The same will be true for higher order preconditioners.

Figures 7 and 8 show that $|\mathbf{r}_k|$, the norm of the residual vector after the k'th iteration, decreases exponentially after every iteration. Even for graphite, with a large refractive index, the number of iterations needed for convergence is only 91, which is 2.3% of the matrix dimension n. For smaller refractive indices the number of iterations for m=1 is half of the number of iterations for

m=0. The time for one iteration however is approximately a factor two higher (four instead of two matrix vector products per iteration). Therefore the total execution time is approximately the same. For higher refractive indices the number of iterations is still decreased by the preconditioner, but not by a factor of two. In that case the total execution time for m=1 is much larger than for m=0. The first order von Neumann preconditioner is too inaccurate to be an effective preconditioner for our type of matrices. In the future we will experiment with higher order von Neumann preconditioners, and other types of polynomial preconditioners. Especially if the size of matrix grows we expect that good preconditioning will be inevitable to obtain realistic simulation times.

4.7 CONCLUSIONS

The preconditioned Conjugate Gradient method, for dense symmetric complex matrices, using polynomial preconditioners, can be parallelised efficiently for distributed memory computers. Both a theoretical analysis of the time complexity and actual implementations support this conclusion. Furthermore, the time complexity analysis shows that a parallel implementation on a one dimensional (ring) topology is as good as an implementation on a more complex two dimensional (cylinder) topology.

Theoretical predictions of the execution time of the parallel implementation agree very well with the experimental results.

Convergence of the PCGNR method, for some typical matrices, is very good. The first order von Neumann preconditioner, used as a test case, parallelises very well; the efficiency of the implementation remains close to 1. However, the total execution time is not decreased. More research with respect to better polynomial preconditioners is required.

4.8 ACKNOWLEDGEMENTS

We wish to thank M.J. de Haan and S.G. Meijns for their contribution to the implementation of the Occam code and the performance measurements.

4.9 REFERENCES

1] C.F. Bohren and D.R. Huffman, *Absorption and Scattering of Light by Small Particles* (John Wiley & Sons, 1983).

2] P.M.A. Sloot, A.G. Hoekstra, H. van der Liet, and C.G. Figdor, "Scattering matrix elements of biological particles measured in a flow through system: theory and

practice," Applied Optics **28**, 1752-1762 (1989).
3] D.W. Shuerman, *Light Scattering by Irregularly Shaped Particles* (Plenum Press, New York, 1980).
4] E.M. Purcell and C.R. Pennypacker, "Scattering and absorption of light by nonspherical dielectric grains," The Astrophysical Journal **186**, 705-714 (1973).
5] G.H. Golub and C.F. van Loan, *Matrix Computations* (The John Hopkins University Press, 1989).
6] B.T. Draine, "The discrete dipole approximation and its application to interstellar graphite grains," Astrophys. J. **333**, 848-872 (1988).
7] M.R. Hestenes and E. Stiefel, "Methods of Conjugate Gradients for solving linear systems," Nat. Bur. Standards J. Res. **49**, 409-436 (1952).
8] S.F. Ashby, T.A. Manteuffel, and P.E. Saylor, "A taxonomy for conjugate gradient methods," Siam J. Numer. Anal. **27**, 1542-1568 (1990).
9] J.A. Meijerink and H.A. van der Vorst, "An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix," Math. Comp. **31**, 148-162 (1977).
10] O.G. Johnson, C.A. Micchelli, and G. Paul, "Polynomial preconditioners for conjugate gradient calculations," Siam J. Numer. Anal. **20**, 362-376 (1983).
11] S.F. Ashby, "Minimax polynomial preconditioning for Hermitian linear systems," Siam J. Matrix Anal. Appl. **12**, 766-789 (1991).
12] Y. Saad, "Practical use of polynomial preconditionings for the conjugate gradient method," Siam J. Sci. Stat. Comput. **6**, 865-881 (1985).
13] C. Tong, "The preconditioned conjugate gradient method on the connection machine," International Journal of High Speed Computing **1**, 263-288 (1989).
14] S.F. Ashby, T.A. Manteuffel, and J.S. Otto, "A comparison of adaptive Chebyshev and least squares polynomial preconditioning for Hermitian positive definite linear systems," Siam J. Stat. Comput. **13**, 1-29 (1992).
15] A.G. Hoekstra, P.M.A. Sloot, W. Hoffmann, and L.O. Hertzberger, *Time complexity of a parallel Conjugate Gradient solver for light scattering simulations: theory and SPMD implementation,* Tech. Rept. CS-92-06, Faculty of Mathematics and Computer Science, University of Amsterdam, 1992.
16] 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, 1988).
17] W. Hoffmann and K. Potma, "Implementing linear algebra algorithms on a Meiko Computing Surface," Appl. Numer. Math. **8**, 127-148 (1991).
18] Inmos Ltd, *OCCAM® 2 Reference Manual* (Prentice Hall, 1988).
19] M.A. Baker, K.C. Bowler, and R.D. Kenway, "MIMD implementation of linear solvers for oil reservoir simulation," Parallel Computing **16**, 313-334 (1990).
20] J.I. Hage, M. Greenberg, and R.T. Wang, "Scattering from arbitrarily shaped particles: theory and experiment," Applied Optics **30**, 1141-1152 (1991).

Chapter 5

A PARALLEL IMPLEMENTATION OF THE COUPLED DIPOLE METHOD OF ELASTIC LIGHT SCATTERING

A.G. Hoekstra, L.O. Hertzberger, and P.M.A. Sloot

Submitted to Computers in Physics

A Parallel Implementation of the Coupled Dipole Method of Elastic Light Scattering

ABSTRACT

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 bloodcells, 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 Massively Parallel computer, a Parsytec GCel-3, containing 512 T805 transputers. The efficiency of the parallel implementation is investigated for simulations of model particles containing a small number of dipoles, and the results are extrapolated to large scale simulations. 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 bloodcells, is investigated.

5.1 INTRODUCTION

Elastic light scattering (ELS) from arbitrary particles has many important applications, both in exact sciences and industrial or environmental utilisations. Examples are ELS from human white bloodcells [e.g. 1, 2], from interstellar and interplanetary dust particles [e.g. 3, 4], from soot particles in combustion flames [e.g. 5, 6], or from airborne particles [e.g. 7, 8]. 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 [9, chapter 3]. Furthermore, 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. 10]. 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 [3]. 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 bloodcells [1]. More examples can be found in [11].

This observation, the need to calculate ELS from arbitrary particles, prompted much research to methods allowing a numerical solution of the ELS problem. One such method is the Coupled Dipole (CD) method [12], which is equivalent with the VIEF method[13, 14]. The original CD method is due to

Purcell and Pennypacker. Afterwards many authors have contributed to the method [15, 16, 17, 18, 19, 20]. However, these additions to, and refinements of the original CD method have no major consequence to the main subject of this report, as will become clear in the sequel.

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. For this a system of 3N equations with 3N unknowns, N being the number of dipoles, must be solved. In our application, simulation of ELS from human white bloodcells, 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 [21, 22].

We have implemented a Conjugate Gradient method, suited for the linear system from the CD method, on a parallel MIMD computer [22]. Here we report on an implementation of the CD method, using this parallel kernel, on a 512 node transputer Parsytec GCel MIMD computer. After a short explanation 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 bloodcells?

5.2 THE COUPLED DIPOLE METHOD

Consider an arbitrary particle, located at the origin of a Cartesian coordinate system. The particle is illuminated by a monochromatic electromagnetic field $\mathbf{E}^0(\mathbf{r})$. The wavelength is λ, and for simplicity 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 [10].

The CD method divides the particle in N subvolumes. The size of a subvolume 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 $\lambda/20 < d < \lambda/10$, with d the size of a subvolume [23]. The field at $\mathbf{r}_i$ radiated by a dipole located at $\mathbf{r}_j$ is [24]

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

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

$\mathbf{n} = \mathbf{r} / r$, 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:

$$\mathbf{p} = \boldsymbol{\gamma}\mathbf{E}, \quad [2]$$

$\boldsymbol{\gamma}$ is the polarizability tensor. We will assume an isotropic polarizability: $g = \gamma\mathbf{I}$, with $\mathbf{I}$ the identity matrix and γ a scalar polarizability. This means that we take spherical subvolumes 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

$$\mathbf{E}(\mathbf{r}_i) = \gamma_j \mathbf{F}_{ij} \mathbf{E}_j \ , \quad [3]$$

with the functional $\mathbf{F}_{ij}$ defined by equations 1 and 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

$$\mathbf{E}(\mathbf{r}_i) = \mathbf{E}^0(\mathbf{r}_i) + \sum_{j\neq i}^{\mathbf{N}} \gamma_j \mathbf{F}_{ij} \mathbf{E}_j \ , \ 1 \le i \le N. \quad [4]$$

The summation in equation 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. 13, 25], 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 [25]. The exact form of this eigenterm still is under discussion [see e.g. 25, 26]. Introduction of an eigenterm in the numerical simulation is trivial. Keeping all this in mind, 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.

Equation 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

$$\mathbf{x} = \begin{pmatrix} \mathbf{E}(\mathbf{r}_1) \\ \vdots \\ \mathbf{E}(\mathbf{r}_N) \end{pmatrix}, \mathbf{b} = \begin{pmatrix} \mathbf{E}^0(\mathbf{r}_1) \\ \vdots \\ \mathbf{E}^0(\mathbf{r}_N) \end{pmatrix} \text{ and } \mathbf{A} = \begin{pmatrix} \mathbf{I} & -\gamma\mathbf{F}_{12} & \cdots & & -\gamma\mathbf{F}_{1N} \\ -\gamma\mathbf{F}_{21} & \mathbf{I} & & & \\ \vdots & & \ddots & & \vdots \\ & & & \mathbf{I} & -\gamma\mathbf{F}_{N-1,N} \\ -\gamma\mathbf{F}_{N1} & & \cdots & -\gamma\mathbf{F}_{N,N-1} & \mathbf{I} \end{pmatrix}. \quad [5]$$

The matrix **A** is the 3N×3N 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 $\mathbf{r}_{obs}$:

$$\mathbf{E}^S(\mathbf{r}_{obs}) = \gamma \sum_{i=1}^{N} \mathbf{F}_{obs,i}\, \mathbf{E}(\mathbf{r}_i) \ . \quad [6]$$

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 Cartesian co-ordinates of the middle point of the dipoles is [(k+1/2)d, (l+1/2)d, (m+1/2)d], with (k,l,m) integer numbers. The scalar polarizability is calculated using the Claussius-Mossotti relation [24]

$$\gamma = \frac{4\pi\varepsilon_0}{3n} \frac{m^2 - 1}{m^2 + 2} \ , \quad [7]$$

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. 27, 28]. Once again, these adaptations of the original CD method have no consequences for the parallel implementation of the CD method.

Figure 1 gives an estimate of the number of dipoles needed to describe a compact particle, as a function of the size parameter α, with d equal to λ/20, λ/10, and λ/5. Even for modest size parameters the number of dipoles is $O(10^4)$ or larger. In our specific application, simulation of ELS from human white bloodcells, α is in the range of 20 to 60. This means that the number of dipoles should be larger than 100.000.

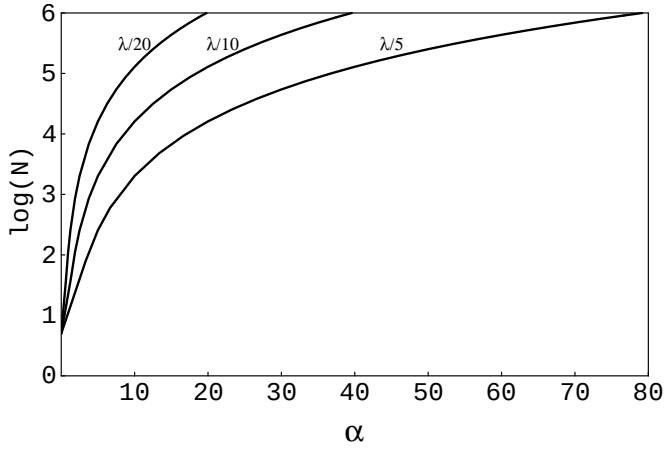


Figure 1: Estimation of the number of dipoles needed to model a compact particle.

5.3 PARALLEL IMPLEMENTATION

5.3.1 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, equation 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×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) [27], 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 [16], and Draine applied a Conjugate Gradient iteration [15].

Direct methods require $O(n^3)$ floating-point operations to find a solution, whereas iterative method 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 ~ 500), the Jacobi iteration becomes non-convergent [28]. A very efficient iterative method is the Conjugate Gradient method [29]. Draine [15] showed that the Conjugate Gradient method

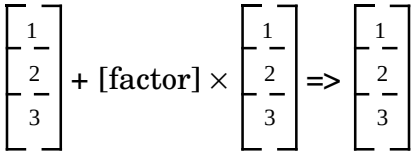
is very well suited for solving the coupled dipole equations. The number of iterations needed to find the solution is much smaller than the dimension of the matrix. For instance, for a typical small particle with 2320 dipoles (n = 6960) the Conjugate Gradient method only needs 17 iterations to converge. We apply a Conjugate Gradient method, the so-called CGNR method [30], to find the electric field on the dipoles. In reference [22] we describe how this method was parallelised for distributed memory computers.

Calculation of the scattered field, using equation 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 [22] as a basis of the parallel CD method. We adapted the processor ring topology and rowblock matrix decomposition of the 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 important details of the complete parallel implementation of the CD method are highlighted.

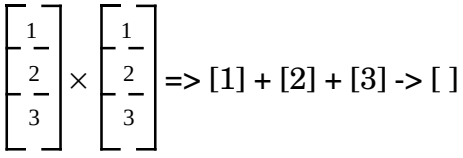
5.3.2 Parallel calculation of the dipole fields

The CGNR method was parallelised 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 [22]. Based on this analysis the CGNR method was 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.

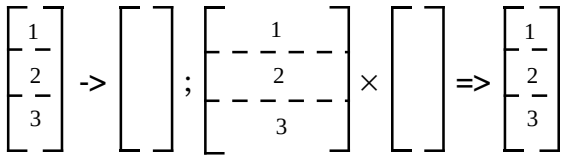
The CGNR method contains two matrix vector products, three vector updates and three inner products per iteration. Figure 2 schematically shows how these operations are performed in parallel [22]. The vector update can be performed completely in parallel, all data is present in local memory of the processors. 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 in 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 vector gather operation the matrix vector product can be performed in parallel. The result 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.

Figure 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 parallel CGNR uses two communication routines: the scalar accumulate operation (see figure 2.b) and the vector gather operation (see figure 2.c). These operations are very similar. In both operations every processor must send data from local memory to all other processors, and receive data from all other processors. On the bidirectional ring this is achieved as follows: first every processor sends its local data to the left and the right processor and, at the same time, receives from the left and the right processor their local data; in the following steps, the parts received in a previous step, are passed on from left to right and vice versa, and in parallel, parts coming from left and right are received and stored.

The efficiency of a parallel program is defined as [31]

$$\varepsilon = \frac{T_{par}(1)}{pT_{par}(p)}, \qquad [8]$$

where $T_{par}(p)$ 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 $O(n^2/p)\tau_{calc}$, the communication time is $O(n)\tau_{comm}$. [22]. 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

$$\varepsilon \approx \left(1 + O\left(\frac{p}{n}\right)\frac{\tau_{comm}}{\tau_{calc}}\right)^{-1}. \qquad [9]$$

Here we neglected many details of the communications and computations, for this see reference [22]. Still, equation 9 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 [22].

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 some 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.

5.3.3 Parallel calculation of the scattered fields

The scattered electric field is calculated according to equation 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 in the root processor (see next subsection), which writes the results 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. However, the parallel calculation of the scattered fields also has a very good parallel efficiency on its own right, as a straightforward analysis reveals.

5.3.4 Details of the implementation

The parallel CD method is implemented on IC^{3}A's* Parsytec GCel-3 massively parallel MIMD computer, with 512 T805 transputers. The implementation was brought about under Parsytec's programming

---

* IC^{3}A is the Interdisciplinary center of Computer based Complex systems Research Amsterdam; for more information contact the authors.

environment Parix** ; the programming language was C.

Parix' programming model is Single Program Multiple Data (SPMD) [32]; 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 transputer has a very efficient context switching mechanism build in hardware. Therefore a transputer can be programmed very efficiently in a "multi-process" way, where each process has a specific task (i.e. communication of data through a link or numerical tasks). Parix offers this feature in the form of a threads library.

On a bare transputer, programmed in its native language Occam [33], communication is done explicitly over the transputer hardware links. This point-to-point communication is synchronous. Parix offers a more abstract view of the transputer, based on the possibilities which are realised in hardware of the new T9000 transputer [34]. Communication is performed through so-called virtual links. A set of virtual links between different processors can be grouped into a virtual topology. Users can define their own virtual topologies, or can use predefined virtual topologies, such as rings, meshes, or trees, by calling appropriate functions of the virtual topology library of Parix. These library functions guarantee an optimal mapping of the virtual topology on the actual physical two dimensional network of the GCel computer.

The main program of the parallel CD method, which runs on each processor, is shown below.

```
main ()
{                                       /* main of parallel CD method */
        MakeRing (...);                 /* Create Ring Topology */
        GetRing_Data (...);             /* Extract information of Ring
                                           topology */
        StartThread (Calculator,...);   /* Start the Calculator thread */
        StartThread (Router,...);       /* Start the Router thread */
        Wait for threads to terminate
}
```

First, the ring topology is established by a call to the MakeRing function, which installs a ring topology on the booted partition, and numbers all processors in the ring from zero to p, where p is the total number of processors in the ring. The GetRing_Data call makes this information available to the program. In our implementation processor zero is assigned a special role; i/o is performed by this processor. Strictly speaking this is not necessary, all processors can perform i/o operations using Parix' remote procedure calls to the Unix system of the host computer.

Next two threads are started, a calculator and a router. These two processes are connected via an internal link, see figure 3. The Calculator performs all the numerical work. If data should be transferred to other processors (e.g. a vector gather operation), the calculator sends a command to the Router, which calls appropriate communication routines. After

** We used Parix version 1.1

termination of the communication, a 'ready' signal is send to the calculator.

This style of programming has some advantages. The Calculator code strongly resembles the 'sequential' code, with occasional calls to the router. In this way the complete parallel program can be easily maintained and adapted, if necessary. Furthermore, the Router and the Calculator can actually run in parallel on a transputer. The CPU and link hardware runs in parallel, thus enabling to communicate and calculate at the same time. Therefore, instead of waiting for the 'ready' signal, the Calculator could perform some calculations, and then wait for the 'ready signal'. Finally, many parallel scientific codes share many generic global communication routines. If the Router contains all these calls, a reusable harness for parallel programming on a ring (or any other topology) can be created, and application programmers only need to develop the Calculator part of the code.

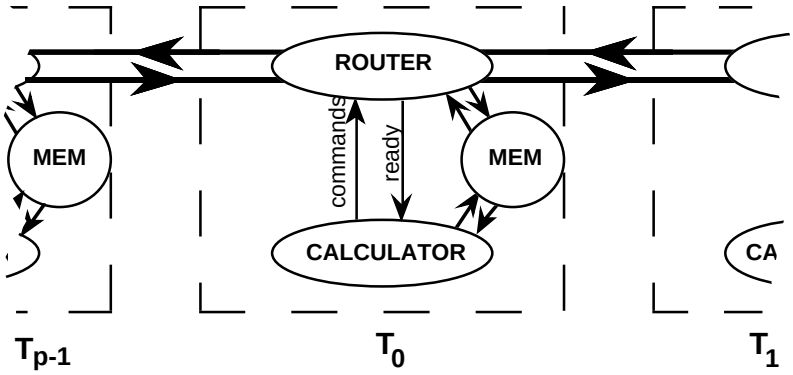


Figure 3: Threads running on each transputer; the Router connects to the neighbouring processors in the ring, the calculator performs the numerical tasks and issues commands to the Router to perform specific communications. Mem is the memory of a transputer

The kernel of the parallel CD method, the CGNR method, was originally implemented in Occam [22]. Porting of the original Occam code to Parix-C turned out to be very straightforward, because many powerful concepts in Occam to express parallelism are mimicked by functions in Parix libraries.

5.4 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, using the high priority clock of the transputer (ticks every μs).

As a test problem we simulate scattering by a homogeneous sphere, whose scattering properties are known analytically (the so-called Mie theory [10]). In the CD method we discretised 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

$$(i+1/2)^2 + (j+1/2)^2 + (k+1/2)^2 \leq l^2 \qquad [10]$$

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 = λ/10. The wavelength was 488 nm and the refractive index was 1.05.

For small problem sizes (N < ~2200) the execution times 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.

Figure 4 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. Figure 5 and 6 show these results for the calculation of the scattered field, and for the complete parallel CD simulation, respectively. Note that in figure 6 the maximum problem size is smaller than in figure 4 and 5.

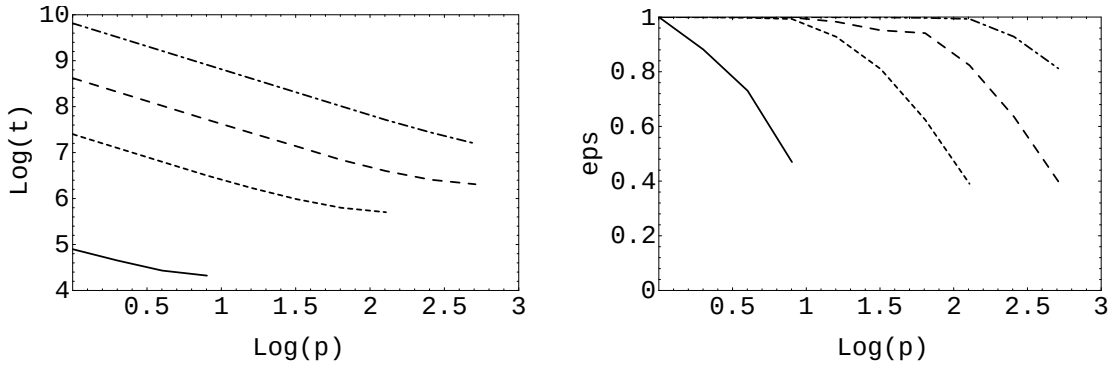


Figure 4: The total execution time (left figure), and efficiency (right figure) of one iteration of the CGNR as a function of the number of processors p; the execution time is in seconds. The solid line is for a simulation with 8 dipoles, the dotted line is with 136 dipoles, the dashed line is with 512 dipoles and the dotted-dashed line is with 2176 dipoles.

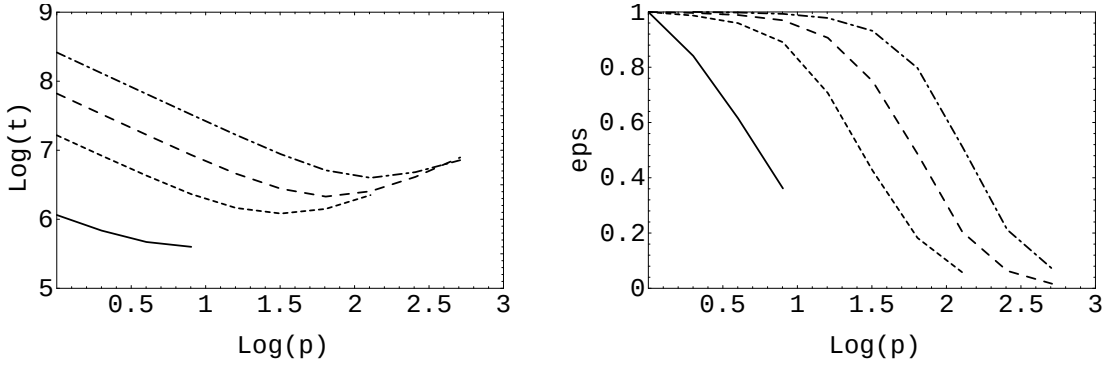


Figure 5: The total execution time (left figure), and efficiency (right figure) of the calculation of the scattered electric field as a function of the number of processors p; the execution time is in seconds; the field was calculated for scattering angles from 0^o to 180^o, with steps of 0.1^o. The solid line is for a simulation with 8 dipoles, the dotted line is with 136 dipoles, the dashed line is with 512 dipoles and the dotted-dashed line is with 2176 dipoles.

