

Journal of Quantitative Spectroscopy & Radiative Transfer 109 (2008) 1543–1548

Journal of Quantitative Spectroscopy & Radiative Transfer

www.elsevier.com/locate/jqsrt

Note

Mie theory 1908, on the mobile phone 2008

Thomas Wriedt*

FB4, Verfahrenstechnik, Universität Bremen, Germany

Received 19 November 2007; received in revised form 8 January 2008; accepted 9 January 2008

Abstract

Hundred years ago Gustav Mie published his famous paper on the theory of light scattering by spherical particles. It has been programmed in many programming languages since invention of the computer. Nowadays with the mobile phone becoming an educational tool it would be of interest to port a Mie scattering application to a Java enabled hand set. The objective of this paper is to investigate the feasibility of porting existing Mie software to a hand held device. We will show that the mobile phone can be an advanced programming environment for the Mie theory and we will present some exemplary computational results.

© 2008 Elsevier Ltd. All rights reserved.

Keywords: Light scattering; Mie theory; Gustav Mie; Mobile phone; Matlab; Java ME

1. Introduction

In 1908 Gustav Mie published his famous paper on simulation of the colour effects connected with colloidal Gold particles [1], which has been translated in 1976 [2]. In this paper he gave a first outline of how to compute light scattering by small spherical particles. Hundred years after publication his paper is still much cited [3]. The citations show that the applications of Mie theory cover a wide range of subjects from climate research, optical particle characterization to interstellar dust in astrophysics. Mie theory has been implemented in various traditional programming languages starting of cause from FORTRAN. But Mie programmes are also available in Pascal, C + + and Java. It has also been programmed in computer algebra systems such as Maple and Mathematica. Additionally it is available in numerical mathematics systems such as IDL and Matlab [4]. In recent times, the mobile phone or smart phone has gained popularity as an active educational tool for various purposes [5]. So for students to quickly obtain a good intuition and a good working knowledge of Mie's theory a port to the mobile phone might be useful. For the purpose of teaching students in the application of Mie theory a small Mie scattering program was ported to the mobile phone. This article discusses the different way to port Mie theory onto a Java enabled mobile phone. We will present some computational results to check the application and to demonstrate its capability using an emulator and a real phone.

^{*}Tel.: +49 421 218 2507; fax: +49 421 218 5378.

E-mail address: thw@iwt.uni-bremen.de

 $^{0022\}text{-}4073/\$$ - see front matter @ 2008 Elsevier Ltd. All rights reserved. doi:10.1016/j.jqsrt.2008.01.009

2. Mobile phone programming

For programming of the mobile phone there are various systems available. Java is the most popular choice for development. Almost all manufacturers of mobile phones provide some Java Micro Edition (Java ME) platform for program development. Java ME (also called J2ME) enables writing Java applications (Java Midlets) for cellular phones.

The Java runtime environment is especially adapted to mobile devices that have some limitations compared to standard desktop computers. These include limited memory, small screen sizes, reduced input methods, and slow processors. There are a number of advantages with Java ME. Java ME is an industry standard for Java enabled mobile phones. It is a free and open platform. Java applications are of portable nature which ensures that an application written for one brand of mobile phone should work with all other brands.

Based on Java ME there are also other integrated development environments such as the Mobile BASIC [6] or the MIDLetPascal [7] compiler and development environment. Both generate Java bytecode that runs on any J2ME device. Additionally both feature float variables which will be needed for programming of Mie theory. There is also a C programming environment call Cibyl [8] based on GNU compiler tools and also including floating point support. Cibyl allows to compile C programs and other languages supported by GCC to be executed on Java ME capable phones [9]. All these three environments as Java still miss complex arithmetics such that a Mie program would have to be converted to real arithmetics.

But there are other mathematics programs which have been ported to mobile phones using Java ME that include complex arithmetics. The first one is the symbolic algebra software Jasymca [10] which ports Maxima to Jave ME. Maxima is based on the computer algebra system Macsyma. There is also a Matlab clone called FnattLabME [11], a numerical mathematics software. This ports FnattLab, a Java-based Matlab clone, to Java ME. The original FnattLab [12] runs on standard Java. The parser of FnattLab follows the original Matlab grammar as closely as possible. For the subset of Matlab commands and operators that are implemented in FnattLabME we refer the interested reader to an electronic paper by Dersch [13]. As there are a number of Matlab Mie scattering applications, FnattLabME has been chosen as numerical mathematics systems for the port. FnattLabME also includes a basic PLOT command which helps to visualize computational results in a diagram. A further advantage is that no Java programming is needed to port the application.

