User notes for DDSCAT

Version 4c

Bruce T. Draine
Princeton University Observatory
Princeton, New Jersey 08544-1001
draine@astro.princeton.edu

Piotr J. Flatau
Scripps Institution of Oceanography
University of California, San Diego
La Jolla, California 92093-0221
pflatau@ucsd.edu

December 18, 1994

Preface

If you are reading this report, you have presumably succeeded in getting the source code for DDSCAT.4c following the instructions in the README file. We refer you to the list of references for discussions of the theory and accuracy of the Discrete Dipole Approximation.

What follows are instructions for:

- compiling and linking the code
- running a sample calculation,
- understanding the output from the sample calculation,
- how to modify the parameter file to do your desired calculations,
- how to specify target orientation,
- how to change the DIMENSIONs in the source code to do your desired calculations.

The instructions for compiling, linking, and running will be appropriate for a UNIX system; slight changes will be necessary for non-UNIX sites, but they are quite minor and should present no difficulty.

Acknowledgements

The routine ESELF making use of the FFT was originally written by Jeremy J. Goodman. The FFT routine FOURX is based on a FFT routine written by Norman Brenner (Brenner 1969). The FFT routine CXFFT3 is based on a FFT routine written by Clive Temperton (Temperton 1983). The routine REFICE was written by Steven B. Warren, based on Warren (1984). The routine REFWAT was written by Eric A. Smith. We are indebted to all of these authors for making their code available.

Development of DDSCAT was supported in part by National Science Foundation grants AST-8341412, AST-8612013, AST-9017082, and in part by Air Force Office of Scientific Research grant AFOSR-88-0143.

Contents

Chapter 1

DDSCAT User Guide

1.1 Compiling and linking

To compile the `DDSCAT` code on a Unix system, position yourself in the source directory `../DDA/src`. The `Makefile` supplied has compiler options appropriate for Sun Fortran on SunOS 4.1. If you have a different Fortran compiler, you will probably need to edit the `Makefile` to provide the desired compiler options. Samples are given of compiler options for HP, IBM, and SGI operating systems.

So far as we know, there are two operating-system-dependent aspects of `DDSCAT.4c` : the device number to use for "standard output", and the `TIMEIT` routine.

The variables `IDVOUT` and `IDVERR` specify device numbers for "running output" and "error messages". Normally these would both be set to the device number for "standard output" (e.g., writing to the screen if running interactively). The variables `IDVOUT` and `IDVERR` are set by `DATA` staments in the "main" program `DDSCAT`. Under Sun Fortran, `DATA IDVOUT/0/` results in unbuffered output to "standard output"; unbuffered output (if available) is nice so that if you choose to direct your output to a file (e.g., using `ddscat >& ddscat.out &`) the output file will contain up-to-date information. Other operating systems or compilers may not support this, and you may need to edit `DDSCAT.f` to change the two data statements to `DATA IDVOUT/6/` and `DATA IDVERR/6/`.

The only other operating system-dependent part of `DDSCAT.4c` is (we hope!) the single subroutine `TIMEIT`. You have been provided with several versions of `TIMEIT`:

- timeit_sun.f uses the SunOS system call `etime` ,
- timeit_convex.f uses the Convex OS system call `etime`,
- timeit_cray.f uses the system call `second`,
- timeit_hp.f uses the HP-AUX system calls `sysconf` and `times`,
- timeit_ibm6000.f uses the AIX system call `mclock`,
- timeit_sgi.f used the IRIX system call `etime`,
- timeit_vms.f uses the VMS `LIB$INIT_TIMER` and `LIB$SHOW_TIMER`,

- timeit_titan.f uses the system call `cputim`,
- timeit_null.f is a dummy routine which provides no timing information, but can be used under any operating system.

If you are on a VMS system, you will want to compile and link ALL of the source code in the files `SRCn.FOR`, for $n = 0 - 9$. `SRC9.f` includes the source code for a version of `TIMEIT` (`timeit_vms.f`) which should be VMS-compatible. Non-Unix and non-VMS sites: you must compile and link one of the `timeit` routines, possibly after modifying it to work on your system; `timeit_null.f` is the easiest choice, but it will return no timing information.