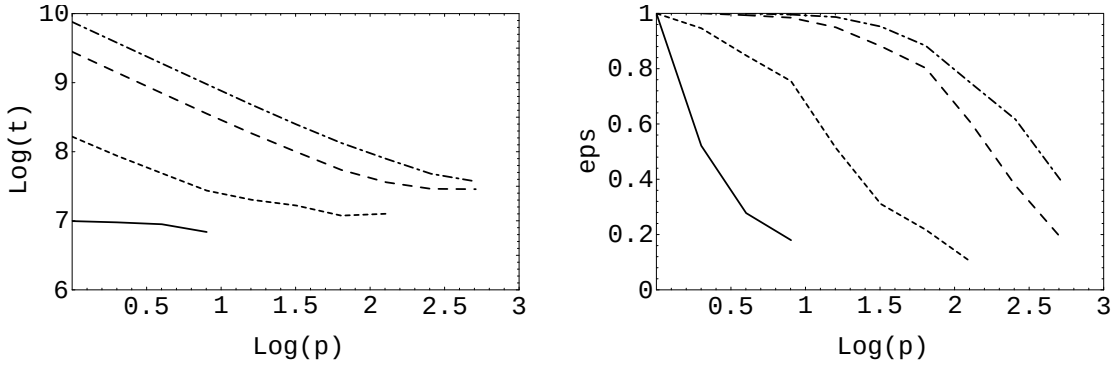


Figure 6: The total execution time (left figure), and efficiency (right figure) of a total coupled dipole simulation, including initialisations and i/o, as a function of the number of processors p; the execution time is in seconds. The solid line is for a simulation with 8 dipoles, the dotted line is with 136 dipoles, the dashed line is with 512 dipoles and the dotted-dashed line is with 912 dipoles.

Figure 7 shows the result of a CD simulation of scattering by sphere, together with the analytical Mie result. The number of dipoles was 33552 (corresponding to l = 17), the diameter of the dipoles was $\lambda/10$, resulting in a size parameter α = 12.6. This calculation required 11 hours on 512 processors. The parallel speedup was estimated to be larger than 500, the computational speed was 250 Mflop/s. As soon as new updates of the C compilers are available, this number is expected to be increased with a factor 2 (due to much better code optimisers).

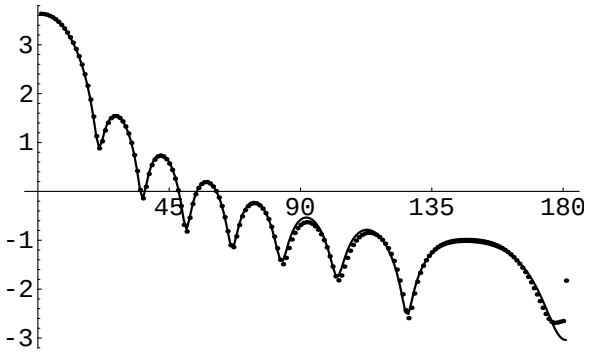


Figure 7: Coupled Dipole simulation (dots) of scattering by a sphere, and Mie calculations (line). The S_{11} element of the scattering matrix S, as a function of the scattering angle is shown. The size parameter of the sphere is α = 12.6, the refractive index is m = 1.05. The number of dipoles in the CD simulation was 33552, the size of the dipoles was $\lambda/10$.

5.5 DISCUSSION AND CONCLUSIONS

We want to simulate ELS from human white bloodcells and bonemarrow 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 bloodcells (e.g. between the three forms of Granulocytes), or between malign and healthy cells.

Approximate theories for ELS, such as Rayleigh-Debye-Gans (RDG) scattering [10], are sometimes very successful to describe certain properties of ELS from white bloodcells. For instance, we showed that a modified form of RDG scattering [35] can describe anomalous scattering behaviour of osmotically active Lymphocytes very well [36], even giving rise to new biophysical insights concerning the biomechanics of Lymphocytes [37]. However, more subtle properties of ELS, involving the polarisation of the scattered light, cannot be described with these approximate theories [1]. It are exactly these properties that play a key role to distinguish between subsets of white bloodcells, 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 promted 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 [14], 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.

Figure 1 shows that our simulations require $O(10^5)$ or more dipoles (white bloodcells: $20 < \alpha < 60$). This implies that we have to solve a very large, dense system of equations (equation 4). According to Edelman [38] our desired simulation can be shared among the largest Computational Electromagnetics simulations reported to date. To solve these large simulations requires 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 very 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 bloodcells, including biological variability?

The calculation of the electric field on the dipoles, i.e. solving equation 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 on a 64 node Meiko Computing Surface [22]. The main conclusion of this work is summarised by equation 9. If each processors contains many dipoles (i.e. the grainsize $n/p >> 1$), then the efficiency can be very close to 1. On the Parsytec we measured GCel $\tau_{comm}/\tau_{calc} = O(1)$ (data not shown). Therefore, if $n/p = 3N/p > \sim 100$, then $\varepsilon > 0.99$; i.e. the efficiency will be very close to 1 if each processor contains approximately 30 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 good on the full domain of the Parsytec ($p = 512$ processors). The performance measurements shown in figure 4 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 grainsizes

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 quadratic with the number of dipoles and is inversely proportional with the number of processors (see section 5.3.2). Note that the measured efficiencies for the simulations with 8 dipoles on 8 processors, 136 dipoles on 128 processors, and 512 dipoles on 512 processors all are approximately the same: $\varepsilon \sim 0.4$. This is a nice demonstration of the fact that the efficiency is, in this case, mainly determined by the grainsize of the parallel program; the number of dipoles per processor was, in all cases, one. The largest simulation for which we measured the time for one iteration on one - and more processors was with 2176 dipoles. For p = 128 (17 dipoles per processor) we measured $\varepsilon = 0.99$, and even on the full domain (p = 512, 4 or 5 dipoles per processor) we measured $\varepsilon = 0.81$.

The parallel calculation of the scattered field is not as efficient as the parallel implementation of the CGNR (see figures 4 and 5). 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 5.3.3, 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 dipoles. Therefore, the total time needed for the parallel calculation of the scattered fields is: $T = c_1 N/p + c_2 (p-1)$, 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 figure 5. 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 figure 5. 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 grainsize, 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, booting a partition of the GCel machine is only weakly depended on the number processors in the partition. 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 figure 6, show the expected behaviour. For a very small simulation (N = 8, the solid line) the total execution time is completely dominated by i/o: the execution time is constant. However, increasing N leads to still better efficiencies, and the largest simulation which was timed (N = 912, 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 grainsize), the efficiency can be made better than 0.99, even for 512 processors.

Figure 7 shows the results of a large simulation. We could not carry out simulations with more dipoles due to memory limitations. The simulation results compare very well with the analytical Mie calculations (this is also true for the other matrix elements; data not shown). Only in the backscattering 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.

A huge drawback is the execution time. Currently we achieve speeds of 250 Mflop/s, resulting in a simulation time of 11 hours. The execution time of one iteration of the CGNR method scales quadratic 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. Currently we are investigating this function, by looking at the condition number of the interaction matrix **A** (see equation 5). However, in the sequel of this discussion we will neglect this function.

Realistic simulations of white bloodcells require a factor three to ten more dipoles, i.e. a 10 to 100 times increase in execution time. On our Parsytec this would lead to execution times of 100 to 1000 hours. Even on the fastest computers available today, running 1000 times faster than the Parsytec, simulation times will be in the order of hours. As was shown by Singham [28], 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 is that CD simulations containing $O(10^4)$ dipoles can be performed on powerful MPP systems. For instance, the simulation of ELS from the sphere, as presented in section 5.4, would take approximately 8 minutes on e.g. a 4096 node T9000 system. Here we assume that the T9000 transputer runs 10 times faster than the T805 transputers [34], and that the efficiency of the parallel CD is very close to 1 on this machine. However, simulations with $O(10^5)$ or even $O(10^6)$ dipoles, as we require, are currently not possible.

The execution time of CD simulations, using iterative solvers, scales as N^2. This is due to the matrix vector products in the CGNR method. From a

physical point of view this 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 LogN) or even to O(N), are the so-called hierarchical tree methods [39, 40]. 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. We will develop such a hierarchical tree algorithm for the vector potential of radiating dipoles (in three dimensions). This algorithm replaces the matrix vector products in the iterative solver of the CD simulation. Now the interaction between the dipoles is not calculated for each dipole pair directly, but the dipoles are grouped together in a hierarchical way, and the interaction between single dipoles and this hierarchy of dipolar groups is calculated. In this way the complexity of the complete CD simulation can be reduced to O(N). This would facilitate CD simulations with $O(10^5)$ or more dipoles on high end parallel computing systems, and light scattering simulations form smaller arbitrary particles on workstations or personal computers.

5.6 REFERENCES

1] P.M.A. Sloot, A.G. Hoekstra, H. van der Liet, and C.G. Figdor, "Scattering matrix elements of biological particles measured in a flow through system: theory and practice," Applied Optics **28**, 1752-1762 (1989).

2] B.G. de Grooth, L.W.M.M. Terstappen, G.J. Puppels, and J. Greve, "Light-Scattering Polarisation Measurements as a New Parameter in Flow Cytometry," Cytometry **8**, 539-544 (1987).

3] J.I. Hage and J.M. Greenberg, "A model for the optical properties of porous grains," Astrophysical Journal **361**, 251 (1990).

4] J.I. Hage, M. Greenberg, and R.T. Wang, "Scattering from arbitrarily shaped particles: theory and experiment," Applied Optics **30**, 1141-1152 (1991).

5] T.T. Charampopoulos, D.W. Hahn, and H. Chang, "Role of metal additives in light scattering from flame particulates," Applied Optics **31**, 6519-6528 (1992).

6] C.M. Sorensen, J. Cai, and N. Lu, "Light scattering measurements of monomer size, monomer per aggregate, and fractal dimension for soot aggregates in flames," Applied Optics **31**, 6547-6557 (1992).

7] W.L. Eberhard, "Ice-cloud depolarisation of backscatter CO_2 and other infrared lidars," Applied Optics **31**, 6485-6490 (1992).

8] I. Colbeck, E.J. Hardman, and R.M. Harrison, "Optical and dynamical properties of fractal clusters of carbonaceous smoke," J. Aerosol Sci. **20**, 765-774 (1989).

9] P.W. Barber and S.C. Hill, *Light Scattering by Particles: Computational Methods* (World Scientific, 1990).

10] C.F. Bohren and D.R. Huffman, *Absorption and Scattering of Light by Small Particles* (John Wiley & Sons, 1983).

11] D.W. Shuerman, *Light Scattering by Irregularly Shaped Particles* (Plenum Press, New York and London, 1980).

12] E.M. Purcell and C.R. Pennypacker, "Scattering and absorption of light by nonspherical dielectric grains," The Astrophysical Journal **186**, 705-714 (1973).

13] J. Hage, *The optics of porous grains and the nature of comets*, Ph.D. dissertation, University of Leiden, Leiden, the Netherlands, 1991.

14] A. Lakhtakia, "Macroscopic theory of the coupled dipole approximation method," Opt. Comm. **79**, 1-5 (1990).
15] B.T. Draine, "The discrete dipole approximation and its application to interstellar graphite grains," Astrophys. J. **333**, 848-872 (1988).
16] S.B. Singham and C.F. Bohren, "Light scattering by an arbitrary particle: a physical reformulation of the coupled dipole method," Optics Letters **12**, 10-12 (1987).
17] P.J. Flatau, G.L. Stephens, and B.T. Draine, "Light scattering by rectangular solids in the discrete-dipole approximation: a new algorithm exploiting the Block-Toeplitz structure," J. Opt. Soc. Am. A **7**, 593-600 (1990).
18] A. Lakhtakia, "Strong and weak forms of the method of moments and the coupled dipole method for scattering of time-harmonic electromagnetic fields," Int. J. Mod. Phys. C **3**, 583-603 (1992).
19] A. Lakhtakia, "General theory of the Purcell-Pennypacker scattering approach and its extension to bianisotropic scatterers," Astrophysical Journal **394**, 494-499 (1992).
20] C.E. Dungey and C.F. Bohren, "Light scattering by nonspherical particles: a refinement to the coupled-dipole method," J. Opt. Soc. Am. **8**, 81-87 (1991).
21] A.G. Hoekstra and P.M.A. Sloot, "Implementation of a parallel conjugate gradient method for simulation of elastic light scattering," Computers in Physics **6**, 323 (1992).
22] A.G. Hoekstra, P.M.A. Sloot, W. Hoffmann, and L.O. Hertzberger, *Time complexity of a parallel Conjugate Gradient solver for light scattering simulations: theory and SPMD implementation,* Tech. Rept. CS-92-06 Faculty of Mathematics and Computer Science, University of Amsterdam, 1992.
23] A.G. Hoekstra and P.M.A. Sloot, "Dipolar Unit Size in Coupled-Dipole Calculations of the Scattering Matrix Elements," Optics Letters **18**, 1211-1213 (1993).
24] J.D. Jackson, *Classical Electrodynamics, second edition* (John Wiley & Sons, New York, Chichester, Brisbane, Toronto, Singapore, 1975).
25] G.H. Goedecke and S. O'Brien, "Scattering by irregular inhomogeneous particles via the digitised Green's function algorithm," Applied Optics **27**, 2431-2438 (1988).
26] D.E. Livesay and C. Kun-Mu, "Electromagnetic fields induced inside arbitrarily biological bodies," IEEE trans. microwave theory **mtt-22**, 1273-1280 (1974).
27] S.B. Singham and G.C. Salzman, "Evaluation of the scattering matrix of an arbitrary particle using the coupled dipole approximation," J. Chem. Phys. **84**, 2658-2667 (1986).
28] S.B. Singham and C.F. Bohren, "Light scattering by an arbitrary particle: the scattering-order formulation of the coupled-dipole method," J. Opt. Soc. Am. A **5**, 1867-1872 (1988).
29] G.H. Golub and C.F. van Loan, *Matrix Computations* (The John Hopkins University Press, Baltimore and London, 1989).
30] S.F. Ashby, T.A. Manteuffel, and P.E. Saylor, "A taxonomy for conjugate gradient methods," Siam J. Numer. Anal. **27**, 1542-1568 (1990).
31] 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).
32] Parsytec GmbH, *Parix 1.2 User Manual*, Germany, 1993.
33] Inmos Limited, *Occam®2 Reference Manual* (Prentice Hall, 1988).
34] M.D. May, P.W. Thompson, and P.H. Welch, *Networks, Routers and Transputers* (IOS Press, Amsterdam, Oxford, Burke, 1993).
35] P.M.A. Sloot and C.G. Figdor, "Elastic light scattering from nucleated blood cells: rapid numerical analysis," Applied Optics **25**, 3559-3565 (1986).
36] P.M.A. Sloot, A.G. Hoekstra, and C.G. Figdor, "Osmotic Response of Lymphocytes Measured by Means of Forward Light Scattering: Theoretical Considerations," Cytometry **9**, 636-641 (1988).
37] A.G. Hoekstra, J.A. Aten, and P.M.A. Sloot, "Effect of aniosmotic media on the T-lymphocyte nucleus," Biophysical Journal **59**, 765-774 (1991).
38] A. Edelman, "Large Dense Numerical Algebra in 1993: the parallel computing influence," Intern. J. Supercomputer Appl. **7**, 113-128 (1993).
39] L.L. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems* (The MIT Press, 1988).
40] J.K. Salmon, *Parallel Hierarchical N-Body Methods*, Ph.D. dissertation, California Institute of Technology, Pasadena, USA, 1991.

Chapter 6

A COMPARISON OF NATIVE AND GENERIC COMPUTING ENVIRONMENTS FOR A TRANSPUTER PLATFORM

A comparison of native and generic programming environments for a transputer platform

ABSTRACT

We compare the Iserver-Occam, Parix, and Express parallel programming environments on a 512 node Parsytec GCel. The comparison is made by a detailed analysis of the performance of a particular application. In our approach we start with the application and isolate the basic (environment) dependent building blocks. These basic building blocks, which depend on floating point performance and communication capabilities of the environments, are analysed independently. We have measured point to point communication times, global communication times and floating point performance. All information is combined into a time complexity analysis, allowing us to compare the environments on all relevant degrees of functionality. Together with demands for portability of the code, and development time (i.e. programmability), an overall judgement of the environments can be made.

6.1 INTRODUCTION

Real success of Massively Parallel Processing critically depends on programmability of the parallel computers and on portability of parallel programs. We are made to believe that “parallel computing has come to age”. Although it is safe to say that parallel hardware has reached a convincing stage of maturity, both programmability of the parallel hardware and portability of parallel programs still pose serious problems to developers of parallel applications.

Today, an application programmer is usually faced with a situation as drawn in figure. 1. A parallel computing platform supports native environments, which allow very low level programming, or allow a more abstract view of the hardware. Furthermore, generic environments, also available on other platforms, can be used. These environments can be grouped in order of decreasing hardware visibility and increasing portability. Of course, one expects that the price to be paid for portability is a decrease of control of the hardware and associated degradation of performance.

A typical example is provided by the developments in transputer based parallel processing, to which we will restrict ourselves in this paper. The first generation of these parallel systems consisted of transputers hardwired into grids, and had to be programmed in the transputer’s native language Occam.

The programmer had to now all the details of the parallel hardware, routing of messages had to be done explicitly, and the topology of the network was fixed. The next generation allowed software reconfigurable topologies and programming of the system in the standard languages C and Fortran 77, extended with message passing primitives. Furthermore, programming environments, like e.g. CS-Tools [1], allowed to send messages between processes, not necessarily located on adjacent processors: the routing was done implicitly by the system. The next step should have been systems based on the new T9000 transputer. The delay in production of this chip forced manufacturers to turn to other chips, or downgrade their new systems to T805 transputers and emulate the T9000 Virtual Channel Routing [2] in software. This resulted in e.g. the GCel series of Parstyec [3], with its programming environment Parix [4]. Here concurrent processes communicate via virtual channels and the machine can be configured into virtual topologies.

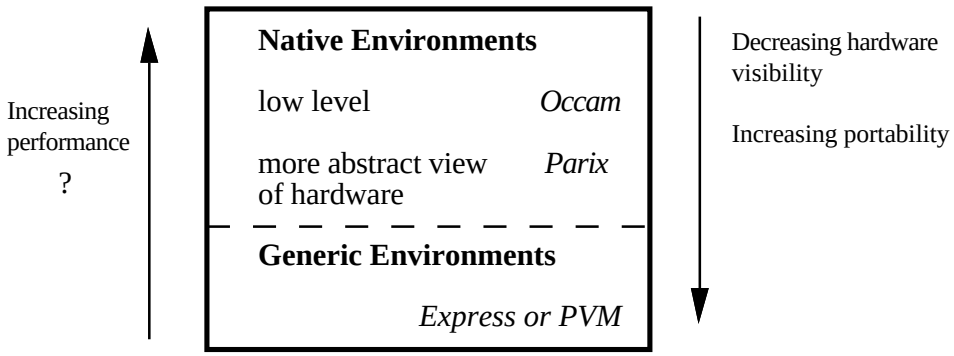


Figure 1: a typical situation encountered by application programmers of parallel systems.

Despite these efforts, portability of parallel programs developed in native transputer environments, to other parallel computing systems, is rather poor. Fortunately, parallel programming environments have been developed which are supported on a large number of different parallel computers. These environments carry the potential of true source level portability of parallel programs between very different types of parallel systems, including clusters of workstations and heterogeneous systems. Two very popular environments are now also available on transputer based systems. First, the Express [5] system of Parasoft, which is available on e.g. networks of workstations, or the Intel Paragon, is also available on transputer systems. Secondly, the PVM environment [6], which seems to become *the* standard parallel programming environment, has been ported to the Parsytec GCel system by our group [7].

In this paper we will compare two native environments, Iserver-Occam and Parix, with one generic environment, Express. All experiments are executed on the Parsytec GCel. In this way we can judge the trade-off which clearly exists between native environments, usually offering a better performance at the price of extensive programming effort, and generic environments which allow to develop more portable programs. Currently we are running benchmarks of PVM on the Parsytec GCel, and these results will be available soon.

As a representative case study we have implemented an application from Physics, i.e. Elastic Light Scattering simulations using the Coupled Dipole

method [8, 9], in these three environments on the Parsytec GCel. This application has the following characteristics:

- It is a real application. This means that the application is actually used for simulations.
- The time complexity of the program is predictable. The execution time of the program can be expressed in terms of problem size, number of processors, and a small set of basic system parameters. This allows a first comparison of the environments by measuring this very limited set of parameters.
- It contains global communication routines. Global communication requires routing of messages to all processors. Explicit coding of this in e.g. Occam is an extensive programming effort, which is not necessary in environments like Express.
- The implementation does not exhibit (severe) load imbalance. Load balancing is a research area by itself, and would obscure our current experiment. The Parsytec GCel is a monolithic platform, i.e. all processors have the same capabilities, unlike e.g. networks of workstations. Therefore, the only possible source of load imbalance in our experiment would be the program itself.

We will analyse the behaviour of the parallel Coupled Dipole method in the three environments by analysing basic and global communication routines, floating point performance, and actual execution times of the parallel program as a function of problem size and number of processors. We will investigate if the basic measurements can predict the runtime of the application, and if such basic measurements can be used as a heuristic to assess the merits of a programming environment.

6.2 THE HARDWARE AND SOFTWARE ENVIRONMENT

6.2.1 The Parsytec GCel

The Parsytec GCel-3/512, which was installed at IC^{3}A* in Amsterdam in January 1993, consists of 512 T805 transputer, running at 30 MHz, with 4 Mbyte external RAM each. The transputers are physically connected into a 16×32 two dimensional grid. The architecture of the machine is drawn schematically in figure 2.

* IC^{3}A is the Interdisciplinary Centre for Computer based Complex systems research Amsterdam. For more information, please contact the authors

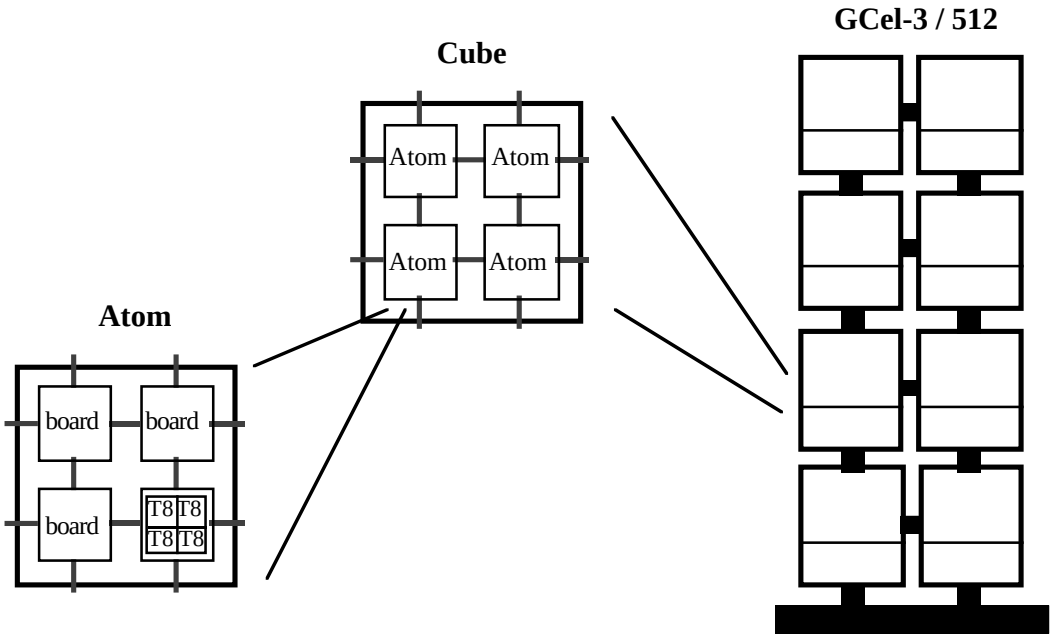


Figure 2: Schematic drawing of architecture of the Parsytec GCel-3/512 installation.

The basic logical unit of the GCel is an “atom”, constituting of a total of 16 transputer. An atom consists of 4 boards each equipped with 4 transputers. The basic building block is a “cube”, containing 4 atoms, i.e. 64 transputers. GCel installations consist of multiple cubes, in our case 8.

All atoms are arranged in a large number of overlapping partitions. Users can request a partition. In this way the user can get parts of the machine, containing multiples of 16 transputers. Users enter the machine via a front-end, a Sun Sparc 10 workstation.

6.2.2 Parix

The standard programming environment for the GCel is Parix [4], an acronym for parallel extensions to Unix. All experiments described here are performed under Parix release 1.2. Parix is a cross-development system. Applications are developed and compiled on the front-end, and subsequently downloaded on the transputers. The programming model is Single Program Multiple Data (SPMD) [4]; 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. Furthermore, each processor can load different code segments during runtime.