3. Porting of Mie program

There are various Matlab programs implementing Mie's theory. For porting to FnattLabME the program published by Barnett in September 1997 [14] has been chosen as it is a straightforward version of the FORTRAN program published in the book by Bohren and Huffman [15]. For a profound explanation of Mie's theory the interested reader is referred to this book.

This program immediately ran on FreeMat (a Matlab clone) [16]. But some modifications had to be made to the original application to obtain correct computational results within FnattLabME on the phone concept emulator included with the SUN J2ME Wireless Toolkit [17]. The J2ME emulator is used to test the application on a desktop computer without the inconvenience of downloading and running it on a real mobile phone [18]. Debugging an application with the emulator will help to develop it such that it will run correctly on a variety of mobile devices. One can also run an application using different devices in the emulator to identify and understand portability issues to other devices.

A simple way to include the scattering application in FnattLabME is to integrate the new m-files to FnattLabME directly by adding the files to the m-file directory of the FnattLabME.jar archive. Of cause in the FnattLabME.jad file the size of the new FnattLabME.jar has to be adapted accordingly. With FnattLabME you can also read and edit m-files from a memory cards using FnattLabME's file browser and editor.

The adaptations needed to run the application include conversion of two lines of complex arithmetics to real arithmetics because FnattLabME does not support complex array–array multiplication. This problem originates from the Jama [19] numerical library, which is a basic linear algebra package for Java, used in the second edition of FnattLabME [13].

The FnattLabME wrongly interpreted a vector for an array in one instance. So some adaptation of the original program was needed. Some multiple calls of the spherical Bessel function routine were needed because the array including the results was somehow over written by some following instruction such that the array including the spherical Bessel function values was no longer available. This problem was solved by an additional call of the spherical Bessel function. Actually our port of the Mie scattering application is only a basic port of the original Matlab Mie program. Only minor modifications were made to the original Matlab application to retain core functionality. No efforts were made to optimize the program with respect to computational time or use of computational resources. Neither did we recompile the original FnattLabME Java program. We only added the new m-files.

There are many ways for optimization to reduce the size of the code. The original FnattLabME Java code could be adapted by deleting included m-files that are not needed. Additionally FnattLabME could be recompiled with only those Matlab commands needed for the Mie scattering application.

4. Computational results

In this section we would like to present some exemplary computational results using the Mie scattering application. This has been tested on the phone emulator of the SUN J2ME Wireless Toolkit and additionally on a real phone, the Nokia 6680. The computational results were checked by comparing to results obtained using FreeMat.

FnattLabME including the Mie scattering application can be installed on a mobile phone or the phone emulator from a web page [20]. FnattLabME is distributed under GNU-license. Following installation you can call FnattLab and test the application using some basic Matlab commands. The Mie scattering application to plot a scattering diagram is started using the following command mieplot (m, k, pol) with m the refractive index, k the size parameter and pol the incident polarization 0 meaning s (orthogonal) polarization and 1 standing for p (parallel) polarization. The size parameter is defined by $k = \pi d/\lambda$ with d the diameter of the sphere and λ the incident wavelength. The scattering diagram is computed from 0° to 180° with 181 plot points.

In the SUN phone emulator the input window looks like shown in Fig. 1. In Figs. 2 and 3 the computed scattering diagrams are plotted for p (parallel) and s (orthogonal) polarization. The scattering patterns obtained using the Nokia 6680 mobile phone are shown in Figs. 4 and 5. With this mobile phone the

| Tatt | REC | (CC) |
|----------------|------|------|
| FnattLab | | |
| Output | | |
| | | |
| | | |
| | | |
| FnattLab Versi | on 1 | |
| >> m = 1.5; | | |
| >> k = 10; | | |
| >> pol = 0; | | |
| | | |
| Input | | |
| mieplot(m,k,p | oj. | |
| | .1 | |
| | | |
| | | |

Fig. 1. Calling the Mie scattering application in FnattLab.



Fig. 2. Scattering diagram on emulator for m = 1.5, k = 10 and s polarization.



Fig. 3. Scattering diagram on emulator for m = 1.5, k = 10 and p polarization.

computational time was about 10 min for one diagram. In Fig. 6 the duration of a simulation run on the Nokia 6680 mobile phone is plotted versus the size parameter.