After editing the `Makefile` (while still positioned in `~/DDA`) simply type

```
make ddscat
```

This should create an executable file `~/DDA/src/ddscat`. Non-UNIX sites: the objective is to compile and link all of the Fortran code (except for (1) the program `CALLTARGET.f` , and (2) the multiple versions of `TIMEIT`) to create an executable with a name like `ddscat` or `DDSCAT.EXE` or whatever.

1.2 Moving the executable

Now reposition yourself into the directory `~/DDA`, and copy the executable from `src/ddscat` to the `~/DDA` directory by typing

```
cd ~/DDA
cp src/ddscat ddscat
```

This should copy the file `~/DDA/src/ddscat` to `~/DDA/ddscat`. Similarly, copy the sample parameter file `ddscat.par` and the file `diel.tab` to the `~/DDA` directory by typing

```
cp doc/ddscat.par ddscat.par
cp doc/diel.tab diel.tab
```

1.3 The sample parameter file "ddscat.par"

The directory `~/DDA` should now contain a sample file `ddscat.par` which provides parameters to the program `DDSCAT`. The file `ddscat.par` as provided is set up to calculate DDA scattering by a $8 \times 6 \times 4$ rectangular array of 192 dipoles, with an effective radius of 1μm, at a wavelength of 6.2832μm.

The dielectric function of the target material is provided in the file `diel.tab`, which is a sample file in which the refractive index is set to $m = 1.33 + 0.01i$ at all wavelengths; the name of this file is provided to `DDSCAT` by the parameter file `ddscat.par`. It is read by subroutine `DIELEC`.

The sample parameter file as supplied calls for Brenner's FFT routine to be employed in the complex conjugate gradient calculations.

The sample parameter file specifies (via option `LATTDR`) that the "Lattice Dispersion Relation" of Draine and Goodman (1993) be employed to determine the dipole polarizabilities; other possible choices are `DRAI88` (prescription used by Draine 1988) and `GOBR88` (prescription used by Goedecke and Obrien 1988 and Hage and Greenberg 1990). In the limit $|m|kd \ll 1$ (where k is the wave vector and d is the lattice spacing) all three options converge to the same limit; for |m|kd < 0.1 there are significant differences among them. Consult the paper by Draine and Goodman (1993) for a discussion and comparison of these three prescriptions. *Option* `LATTDR` *is recommended for general use, based upon the tests presented by Draine and Goodman (1993)* .

The sample `ddscat.par` file specifies that the calculations be done for a single wavelength ($6.2832\mu m = 2\pi$) and a single effective radius ($1.00\mu m$). Note that in `DDSCAT` the "effective radius" $a_{\rm eff}$ is the radius of a sphere of equal volume – i.e., a sphere of volume Nd^3 , where d is the lattice spacing and N is the number of occupied (i.e., non-vacuum) lattice sites in the target. Thus the effective radius $a_{\rm eff} = (3N/4\pi)^{1/3}d$.

The incident radiation is always assumed to propagate along the x axis in the "Lab Frame". The sample `ddscat.par` file specifies incident polarization state $\mathbf{e}_{01}$ to be along the y axis (and consequently polarization state $\mathbf{e}_{02}$ will automatically be taken to be along the z axis). `IORTH=2` in "ddscat.par" calls for DDA calculations to be carried out for both incident polarization states ($\mathbf{e}_{01}$ and $\mathbf{e}_{02}$).