The processors synchronise, and exchange data, by means of message passing. Communication is performed through so-called virtual links, and can be synchronous and a-synchronous. Every processor can define a virtual link to any other processor in the network. Parix takes care of routing the messages through the network. A set of virtual links between different processors can be grouped into virtual topologies. Users can define their own virtual topologies, or can use existing virtual topologies, such as rings, meshes, or trees, by calling appropriate functions from the virtual topology library of Parix. These library

functions guarantee an optimal mapping of the virtual topology on the actual physical two dimensional network of the GCel computer. Messages can also be exchanged without using virtual links, however this mechanism, which heavily relies on the Virtual Channel Routing capabilities of the T9000 transputer, is not very efficient on the T805 hardware. Furthermore, new releases of Parix contain global communication routines, such as broadcasts, global sums etc. However, during development of the application these routines were not yet available, and are therefore not included in the experiments described in this paper.

Parix programs can be developed in ansi C or Fortran 77, and all standard Unix tools can be used. Communication between processors is performed through library calls. Input and output to the front-end is performed through remote procedure calls to appropriate Unix system calls, and communication with the front-end is also possible through Unix sockets. Parix 1.2 is equipped with a performance analyser called Patop. A parallel debugger is not yet present.

6.2.3 Iserver-Occam

The Iserver-Occam environment consists of the Occam toolset to develop Occam 2 programs and run them on, in our case, the Parsytec GCel. Occam and the transputer were developed together, and Occam can be considered the native language of the transputer. Occam allows maximal exploitation of the transputer hardware. Occam programs are developed and compiled (compiler version d7205) on the Unix front end, and subsequently downloaded on the Parsytec. Mapping of transputer links to Occam channels, and configuration of the network is carried out by a special configuration file, which is included in the Occam program.

Occam can only handle synchronous communication through Occam channels. If such a channel is mapped onto a transputer link, parallel processes running on neighbouring transputers can exchange data. In order to send data to an arbitrary processor the programmer has to route the data explicitly through the network.

6.2.4 Express

The Express system [5] is a product of Parasoft, and is based on the early work of Fox et al. [10]. Express is available on a wide range of platforms including Suns, Cray systems, and Meiko systems. Currently we have a β version running directly on the transputers of the Parsytec GCel. Currently we are investigating the possibility to put Express on top of Parix. In this way, Express itself will be portable to all platforms running Parix.

Express consists of a set of libraries to describe the parallelism in the program. For instance, a communication library is present, offering primitives for sending messages between nodes, and high-level global communication routines such as broadcast and global data gathering. Furthermore, Express contains a parallel graphics system, that offers a variety of graphical functions to all nodes in the system. Express supports C and Fortran77 programming,

and contains a number of programming tools such as a parallel debugger and a graphical performance analyser.

Express offers two different working models: the host-node model and the so-called cubix model. In the host-node model one dedicated node, usually the front-end machine of the parallel system, starts and controls the computation on the parallel nodes, and all I/O operations have to be performed by the host. In the cubix model a dedicated host node does not exist, and all work is performed by the parallel nodes. Operating systems services that the parallel nodes may require are transparently redirected to the front-end. The cubix model implies SPMD programming and is comparable with Parix' programming model. The Coupled Dipole program is implemented in the cubix model.

A cubix program is executed as follows. First a partition is booted in the Parsytec GCel. Next, Express is initialised by loading the Express kernel and the routing tables on each transputer in the partition. Finally the program is started by loading it on a user specified number of processors inside the Express partition.

On the Parsytec version the user has no control over the mapping of the parallel processes into the partition. Furthermore, the physical location in the partition is unknown, only process identification numbers can be assessed. However, by using e.g. the exgridinit call the parallel processes can order themselves into N dimensional grids and communicate in for instance the left or north direction. This is comparable to the virtual topologies of Parix.

6.3 THE COUPLED DIPOLE APPLICATION

6.3.1 Functional aspects

The Coupled Dipole (CD) method [11] simulates Elastic Light Scattering from arbitrary particles. The particle is discretised in N small subvolumes, called dipoles. The simulation consists of two stages. First, the internal electric field on the dipoles is calculated, and secondly the scattered electric field is calculated using the previously obtained internal field. The first stage can be formulated as a matrix equation $\mathbf{Ax} = \mathbf{b}$, with $\mathbf{A}$ a dense, complex symmetric $3N \times 3N$ interaction matrix, $\mathbf{b}$ a known vector of length 3N, and $\mathbf{x}$ the wanted vector of length 3N containing the internal electric field. To solve this matrix equation is the most demanding part of a CD simulation. The equation is solved by means of the CGNR method [12], which is a Conjugate Gradient method suited for complex non-Hermitian matrices.

The scattered field is calculated by summation of the radiation of all N dipoles at the observation point. For every dipole one must calculate a matrix vector product $\mathbf{Bx}_i$, with $\mathbf{B}$ a 3×3 complex matrix and $\mathbf{x}_i$ a subvector from the large vector $\mathbf{x}$, with length 3. The main computational difficulty of the CD method lies in the fact that for realistic simulations the number of dipoles N has to be very large [8]; typical values for N are 10^4 to 10^5. The calculation of the internal field is the most demanding part of the CD simulation.

We have developed a parallel version of the CD method [8]. Parallelism was introduced by means of a data decomposition: each processor receives N/p dipoles (p is the number of processors). In conjunction with the CGNR method, this decomposition is equivalent with a row-block decomposition of the interaction matrix **A** [8]. We have studied in great detail parallel versions of the CGNR method [13], and concluded that parallel CGNR with a rowblock decomposition is very efficient if the number of rows (i.e. the number of dipoles) per processor is large enough.

The CGNR method is an iterative method to solve matrix equations. One iteration contains two matrix vector products, three vector updates (**y** = a**x** + **y**) and three inner products (r = **x** • **x**). The parallel implementation of the CGNR method and the complete Coupled Dipole method, using a rowblock decomposition of the interaction matrix, are described in chapter 4 and 5 of this thesis.

Assume that one floating point operations takes τ_{calc} μs. The execution time for the parallel vector update is (remember, all elements are complex numbers)

$$T^{vu}(N,p) = 24\left\lceil \frac{N}{p} \right\rceil \tau_{calc} , \qquad [1]$$

with $\lceil x \rceil$ the ceiling function of x. The parallel inner product has an execution time

$$T^{ip}(N,p) = \left(24\left\lceil \frac{N}{p} \right\rceil - 2 \right) \tau_{calc} + t_{sa} , \qquad [2]$$

with t_{sa} the time to perform the scalar accumulate operation. Finally, the execution time of the parallel matrix vector product with row block decomposition of the matrix is

$$T^{mv}(N,p) = 3\left\lceil \frac{N}{p} \right\rceil (24N-2) \tau_{calc} + t_{vg} , \qquad [3]$$

with t_{vg} the time for the vector gather operation. In real CD simulations, the decomposed matrix **A** cannot be kept in local memory of the processors. Fortunately we can compute the matrix elements as they are needed. This is exactly what happens in the experiments described in this paper. Therefore, an additional time to compute the matrix elements has to be included in Eq. 3. Every 3×3 block in **A**, describing an interaction between two dipoles, requires 107 floating point operations to be calculated. This results in

$$T^{mv}(N,p) = 3\left\lceil \frac{N}{p} \right\rceil (24N-2) \tau_{calc} + 107N\left\lceil \frac{N}{p} \right\rceil \tau_{calc} + t_{vg} . \qquad [4]$$

Assume that the scattered field is calculated at w observation points (for instance scattered field as a function of scattering angle). This means that

every processor has to calculate the contribution of its local dipoles to the scattered field at w observation points. To find the total scattered field, the fields calculated in all processors must be accumulated, and summed. The total execution time for the parallel calculation of the scattered fields therefore is

$$T^{sf}(N,p) = w\left\lceil \frac{N}{p} \right\rceil (66+107)\tau_{calc} + t_{va} \ . \qquad [5]$$

The factor 66 is due to the complex matrix vector product $\mathbf{Bx}_i$, and the factor 107 is due to the calculation of the matrix elements of $\mathbf{B}$.

The time for one iteration of the CGNR method is

$$T^{iter} = 2T^{mv} + 3T^{vu} + 3T^{ip} \ . \qquad [6]$$

Assume that the CGNR method requires v iteration to find the solution vector $\mathbf{x}$. Then, the total execution time for one Coupled Dipole simulation T^{CD} is

$$T^{CD} = vT^{iter} + T^{sf} + T^{IO} \ . \qquad [7]$$

Here we included the time T^{IO} needed for Input-Output operations.

We have now specified the execution time of the parallel Coupled Dipole simulation in terms of floating point operations and global communications. Both the global communications and the cost of a floating point operation depend on the specific programming environment. If we can measure τ_{calc} and the global communication cost we can estimate the performance and scalabilty of the parallel simulation.

6.3.2 Implementation

6.3.2.1 General

Starting with the sequential Coupled Dipole method C-source code, the parallel implementation in Parix and Express is straightforward. First, some bookkeeping code to establish the data decomposition must be implemented. Subsequently, the loop indices in the matrix vector products, inner products and vector updates must be adapted to match the data decomposition. These additions to the source code are independent of the programming environment, except for one call to get the number of processors available in the network, and to find the location of each processor in the network.

Next, the communication routines must be included into the code. Before executing a parallel matrix vector product the argument vector must be gathered in each processor. After calculating partial inner products in parallel a scalar accumulate operation must be performed. Finally, after calculating the scattered field of the local dipoles in parallel, the resulting field must be accumulated. The implementation of these routines strongly depends on the programming environment, and will be discussed in the following sections.

The parallel Coupled Dipole method was first implemented on a Meiko Computing surface, containing 64 T800 transputers [13]. This implementation

was in Occam, in the Occam Programming System. This code was ported to the Iserver environment to run on the Parsytec GCel. The structure of the implementation in Parix (in the language C) strongly resembles the Occam version. The Parix version was ported to Express, by adapting the communication routines and part of the bookkeeping code. We will first discuss the Occam version, followed by the Parix and Express versions of the program.

6.3.2.2 The Iserver-Occam program

The Coupled Dipole method with the rowblock decomposition of the interaction matrix was implemented on a bi-directional ring topology. The vector gather operation is implemented as follows:
1) In the first step every processor sends its local part of the vector to the left and the right processor and, at the same time, receives from the left and the right processor their local part.
2) In the following steps, the parts received in a previous step, are passed on from left to right and vice versa, and in parallel, parts coming from left and right are received and stored.
After $\lfloor p/2 \rfloor$ steps (with $\lfloor x \rfloor$ the floor function of x), every processor in the ring has received the total vector. The scalar - and vector accumulate are implemented likewise, but in addition the processor must summate the received data with its local data.

The pseudo Occam code for the vector gather operation is given below:

```
Vector Gather
PAR
  SEQ -- from right to left
    PAR -- start the communication
      send to left
      receive from right
    PAR i=1 FOR "p/2-1" -- pass data from right to left
      send to left which was received from right in the previous step
      receive from right
  SEQ -- from left to right
    PAR -- start the communication
      send to right
      receive from left
    PAR i=1 TO "p/2-1" -- pass data from left to right
      send to right which was received from left in the previous step
      receive from left
```

The operation consists of two parallel branches, one receiving data from the right and sending data to the left, the other vice-versa. Each branch consists of a starting up of the communication, which is the first PAR inside the SEQ branches, subsequently the gather operation is carried out in the replicated PAR operation. The Occam language is very powerful to express such complicated communication patterns. Due to the close relationship between Occam and the transputer, this part of the code runs very efficiently on a transputer (see section 6.4). The complete Occam source code of the vector gather operation however is much more complicated due to the special way we

had to store the vector with complex variables and due to the small load imbalance which is introduced if the number of dipoles N is not exactly divisible by the number of processors p.

In our application, all transputers run two processes, a router and a calculator. Router processes on neighbouring transputers are connected via a channel. These channels are associated with hardware transputer links via a configuration file. The router process calls communication routines, such as the vector gather operation. The calculator process performs the work on the decomposed data. This work is divided in cycles, at the end of every cycle a communication step occurs. The calculator sends a command, in the form of a single character, to the router process. The router process receives this character, interprets it and issues the desired communication routine. During this communication step the calculator process is idle. After finishing the communication, the router process sends a 'ready' signal to the calculator process, which then proceeds with the next cycle in the algorithm.

The original implementation of the Occam code for the Meiko system was carried out in approximately 8 months, porting of the code to the Iserver environment took approximately 3 weeks. The total Iserver-Occam source code contains 4500 lines.

6.3.2.3 The Parix program

The structure of the Parix implementation, developed in C, is similar to the Occam program. A virtual ring topology is defined by a call to the virtual topology library, and two threads are started on each transputer, a calculator and a router thread. The main program is globally shown below:

```
main ()
{
        /* main of parallel CD method */
        MakeRing (...)          /* Create Ring Topology */
        GetRing_Data (...);     /* Extract information of Ring topology */
        StartThread (Calculator,...);       /* Start the Calculator thread */
        StartThread (Router,...);           /* Start the Router thread */
        Wait for threads to terminate
}
```

First, the ring topology is established, next the location in the ring is found by the GetRing_Data call, and the two threads are started.

We could have mimicked the vector gather operation as described in the previous section using Parix' asynchronous communication calls, as shown below:

```
Vector Gather
{
        /* start the communication */
        Asend (to right);
        ARecv (from left);
        ASend (to left);
        ARecv (from right)
```

```
        Synchronise, wait until all asynchronous communication is ready

        /* pass data from left to right and vice-versa */
        for (i=1 to "p/2 -1") {
                Asend (to right);
                ARecv (from left);
                ASend (to left);
                ARecv (from right)
                Synchronise, wait until all asynchronous communication is ready
        }
}
```

To start an asynchronous communication in Parix is very expensive, and would result in prohibitive startup costs of this vector gather operation. Therefore we redesigned the vector gather operation. First, the data is sent in just one direction, from left to right, and the send and receive operations are synchronous. To achieve this, the vector gather operation must start two threads: one to receive data from the left and a second to send the data to the right. Furthermore, a mechanism to synchronise the receiving and sending thread must be implemented.

The final vector gather operation is shown below:

```
Vector Gather
{
        StartThread (Send_to_Right,...);    /* Start sending thread */
        StartThread (Recv_from_Left,...); /* start receiving thread */
        Wait for threads to terminate
}

Send_to_Right
{
        /*start the communication */
        Send (to right);
        synchronise with receiving thread

        /* pass data from left to right */
        for (i=1 to p-1) {
                Send (to right);
                synchronise with receiving thread
        }
}

Recv_from_Left
{
        /*start the communication */
        Recv (from left);
        synchronise with sending thread

        /* pass data from left to right */
        for (i=1 to p-1) {
```

```
            Recv (from left);
            synchronise with sending thread
        }
    }
```

The synchronisation is implemented by establishing a virtual link between the sending and receiving thread, and communicating *ready* signals between both threads. This version is much more efficient, and is used in the actual Coupled Dipole implementation.

This parallel program was developed using the existing sequential C code, and the experience we gained during the development of the Occam code. The parallel program was developed in two weeks, the final version of the communication routines, as described above consumed two more weeks, and optimisation of the numerical parts of the application, especially the matrix vector product, lasted another three weeks. The complete C source code contains 3800 lines.

6.3.2.4 The Express program

The Express program is based on the optimised Parix C program. The main structure of the Express program however is very different from the Parix implementation. Express offers global communication routines which are actually used in our implementation. These global communications free the application programmer from the notion of a topology. The Express program consists of p parallel processes communicating via (global) message passing routines. The Express kernel takes care of placing the processes on processors and of the actual routing of messages through the network.

The Express program solely consists of the calculator process. The calls in the original Parix calculator to the router are replaced by calls to global communication routines of the Inter Process Communication library of Express. Of course we could have implemented the communication routines using constructs like the Parix program, but we feel that a typical application programmer *will* use the global communication routines if they are available. The numerical part of the code remained unchanged.

The vector gather operation is implemented using the exconcat function. This function concatenates the data of each participating node into a single buffer. If the data of each participating node is not equally sized (that is if N / p is not an integer), the resulting buffer must be compacted. The vector gather operation is now reduced to

```
Vector Gather
{
    exconcat ();
    if necessary, compact the resulting buffer
}
```

The accumulate operations are implemented using the excombine function which allows each node to combine data from all other nodes using a user supplied function (i.e. in this case addition).

Express has buffers for the communication routines. These buffers can be

adjusted by the user, and large buffers result in better communication characteristics. However, it turned out that we had to choose very small buffers of 1 kbyte to be able to run Express on large partitions. This resulted in a small degradation of the communication routines.

The time to port the Parix program to Express was three days, including reorganising the original code as described above. The resulting Express version of the Coupled Dipole code runs without change on a cluster of workstations, on the Meiko Computing Surface and the Parsytec GCel. The total source code contains 2400 lines.

6.4 RESULTS

6.4.1 Description of the measurements

The run time of the parallel Coupled Dipole implementation is determined by τ_{calc}, and the global communication times (see section 6.3.1). In the Iserver-Occam and Parix implementation the global routines are explicitly implemented using (nearest neighbour) point to point communications. The global routines in Express are also implemented using basic point to point communication. However, as an application programmer we do not exactly know how these global routines are actually implemented.

In order to compare the environments we have measured floating point performance (i.e. τ_{calc}), the basic communication routines, the global communication routines and finally the execution time of the parallel Coupled Dipole method.

6.4.2 Floating point performance

We have modelled the floating point performance with just one parameter τ_{calc} (section 6.3.1). Beside the raw power of the floating point unit of the T805, this parameter should also include the effects of loop overhead, memory access, cache behaviour etc. Clearly τ_{calc} depends on the type of operation that is performed, and one can argue that this single parameter model is to simple to predict the floating point performance. However, by keeping our specific application in mind, we can use this single parameter.

The numerical work in the Coupled Dipole method consists of matrix vector products, inner products, and vector updates. Since the largest portion of the numerical work is the matrix vector product, we have measured the execution time of the matrix vector product on 1 processor, as a function of the number of dipoles N. The parameter τ_{calc} was extracted from the measurements by fitting them with equation 4.

Figure 3 shows the results of the measurements, together with the fitted functions. The τ_{calc} that resulted from the fits are; for Parix τ_{calc} = 1.28 μs/flop, for Iserver-Occam τ_{calc} = 2.63 μs/flop, and for Express τ_{calc} = 1.72 μs/flop.

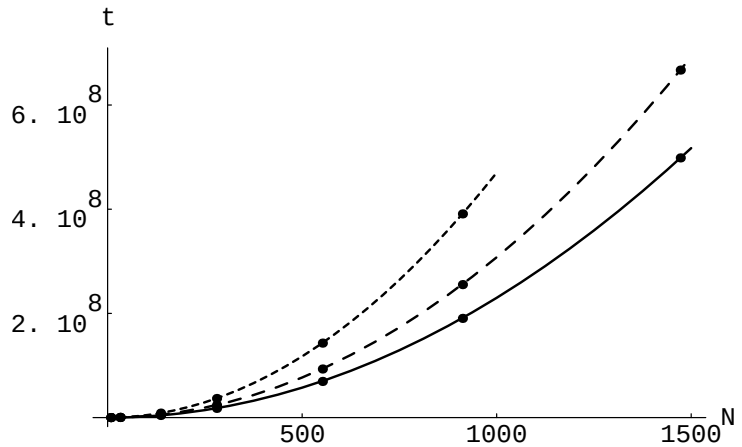


Figure 3: The execution time (in μs) of the matrix vector product on 1 processor, as a function of the number of dipoles. The dots are the measurements, the lines are the fits; the solid line is for Parix, the short-dashed line for Iserver-Occam, and the long-dashed line for Express.

6.4.3 Basic point to point communication

Point to point communication can be described with a linear two parameter model. The point to point communication time t_{pp} is

$$t_{pp} = \tau_{setup} + n\,\tau_{send}, \qquad [8]$$

with n the number of bytes, τ_{setup} a setup time to initialise and start the communication, and τ_{send} the send time to transfer 1 byte. Communication in Occam, and communication on virtual links and virtual topologies in Parix can be modelled with equation 8. In Express the situation is slightly complicated by the buffering which occurs during communication. If the message size is larger than the buffer size an extra overhead is introduced, which can be modelled by adding a term $\lfloor n_{buffer} / n \rfloor \tau_{buffer}$, with n_{buffer} the buffer size and τ_{buffer} the extra overhead time.

In Occam all communication is synchronous, and between neighbouring transputers. We have measured t_{pp} as a function of n. The result were perfectly linear (data not shown), and were fitted to equation 8 with a least squares method. We could distinguish three different situations: communication between transputers inside one atom (see figure 2), communication between neighbouring transputer in two adjacent atoms inside one cube, and communication between neighbouring transputer in two adjacent cubes. Table 1 shows the results of the fits, together with estimates of the error. The setup time is almost constant, the send time increases with increasing "distance" between the nodes.

which communication	τ_{send} (μs/byte)	τ_{setup} (μs)
inside atom	0.71 ± 0.01	3.73 ± 0.02
between atoms inside cube	0.87 ± 0.01	3.82 ± 0.02
between cubes	0.90 ± 0.01	3.84 ± 0.02

Table 1: Send - and setup times for nearest neighbour point to point communication for Iserver-Occam.

Parix's virtual links allow point to point communication between any node, the kernel of Parix routes the messages through the hardware. However, in the Coupled Dipole implementation (see section 6.3.2.3) the only point to point communication consists of synchronous send/receive pairs between adjacent processors in the virtual ring topology. Therefore we have only measured synchronous point to point communication between neighbouring processors in a virtual ring topology. In most cases the virtual ring can be mapped onto the physical two dimensional mesh such that neighbouring processors in the virtual ring are also neighbouring processors in the physical two dimensional mesh. We have analysed the same nearest neighbour communications as with Iserver-Occam. However, we could not distinguish significant differences between the results; in all cases we find $\tau_{send} = 0.92 \pm 0.02$ μs/byte and $\tau_{setup} = 67 \pm 2$ μs.

The analysis of point to point communication in Express is complicated by two effects. First, the localisation of the parallel processes is not known, and therefore we do not know if a communication is between physical neighbouring processors. Secondly, in the Express program we only use global communication primitives. Although these global communication functions are implemented using point to point communication, we do not know in detail *how* these global functions operate. In order to get a picture of the basic communication performance of Express we therefore measured *all* possible point to point communications between node zero and all other nodes in an Express partition, and analysed the distribution in the resulting setup - and send times.

Figure 4 shows in one figure the t_{pp} as a function of the number of transferred bytes for point to point communication between node 0 and nodes 1, 2, and 3 in a 4 processor partition. We can see one fast communication and two slower communication. Note that we can actually observe two different types of behaviour in the slower communication if the message size exceeds the buffer size. The nature of this behaviour is unknown to us. Table 2 shows the results of fitting the three data sets with equation 8.

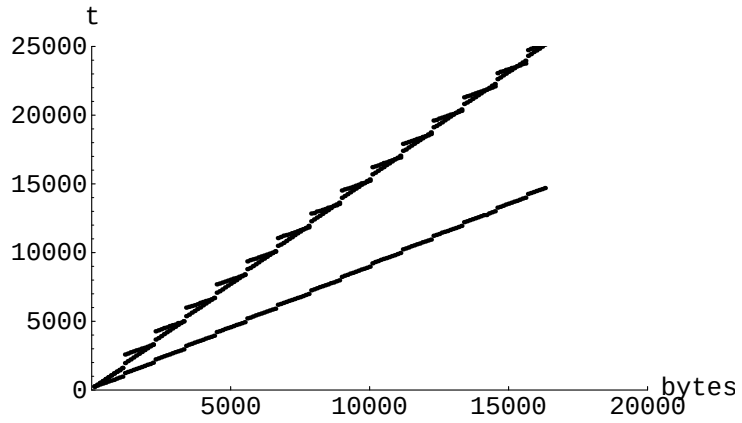


Figure 4: Point to point communication time (in μs) as a function of the number of transferred bytes, for communication from node 0 to node 1, 2, and 3 in a 4 node Express partition.

which communication	τ_{send} (μs/byte)	τ_{setup} (μs)
node 0 -> node 1	0.90 ± 0.01	88.2 ± 0.2
node 0 -> node 2	1.54 ± 0.01	21.7 ± 0.2
node 0 -> node 3	1.54 ± 0.01	290 ± 3

Table 2: Send and setup times for the point to point communications in a 4 node Express partition, see figure 4.