With the phone emulator the largest size parameter we got correct results was 10. On the Nokia 6680 there was no problem with a size parameter as large as 20. From the diagrams you see that the graphical resolution of the mobile phone is quite low whereas the diagrams displayed on the emulator are almost perfect. Nevertheless there is still an error in the plotting routine of FnattLabME as seen in the numerical labelling of the *x*-axis. Numbers higher than 100 are not displayed correctly on the *x*-axis of all Figs. 2–5.

5. Conclusion

In this paper we presented a port of a Matlab Mie scattering application on a Matlab clone of a mobile phone. On a real phone we obtained correct scattering diagrams for size parameters as large as 20. We hope that this application will be of interest in teaching students.



Fig. 4. Scattering diagram on Nokia 6680 phone for m = 1.5, k = 10 and s polarization.



Fig. 5. Scattering diagram on Nokia 6680 phone for m = 1.5, k = 10 and p polarization.



Fig. 6. Duration of simulation on Nokia 6680 mobile phone versus size parameter k.

Acknowledgement

I would like to acknowledge Helmut Dersch for his comments on a draft version of this paper and Osman Kurt for help with the real phone computations.

References

- [1] Mie G. Beiträge zur Optik trüber Medien, speziell kolloidaler Metallösungen. Ann Phys Vierte Folge 1908; Band 25(3):377-445.
- [2] Mie G. Contributions to the optics of turbid media, particularly of colloidal metal solutions. Royal Aircraft Establishment, Library Translation 1873, 1976, RAE-Lit-Trans-1873 (http://diogenes.iwt.uni-bremen.de/vt/laser/papers/RAE-LT1873-1976-Mie-1908-translation.pdf).
- [3] Marx W. Dornröschen und Mauerblümchen. Nachzügler und vergessene Arbeiten in der Physik. Phys Unserer Zeit 2007;1:34-9.
- [4] Wriedt T. Electromagnetic scattering programs (www.T-Matrix.de).
- [5] McNeal T, van't Hooft M. Anywhere, anytime: using mobile phones for learning. J Res Center Educ Technol Winter 2006 (http:// www.rcetj.org).
- [6] Mobile BASIC 2.1---off phone development environment (http://www.mobilebasic.com/DesktopEdition.html).
- [7] MIDletPascal (http://en.wikipedia.org/wiki/MIDletPascal).
- [8] Cibyl, C programming environment (http://spel.bth.se/index.php/Cibyl).
- [9] Kågström S, Grahn H, Lundberg L. Cibyl—an environment for language diversity on mobile devices. In: Proceedings of the 3rd international conference on virtual execution environments, San Diego, CA, USA, June 13–15, 2007. p. 75–82.
- [10] Dersch H. Jasymca: symbolic calculator for mobile devices, following GNU-Maxima (http://webuser.fh-furtwangen.de/~dersch/).
- [11] Dersch H. FnattLabME: Java based Matlab-compatible calculator for cellphone and PDA (http://webuser.fh-furtwangen.de/ ~dersch/).
- [12] Fnattlab, Matlab clone written in Java (http://sourceforge.net/projects/fnattlab/).
- [13] Dersch H. FnattLabME 2. Release, March 22, 2006 (http://webuser.hs-furtwangen.de/~dersch/FnattLabME/FnattLabME2.pdf).
- [14] Barnett D. Matlab Mie functions (http://www.lboro.ac.uk/departments/el/research/photonics/matmie/mfiles.html).
- [15] Bohren CF, Huffman DR. Absorption and scattering of light by small particles. New York: Wiley-Interscience; 1993.
- [16] FreeMat, free environment for rapid engineering and scientific prototyping and data processing (http://freemat.sourceforge.net).
- [17] Sun Java Wireless Toolkit (formerly known as Java 2 Platform, Micro Edition (J2ME) Wireless Toolkit) (http://java.sun.com/ products/sjwtoolkit/).
- [18] Yuan MJ. Enterprise J2ME: developing mobile Java applications. Englewood Cliffs, NJ: Prentice-Hall; 2003.
- [19] Hicklin J, et al. Jama: A Java Matrix Package (http://math.nist.gov/javanumerics/jama/), July 13, 2005.
- [20] FnattLabME including Mie scattering application (http://diogenes.iwt.uni-bremen.de/vt/laser/wap/FnattLab.jar).

1548