The target is assumed to have two vectors $\mathbf{a}_1$ and $\mathbf{a}_2$ embedded in it; $\mathbf{a}_2$ is perpendicular to $\mathbf{a}_1$. In the case of the $8 \times 6 \times 4$ rectangular array of the sample calculation, the vector $\mathbf{a}_1$ is along the "long" axis of the target, and the vector $\mathbf{a}_2$ is along the "intermediate" axis. The target orientation in the Lab Frame is set by three angles: `BETA`, `THETA`, and `PHI`, defined and discussed below in section 1.7 ("Target Orientation"). Briefly, the polar angles `THETA` and `PHI` specify the direction of $\mathbf{a}_1$ in the Lab Frame. The target is assumed to be rotated around $\mathbf{a}_1$ by an angle `BETA`. The sample `ddscat.par` file specifies `BETA` and `PHI` to be zero, and calls for three values of the angle `THETA`. `DDSCAT` assumes that the `THETA` values should be uniformly spaced in `cos(THETA)`; thus, asking for three values of `THETA` between 0 and 90deg yields `THETA`= 0, 60, and 90deg. For the `THETA`= 0 calculation (`w00r00k000.sca`) the target is oriented in the Lab Frame with the long axis in the x-direction, and the intermediate axis in the y-direction. For the `THETA`= 90 calculation (`w00r00k002.sca`) the target is oriented in the Lab Frame with the long axis in the y-direction, and the intermediate axis in the x-direction.

The present version, `DDSCAT.4c` , chooses the angles `BETA`, `THETA`, and `PHI` to sample the intervals (`BETAMI`, `BETAMX`), (`THETMI`, `THETMX`), (`PHIMIN`, `PHIMAX`), where `BETAMI`, `BETAMX`, `THETMI`, `THETMX`, `PHIMIN`, `PHIMAX` are specified in `ddscat.par`. The prescription for choosing the angles is to:

- uniformly sample in `BETA`,
- uniformly sample in `PHI`,
- uniformly sample in cos(`THETA`).

This prescription is appropriate for "random" orientation of the target, within the specified limits of `BETA`, `PHI`, and `THETA`.

Note that when `DDSCAT.4c` chooses angles it handles `BETA` and `PHI` differently from `THETA` (this is a *change* from `DDSCAT.4a`). The range for `BETA` is divided into `NBETA` intervals, and the midpoint of each interval is taken. Thus, if you take BETAMI= 0, BETAMX= 90, NBETA= 2 you will get BETA=22.5 and 67.5. Similarly, if you take PHIMIN= 0, PHIMAX= 180, NPHI=2 you will beta ???? PHI= 45 and 135.

Sampling in THETA is done quite differently from sampling in BETA and PHI. First of all DDSCAT.4c samples uniformly in cos(THETA), not THETA. Secondly, the sampling depends on whether even or odd NTHETA is specified. If NTHETA is odd, then the values of THETA selected include the extreme values THETMI and THETMX; thus, THETMI= 0, THETMX= 90, NTHETA= 3 will give you THETA= $0, 60, 90$. If, on the other hand, you specify an even value of NTHETA, then the range of cos(THETA) will be divided into NTHETA intervals, and the midpoint of each interval will be taken; thus, THETMI=0,THETMX=90,NTHETA=2 will give you THETA=41.41 and 75.52 (cos(THETA) = 0.25 and 0.75). The reason for this is that if odd NTHETA is specified, than the "integration" over cos(THETA) is performed using Simpson's rule for greater accuracy. If even NTHETA is specified, then the integration over cos(THETA) is performed by simply taking the average of the results for the different THETA values.

If averaging over random orientation is desired, it is recommended that the user specify an *odd* value of NTHETA so that Simpson's rule will be employed.

This prescription represents "random" orientation within the specified limits of BETA, THETA, and PHI. True "random" orientation would in principle be represented by BETA, THETA, PHI in the intervals $(0, 360)$, $(0, 180)$, $(0, 360)$ (degrees, of course); for symmetric targets, it may not be necessary to fully sample this range (e.g., (THETA,PHI) and (180-THETA, PHI+180) are physically identical cases for a target with reflection symmetry through a plane normal to the vector $\mathbf{a}_1$). It is important to keep in mind that solutions for different PHI are obtained relatively inexpensively from the solution (two polarizations) for the first PHI value; thus one can generally afford to specify finer sampling in PHI than for BETA or THETA.

Chapter 2 provides a detailed description of the file ddscat.par.