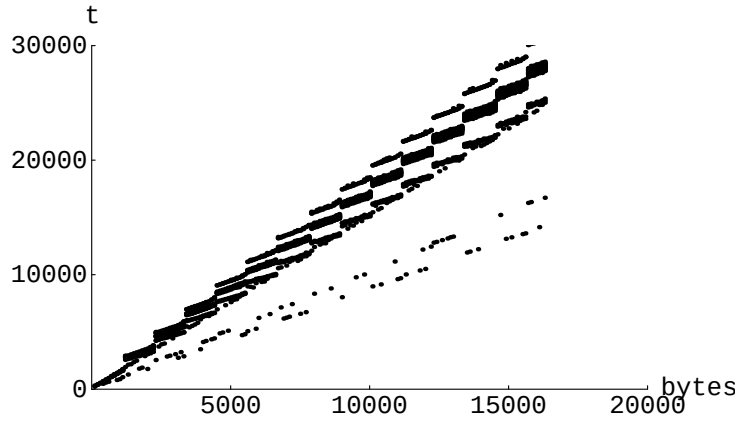


Figure 5: All measured point to point communication times (in μs) as a function of the number of transferred bytes for the 256 node Express partition.

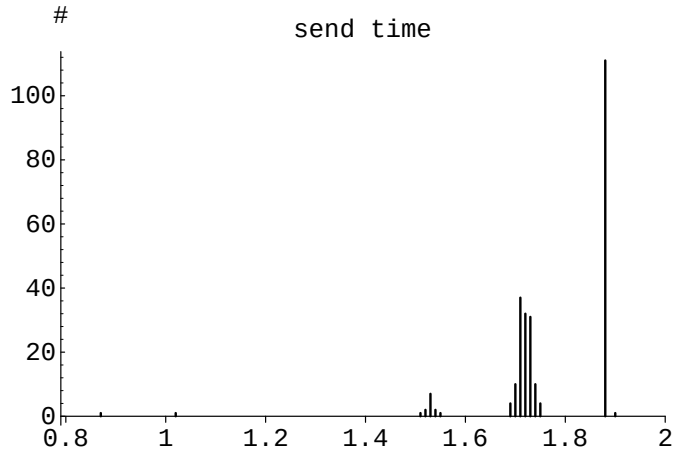


Figure 6: The histogram of the occurrences of send times (in μs/byte) in the 256 node Express partition; the step size is 0.01 μs/byte.

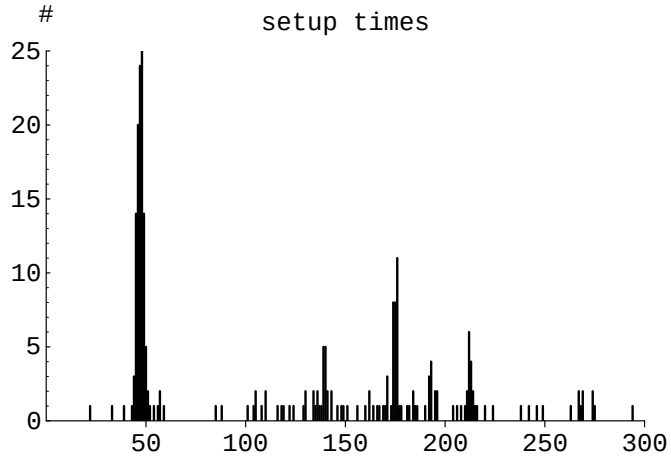


Figure 7: The histogram of the occurrences of setup times (in μs) in the 256 node Express partition; the step size is1 μs.

If the experiment is performed in a 16 processor partition we observe the same behaviour. After fitting we find 2 links with a fast τ_{send} of 0.9 μs/byte and a τ_{setup} of 85 μs. All other links have a τ_{send} of ±1.5 μs/byte, and have setup times clustered around 20 μs and 290 μs. By increasing the partition size we continue to observe these clear clustering of τ_{send} and τ_{setup}. Figure 5 shows all measured point to point communications in the 256 node partition. We have fitted all 255 experiments and generated histograms of τ_{send} and τ_{setup}. The histograms are drawn in figures 6 and 7.

We clearly observe the clustering of especially the τ_{send}. The two fast connections have a τ_{send} of 0.88 μs/byte and 1.05 μs/bytes. The rest of the connections clusters around 1.53 μs/byte, 1.73 μs/byte, and 1.88 μs/byte. Clustering of τ_{setup} is less obvious, a large peak around τ_{setup} = 48 μs can be seen, and a very broad distribution between 100 and 300 μs.

6.4.4 Global communication

We have measured the time for the vector gather operation as a function of the number of processors and as a function of the total vector length n in bytes. For Iserver-Occam and Parix we exactly know how the vector gather operation is implemented, and we can formulate a model for the execution time of the vector gather. In Iserver-Occam we have to perform $\lfloor p/2 \rfloor$ times point to point communications between neighbours on the ring. In each communication step $\lceil n/p \rceil$ bytes must be transferred. Therefore,

$$(t_{vg})^{Iserver} = \lfloor p/2 \rfloor (\tau_{setup} + \lceil n/p \rceil \tau_{send}) + \tau_{init} . \qquad [9]$$

In equation 9 we have included a term τ_{init} which describes the initialisation time for the vector gather operation. The setup - and send time are known from the point to point measurements.

In Parix the vector gather operation is almost identical to the Iserver version, but now the data is transferred through a mono directional ring and therefore we have to carry out (p-1) point to point communications. This results in

$$(t_{vg})^{Parix} = (p-1)\,(\tau_{setup} + \lceil n/p \rceil \tau_{send}) + \tau_{init}\ . \qquad [10]$$

Although Express' exconcat function is described for hypercube architectures by Fox et al. [10, chapter 14], we will not attempt to find a detailed model for the exconcat as implemented on the two dimensional grid of the Parsytec GCel. We assume that the exconcat is linear in n because it uses point to point communications which are linear in n.

For each value of p we have plotted the measured communication time as a function of n. In all cases the results are linear. We have fitted the measurements to $t_{vg} = \tau_a\, n + \tau_b$; table 3 shows the results of the fit.

p	τ_a (µs/byte)			τ_b (µs)		
	Iserver	Parix	Express	Iserver	Parix	Express
2	-	0.98	0.59	-	$9.80\ 10^2$	$2.79\ 10^2$
4	0.34	0.91	1.75	$5.00\ 10^2$	$1.24\ 10^3$	-92
8	0.42	0.92	4.22	$5.34\ 10^2$	$1.98\ 10^3$	$8.32\ 10^2$
16	0.45	0.93	5.62	$6.85\ 10^2$	$3.42\ 10^3$	$1.46\ 10^4$
32	0.49	1.07	13.2	$1.30\ 10^3$	$6.06\ 10^3$	$8.37\ 10^4$
64	0.49	1.08	19.8	$2.82\ 10^3$	$1.21\ 10^4$	$3.13\ 10^5$
128	0.48	1.13	31.9	$6.13\ 10^3$	$2.27\ 10^4$	$1.25\ 10^6$
256	0.48	1.12	59.3	$1.30\ 10^4$	$5.28\ 10^4$	$5.96\ 10^6$
512	0.48	1.11	-	$2.56\ 10^4$	$9.98\ 10^4$	-

Table 3: Result of fitting the global communication time to a linear model $t_{vg} = \tau_a\, n + \tau_b$.

A goodness of fit analysis revealed that the linear model is significant to describe the experimental data. Only the Express results for p = 4 and p = 8 cannot be fitted accurately with a linear model. Furthermore, for the Express results on large partitions, the noise level (estimated by repeatedly measuring the communication time) was very high and the resulting fit parameters also posses large (± 10 %) errors. The errors in τ_a and τ_b for the Iserver and Parix results are estimated to be smaller than 1 %.

Comparison of the τ_a and τ_b for Iserver-Occam and Parix with equations 9 and 10 shows that the observed behaviour is quantitatively in agreement with the models for the vector gather operation. The parameter τ_b is in both case linearly dependent on p. However, fitting τ_b as a function of p to a line results in much larger values for e.g. τ_{setup} as those expected from the point to point communication measurements (data not shown).

The fitted values for τ_a of Iserver-Occam agree very well with the model, as can be seen by comparing table 3 with table 1 and equation 9. The agreement in case of Parix is also satisfying, especially for large p.

Finally, the execution time of the vector gather operation in Express (i.e. the exconcat function) is, as expected, linear in p. However, we can observe two fundamental differences with the Iserver-Occam and Parix results. First, the τ_a depends on p, and increases sub linear (probably as Log(p)) in p. Furthermore, the initialisation τ_b increases faster than linear with p, which results for large p in (unacceptably) high initialisation times of e.g. 6 seconds on a 256 partition. This result is a very good example of the trade-off of programmability against performance. The Express vector gather operation

consists of just one call to exconcat, and a trivial buffer compacting. However, the price to be paid is a very bad scaling behaviour, as compared to Iserver-Occam or Parix.

6.4.5 Performance of the Coupled Dipole implementation

We have measured the execution time for one Conjugate Gradient iteration, for the calculation of the scattered field and for a total Coupled Dipole simulation (including I/O) as a function of the number of dipoles N in the simulation and as a function of the number of processors. The number of dipoles was limited to 2176 to prevent prohibitive execution times on 1 processor.

Figures 8, 9, and 10 give the execution times as a function of the number of processors. The time for one iteration is shown for N = 32, N = 552, and N = 2176, and the time for the calculation of the scattered field is shown for N = 32 and N = 2176. The time for the total Coupled Dipole simulation is shown for N = 32 and N = 912. All other problem sizes exhibit a comparable behaviour. Note that in all following figures the solid lines are for Parix, the short-dashed lines for Iserver-Occam, and the long-dashed lines for Express. Furthermore, note that in figures 8, 9, and 10 both the number of processors and the execution time are drawn on a logarithmic scale.

From these execution times one can calculate speedups and efficiencies. We will only show the efficiencies of one Conjugate Gradient (N = 32 and N = 2176, figure 11) and of the total Coupled Dipole simulation (N = 32 and N = 912, figure 12). The efficiency is defined as T(p=1)/(p T(p)), with T(p) the execution time on p processors.

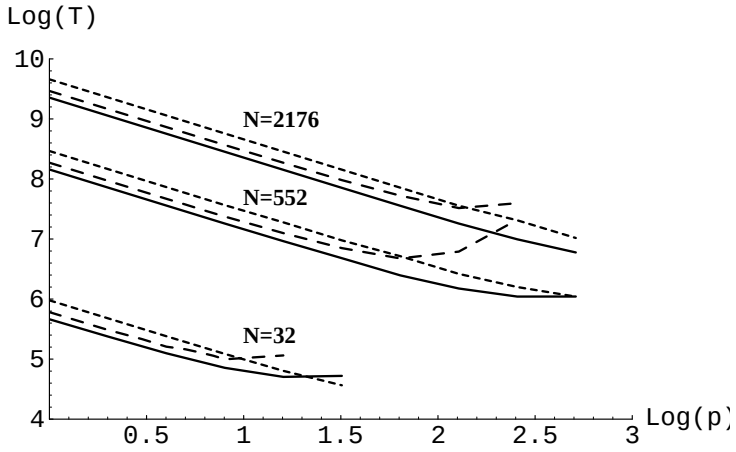


Figure 8: The execution time of one iteration of the Conjugate Gradient method, as a function of the number of processor, for N = 32, N = 552, and N = 2176; the solid line is for Parix, the short-dashed line for Iserver-Occam, and the long-dashed line for Express.

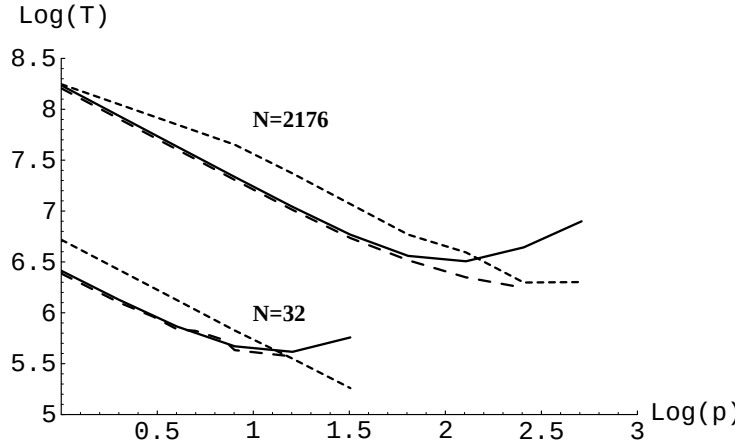


Figure 9: The execution time of scattered field calculation, as a function of the number of processor, for N = 32 and N = 2176; the solid line is for Parix, the short-dashed line for Iserver-Occam, and the long-dashed line for Express.

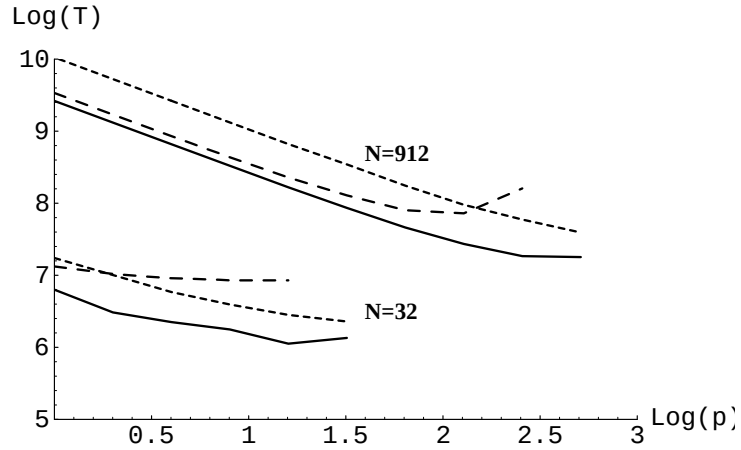


Figure 10: The execution time of a total Coupled Dipole simulation (including I/O), as a function of the number of processor, for N = 32 and N = 912; the solid line is for Parix, the short-dashed line for Iserver-Occam, and the long-dashed line for Express.

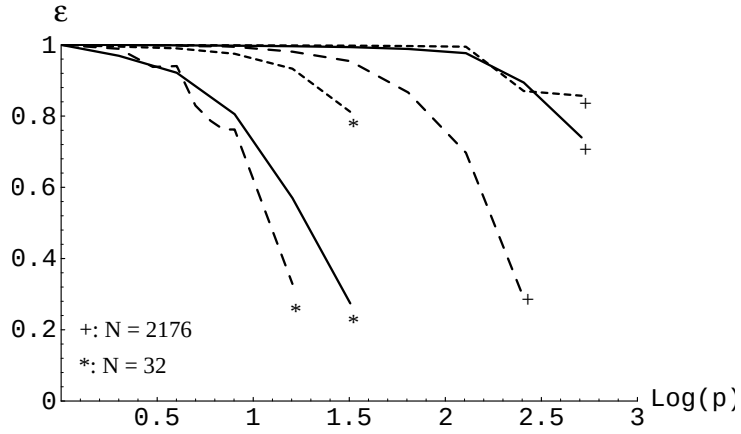


Figure 11: The efficiency of one Conjugate Gradient iteration, as a function of the number of processor, for N = 32 and N = 912; the solid line is for Parix, the short-dashed line for Iserver-Occam, and the long-dashed line for Exress.

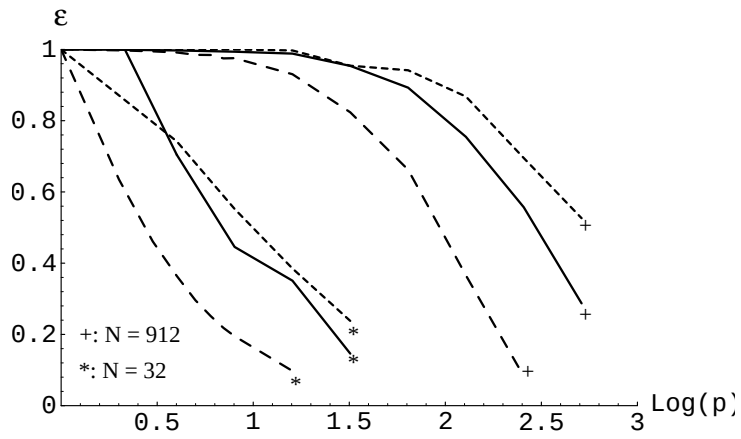


Figure 12: The efficiency of a total Coupled Dipole simulation (including I/O), as a function of the number of processor, for N = 32 and N = 912; the solid line is for Parix, the short-dashed line for Iserver-Occam, and the long-dashed line for Exress.

6.5 DISCUSSION

This paper compares native and generic parallel programming environments; i.e. Iserver-Occam and Parix versus Express. An exhaustive comparison between the complete functionality of all three environments is not very useful. We need a guideline in the form of an application that must be, or is implemented in these environments. Our particular application, the Coupled Dipole simulation, relies on a representative subset of the complete functionality of the environments.

We analysed basic - and global communication routines which are needed in the Coupled Dipole program, floating point performance, and the execution time of the application. In this discussion we will compare the environments on the first three levels, and investigate if we can predict the performance of the Coupled Dipole program. As will be shown, this is indeed possible and this result shows that our basic measurements can be used as a heuristic to assess the merits of parallel programming environments.

The floating point performance, which in our definition is the lumped sum of floating point unit performance, loop overheads, memory access overheads and others, basically is a test of the compilers. Remember that we timed a computational kernel of the Coupled Dipole method, and therefore the resulting numbers should not be compared with other results for floating point performance. We can however compare the results of the same computational kernel compiled by the three different compiles. Parix has the best performance, with a τ_{calc} of 1.28 μs/flop. Express has the second best performance with τ_{calc} = 1.72 μs/flop and surprisingly the Iserver-Occam result is much worse with a τ_{calc} = 2.63. This large τ_{calc} for Iserver-Occam is probably due to the fact that the code was compiled such that all runtime cheques are maintained. In the future we will also perform experiments with the Occam code without runtime cheques and we expect that the results will

compare with the Parix results.

The point to point communication results show the power of Iserver-Occam. A setup time of only 3.8 μs, compared to 67 μs for Parix, and 50 μs < τ_{setup} < 300 for Express, shows that the Occam channels are directly mapped onto the transputer hardware. The abstractions of Parix and Express induce much larger setup times. The same is true for the sending time, but here the difference between Iserver-Occam and Parix is not that large, and the fastest connections in the Express partition also compare very good with Iserver-Occam and Parix. Still, the situation as drawn in figure 1 is confirmed for the point to point communication, and the question mark can in this be replaced by an exclamation mark.

The global communication routine reveals the same picture: decreasing performance as the price to be paid for decreasing hardware visibility and increasing programmability. The Iserver-Occam implementation has the smallest initialisation time and fasted transfer rate. The initialisation time in Parix is a factor 2 to 4 larger than in Iserver-Occam, and the transfer rate is a factor 2 smaller.

Except for very small partitions the Express vector gather operation is always slower than Iserver-Occam and Parix. Furthermore, as shown in section 6.4.4, the exconcat function has a fundamental different scaling behaviour, i.e. an increasing τ_a as a function of p, and a faster than linear increase of τ_b as a function of p. This results in transfer rates of 60 μs/byte[**] and initialisation times in the order of seconds! In compute intensive applications like the Coupled Dipole simulation, where the computational work scales with N^2, this does not need a problem (as can be seen in the sequel). However, for applications where the computational work is linear in N, and which need such global communications, this scaling behaviour will result in very bad performance on real massively parallel systems with much more than say 32 processors.

The global communication can be modelled quantitatively for Iserver-Occam and Parix. The observed scaling behaviour agrees with the model. However, using the results of the point to point communication experiments in the model will result in an under estimation of the global communication time. Still we are able to estimate the order of magnitude of the global communication time. We did not attempt to model the exconcat function of Express. However, based on the relative good results of the Express for the point to point communication, as compared to e.g. Parix, we did not expect the relative poor behaviour of the exconcat function.

Figures 8 - 10 show the execution time of one iteration, the scattered field calculation, and the total simulation, as a function of p and N. The calculation time is of order $O(N^2 / p)$ (see section 6.3.1), whereas the total communication time is of order O(N). Therefore, for large N and small p we actually observe the floating point performance of the environments. This is most obvious in figure 8 and 10. The Parix implementation is the fastest, secondly is Express and due to its poor floating point performance the Iserver-Occam implementation is the slowest. However, if p is increased for a constant N, the communication time

---

[**] Remember however how this transfer rate τ_a is defined in section 6.4.4. It is not the same as the linkspeed of the point to point communication, but is defined as the rate at which the total vector is transferred through the network in the vector gather operation.

becomes more important, and at some point becomes larger than the calculation time. Due to the very poor performance of the exconcat function the Express results already show an increase in execution time of one iteration for e.g. p = 100 and N = 552, whereas the execution time of the Iserver-Occam and Parix implementation still decrease. For an even larger number of processors the Parix implementation also shows an increase in execution time and for N = 552 the Iserver-Occam implementation has a comparable execution time as the Parix implementation.

Scalabilty of the three parallel implementations is much better observed by looking at the efficiency plots. From figures 11 and 12 we can see that in all cases the Iserver-Occam has the best efficiency, followed by Parix and finally by Express. We have to be very careful with this comparison. The Iserver-Occam spends much more time in the parallel calculations and will therefore have a better efficiency. Or, if we can decrease τ_{calc} of Iserver-Occam, both the runtime - and efficiency results will be comparable to those of Parix.

The efficiency plots however allow one important conclusion, which holds for the parallel Coupled Dipole implementation in all three environments. If the number of dipoles is very large compared to the number of processors (N / p >> 1), the efficiency can be very close to 1. However, for Express N/p has to be much larger than e.g. Iserver-Occam to reach efficiencies close to 1. In that situation, which is frequently encountered in real production runs, the execution time is mainly determined by τ_{calc}, and the Parix implementation is the fastest.

We should note that in principle we can implement the vector gather operation in Express in the same way as in Parix. This would induce a large programming overhead, but we may expect from the histograms as reported in figures 6 and 7 that the vector gather operation will then be comparable to the Parix results. Finally, next versions of Parix also contain global communication routines. This greatly increases the programmability of global communications in Parix, but experiments will have to establish if the same price as in Express has to be paid.

Knowledge of the floating point performance and the global communication capabilities of the environments allowed us to interpret and understand the execution time and efficiency of the application. Let us now investigate if we can predict the execution time of the Coupled Dipole application. We only show results for the execution time of one Conjugate Gradient iteration. We use equations 1, 2, 4 and 6 to calculate T^{iter} as a function of p and N. The times for the vector gather operation, and for the vector - and scalar accumulate operation are calculated by using the results of table 3, and finally τ_{calc} as reported in section 6.4.2 is used. Figures 13 and 14 show the comparison between the measurements (the points) and theory (the lines: solid line for Parix, short-dashed line for Iserver-Occam, and long-dashed line for Express) of the execution time of one Conjugate Gradient iteration, as a function of the number of dipoles N, for p = 32 and p = 256 respectively. Furthermore, figure 15 shows the results for Parix and Iserver-Occam for p = 512.

In all cases we can accurately predict the execution time of the Parix and Iserver-Occam implementation. For small partitions (i.e. p = 32, see figure 13, but also others) we can also accurately predict the execution time of the Express implementation. However for larger partitions errors between theory

and experiment as large as 30 % are observed (see figure 14). These errors can be traced back to the uncertainties in the fit parameters of the exconcat function, as reported in section 6.4.4.

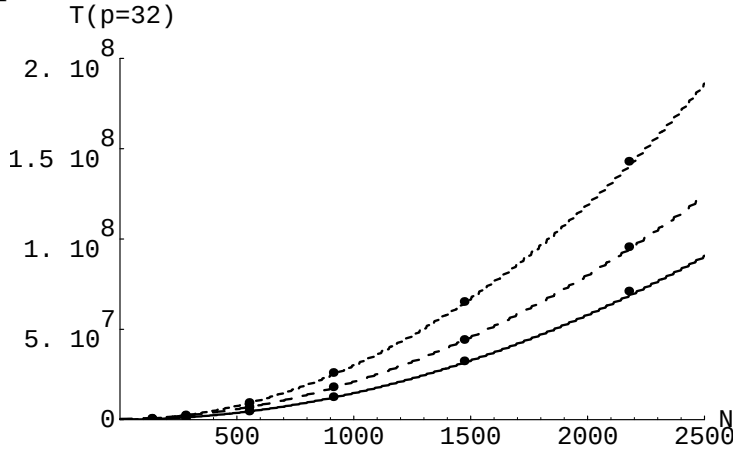


Figure 13: The execution time (in μs) on 32 processors, for one Conjugate Gradient iteration, as a function of the number of dipoles N ; the dots are the measurements, the lines are the theoretical predictions; solid line for Parix, short-dashed line for Iserver-Occam, and long-dashed line for Express.