1.4 Running DDSCAT using the sample ddscat.par file

To execute the program on a UNIX system, simply type

```
ddscat >& ddscat.out &
```

which will redirect the "standard output" to the file ddscat.out, and run the calculation in the background. The sample calculation ($8 \times 6 \times 4 = 192$ dipole target, 3 orientations, two incident polarizations, with scattering calculated for 14 distinct scattering directions), requires 41 cpu seconds on a Sun 4/50 ("Sparcstation IPX").

1.5 Output files

If you run DDSCAT using the command ddscat >& ddscat.out & you will have various types of files when the computation is complete:

- a file ddscat.out,
- a file mtable

- a file `qtable`
- a file `qtable2`
- files `wAArBBori.avg` (one, `w00r00ori.avg`, for the sample calculation);
- files `wAArBBkCCC.sca` (3 for the sample calculation: `w00r00k000.sca`, `w00r00k001.sca`, `w00r00k002.sca`). Only if `ddscat.par` specified `IWRKSC`=1,

The file `ddscat.out` will contain any error messages generated as well as a running report on the progress of the calculation, including creation of the target dipole array. During the `CCG` calculations, $Q_{\rm ext}$, $Q_{\rm abs}$, and $Q_{\rm pha}$ are printed after each iteration; you will be able to judge the degree to which convergence has been achieved. Unless `TIMEIT` has been disabled, there will also be timing information.

The file `mtable` contains a summary of the dielectric constant used in the calculations.

The file `qtable` contains a summary of the orientationally-averaged values of $Q_{\rm ext}$, $Q_{\rm abs}$, $Q_{\rm sca}$, $g = < \cos(\theta_{\rm s}) >$, and $Q_{\rm bk}$. Here $Q_{\rm ext}$, $Q_{\rm abs}$, and $Q_{\rm sca}$ are the extinction, absorption, and scattering cross sections divided by $\pi a_{\rm eff}^2$. $Q_{\rm bk}$ is the differential cross section for backscattering divided by $\pi a_{\rm eff}^2$.

The file `qtable2` contains a summary of the orientationally-averaged values of $Q_{\rm pha}$, $Q_{\rm pol}$, and $Q_{\rm cpol}$. Here $Q_{\rm pha}$ is the "phase shift" cross section divided by $\pi a_{\rm eff}^2$ (see definition in Draine 1988). $Q_{\rm pol}$ is the "polarization" efficiency, equal to the difference between $Q_{\rm ext}$ for the two orthogonal polarization states. $Q_{\rm cpol}$ is defined to be the product $Q_{\rm pol}Q_{\rm pha}$, since an optically-thin medium with a small twist in the alignment direction will produce circular polarization in initially unpolarized light in proportion to $Q_{\rm cpol}$.

For each wavelength and size `DDSCAT.4c` produces a file with a name of the form `wAArBBori.avg`, where `AA` (`=00,01,02`...) designates the wavelength and `BB` (`=00,01,02`...) designates the "radius"; this file contains `Q` values and scattering information averaged over however many target orientations have been specified. An (annotated) sample file (the file `w00r00ori.avg` produced by the sample calculation) is provided below in Appendix A.

In addition, if `ddscat.par` has specified `IWRKSC`=1 (as for the sample calculation), `DDSCAT.4c` will generate files with names of the form `wAArBBkCCC.avg`, where `AA` and `BB` are as before, and `CCC` =(`000,001,002`...) designates the target orientation; these files contain `Q` values and scattering information for each of the target orientations. The structure of each of these files is very similar to that of the `wAArBBori.avg` files. Because these files may not be of particular interest, and take up disk space, you may choose to set `IWRKSC`=0 in future work. However, it is suggested that you run the sample calculation with `IWRKSC`=1. The sample `ddscat.par` file specifies `IWRKSC`=1 and calls for use of 1 wavelength, 1 target size, and averaging over 3 target orientations. Running `DDSCAT.4c` with the sample sample `ddscat.par` file will therefore generate files `w00r00k000.sca`, `w00r00k001.sca`, and `w00r00k003.sca`. To understand the information contained in one of these files, please consult Appendix B, which contains an (annotated) example of a `.sca` file: the file `w00r00k000.sca` produced in the sample calculation.