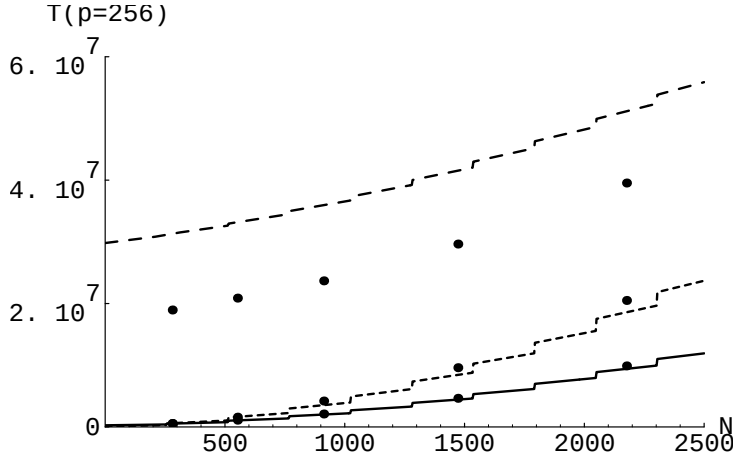


Figure 14: The execution time (in μs) on 256 processors, for one Conjugate Gradient iteration, as a function of the number of dipoles N ; the dots are the measurements, the lines are the theoretical predictions; solid line for Parix, short-dashed line for Iserver-Occam, and long-dashed line for Express.

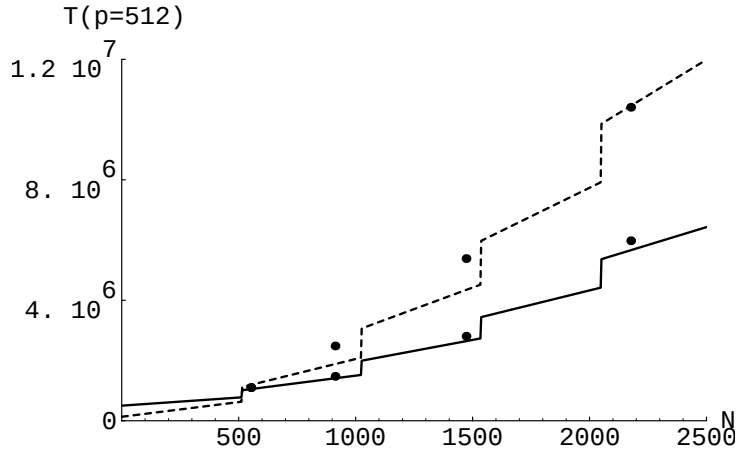


Figure 15: The execution time (in μs) on 512 processors, for one Conjugate Gradient iteration, as a function of the number of dipoles N ; the dots are the measurements, the lines are the theoretical predictions; solid line for Parix, and short-dashed line for Iserver-Occam.

6.6 CONCLUSIONS

We can predict the execution time of our specific application in the different environments using basic performance measurements in these environments. Therefore, these basic performance measurement, combined with a time complexity model as presented in section 6.3.1 and information on programmability and portability, allows us to compare the environments. Table 4 compares the three environments on a scale from good (++) to poor (--). The environments are judged on programmability (Prog.), on source level portability between hardware platforms (Port.), on the availability of tools, such as a performance analyser, debuggers etc. These first three characteristics are independent of the particular application. The rest of the columns are devoted to application dependent characteristics. These characteristics are the floating point performance (F.P.), the communication performance (Comm.), and the scalability, or efficiency, of the resulting parallel program (Scal.). Depending on the importance of e.g. portability versus scalability or floating point performance an overall judgement of the environments can be made.

	Prog.	Port.	Tools	F.P.	Comm.	Scal.
Iserver	--	--	--	-	++	++
Parix	+-	+-	+-	++	+	+
Express	+	++	++	+	--	-

Table 4: A comparison of the Iserver-Occam, the Parix, and the Express environment on programmability (Prog.), on source level portability (Port.), on the availability of tools (Tools), on floating point performance (F.P.), on communication performance (Comm.), and on scalability, or efficiency, of the resulting parallel program (Scal.).

In conclusion, we have compared the Iserver-Occam, Parix, and Express parallel programming environments on a Parsytec GCel, by a detailed analysis of the performance of a particular application. Our approach, in which we start with an application, isolate the basic (environment) dependent building blocks which are analysed independently, and combining all information in a time

complexity analysis, allowed us to compare the environments on all relevant degrees of functionality. Together with demands for portability of the code, and development time (i.e. programmability), an overall judgement of the environments can be made.

In general we observe that increasing portability and programmability, in going from Iserver-Occam, via Parix to Express, results in a degradation of especially the communication capabilities. The global communication routine of Express that we tested has a very bad scaling behaviour which clearly shows up in the larger partitions. This results in poor scalability of the Express implementation. Fortunately, in real production situations, with large problem sizes, the application has an efficiency very close to one, and the run time is mainly determined by the floating point performance. In that situation Parix is the fastest, but Express, offering portable and easily implementable code, also has a very good performance.

6.7 REFERENCES

1] *CS Tools Documentation guide, a technical overview*, Meiko Ltd., 1989.

2] M.D. May, P.W. Thompson, and P.H. Welch, *Networks, Routers and Transputers* (IOS Press, Amsterdam, Oxford, Burke, 1993).

3] F. Langhammer, "Second generation and teraflops parallel computers," In *Parallel Computing and Transputer Applications* (IOS press, M. Valero, E. Onate, M. Jane, J.L. Larriba, and B. Suárez Ed., pp. 62-79, Amsterdam, Washington, 1992).

4] Parsytec, "The PARIX programming environment," In *Transputer Systems - ongoing research* (IOS Press, A. Allen Ed., pp. 218-230, Amsterdam, Oxford, Washington, Tokyo, 1992).

5] J. Flower and A. Kolawa, "A packet history of message passing systems," Physics Reports **207**, 291-304 (1991).

6] J. Dongarra, G.A. Geist, R. Manchek, and V.S. Sundaram, "Integrated PVM Framework Supports Heterogeneous Network Computing," Computers in Physics **7**, 166-175 (1993).

7] P.M.A. Sloot, *private communication*, for more information you can send email to peterslo@fwi.uva.nl.

8] A.G. Hoekstra and P.M.A. Sloot, "New Computational Techniques to Simulate Light Scattering from Arbitrary Particles," in *Proceedings of the 3rd International Congress on Optical Particle Sizing '93 - Yokohama*, Ed. M. Maeda, 1993, pp. 167-172.

9] A.G. Hoekstra and P.M.A. Sloot, "Simulating Elastic Light Scattering Using High Performance Computing Techniques," in *European Simulation Symposium 1993*, Ed. A. Verbraeck and E.J.H. Kerckhoffs, Society for Computer Simulation International, 1993, pp. 462-470.

10] G. Fox, M. Johnson, G. Lyzenga, S. Otto, J. Salmon, and D. Walker, *Solving Problems on Concurrent Processors 1: General techniques and Regular Problems* (Prentice-Hall, Englewood Cliffs, New Jersey, 1988).

11] E.M. Purcell and C.R. Pennypacker, "Scattering and absorption of light by nonspherical dielectric grains," The Astrophysical Journal **186**, 705-714 (1973).

12] S.F. Ashby, T.A. Manteuffel, and P.E. Saylor, "A Taxonomy for Conjugate Gradient Methods," Siam J. Numer. Anal. **27**, 1542-1568 (1990).

13] A.G. Hoekstra, P.M.A. Sloot, W. Hoffmann, and L.O. Hertzberger, *Time complexity of a parallel Conjugate Gradient solver for light scattering simulations: theory and SPMD implementation,* Tech. Rept. CS-92-06 Faculty of Mathematics and Computer

Science, University of Amsterdam, 1992.

Chapter 7

DIPOLAR UNIT SIZE IN COUPLED DIPOLE CALCULATIONS OF THE SCATTERING MATRIX ELEMENTS

A.G. Hoekstra and P.M.A. Sloot,

Optics Letters **18**, 1993, p. 1211 - 1213

Dipolar Unit Size in Coupled Dipole Calculations of the Scattering Matrix Elements

ABSTRACT

The Coupled Dipole method is widely used to calculate the light scattering matrix **S** from arbitrary particles. An important parameter in the model is the size of the dipolar subunits. Usually a size of ~1/10 to ~ 1/20 of the wavelength of the incident light is sufficient for accurate calculations. However, it was noted that accurate $\mathbf{S}_{34}$ calculations require much smaller dipolar subunits. Here we will show that this conclusion is too pessimistic, by examining the sensitivity of the $\mathbf{S}_{34}$ elements on surface roughness of spherical particles. Furthermore we will give an example of an accurate $\mathbf{S}_{34}$ calculation with dipolar subunits as large as 1/10 of the wavelength.

7.1 INTRODUCTION

The Coupled Dipole (CD) method, originally formulated by Purcell and Pennypacker [1], is a very powerful method to calculate Elastic Light Scattering from arbitrary particles. The CD method divides a particle in small subvolumes, which are assumed to behave as ideal dipoles. The electric field on each dipole, due to the external field and the field radiated by all other dipoles is calculated. Next the scattered field is obtained by summing the fields radiated by all dipoles in the observation points. By repeating this calculation for a parallel - and a perpendicular polarised incident electric field, the complete (4×4) scattering matrix **S** of the particle can be computed. Although the basic concepts of the model are straightforward, the model possesses many parameters which are topic of active research. An example is the choice of the polarizability of the dipolar subunits. Lakhtakia gives a review of the CD method [2].

A very important parameter of the CD method is the size of the dipoles. The dipoles are placed on a cubic grid, with grid spacing d. The dipoles therefore are assumed to describe the response of a cube with volume d^3. The computation of the electric field on the dipoles requires solving a set of 3N equations with 3N unknowns, where N is the number of dipoles. Therefore it is most advantageous to choose d as large as possible, thus decreasing N. On the other hand, if d becomes too large the dipole approximation breaks down and the results of CD simulations will no longer be reliable. By comparing CD computations on spheres with the exact Mie results, it was concluded that $\lambda/20 \leq d \leq \lambda/10$ gives a good agreement between the simulated and exact differential

cross sections (e.g. [1]).

Singham [3] calculated the $\mathbf{S}_{34}$ element with the Coupled Dipole method. By comparing the calculations with exact results for a sphere, it was concluded that accurate $\mathbf{S}_{34}$ simulations require much smaller dipoles, with $d \sim \lambda/45$ (depending on the refractive index of the particle), which is two to four times smaller than for accurate $\mathbf{S}_{11}$ simulations. Since N scales as d^{-3}, the total number of dipoles would be 8 to 64 times larger as for accurate $\mathbf{S}_{11}$ calculation. If the system of equations is solved with an $O(N^2)$ iterative method, this would require a factor of 64 to 4096 longer execution times on a computer. This is a very discouraging conclusion. Especially if one realises that the $\mathbf{S}_{34}$ element is known to be very sensitive for slight changes in structure and optical constants of a particle (see e.g. [4, 5]), and therefore of main interest. In this paper we show that Singham's conclusion is too pessimistic.

7.2 RESULTS

Singham simulated a sphere with size parameter α equal to 1.55 and relative refractive index m equal to 1.33. This sphere was simulated with the CD method containing 123 up to 5575 dipoles. The CD results for the $\mathbf{S}_{11}$ elements are already in good agreement with the Mie results for the model with 123 dipoles ($d \sim \lambda/10$). However, even for 5575 dipoles ($d \sim \lambda/45$) the results of the CD simulation of the $\mathbf{S}_{34}$ element are still not in good agreement with the Mie results, although the CD results do approximate the Mie results if the number of dipoles is gradually increased [3]. Based on this result, and on simulations with m = 1.1 and 1.02 it was concluded that accurate $\mathbf{S}_{34}$ computation require much smaller dipolar subunits.

In order to verify these results we conducted CD simulations with a much larger number of dipoles, but with $d = \lambda/10$. Figure 1 shows a CD calculation† (the dots) of the $\mathbf{S}_{34}$ element of a sphere with $\alpha = 10.7$ and m = 1.05; the number of dipoles was 20672 and $d = \lambda/10$. The agreement with the Mie calculation (line) is excellent. This example, and other calculations indicate that if the number of dipoles in the simulation is large enough, $\mathbf{S}_{34}$ can be calculated with the same accuracy as the other matrix elements, with $\lambda/20 \leq d \leq \lambda/10$.

† the calculation was performed with our implementation of the CD method on a 512 node Parsytec GCel computer.

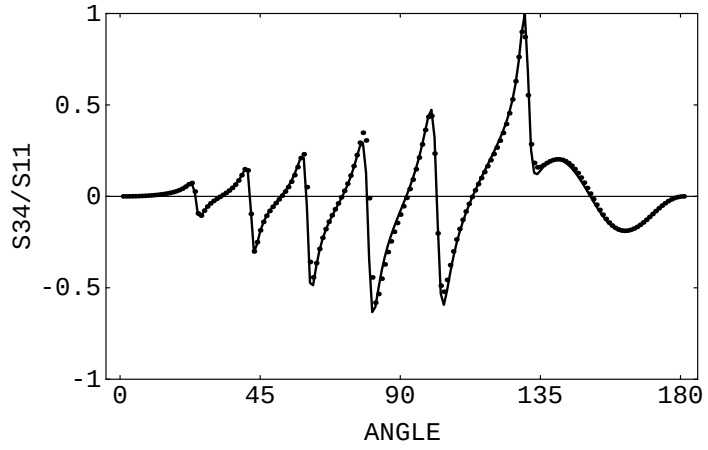


Figure 1: $\mathbf{S}_{34}/\mathbf{S}_{11}$ as a function of the scattering angle, for a sphere with α = 10.7 and m = 1.05.; the line is the Mie calculation, the dots are the CD calculations with d = λ/10 (20672 dipoles).

This counter example shows that another, overlooked argument must enter the discussion. The coarseness of the discretization seems to be the key issue. The CD method discretizes a particle with small cubes (assuming dipoles on a cubic grid). This means that we simulate the sphere by a spherical particle with a rough surface. If the same particle is discretised using more dipoles, the surface roughness will be smaller and the particle will be more closely approximated. We investigated the influence of surface roughness on the α = 1.55 sphere, and show that the decrease of the surface roughness of the discretised particle, as the number of dipoles is increased, obscures the conclusions of Singham.

In the CD method we discretised 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

$$(i+1/2)^2 + (j+1/2)^2 + (k+1/2)^2 \leq l^2 \, . \qquad [1]$$

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. The size of the dipoles determines the radius of the simulated sphere, via $r_{sphere} = (3N/4\pi)^{1/3}d$ (the equal volume sphere [1]). The discretised sphere has a radius $r = r_{sphere}[1 \pm O(\varepsilon)]$ with ε a measure of the surface roughness. In our case the radius is modulated with an amplitude of approximately d/2, therefore $\varepsilon = 0.5\,(3N/4\pi)^{-1/3}$.

Consider a particle with "radius"

$$r = r_0\,[1.0 - (2l)^{-1}\cos(4l\theta)], \qquad [2]$$

with θ an azimuthal angle. This is a spherical particle with mean radius r_0 and roughness 1/(2l). If l is increased, the roughness decreases, but at the same time the frequency of the modulation is increased (the cosine term). This is exactly what happens if we increase the number of dipoles in the discretization of the sphere by increasing l in equation 1. Figure 2 shows a cross section of the rough spheres for l = 3 and l = 5.

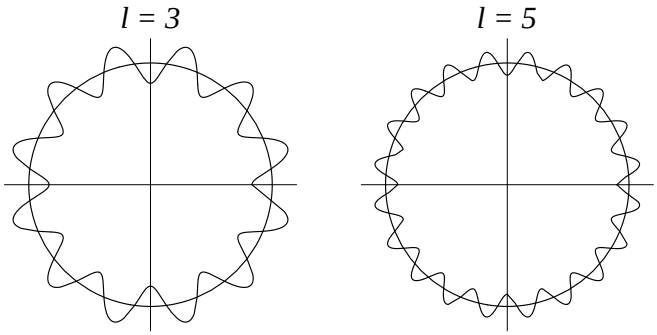


Figure 2: cross section of rough spheres, defined by equation 2, with l equals 3 and 5 respectively.

Note that $(2l)^{-1}$ is approximately equal to the estimated roughness of the discretised sphere. We calculated the $\mathbf{S}_{11}$ and $\mathbf{S}_{34}$ elements of a random distribution of this rough sphere for l = 3, l = 5, and l = 17, and adjusted r_0 such that in all cases the volume of the particle was the same and equal to the test sphere with α = 1.55. The scattering properties of this axis-symmetric particle were calculated with the T-matrix method, using the computer programs of Barber and Hill [6]. This calculation serves as a model of the surface roughness of the discretised sphere.

Figure 3 and 4 give the results for $\mathbf{S}_{34}$ and $\mathbf{S}_{11}$ for the l = 3 and l = 5 rough sphere, together with Mie calculation for the α = 1.55 sphere. The l = 17 calculation is indistinguishable from the Mie calculation.

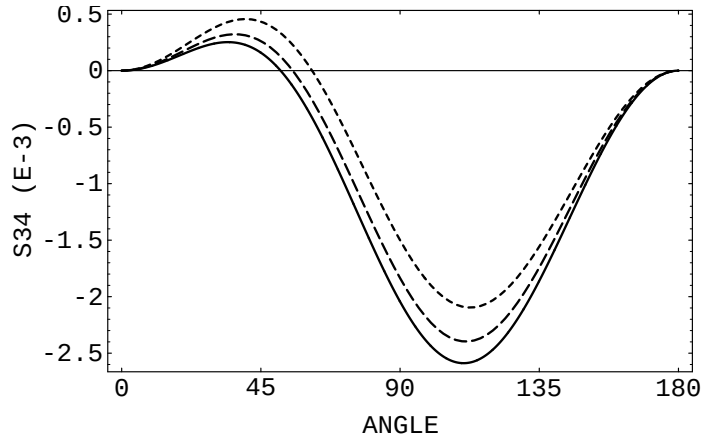


Figure 3: The $\mathbf{S}_{34}$ element as a function of the scattering angle for a sphere with α = 1.55 (line), and of equal volume rough spheres with l = 3 (short dash) and l = 5 (long dash); the refractive index was 1.33.

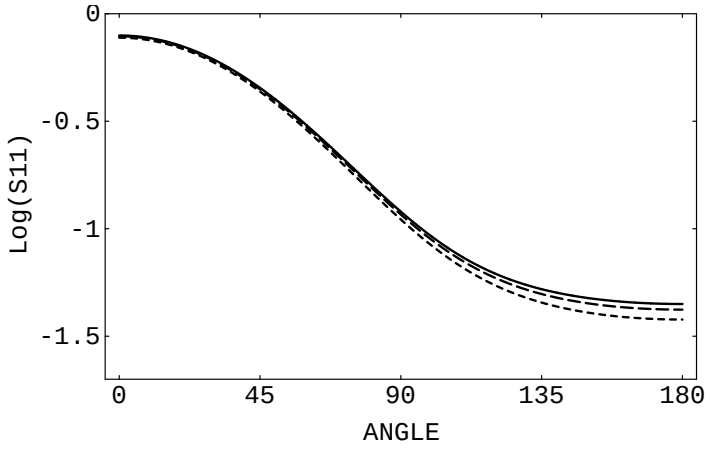


Figure 4: The $\mathbf{S}_{11}$ element as a function of the scattering angle; rest as in figure 3.

7.3 DISCUSSION AND CONCLUSIONS

The influence of surface roughness is most obvious for the $\mathbf{S}_{34}$ element. The $\mathbf{S}_{11}$ element of the rough sphere deviates slightly from the sphere in the backscattering direction. For l = 5 the results are almost equal to the sphere. The same is true for the $\mathbf{S}_{12}$ and $\mathbf{S}_{33}$ element (data not shown). However the roughness has a much more pronounced effect on the $\mathbf{S}_{34}$ element, as is obvious from figure 3. Therefore, a coarse discretization of the sphere in the CD method can result in larger errors in the $\mathbf{S}_{34}$ element, compared to errors in the other scattering matrix elements.

If we compare the calculations on the rough sphere with the CD calculations of Singham (figures 1 and 2 in reference [3]) we see the same trends. For a small number of dipoles (123, comparable with l = 3), the $\mathbf{S}_{34}$ computation deviates significantly from the exact Mie result, in the same way as the rough sphere (see figure 3). The $\mathbf{S}_{11}$ result of the CD computation already is very good, with only a deviation from the exact Mie result in the backscattering directions. If the number of dipoles is increased in the CD calculations, the $\mathbf{S}_{11}$ result is in excellent agreement with the exact Mie calculation. The $\mathbf{S}34$ result is approaching the exact results, but the agreement is far from good.

Surface roughness has a strong effect on the $\mathbf{S}_{34}$ element. Therefore, if one wants to exploit the CD method to calculate the $\mathbf{S}_{34}$ element of a smooth particle, surface roughness induced by the discretization of the particle must be very small. This is achieved by using a large number of dipolar subunits (large l). If the particle is small (e.g. the α = 1.55 sphere), this results in relative small dipolar subunits. However, for larger particles (see figure 1) the subunits can be much larger. In that case $\mathbf{S}_{34}$ calculations with comparable accuracy as $\mathbf{S}_{11}$ calculations can be achieved with dipolar subunits with $\lambda/20 \leq d \leq \lambda/10$.

7.4 REFERENCES

1] E.M. Purcell and C.R. Pennypacker, "Scattering and absorption of light by nonspherical dielectric grains," Astrophys. J. **186**, 171-192 (1973).

2] A. Lakhtakia, "Macroscopic theory of the coupled dipole approximation method," Opt. Commun. **79**, 1-5 (1990).

3] S.B. Singham, "Theoretical factors in modelling polarised light scattering by arbitrary particles," Applied Optics **28**, 5058-5064 (1989).

4] W.S. Bickel, J.F. Davidson, D.R. Huffman, and R. Kilkson, "Application of polarisation effects in light scattering: a new biophysical tool," Proc. Natl. Acad. Sci. U.S.A. **73**, 486-490 (1976).

5] P.M.A. Sloot, A.G. Hoekstra, H. van der Liet, and C.G. Figdor, "Scattering matrix elements of biological particles measured in a flow through system: theory and practice," Applied Optics **28**, 1752-1762 (1989).

6] P.W. Barber and S.C. Hill, *Light Scattering by Particles: Computational Methods* (World Scientific, 1990).

Chapter 8

COUPLED DIPOLE SIMULATIONS OF SCATTERING OF SPHERES ILLUMINATED BY A HIGHLY FOCUSED LASERBEAM

FIRST RESULTS AND COMPARISON WITH GENERALISED LORENZ-MIE THEORY

Coupled Dipole Simulations of Scattering of Spheres illuminated by a highly focused Laserbeam

first results and comparison with generalised Lorenz-Mie theory

ABSTRACT

We have included a new element into Coupled Dipole simulations of Elastic Light Scattering: scattering by a Gaussian beam. The feasibility of the Coupled Dipole method to accurately describe this situation is investigated. We compare Coupled Dipole simulations of scattering from spheres located in a Gaussian beam with the generalised Lorenz-Mie theory. Although the range of these first tests is limited, it can be concluded that the Coupled Dipole method is capable to describe scattering from Gaussian beams by arbitrary particles.

8.1 INTRODUCTION

Elastic Light Scattering (ELS) is an important technique to assess morphological or optical information from scattering particles. Especially if measurements must be non-invasive, or if particles cannot be sampled and placed under a microscope (e.g. white bloodcells in Centrifugal Elutriation equipment [1], or interstellar dust particles [2]), ELS is a very powerful method. Unfortunately, in most cases the particles in question are not spherical [see e.g. 3] and Mie theory to describe the ELS cannot be used. Furthermore, in many cases approximate theories to describe the scattering process are also not appropriate [see e.g. 4].

For example, we want to simulate ELS from human white bloodcells and bonemarrow cells. The simulations will assist to define optimal light scattering experiments to distinguish between subsets of white bloodcells (e.g. between the three forms of Granulocytes), or between malign and healthy cells. Approximate theories for ELS, such as Rayleigh-Debye-Gans (RDG) scattering [5], are sometimes very successful to describe certain properties of ELS from white bloodcells. We showed that a modified form of RDG scattering [6] can describe anomalous scattering behaviour of osmotically active Lymphocytes very well [7], even giving rise to new biophysical insights concerning the biomechanics of Lymphocytes [8]. However, the polarisation of the scattered light cannot be described with these approximate theories [4]. It are exactly these properties that play a key role to distinguish between subsets of white

bloodcells, as was shown very elegantly by for instance de Grooth et al. [9], who measured depolarisation of the scattered light to differentiate between Eosinophilic - and Neutrophilic Granulocytes.

Observations like those above initiated much research to methods capable to describe ELS from arbitrary shaped particles. One such method is the well-known Coupled Dipole (CD) method, which was originally developed by Purcell and Pennypacker [10]. We want to use this method for simulations of ELS from human white bloodcells. We developed a version of the CD method, suitable for parallel computing [11, 12], and showed that the complete scattering matrix can be calculated accurately with simulations containing a large number of dipoles [11, 13, 14].

Another constraint implicitly present in most ELS theories was also removed during the last two decades. Textbook theories all assume an incident plane wave [see e.g. 5]. In many practical situations however this assumption is not valid. For instance, in our experimental studies of ELS from human white bloodcells we apply flowcytometry [4, 8]. In a flowcytometer cells are forced, by means of a technique called hydro-focusing, to pass through the focal point of a strongly focused laser beam one-by-one [15]. Therefore, our goal is to simulate ELS from arbitrary particles in focused beams. Other examples can be found in scattering from particles in optical traps [e.g. 16, 17, 18, 19].

A number of groups have published methods and results of scattering of beams of light by particles, with most emphasis on laserbeams operating in the (Gaussian) TEM_{00} mode. For instance, Yeh et al. [20] introduced an incident focused beam into the T-matrix method, thus developing a method to find solutions of scattering of sharply focused beams by arbitrary particles. Barton et al. [21] developed a method to find internal and near surface fields of spheres irradiated by a focused beam, and Gouesbet et al. [22] developed a generalisation of the Lorenz-Mie theory (coined GLMT), to describe scattering from spheres located at an arbitrary position in a Gaussian beam. Chevaillier et at. [23] included a Gaussian beam into the Fraunhofer approximation, and calculated near-forward scattering of glass spheres located in any position in the beam. Later this approximation was compared with the GLMT theory [24, 25]. Esam et al. [26] studied Morphology Dependent Resonances in droplets situated off-axis in a Gaussian beam, and Schaub et al. [27] studied the internal field in methanol droplets as a function of the local diameter of a focused laserbeam. Recently, Lock [28] has reviewed much of the work done in this area and has applied his formalism, based on the work of Gouesbet et al. and Barton et al. to find high-order rainbows in the far-field scattered intensities by illuminating a sphere with an off-axis highly focused laserbeam.

In principle the CD method allows to simulate scattering by an arbitrary incident field* (see section 8.2). However, all CD simulations published to date** assume incident plane waves. Therefore, in this paper we study the feasibility of the CD method to simulate scattering by focused laserbeams. We simulate scattering of strongly focused laserbeams from a sphere with the CD method, and compare the results with analytical calculations using the generalised Lorenz-Mie theory (GLMT).

---

* Arbitrary in the sense that the incident field must be a solution of the Maxwell equations.

** Known to the authors.

Section 8.2 shortly introduces the Coupled Dipole method and the generalised Lorenz-Mie theory, the results of CD simulations and comparisons with GLMT calculations are made in section 8.3. Finally, section 8.4 and 8.5 give a discussion and conclusions.

8.2 THEORY

8.2.1 The Coupled Dipole Method

Consider an arbitrary particle, located at the origin of a Cartesian coordinate system. The particle is illuminated by a monochromatic electromagnetic field $\mathbf{E}^0(\mathbf{r})$. This external field can have any form which is consistent with the Maxwell equations. The exact form is given in section 8.2.2. The wavelength is λ, and for simplicity 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.

The CD method divides the particle in N subvolumes. The size of a subvolume 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 $\lambda/20 < d < \lambda/10$, with d the size of a subvolume [13]. The field at $\mathbf{r}_i$ radiated by a dipole located at $\mathbf{r}_j$ is [29]

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

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

$\mathbf{n} = \mathbf{r} / r$, 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:

$$\mathbf{p} = \boldsymbol{\alpha}\mathbf{E}, \qquad [2]$$

$\boldsymbol{\alpha}$ is the polarizability tensor. We will assume an isotropic polarizability :a = $\alpha\mathbf{I}$, with $\mathbf{I}$ the identity matrix and α a scalar polarizability. The field at $\mathbf{r}_i$ radiated by a dipole located at $\mathbf{r}_j$, with an isotropic polarizability α_j, can now be written as

$$\mathbf{E}(\mathbf{r}_i) = \frac{\alpha_j}{4\pi\varepsilon_0}\mathbf{F}_{ij}\mathbf{E}_j \ , \qquad [3]$$

with the functional $\mathbf{F}_{ij}$ defined by equations 1 and 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

$$\mathbf{E}(\mathbf{r}_i) = \mathbf{E}^0(\mathbf{r}_i) + \frac{1}{4\pi\varepsilon_0}\sum_{j\neq i}^{N}\alpha_j\mathbf{F}_{ij}\mathbf{E}_j \ , \ 1 \le i \le N. \qquad [4]$$

The summation in equation 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. 30, 31], 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 [31]. The exact form of this eigenterm still is under discussion [see e.g. 31, 32, 33]. Introduction of an eigenterm in the numerical simulation is trivial. Keeping all this in mind, the eigenterm will be neglected in the sequel of this paper. To simplify even more, we assume that the particle is homogeneous, i.e. $\alpha_i = \alpha$, for all values of i.

Equation 4 defines a set of 3N equations for the 3N unknowns ($\mathbf{E}_x(\mathbf{r}_i)$, $\mathbf{E}_y(\mathbf{r}_i)$, $\mathbf{E}_z(\mathbf{r}_i)$). After solving these equations, the scattered electric field $\mathbf{E}^s$ is calculated by summing the fields, radiated by the dipoles, at the observation point $\mathbf{r}_{obs}$:

$$\mathbf{E}^S(\mathbf{r}_{obs}) = \frac{\alpha}{4\pi\varepsilon_0}\sum_{i=1}^{N}\mathbf{F}_{obs,i}\mathbf{E}(\mathbf{r}_i) \ . \qquad [5]$$

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 Cartesian co-ordinates of the middle point of the dipoles is [(k+1/2)d, (l+1/2)d, (m+1/2)d], with (k,l,m) integer numbers. Here we will simulate scattering by a homogeneous sphere. In that case the (k, l, m) numbers are found demanding that

$$(i+1/2)^2 + (j+1/2)^2 + (k+1/2)^2 \le l^2 \qquad [6]$$

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 = λ/10 [13]. The wavelength was 488 nm and the refractive index was 1.05. The radius of the simulated sphere is found by demanding that the volume of the sphere equals Nd^3 (i.e. the equivalent sphere [10]). Therefore by adapting l and/or d any sphere can be simulated.

The scalar polarizability is calculated using the Claussius-Mossotti relation [29]

$$\alpha = \frac{4\pi\varepsilon_0}{3n}\frac{m^2-1}{m^2+2} \ , \qquad [7]$$

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

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

8.2.2 Description of a Gaussian beam

We use the theory of electromagnetic beams of Davis [34] to describe the electric field of a focused laserbeam. This theory embeds the very successful theory of laserbeams which is based on a paraxial approximation to the scalar wave equation [e.g. 35]. Furthermore, it resolves an apparent paradox in traditional paraxial beam theory which was first noticed and solved by Lax et al. [36]. In paraxial beam theory one assumes that the electric field if plane polarised in e.g. the x-direction. Under this assumption, the Maxwell equations dictate that the field must be independent of the x-coordinate. For plane waves this condition is fulfilled, but in the paraxial approximation Gaussian solutions are found in the transverse direction, in spite of the fact that $\partial \mathbf{E}_x/\partial x$ has to be zero. Lax et al. showed that these Gaussian solutions are consistent zero-order solutions of the Maxwell equation [36]. Furthermore, they established a first order correction, a small longitudinal component in the field.

Davis' theory is more attractive, because of its relative simplicity compared to the work of Lax et al. Furthermore, it reproduces the results of Lax et al., and it gives explicit expressions for second and third order corrections to the field and a method to get to even higher order correction if necessary. Barton et al. derive expressions for the fifth order corrections [37]. Davis starts with the proposal that the electromagnetic vector potential **A** is linearly polarised in e.g. the x direction; **A** = (A,0,0). The only non-vanishing component A obeys a scalar wave equation. The equation is solved by putting $A = \psi(\mathbf{r})\exp(-ikr)$, subsequent scaling of the equation in the transverse (x and y) direction with the beamwaist radius ω_0 and in the longitudinal (z) direction with the diffraction - or spreading length $l = k\omega_0^2$, and expanding ψ as

$$\psi = \psi_0 + s^2\psi_2 + s^4\psi_4 + \cdots,$$

with s a small parameter defined as $s = \omega_0/l = 1/k\omega_0$. The function ψ_0 is the well known fundamental mode solution from paraxial beam theory. Davis also gives expressions for ψ_2, and expressions for the electric fields up to third order in s.

In this paper we only use the fields to first order in s to be consistent with the GLMT (see section 8.2.3), but higher order corrections are easily included into the CD simulations (see also section 8.4). Realising that we assume an $\exp(i\mathbf{kr})$ spatial dependence for the electric fields in the CD method, and converting the results of Davis to S.I. units, the electric field to first order in s is,

$$\mathbf{E}_x = \psi_0 \exp(ikz) , \qquad [8]$$

$$\mathbf{E}_y = 0 , \qquad [9]$$

$$\mathbf{E}_z = -\frac{2Qx}{l}\mathbf{E}_x \ , \qquad [10]$$

with

$$\psi_0 = iQ\exp(-iQr_\dagger^2) \ , \qquad [11]$$

$$Q = \frac{1}{i + 2z_\dagger} \ , \qquad [12]$$

$$r_\dagger^2 = x_\dagger^2 + y_\dagger^2 \ , \qquad [13]$$

and $x_\dagger$, $y_\dagger$, and $z_\dagger$ the scaled co-ordinates

$$x_\dagger = x / \omega_0 \ , \ y_\dagger = y / \omega_0 \ , \ z_\dagger = z / l \ . \qquad [14]$$

The focal point is assumed to be located at the origin. In the CD simulations however the particle is located in the origin of the coordinate system, relative to which the scattered fields are calculated, and the focal point is shifted to another position (x_0, y_0, z_0). The expressions for the electric field change in an obvious way by this translation of the position of the focal point. As shown by Gouesbet et al. [22], this description of the electric field will give errors of $O(s^2)$.

8.2.3 The Generalised Lorenz-Mie Theory

The Generalised Lorenz-Mie theory (GLMT), developed by professor Gouesbet and his group, describes elastic light scattering from a homogeneous, isotropic, non-magnetic sphere located at an arbitrary position in a Gaussian laserbeam [22]. This theory introduces a new set of coefficients, the $g_{n,TE}^m$ and $g_{n,TM}^m$ (n = 1,…,∞; m = -n,…,n), into the familiar expressions for Lorenz-Mie scattering. These GLMT coefficients describe the effect of the Gaussian beam. Gouesbet et al use Davis' description of a Gaussian beam (see previous section) to lowest order in s (equations 8 - 10). This is called order L of approximation. As was shown by Gouesbet et al., this order L of approximation will result in errors of $O(s^2)$, which will be negligible in most cases.

In general, the complete double set of GLMT coefficients is needed, however, if the particle is located on the axis of the beam it is found that

$$g_{n,TM}^1 = g_{n,TM}^{-1} = \frac{1}{2}g_n \ , \ g_{n,TE}^1 = g_{n,TE}^{-1} = -\frac{i}{2}g_n \ ,$$

and for $|m| > 1$ the coefficients are zero. Furthermore, in the limit of an infinite beamwaist radius (a plane wave), the g_n coefficients are equal to 1, and the GLMT reproduces the Lorenz-Mie formula. We will not reproduce the expressions for the scattered electric fields, and the $g_{n,TM,TE}^m$ coefficients, as they

can be found in the original papers on the subject (reference 22, and references therein). However, it should be noted that the $g^m_{n,TM,TE}$ coefficients are double quadratures, containing highly oscillatory functions. Computation of the coefficients is far from trivial, e.g. direct evaluation of the quadratures on a IBM 3090 mainframe takes 30 seconds up to more than 2 hours of CPU time [38].

Beside direct evaluation, two more efficient methods were introduced to calculate the GLMT coefficients [38, 39]. The first method is rigorous, and relies on a finite series description, and the second method is an approximation which relies on van de Hulst's localisation principle. The localised formulation is most simple and efficient, and is readily included into existing Lorenz-Mie programs.

In the localised approximation, the small longitudinal element of the incident beam is not present, and therefore it should be seen as an approximation of the order L description of the GLMT. Gouesbet et al. call this the order L^- description. According to van de Hulst's localisation principle [40] a term of order n in the Mie series corresponds to a ray passing the origin at a distance $(n + 1/2)(\lambda/2\pi)$. In the original formulation of the localised approximation [41], for a sphere located in the centre of the beam, the g_n factors were associated with the amplitude of the Gaussian beam at a distance $(n + 1/2)(\lambda/2\pi)$ from the origin. Later, more rigorous methods to derive the expressions for the localised approximation were introduced. The localised approximation was extensively studied for a sphere located on-axis of the beam [38]. The g_n factors in the localised approximation compare very well with rigorous calculations, except for extreme small values of the beamwaist radius ($\omega_0 = \lambda$).

In case of an off-axis location of the sphere, a localised approximation was also developed [39]. This formulation relies on a rather unorthodox method to find correct expressions for needed normalisation constants. However, the validity of the conjecture leading to the normalisation constants, and the validity of the resulting localised approximation was shown numerically.

In the sequel of this paper we will present GLMT calculations using a computer program which was kindly provided to us by professor Gouesbet. This program includes the localised approximation to calculate the GLMT coefficients.

8.3 RESULTS

The difference between plane wave scattering and scattering from a focused beam is most prominent if ω_0/r is small, with r the radius of the sphere [see e.g. 41]. Therefore we should try to simulate the largest possible sphere, in combination with the smallest possible beamwaist radius.

The largest CD simulation that we can carry out at this moment is for l = 20, yielding a sphere discretised with 33552 dipoles [11]. This limit is due to memory limitations of the parallel computer. On the 512 node Parsytec GCel of

IC^{3}A* the l = 20 simulations runs for approximately 11 hours. As was shown in e.g. reference 13 the size of the dipoles can be as large as $\lambda/10$. With this value of d we find for the radius of the equivalent sphere r =2.0λ. This corresponds to a scattering parameter $\alpha = 2\pi r/\lambda = 12.556$. In the sequel we put λ = 488 nm, corresponding to the most intense blue line of an Ar-ion laser.

According to van de Hulst [40], the amplitudes of a_n and b_n in the Mie series are large for $n < \alpha$, drop sharply for n near α, and become virtually zero when n exceeds α by 2 or 3. This observation is clarified by his localisation principle. From this we can conclude that if it is possible to calculate the $g^m_{n,TM,TE}$ coefficients accurately for $n \leq \alpha + 3$, the resulting scattered field is also very accurate, since the contributions of $n > \alpha + 3$ in the Mie series, and therefore also in the GLMT series, can be neglected.

As was shown by Gouesbet et al. [reference 38, table V], the localised approximation to calculate g_n, with the sphere positioned at the beam waist centre and λ = 500 nm, gives errors smaller than 1 % for $n \leq 1$ if $\omega_0 = \lambda$, for $n \leq 10$ if $\omega_0 = 2\lambda$, $n \leq 28$ if $\omega_0 = 4\lambda$, and $n \leq 80$ if $\omega_0 = 8\lambda$.

Therefore, if we put $\omega_0 = 4\lambda = 1.952$ μm, the localised approximation of the GLMT will give good results for our sphere with $\alpha = 12.556$ located at the beamwaist centre. Furthermore, from reference 38 and 39 we may conclude that in this case the localised approximation still yields reliable results for small displacements of the sphere. To find a numerical value for these small displacements we rely once more on the localisation principle. A term with n = 28, for which the localised approximation still yields accurate results, corresponds to a ray at a distance of $(n + 1/2)(\lambda/2\pi) = 2.2$ μm from the z axis. Now we assume that if the particle is shifted in say the x direction with no more than 2.2 μm, the resulting $g^m_{n,TM,TE}$ coefficients calculated with the localisation principle are still accurate.

Therefore, we simulate scattering of a focused laserbeam with λ = 488 nm from a sphere with r = 0.976 μm, resulting in $\alpha = 12.556$. The refractive index of the sphere was 1.05. The focused laserbeam is travelling in the positive z-direction, and the beamwaist centre (the focal point) has co-ordinates (x_0, y_0, z_0). The laserbeam is plane polarised in the x-direction and the field is described by equations 8 - 10. The middle point of the sphere is located at the origin. We calculate the intensity of the scattered field in the far field region in the y-z plane, as a function of the scattering angle θ. The configuration is drawn in figure 1. Table 1 gives an overview of the simulations that have been carried out.

Number	ω_0 (μm)	x_0 (μm)	y_0 (μm)	z_0 (μm)
1	1000	0.0	0.0	0.0
2	1.952	0.0	0.0	0.0
3	1.952	0.976	0.0	0.0
4	1.952	1.952	0.0	0.0
5	1.952	1.952	1.952	1.952

* IC^{3}A is the Interdisciplinary centre of Computer based Complex systems Research Amsterdam; for more information contact the authors.

6	1.952	3.904	0.0	3.904
7	0.976	0.0	0.0	0.0
8	0.976	0.976	0.0	0.0
9	0.976	1.952	0.0	0.0

Table 1: An overview of the CD simulations

In experiment number 1 a CD simulation was performed with a very large beamwaist. This should effectively result in plane wave scattering. In figure 2 the results of this CD simulation are presented (the dots), together with the results of a Mie calculation (dashed line). Figures 3 until 10 show the results of experiments 2 until 9. In all these figures the dots are the results of the CD simulation, the straight line the corresponding GLMT calculation and the dashed line the plane wave (Mie) scattering result of figure 2.

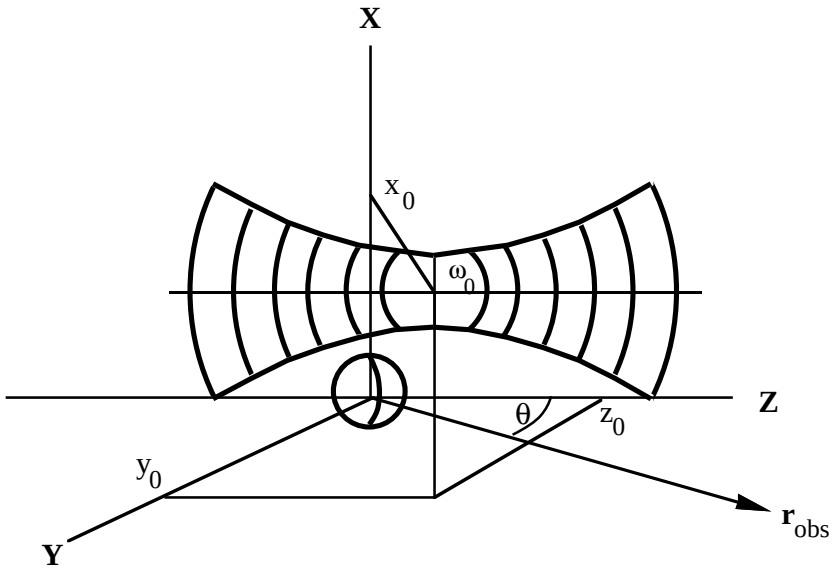


Figure 1: The scattering configuration; the centre of the sphere (radius = 0.976 μm; refractive index = 1.05) is positioned at the origin of a Cartesian coordinate system. The Gaussian laserbeam, with beamwaist radius ω_0, is travelling in the positive z-direction, the wavelength is 488 nm and the beam is plane polarised in the x-direction. The beamwaist centre has co-ordinates (x_0, y_0, z_0). The intensity of the scattered light (in the far-field region) is measured in $\mathbf{r}_{obs}$, as a function of the scattering angle θ.

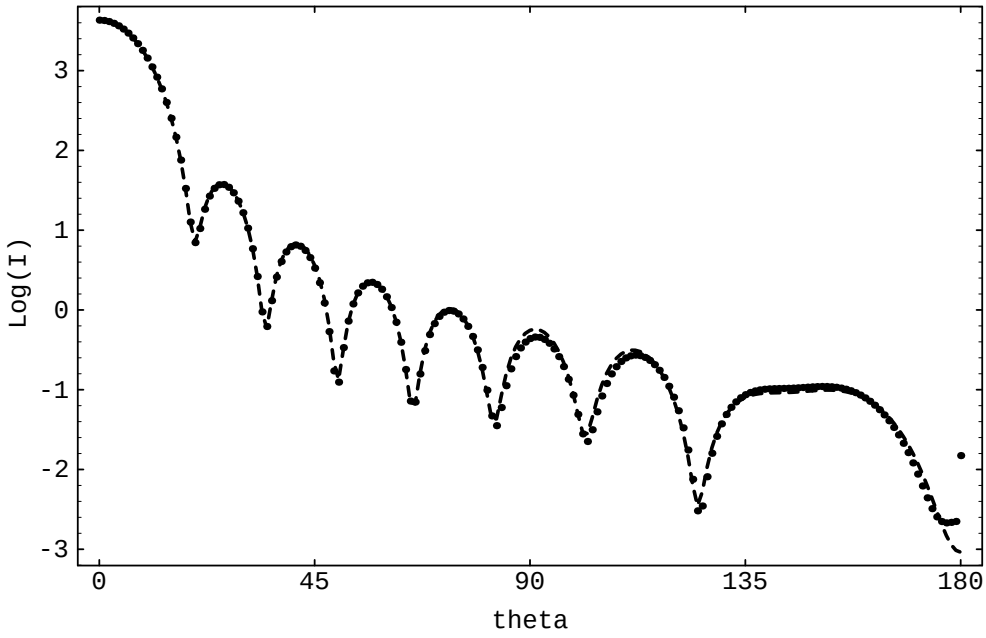


Figure 2: Results of experiment 1 (see table 1); ω_0 = 1000 μm, x_0 = 0.0 μm, y_0 = 0.0 μm, z_0 = 0.0 μm, the dots are the CD simulation, the dashed line is the plane wave Mie calculation.

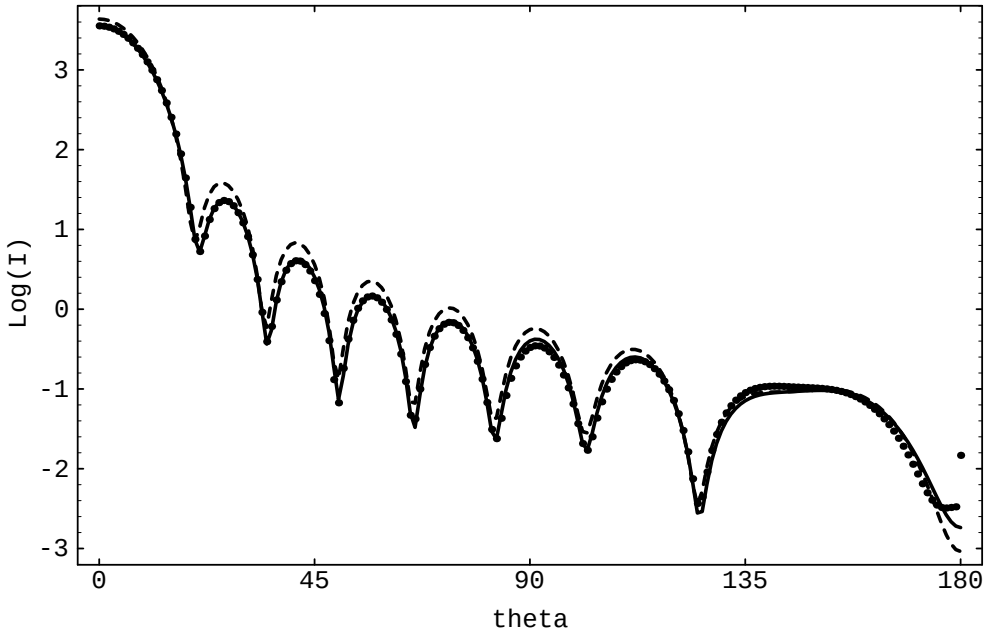


Figure 3: Results of experiment 2 (see table 1); ω_0 = 1.952 μm, x_0 = 0.0 μm, y_0 = 0.0 μm, z_0 = 0.0 μm, the dots are the CD simulation, the solid line is the GLMT calculation, the dashed line is the plane wave Mie calculation.