1.6 Memory requirements

The executable has compiled into it the dimensions for a number of arrays; these array dimensions limit the size of the dipole array which the code can handle. The source code supplied to you (which can be used to run the sample calculation with 192 dipoles) is restricted to problems with targets which are maximum extent of 8 lattice spacings along the x, y, and z directions (MXNX= 8,MXNY= 8,MXNZ= 8; i.e, target must fit within an $8 \times 8 \times 8 = 512$ cube) and involve at most 3 different dielectric functions (MXCOMP=3). With this dimensioning, the executable requires 792 Kbytes of memory to run on a Sparcstation; memory requirements on other hardware and with other compilers should be similar. To run larger problems, you will need to edit the DDSCAT.f to change PARAMETER values and recompile.

All of the dimensioning takes place in the main program DDSCAT – this should be the only routine which it is necessary to recompile. All of the dimensional parameters are set in PARAMETER statements appearing before the array declarations. You need simply edit the parameter statements. Remember, of course, that the amount of memory allocated by the code when it runs will depend upon these dimensioning parameters, so do not set them to ridiculously large values! The parameters MXNX, MXNY, MXNZ specify the maximum extent of the target in the x-, y-, or z-directions. The parameter MXMEM=0 or 1 depending on whether workspace required by Temperton's FFT algorithm is to be reserved. The parameter MXCOMP specifies the maximum number of different dielectric functions which the code can handle at one time. The comment statements in the code supply all the information you should need to change these parameters.

The memory requirement is approximately (557 + 0.527MXNX*MXNY*MXNZ) Kbytes when compiled with MXMEM=0, or (557+0.590MXNX*MXNY*MXNZ) kbytes when compiled with MXMEM=1. These values are for a Sparcstation using the Sun FORTRAN compiler.

1.7 Target orientation

Recall that we define a "Lab Frame" (LF) in which the incident radiation propagates in the $+x$ direction. In ddscat.par one specifies the first polarization state $\mathbf{e}_{01}$ (which obviously must lie in the $y - z$ plane in the LF); DDSCAT automatically constructs a second polarization state $\mathbf{e}_{02} = \mathbf{e}_x \times \mathbf{e}_{01}$ orthogonal to $\mathbf{e}_{01}$ (here $\mathbf{e}_x$ is the unit vector in the +x direction of the LF. For purposes of discussion we will always let unit vectors $\mathbf{e}_x$, $\mathbf{e}_y$, $\mathbf{e}_z$ be the three coordinate axes of the LF. Users will usually find it convenient to let polarization vectors $\mathbf{e}_{01} = \mathbf{e}_y$, $\mathbf{e}_{02} = \mathbf{e}_z$ (although this is not mandatory). Recall that definition of a target involves specifying two unit vectors, $\mathbf{a}_1$ and $\mathbf{a}_2$, which are "frozen" into the target. We require $\mathbf{a}_2$ to be orthogonal to $\mathbf{a}_1$. Therefore we may define a "Target Frame" (TF) defined by the three unit vectors $\mathbf{a}_1$, $\mathbf{a}_2$, and $\mathbf{a}_3 = \mathbf{a}_1 \times \mathbf{a}_2$. For example, when DDSCAT creates a rectangular solid, it fixes $\mathbf{a}_1$ to be along the longest dimension of the solid, and $\mathbf{a}_2$ to be along the next-longest dimension; thus if we created a rectangular target of size $8 \times 6 \times 4$ (as in the sample calculation), in the TF the solid would be located with length 8 in the $\mathbf{a}_1$ direction, length 6 in the $\mathbf{a}_2$ direction, and length 4 in the $\mathbf{a}_3$ direction. Orientation of the target relative to the incident radiation can in principle be determined two ways:

1. specifying the direction of $\mathbf{a}_1$ and $A[2]$ in the LF, or

2. specifying the directions of $\mathbf{e}_x$ (incidence direction) and $\mathbf{e}_y$ in the TF.