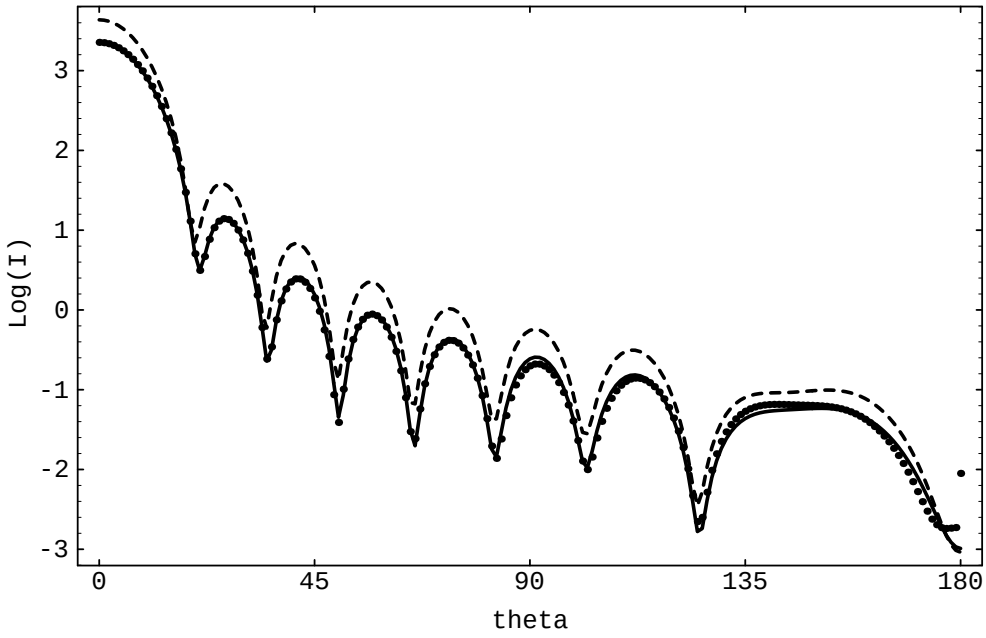


Figure 4: Results of experiment 3 (see table 1); ω_0 = 1.952 μm, x_0 = 0.976 μm, y_0 = 0.0 μm, z_0 = 0.0 μm, the dots are the CD simulation, the solid line is the GLMT calculation, the dashed line is the plane wave Mie calculation.

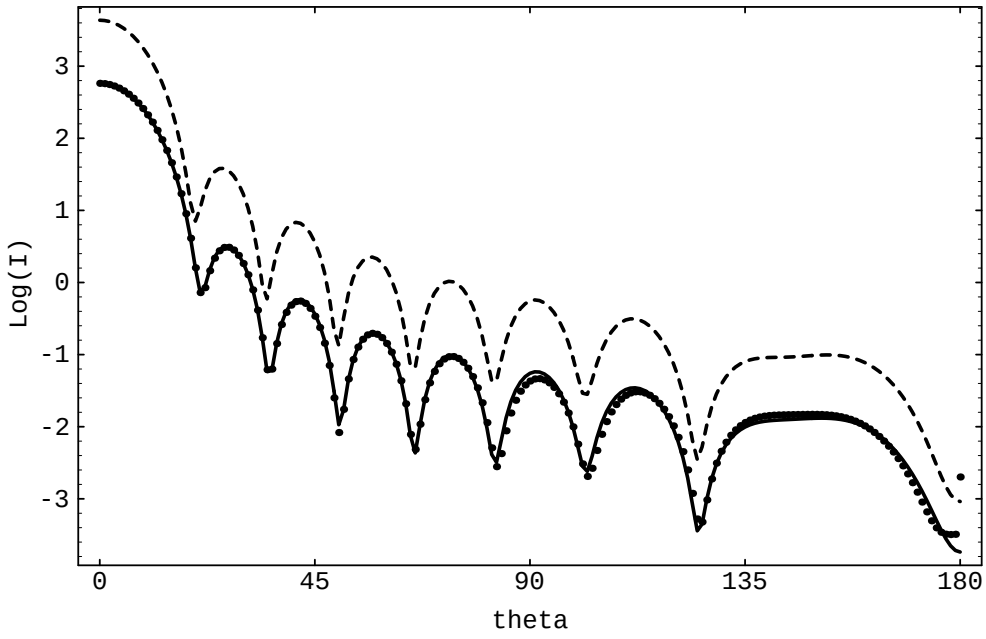


Figure 5: Results of experiment 4 (see table 1); ω_0 = 1.952 μm, x_0 = 1.952 μm, y_0 = 0.0 μm, z_0 = 0.0 μm, the dots are the CD simulation, the solid line is the GLMT calculation, the dashed line is the plane wave Mie calculation.

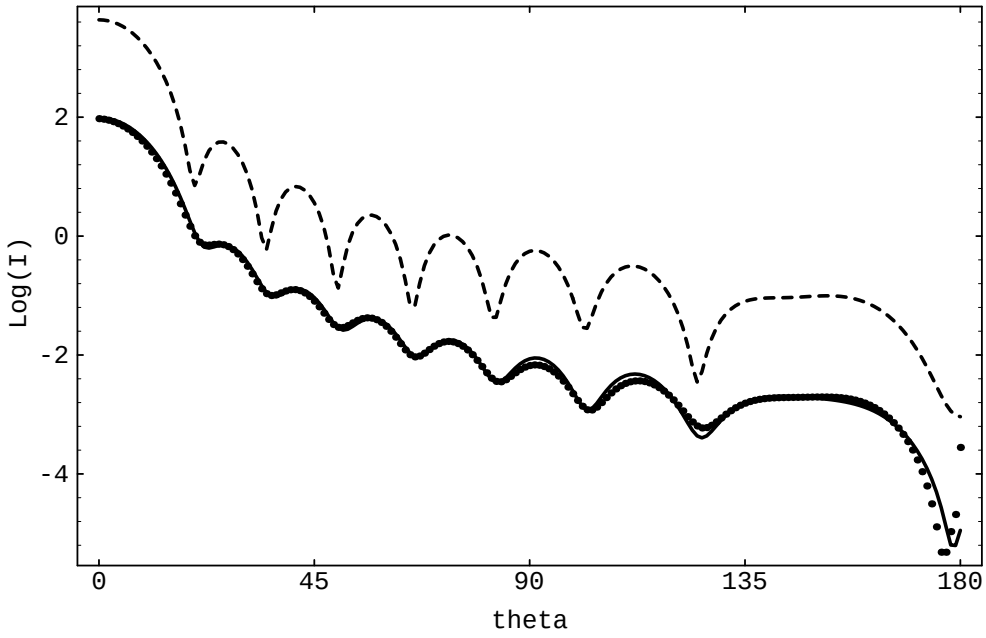


Figure 6: Results of experiment 5 (see table 1); ω_0 = 1.952 μm, x_0 = 1.952 μm, y_0 = 1.952 μm, z_0 = 1.952 μm, the dots are the CD simulation, the solid line is the GLMT calculation, the dashed line is the plane wave Mie calculation.

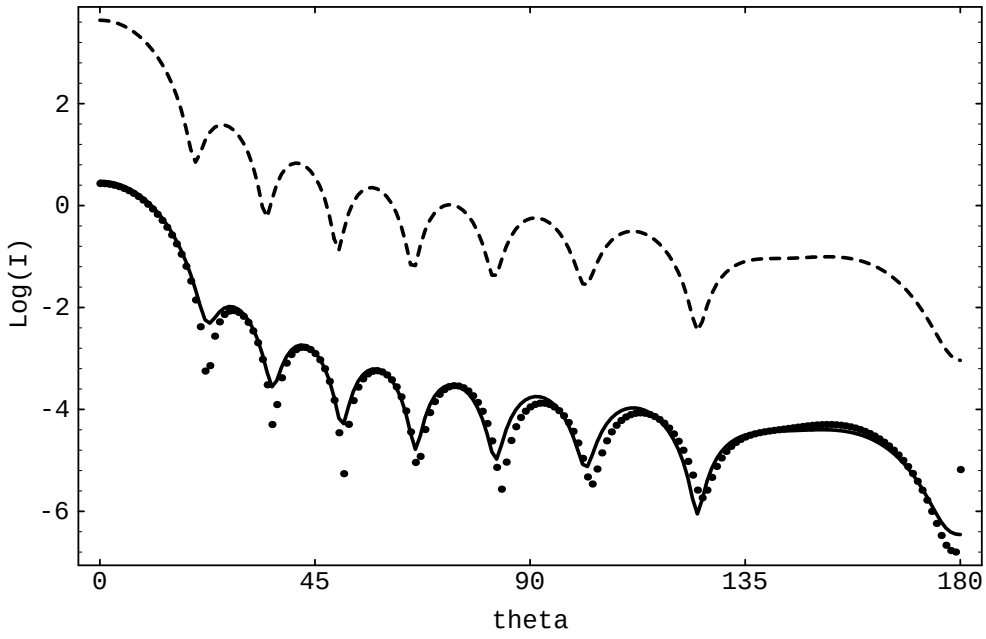


Figure 7: Results of experiment 6 (see table 1); ω_0 = 1.952 μm, x_0 = 3.904 μm, y_0 = 0.0 μm, z_0 = 3.904 μm, the dots are the CD simulation, the solid line is the GLMT calculation, the dashed line is the plane wave Mie calculation.

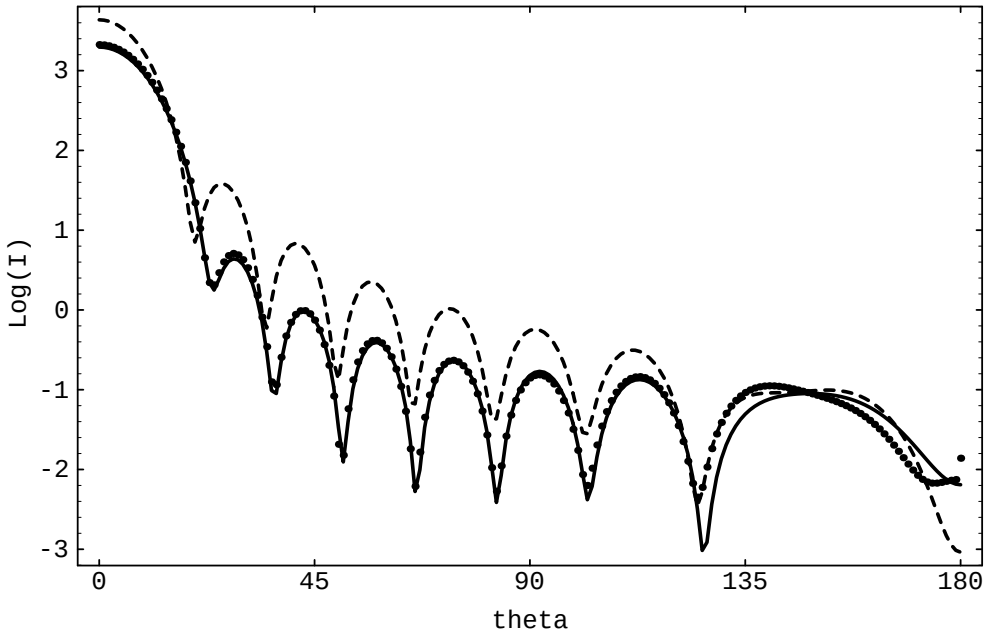


Figure 8: Results of experiment 7 (see table 1); ω_0 = 0.976 μm, x_0 = 0.0 μm, y_0 = 0.0 μm, z_0 = 0.0 μm, the dots are the CD simulation, the solid line is the GLMT calculation, the dashed line is the plane wave Mie calculation.

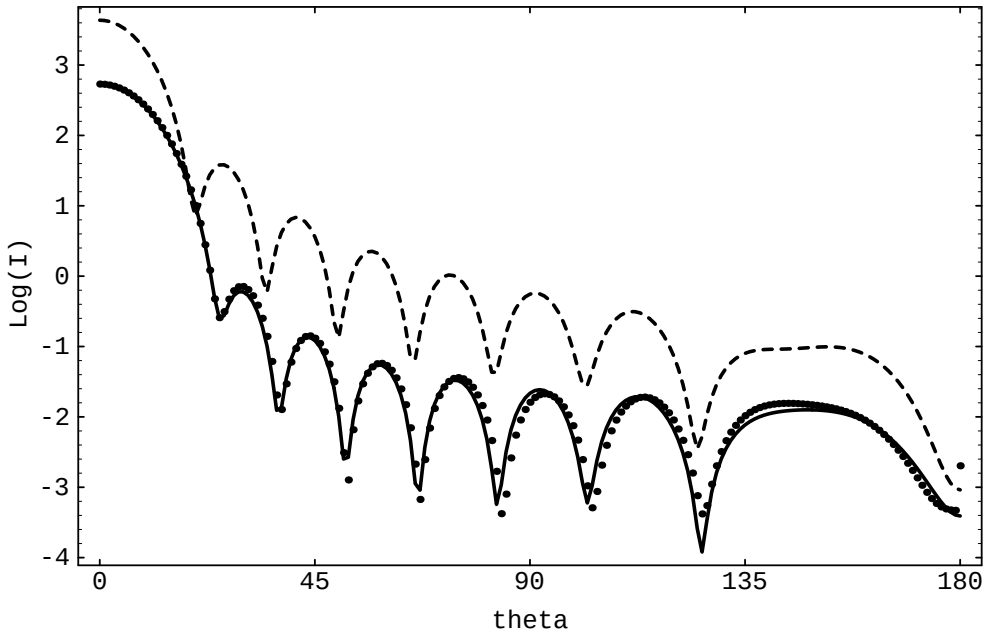


Figure 9: Results of experiment 8 (see table 1); ω_0 = 0.976 μm, x_0 = 0.976 μm, y_0 = 0.0 μm, z_0 = 0.0 μm, the dots are the CD simulation, the solid line is the GLMT calculation, the dashed line is the plane wave Mie calculation.

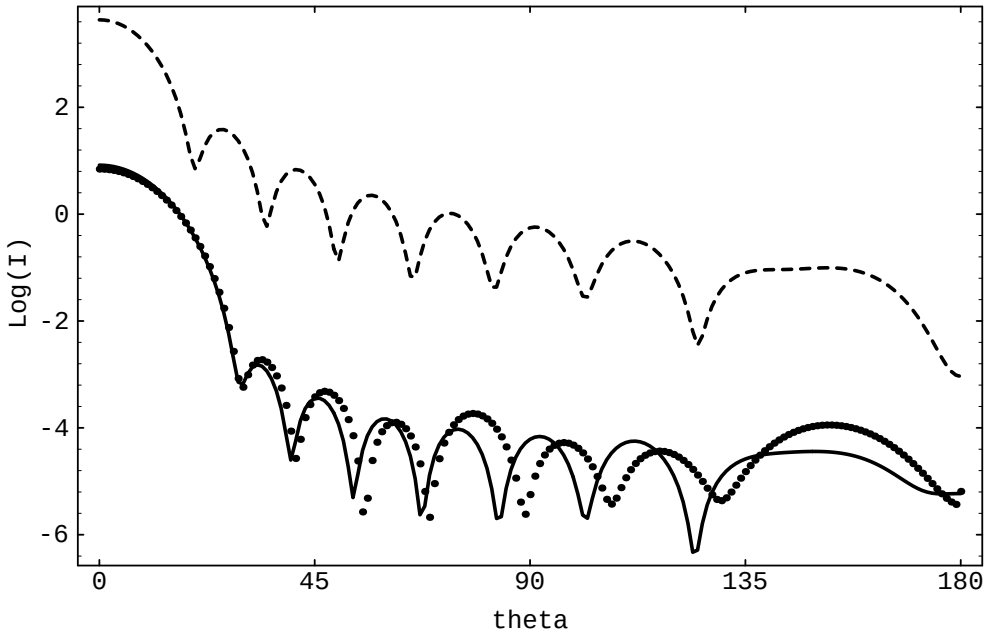


Figure 10: Results of experiment 9 (see table 1); ω_0 = 0.976 μm, x_0 = 1.952 μm, y_0 = 0.0 μm, z_0 = 0.0 μm, the dots are the CD simulation, the solid line is the GLMT calculation, the dashed line is the plane wave Mie calculation.

8.4 DISCUSSION

A need to compare CD simulations with exact solutions definitely exists, as was once again pointed out by Flatau et al. [42]. They specifically want to compare CD simulations with exact solutions for non-spherical particles. However, we concentrate here on another aspect of CD simulations, i.e. the ability to include arbitrary incident fields into the simulations. We have compared CD simulations of a spherical particle illuminated by a Gaussian beam with GLMT calculations, using the localised approximation to calculate the GLMT coefficients.

Unfortunately, both the localised approximation, and our current CD implementation pose their limits on the comparison that we can make. The localised approximation gives a lower limit on the size of the beamwaist radius ω_0, and our current CD simulations are limited to ~35000 dipoles, resulting in a simulated sphere of r = 0.976 μm. Therefore, our results should be viewed as first and limited, but still convincing experiments to show the ability of the CD method to describe scattering from Gaussian beams.

Our experiments can be grouped in four sets. Set one (experiment 1 from table 1) is a check of the plane wave limit of the CD simulation, set two (experiments 2 - 4) are experiments with ω_0 = 1.952 μm and small displacements of the sphere. In this region the localised approximation is known to be valid. Set three (experiments 5 and 6) consists of two experiments with the same ω_0, but with larger displacements of the sphere. Finally set 4 (experiments 7 - 9) comprises simulations with ω_0 = 0.976 μm. The localised

approximation is now no longer guaranteed.

Figure 2 shows that the CD simulations converge to the plane wave limit for very large ω_0. The CD results are identical to CD simulations with a plane wave incident beam (data not shown). The results compare very well with the Mie calculations, except for the backscattering directions.

The effect of focusing the beam to $\omega_0 = 1.952$ μm is not very dramatic (see figure 3). Still, minor shifts in e.g. the position of the first minimum, and a decrease of intensity of the maxima are accurately described by the CD simulation, and agree very well with the GLMT calculations. By gradually shifting the focal point in the x-direction (see figures 4 and 5) the overall intensity of the scattered light decreases, and the position of especially the first few minima shift to larger scattering angles. Again, the CD simulations agree very well with the GLMT calculations. From this set of experiments we can conclude that the CD method is able to accurately describe scattering from a focused laserbeam. However, in future research we will extend the domain of our tests considerably in order to provide more data to support this conclusion.

Figures 6 and 7 show the results for larger displacements of the focal point. In figure 6 the beamwaist centre is shifted in both the x, y and z direction with 1.952 μm, so within the tentatively derived 2.2 μm range. The results compare very well with the GLMT calculations. For more extreme displacements, as in experiment 6 (figure 7) the CD simulations are in error with the GLMT calculations. However, because we expect that the localised approximation is no longer valid in this region, we cannot conclude whether the CD simulations are in error. For this we need better reference data (e.g. GLMT calculations, using finite series to calculate the GLMT coefficients).

Finally figures 8 - 10 show the results for simulations and GLMT calculations for a very small beamwaist radius ($\omega_0 = 2\lambda = 0.976$ μm). In the forward scattering the agreement is good, especially for the case of the particle located at the beamwaist centre (figure 8). The agreement between the CD simulations and GLMT calculations becomes gradually worse if the beamwaist centre is shifted in the x-direction. One should be very careful to draw conclusions from this last set of simulations. The difference between plane wave scattering and the focused beam with the particle in focus (figure 8) is now much more prominent, and thus provides a much better test case for focused beam scattering. Furthermore, the CD simulation is in good agreement with the GLMT calculation , except for $\theta > 120^0$. However, as was argued in the previous section, in this case the localised approximation will introduce errors larger than 1% into the calculation of the g_n factors. It is very tempting to conclude that the localised approximation still yields accurate results, but this would reverse the role of the CD data and the GLMT data. The only accurate conclusion is that the CD simulations and the GLMT calculation using the localised approximation still agree very well for this extremely focused beam. Reliable reference data however is needed to decide whether the results are (indeed) correct. Of course one can speculate, based on the good results of the other experiments (1 - 6), that the CD simulations are correct for this highly focused beam, and that the limits of the localised approximation show up in the last three experiments.

An important point to remember in these calculations is the accuracy of the description of the Gaussian beam itself. In experiment 2 - 6 the beamwaist radius was $\omega_0 = 4\lambda$, thus $s = 1/(k\omega_0) = 1/(8\pi)$. Our first order description of the

incident field is accurate to 1.6E-3. However, in the case of the last 3 experiments, $\omega_0 = 2\lambda$, thus $s = 1/(4\pi)$, and the fields are accurate to 6E-3. In these highly focused beams one should probably have to include higher order terms into the description of the beam. In future research we will investigate the importance of these higher order terms for such highly focused beams, by using the results of Barton et al. [37], and examine if the CD method can handle that situation as accurate as the first order approximation of the Gaussian beam.

However, in practical situations, such as our flowcytometric experiments, ω_0 is much larger than the wavelength and higher order terms in the description of the Gaussian beam can be neglected. In that case we expect, based on the results presented in this work, that the CD method with a first order description of the Gaussian beam can accurately simulate the scattering of arbitrary shaped particles illuminated by a focused beam.

8.5 CONCLUSIONS

We have included a new element into Coupled Dipole simulations, scattering from incident fields other than plane waves. It was shown that the CD method can describe scattering of Gaussian beams from spheres very accurately. Unfortunately, the range of our tests was limited and more research is required. However, from these first results we can conclude that the CD method is able to describe the scattering of particles illuminated by a focused beam.

8.6 REFERENCES

1] P.M.A. Sloot, M.J. Carels, P. Tensen, and C.G. Figdor, "Computer-assisted centrifugal elutriation. I. Detection system and data acquisition equipment," Comp. Meth. Prog. Biomed. **24**, 179-188 (1987).

2] J.I. Hage and J.M. Greenberg, "A model for the optical properties of porous grains," Astrophysical Journal **361**, 251 (1990).

3] D.W. Shuerman, *Light Scattering by Irregularly Shaped Particles* (Plenum Press, New York and London, 1980).

4] P.M.A. Sloot, A.G. Hoekstra, H. van der Liet, and C.G. Figdor, "Scattering matrix elements of biological particles measured in a flow through system: theory and practice," Applied Optics **28**, 1752-1762 (1989).

5] C.F. Bohren and D.R. Huffman, *Absorption and Scattering of Light by Small Particles* (John Wiley & Sons, 1983).

6] P.M.A. Sloot and C.G. Figdor, "Elastic light scattering from nucleated blood cells: rapid numerical analysis," Applied Optics **25**, 3559-3565 (1986).

7] P.M.A. Sloot, A.G. Hoekstra, and C.G. Figdor, "Osmotic Response of Lymphocytes Measured by Means of Forward Light Scattering: Theoretical Considerations," Cytometry **9**, 636-641 (1988).

8] A.G. Hoekstra, J.A. Aten, and P.M.A. Sloot, "Effect of aniosmotic media on the T-lymphocyte nucleus," Biophysical Journal **59**, 765-774 (1991).

9] B.G. de Grooth, L.W.M.M. Terstappen, G.J. Puppels, and J. Greve, "Light-Scattering Polarisation Measurements as a New Parameter in Flow Cytometry," Cytometry **8**, 539-544 (1987).
10] E.M. Purcell and C.R. Pennypacker, "Scattering and absorption of light by nonspherical dielectric grains," The Astrophysical Journal **186**, 705-714 (1973).
11] A.G. Hoekstra and P.M.A. Sloot, *A parallel implementation of the Coupled Dipole method of Elastic Light Scattering*, to be published; see also chapter 5.
12] A.G. Hoekstra and P.M.A. Sloot, "Implementation of a parallel conjugate gradient method for simulation of elastic light scattering," In *Physics Computing '92* (World Scientific, J. Nadrchal and Groot, R. Ed., pp. 345-346, Singapore, 1993).
13] A.G. Hoekstra and P.M.A. Sloot, "Dipolar unit size in Coupled Dipole calculations of the scattering matrix elements," Optics Letters **18**, 1211-1213 (1993).
14] A.G. Hoekstra and P.M.A. Sloot, "New computational techniques to simulate Light Scattering from arbitrary particles," in *proceeding of the Third International Congress on Optical Particle Sizing*, 1993.
15] M.A. van Dilla, P.N. Dean, O.D. Laerum, and M.R. Melamed, *Flow Cytometry: Instrumentation and Data Analysis* (Academic Press, Analytical Cytology, 1985).
16] A. Ashkin, J.M. Dziedzic, and T. Yamane, "Optical trapping and manipulation of single cells using infrared laser beams," Nature **330**, 769-771 (1987).
17] A. Ashkin and J.M. Dziedzic, "Observation of optical resonances of dielectric spheres by light scattering," Applied Optics **20**, 1803-1814 (1981).
18] F. Guilloteau, G. Gréhan, and G. Gouesbet, "Optical levitation experiments to assess the validity of the generalised Lorenz-Mie theory," Applied Optics **31**, 2942-2951 (1992).
19] T. Bakker Schut, G. Hesselink, B.G. Grooth, and J. Greve, "Experimental and theoretical investigations on the validity of the geometrical optics models for calculating the stability of optical traps," Cytometry **12**, 479-485 (1991).
20] C. Yeh, S. Colak, and P. Barber, "Scattering of Sharply focused beams by arbitrarily shaped dielectric particles: an exact solution," Applied Optics **21**, 4426-4433 (1982).
21] J.P. Barton, D.R. Alexander, and S.A. Schaub, "Internal and near-surface electromagnetic fields for a spherical particle irradiated by a focused laser beam," J. Appl. Phys. **64**, 1632-1639 (1988).
22] G. Gouesbet, B. Maheu, and G. Gréhan, "Light scattering from a sphere arbitrarily located in a Gaussian beam, using a Bromwich formulations," J. Opt. Soc. Am. A **5**, 1427-1443 (1988).
23] J. Chevaillier, J. Fabre, and P. Hamelin, "Forward scattered light intensities by a sphere located anywhere in a Gaussian beam," Applied Optics **25**, 1222-1225 (1986).
24] J. Chevaillier, J. Fabre, G. Gréhan, and G. Gouesbet, "Comparison of diffraction theory and generalised Lorenz-Mie theory for a sphere located on the axis of a laser beam," Applied Optics **29**, 1293-1298 (1990).
25] G. Gréhan, G. Gouesbet, F. Guiloteau, and J.P. Chevaillier, "Comparison of the diffraction theory and the generalised Lorenz-Mie theory for a sphere arbitrarily located into a laser beam," Optics Comm. **90**, 1-6 (1992).
26] E. Esam, M. Khaled, S.C. Hill, P.W. Barber, and D.Q. Chowdhury, "Near-resonance excitation of dielectric spheres with plane waves and off-axis Gaussian beams," Applied Optics **31**, 1166-1169 (1992).
27] S.A. Schaub, D.R. Alexander, J.P. Barton, and M.A. Emanuel, "Focused laser beam interactions with methanol droplets: effects of relative beam diameter," Applied Optics **28**, 1666-1669 (1989).
28] J.A. Lock, "Contribution of high-order rainbows to the scattering of a Gaussian laser beam by a spherical particle," J. Opt. Soc. Am. A **10**, 693-706 (1993).
29] J.D. Jackson, *Classical Electrodynamics, second edition* (John Wiley & Sons, New York, Chichester, Brisbane, Toronto, Singapore, 1975).
30] J.I. Hage, M. Greenberg, and R.T. Wang, "Scattering from arbitrarily shaped particles: theory and experiment," Applied Optics **30**, 1141-1152 (1991).
31] G.H. Goedecke and S. O'Brien, "Scattering by irregular inhomogeneous particles via the digitised Green's function algorithm," Applied Optics **27**, 2431-2438 (1988).
32] D.E. Livesay and C. Kun-Mu, "Electromagnetic fields induced inside arbitrarily shaped biological bodies," IEEE trans. microwave theory **mtt-22**, 1273-1280 (1974).
33] J.C. Ku, "Comparisons of coupled-dipole solutions and dipole refractive indices for light scattering and absorption by arbitrarily shaped particles," J. Opt. Soc. Am. A **10**, 336-342 (1993).
34] L.W. Davis, "Theory of electromagnetic beams," Physical Review A **19**, 1177-1179 (1979).

35] H.W. Kogelnik and T. Li, "Laser beams and resonators," Applied Optics **5**, 1550 (1966).
36] M. Lax, W.H. Louisell, and W.B. McKnight, "From Maxwell to paraxial wave optics," Phys. Rev. A **11**, 1365-1370 (1975).
37] J.P. Barton and D.R. Alexander, "Fifth-order corrected electromagnetic field components for a fundamental Gaussian beam," J. Appl. Phys. **66**, 2800-2802 (1989).
38] G. Gouesbet, G. Gréhan, and B. Maheu, "Computations of the gn coefficients in the generalised Lorenz-Mie theory using three different methods," Applied Optics **27**, 4874-4883 (1988).
39] G. Gouesbet, G. Gréhan, and B. Maheu, "Localised interpretation to compute all the coefficients gnm in the generalised Lorenz-Mie theory," J. Opt. Soc. Am. A **7**, 998-1007 (1990).
40] Hulst, H.C., *Light Scattering by Small Particles* (Dover Publications, Inc., second, 1981).
41] G. Grehan, B. Maheu, and G. Gouesbet, "Scattering of laser beams by Mie scatter centres: numerical results using a localised approximation," Applied Optics **25**, 3539-3548 (1986).
42] P.J. Flatau, K.A. Fuller, and D.W. Mackowski, "Scattering by two spheres in contact: comparisons between discrete-dipole approximation and modal analysis," Applied Optics **32**, 3302-3305 (1993).

Chapter 9

CONCLUDING REMARKS

CONCLUDING REMARKS

9.1 INTRODUCTION

Are we able to simulate Elastic Light Scattering from human white blood cells, or more general, from typical biological particles using the Coupled Dipole method? The material presented in this thesis offers several viewpoints to answer this question. In retrospect, and keeping the original project description [1] in mind, we will address the question from a Physical, Numerical, and Computational point of view. In all three cases the answer to the question will be "Yes!", but the road which has to be travelled remains a long one. Therefore, looking into the future, we should also contemplate on the applicability of such Coupled Dipole simulations.

9.2 A PHYSICAL POINT OF VIEW

The largest Coupled Dipole simulations reported to date contain 30.000 to 80.000 dipoles (see chapter 5 and [2]). These simulations were compared with analytical results, and the agreement was very good. One may speculate that this agreement remains as good if the number of dipoles is increased to what is needed to model human white blood cells ($O(10^6)$, see chapter 5).

The Coupled Dipole method has a strong foundation in the macroscopic Maxwell equations, from which it can be derived directly (see chapter 1). Therefore we can expect that the model will yield reliable results for larger particles. The main problem is to find accurate values for the refractive index of the particles of interest.

All our calibration experiments as described in chapters 5 and 8 assume a relative refractive index of 1.05. If we want to simulate white blood cells, a good method to assess the refractive index of the constituents of a white blood cells must be available. Fortunately, Biology provides a plethora of techniques to separate cell compartments. Refractometer techniques can then be used to measure the refractive index of the cell compartments. Furthermore, we must be able to get the morphology of the particle in the computer. Here, advanced microscopy, combined with state of the art image processing technology, can be used to our advantage.

A combination of optical trapping techniques, confocal scanning microscopy of the trapped particle, and measurement of light scattered from the trapped particle, would result in a unique experiment to validate the Coupled Dipole simulations. Experiments to measure light scattering from white blood cells in optical traps are currently being set up by the group of de Grooth of the University of Twente. Furthermore, microwave analogy experiments, which we planned in the original proposal [1], provide important

calibration data for Coupled Dipole simulations. Unfortunately the facilities to perform such experiments are no longer available. However, the data which was gathered in the many microwave analogy experiments can of course be re-used to validate simulation methods like the Coupled Dipole method.

In conclusion, we expect that the Coupled Dipole method is able to simulate scattering from white blood cells. Moreover, comparison with analytical models and techniques to validate Coupled Dipole simulations of real particles can be developed.

9.3 A NUMERICAL POINT OF VIEW

From a numerical point of view, the Coupled Dipole simulation mainly is a very large set of coupled linear equations, with a complex, dense, symmetric, and diagonally dominant system matrix (see chapter 1 and 4). As was argued in chapter 4, merely the size of the matrix forces us to use iterative methods to solve the matrix equation. The question now is: if we increase the dimension of the systems matrix to $O(10^6)$, can we still solve the matrix equation? More specifically, will the iterative method converge and is the result of the iteration accurate enough.

Convergence and accuracy of our iterative method, a Conjugate Gradient method (see chapter 4), is determined by the spectrum of eigenvalues of the system matrix, and its associated condition number. Although we tried to estimate condition numbers for small interaction matrices, the results were not conclusive and could not be extrapolated to large matrix dimensions.

However, indirect evidence shows that the Coupled Dipole interaction matrix is a well conditioned system. First, the number of iterations needed for convergence is very small (see chapter 5: only 19 iteration for a matrix with dimension 10^5). Furthermore, the resulting scattered fields are very accurate compared to the analytical models. This implies that the result of the Conjugate Gradient iteration is also accurate.

The mathematician Alan Edelman publishes yearly accounts of Dense Numerical Linear Algebra. In his 1993 account, entitled "Large Dense Numerical Linear Algebra in 1993, the Parallel Computing Influence" [3], Edelman says "The electromagnetics community is by far the leader for large dense linear systems solving", and "the largest linear systems right now are generated from moment methods; these matrices are quite well-conditioned". The Coupled Dipole interaction matrix is equal to those generated from moment methods in electromagnetics, and therefore we are lead to expect that it is possible to scale the matrix to very large dimensions and still find accurate results.

Off course, as was also mentioned by Edelman, we should include condition estimators into the Coupled Dipole method. These estimates can be used to find an order of magnitude of the numerical error. Furthermore, they can be used to decide if preconditioning techniques have to be used to decrease the number of iterations and increase the numerical accuracy. In chapter 4 we shortly mentioned the polynomial preconditioners, which are promising

candidates as parallel preconditioners. Development of a (polynomial) preconditioner, suited for the Coupled Dipole interaction matrix, will be very important. This is especially true if one wants to use the Coupled Dipole method for large arbitrary particles, with refractive indices larger than those of typical biological material. As shown in chapter 4, in that situation the number of iterations needed for convergence rises sharply.

9.4 A COMPUTATIONAL POINT OF VIEW

9.4.1 Parallel Computing

We have parallelised the Coupled Dipole model for MIMD-DADM systems, and we have shown that the efficiency of our parallel Coupled Dipole method can be very close to 1 if the number of dipoles per processor is large enough (e.g. on the Parsytec GCel, programmed using Parix, larger than 30). Therefore, we can successfully exploit the enormous power of massively parallel systems.

However, the memory efficiency of the current implementation is not very good (see chapter 1). The largest simulations that can be carried out on the Parsytec contain 33.000 dipoles. If we would increase the number of nodes by a factor 2 and with that increase the complete memory in the system also with a factor 2, the maximum number of dipoles can hardly increase. This is due to the fact that most of the local memory of every node is consumed by the large buffer to store the argument vector in the parallel matrix vector product. This implies that we are not able to scale (i.e. increase) the problem size with the number of processors, which can result in very bad scalability (see chapter 1). Furthermore, a situation where the maximum problem size is determined by the local memory of one node instead of the total memory available in the massively parallel system, is off course highly undesirable.

This situation can be solved in two ways. First, we can redesign the parallel implementation to a grid decomposition of the interaction matrix implemented on a cylinder topology. According to chapter 4 this parallel version of the Coupled Dipole method will also posses very good scalability properties. Furthermore, this implementation does not need a buffer of size 3N (with N the number of dipoles), but buffers of size $3N/(p)^{1/2}$. Therefore, if the number of processors is increased with a factor 2, the maximum problem size can be increased with a factor $(2)^{1/2}$.

The second, and in our opinion most preferable solution, is to redesign the parallel matrix vector product with the column block decomposition of the matrix, to a systolic-like matrix vector product as described by e.g. Dontje et al. [4]. Here, one basically interleaves the vector gather operation and the matrix vector product. Every time a packet of 3N/p elements is received it is immediately processed by the matrix vector product. In this way we only need a buffer of length 3N/p instead of length 3N. The memory efficiency of this

implementation is $\varepsilon_m = 1$, resulting in a perfectly scalable algorithm, both from memory usage and performance point of view.

9.4.2 Mflop/s, Gflop/s, or Tflop/s?

Despite all previous arguments, the Coupled Dipole method has one huge drawback (as was shown in chapter 5): execution time. Even on the fastest available systems (1024 node CM-5, capable of 60 Gflop/s on the linpack [5]) the execution times for systems with $O(10^6)$ dipoles will be prohibitive. However, simulation of Light Scattering from human white blood cells requires such large systems. Therefore, is there any hope that we will ever be able to reach our goals?

The prohibitive execution time is due to the quadratic scaling of the execution time with the number of dipoles, which in its turn can be attributed to the matrix vector product. If want to carry out simulations containing $O(10^6)$ dipoles, the simulation must have a much smaller complexity, preferably linear in the number of dipoles.

As was noticed in chapter 5, the matrix vector product is a calculation of the electric field on the dipoles, due to radiation from all other dipoles. Therefore, the Coupled Dipole method can be viewed as a many-body simulation.

Many-body methods possess an algorithmic complexity of $O(N^2/2)$ if all pairwise interactions are calculated (the direct algorithm). For realistic simulations the number of interacting particles has to be very large. The $O(N^2/2)$ complexity of the direct algorithm is a severe restriction for these large scale many-body simulations. Even on the most powerful (massively parallel) supercomputers the execution times of realistic many-body simulations will soon rise above acceptable (or affordable) values.

The conclusion is that the algorithmic complexity of the direct method must be reduced. Some interaction potentials (e.g. Lennard-Jones) allow the use of cut-off techniques, which can reduce the complexity to O(N). However, for long range interaction potentials, such as the dipolar interaction potential, cut-off techniques cannot be applied. A very important class of "clever" many-body algorithms, which reduce the complexity to O(N LogN) or even to O(N), are the so-called hierarchical tree methods [6, 7]. 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.

Appel [8] introduced the first hierarchical tree method, which relies on using a monopole (centre-of-mass) approximation for computing forces over large distances, and on sophisticated data structures to keep track of which particles are sufficiently clustered to make the approximation valid. This method achieves dramatic speedups compared to the direct algorithm, but is less efficient when the distribution of particles is relatively uniform and the required precision is high. Barnes and Hut applied this method in simulations of interacting galaxies [9]. The next step, which was set by Greengard [6], is the use of multipole expansions to compute interaction potentials or forces. This approach is known as the Fast Multipole Method (FMM), and requires an amount of work proportional to N to evaluate all pairwise interactions to any degree of accuracy. Up till now FMM algorithms are developed for scalar 1/r

potentials in two and three dimensions [6, 10]. Salmon presents an overview of hierarchical tree methods [7].

We have to develop a FMM algorithm for the vector potential of radiating dipoles (in three dimensions). This FMM algorithm should replace the matrix vector products in the iterative solver of the Coupled Dipole simulation. Now the interaction between the dipoles is not calculated for each dipole pair directly, but the dipoles are grouped together in a hierarchical way, and the interaction between single dipoles and this hierarchy of dipolar groups is calculated. In this way the complexity of the complete CD simulation can be reduced to O(N). It should be noted that N has to be large to reach a cross over in execution time between the direct algorithm and the FMM algorithm. The FMM algorithm should be build along the same lines as Greengard's FMM algorithm for scalar 1/r potentials in three dimensions [6]. The algorithm will consist of three steps (we omit the mathematical and algorithmic details):

1] form multipole expansions for the vector potentials of the hierarchy of dipolar groups (upward pass);
2] compute the interactions between all dipoles at the coarsest possible level in the hierarchy; for a given group of dipoles in the hierarchy this is accomplished by including interactions between groups which are well separated from each other, and whose interactions are not accounted for at a higher stage in the hierarchy (downward pass);
3] using the resulting vector potential on each dipole, calculate the wanted electric field.

Hierarchical tree methods have proven to be very efficient and accurate, and well suited to be used in realistic many-body simulations. However, efficient implementation on High Performance Computing platforms, specifically massively parallel distributed memory computing systems, if far from obvious. Salmon has successfully implemented the Barnes-Hut method on the Caltech hypercubes [7]. The FMM is implemented on shared memory multicomputers [11], and on the connection machine CM-2 [12]. Furthermore, Leathrum and Board report on FMM implementations on a number of platforms, such as the Intel Touchstone, transputers, Encore Multimax, and distributed workstations running PVM and Linda [13].

The importance of hierarchical many body methods, and especially efficient implementations of such methods on massively parallel systems, is also recognised in the Netherlands. Recently a pilot project was started [14] which will, among others, concentrate on methodologies to find efficient implementations of hierarchical methods on massively parallel distributed memory computers. Off course, this research is directly applicable in the parallel Coupled Dipole method.

9.5 A LOOK INTO THE FUTURE

Do we still need large scale simulations such as the Coupled Dipole

method? Yes, the light scattering - and optical particle sizing community needs these simulations. The most recent Optical Particle Sizing conference [15] contained several applications of the Coupled Dipole method, which all suffered from the fact that they could not simulate large enough systems. Furthermore, many groups rely on analytical or approximate theories to describe scattering from arbitrary particles, although they never were able to test if they are allowed to do so. Coupled Dipole simulations may serve as a validation of such approximate theories.

Our own application of scattering from human white blood cells can also still profit from these simulations. Although flowcytometric experiments are very powerful to distinguish human white blood cells (see chapter 1), the wealth of information hidden somewhere in the light scattering matrix is just waiting to be uncovered, either by experimentation and/or by simulation.

Which role will be reserved for Coupled Dipole simulations? As was argued in chapter 5 and in reference 16, Coupled Dipole simulations up to 1.000 dipoles can be performed routinely on (networks of) powerful workstations. Systems containing $O(10^4)$ dipoles need much more computing power, and orientational averages will become prohibitive. Finally, even larger systems will need innovations as described in section 9.3. Even if we are able to design an efficient parallel implementation of an O(N) Coupled Dipole method, it remains to be seen if such programs can routinely be used to simulate very large system with $O(10^6)$ dipoles. Probably, these simulations will only be carried out sporadically and serve as calibrations or validations of less compute intensive methods.

Finally, is it possible to find alternatives for the Coupled Dipole method? A popular method, which was developed in conjunction with radar-cross section calculations, is direct integration of the Maxwell equations in the time domain [17]. These finite difference time domain (FDTD) methods are very efficient to solve scattering of pulses of electromagnetic radiation. However, this method is also applicable to plane wave scattering. Especially if we are dealing with weak scatterers (i.e. small refractive indices) the FDTD approach can be much more efficient than the Coupled Dipole method. Furthermore, one can speculate if the accuracy which is offered by the Coupled Dipole method is absolutely necessary in our application, where all fine structure in the differential cross section will be damped by biological variability of the particles. If one accepts larger errors in the final solution, one can stop the integration of the FDTD algorithm after only a few cycles of the incident wave, thus reducing the computations drastically.

9.6 CONCLUSIONS

The Coupled Dipole method allows, in principle, simulation of Elastic Light Scattering of arbitrary particles. In practice however the calculation times to solve the Coupled Dipole equations soon become prohibitive. We have developed a parallel Coupled Dipole simulation system, which allows us to perform these simulations on modern massively parallel computing platforms,

but also on clusters of workstations. Now very large CD simulations can be performed on high end systems, and light scattering from smaller arbitrary particles can be simulated on workstations or personal computers. In the near future, hierarchical methods will open the way to simulations containing $O(10^6)$ dipoles, thus allowing us to simulate scattering from human white blood cells.

9.7 REFERENCES

1] P.M.A. Sloot and L.O. Hertzberger, *Simulation of Light Scattering from micron-sized particles by means of computational physics methods,* Tech. Rept. 810-410-04 1 NWO, 1990.

2] P.J. Flatau, K.A. Fuller, and D.W. Mackowski, "Scattering by two spheres in contact: comparisons between discrete-dipole approximation and modal analysis," Applied Optics **32**, 3302-3305 (1993).

3] A. Edelman, "Large Dense Numerical Linear Algebra in 1993: the parallel computing influence," International J. SuperComputer Appl. **7**, 113-128 (1993).

4] T. Dontje, N. Petkov, and K. Schilling, "Computational and Communicational Granularity of Systolic Algorithms on the Connection Machine," In *Parallel Computing '91* (Elsevier Science Publishers B.V., D.J. Evans, G.R. Joubert, and H. Liddell Ed., 1992).

5] J.J. Dongarra, H.W. Meuer, and E. Strohmaier, *TOP500 Supercomputers*, published electronically, for more information send electronic mail to top500@rz.uni-mannheim.de.

6] L.L. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems* (The MIT Press, 1988).

7] J.K. Salmon, *Parallel Hierarchical N-Body Methods*, Ph.D. dissertation, California Institute of Technology, Pasadena, USA, 1991.

8] A.W. Appel, "An Efficient Program for many Body Simulations," SIAM J. Sci. Stat. Comput. **6**, 85 (1985).

9] J. Barnes and P. Hut, "A hierarchical O(N log N) force-calculation Algorithm," Nature **324**, 446 (1986).

10] K.E. Schmidt and M.A. Lee, "Implementing the Fast Multipole Method in Three Dimensions," J. Stat. Phys. **63**, 1223 (1991).

11] L. Greengard and W.D. Gropp, "A Parallel version of the Fast Multipole Method," Computers Math. Applic. **20**, 63 (1990).

12] F. Zhao and S. Lennart Johnsson, "The Parallel Multipole Method on the Connection Machine," SIAM J. Sci. Stat. Comput. **6**, 1420 (1991).

13] J. Leathrum, James F. and J.A. Board, *The Parallel Fast Multipole Algorithm in Three Dimensions,* Tech. Rept. 27706 Duke University, Department of Electrical Engineering, Durham, NC, USA, 1992.

14] P.M.A. Sloot, L.O. Hertzberger, S.W. de Leeuw, H.J.C. Berendsen, and D. Frenkel, *Portable Parallel Simulation of Crystal Nucleation and Growth,* Tech. Rept. 93MPR01/1 FOM, 1993.

15] M. Maeda (Ed). *Proceedings of the3rd International Congress on Optical Particle Sizing '93*, 1993.

16] A.G. Hoekstra and P.M.A. Sloot, "New computational techniques to simulate Light Scattering from arbitrary particles," in *proceeding of the Third International Congress on Optical Particle Sizing*, 1993.

17] A. Taflove and K.R. Umashankar, "Review of FD-TD Numerical Modelling of Electromagnetic Wave Scattering and Radar Cross Section," Proceedings of the IEEE **77**, 682-699 (1989).

SUMMARY

Elastic Light Scattering is a powerful technique to asses morphological or optical properties of small particles. We apply this technique to identify subsets of human white blood cells and bonemarrow cells. Previous work has shown that the polarisation of the scattered light is an indispensable element to achieve this goal, but also that exact - and approximate theories of Elastic Light Scattering cannot accurately describe the complete scattering matrix of arbitrary particles.

The main part of this thesis describes the development and implementation of a computer experiment to simulate Elastic Light Scattering from arbitrary particles. The basis of this simulation is the Coupled Dipole method. Although this method can describe the complete scattering matrix of arbitrary particles, it has very large computational requirements when applied to particles with scattering parameters in the order of 30, like white blood cells scattering visible light. Parallel computers can offer this computational power if the parallel program has good scalability properties.

The scalability of the kernel of the Coupled Dipole method, calculation of the electric field on the dipoles using a Conjugate Gradient iteration method, is thoroughly investigated. Several data decompositions of the interaction matrix, implemented on three different processor network topologies were analysed. This complexity analysis showed that a rowblock decomposition of the matrix, implemented on a ring of processors, will result in efficiencies very close to 1 if the number of rows per processor can be kept large enough. Furthermore, actual run time measurements of the parallel Conjugate Gradient method agree within 5 % with the theoretical time complexity formulas.

The parallel Coupled Dipole method was implemented using the parallel Conjugate Gradient kernel, and the scalability properties were measured. These measurements confirm that our parallel Coupled Dipole implementation has a very good efficiency and therefore is able to exploit the power of massively parallel computers.

The parallel Coupled Dipole method is applied to simulate the scattering of spheres. These simulations serve as a further calibration of the computer experiment. First, the influence of the size of the dipoles on the quality of Coupled Dipole simulations of the scattering matrix elements is investigated. We conclude that the size can be as large as $\lambda / 10$ provided that the surface roughness of the discretised particle is small enough. Secondly a new element is introduced into Coupled Dipole simulations, scattering of focused beams from arbitrary particles. Scattering by focused beams is highly relevant for the simulation of Elastic Light Scattering in flowcytometers. By comparing the simulation results with generalised Lorenz Mie calculations, it is shown that the Coupled Dipole method can predict scattering from focused beams.

Finally this thesis contains a continuation of our experimental studies of the Elastic Light Scattering from human white blood cells. First, we elaborate on the hypothesis that the nuclear volume of osmotically stressed lymphocytes has to change in order to explain observed Elastic Light Scattering data. It is shown that this hypothesis is correct. Furthermore, we propose a possible mechanism of this nuclear growth, a mechanical link between the cell membrane and the nuclear envelope, and model this system by means of a

straightforward one-dimensional visco-elastic system.

Finally, we have continued the effort to measure the complete scattering matrix of particles in flowcytometers. The polarisation properties of the scattered light contain a wealth of information, and are absolutely necessary to distinguish all subsets of human white blood cells and bonemarrow. In co-operation with de Grooth of the University of Twente, the Netherlands, we extended the first experiments carried out by Sloot. During calibration of the new equipment we encountered an exiting phenomenon: Lissajous loops in scatterplots of spheres. These loops are investigated by a computer simulation of the experiment. We show that these loops originate from the extreme dependency of the intensity of the sideward scattered light on the radius of the spheres.