`DDSCAT` uses method (1): the angles `THETA`, `PHI`, and `BETA` are specified in the file `ddscat.par`. The target is oriented such that the polar angles `THETA` and `PHI` specify the direction of $\mathbf{a}_1$ relative to the polar axis $\mathbf{e}_x$, where the $\mathbf{e}_x - \mathbf{e}_y$ plane has `PHI`=0. Once the direction of $\mathbf{a}_1$ is specified, the angle `BETA` than specifies how the target is to rotated around the axis $\mathbf{a}_1$ to fully specify its orientation. A more extended and precise explanation follows:

1.7.1 Orientation of the Target in the Lab Frame

`DDSCAT` uses three angles, `THETA`, `PHI`, and `BETA`, to specify the directions of unit vectors $\mathbf{a}_1$ and $\mathbf{a}_2$ in the LF. The angle `THETA` is the angle between $\mathbf{a}_1$ and $\mathbf{e}_x$. When `PHI`=0, $\mathbf{a}_1$ will lie in the $\mathbf{e}_x - \mathbf{e}_y$ plane. When `PHI` is nonzero, it will refer to the rotation of $\mathbf{a}_1$ around the x axis: e.g., `PHI`=90 puts $\mathbf{a}_1$ in the $\mathbf{e}_x - \mathbf{e}_z$ plane. When `BETA`=0, $\mathbf{a}_2$ will lie in the $\mathbf{e}_x - \mathbf{a}_1$ plane, in such a way that when `THETA`=0 and `PHI`=0, $\mathbf{a}_2$ is in the $\mathbf{e}_y$ direction: e.g, `THETA`=90,`PHI`=0, `BETA`=0 has $\mathbf{a}_1 = \mathbf{e}_y$ and $\mathbf{a}_2 = -\mathbf{e}_x$. Nonzero `BETA` introduces an additional rotation of $\mathbf{a}_2$ around $\mathbf{a}_1$: e.g., `THETA`=90,`PHI`=0,`BETA`=90 has $\mathbf{a}_1 = \mathbf{e}_y$ and $\mathbf{a}_2 = \mathbf{e}_z$. Mathematically:

$$\begin{aligned}\mathbf{a}_1 = {} & \mathbf{e}_x \cos(\theta) \\ & + \mathbf{e}_y \sin(\theta)\cos(\varphi) \\ & + \mathbf{e}_z \sin(\theta)\sin(\varphi)\end{aligned} \tag{1.1}$$

$$\begin{aligned}\mathbf{a}_2 = {} & -\mathbf{e}_x \sin(\theta)\cos(\beta) \\ & + \mathbf{e}_y[\cos(\theta)\cos(\beta)\cos(\varphi) - \sin(\beta)\sin(\varphi)] \\ & + \mathbf{e}_z[\cos(\theta)\cos(\beta)\sin(\varphi) + \sin(\beta)\cos(\varphi)]\end{aligned} \tag{1.2}$$

$$\begin{aligned}\mathbf{a}_3 = {} & \mathbf{e}_x \sin(\theta)\sin(\beta) \\ & - \mathbf{e}_y[\cos(\theta)\sin(\beta)\cos(\varphi) + \cos(\beta)\sin(\varphi)] \\ & - \mathbf{e}_z[\cos(\theta)\sin(\beta)\sin(\varphi) - \cos(\beta)\cos(\varphi)]\end{aligned} \tag{1.3}$$

or, equivalently:

$$\begin{aligned}\mathbf{e}_x = {} & \mathbf{a}_1 \cos(\theta) \\ & - \mathbf{a}_2 \sin(\theta)\cos(\beta) \\ & + \mathbf{a}_3 \sin(\theta)\sin(\beta)\end{aligned} \tag{1.4}$$

$$\begin{aligned}\mathbf{e}_y = {} & \mathbf{a}_1 \sin(\theta)\cos(\varphi) \\ & + \mathbf{a}_2[\cos(\theta)\cos(\beta)\cos(\varphi) - \sin(\beta)\sin(\varphi)] \\ & - \mathbf{a}_3[\cos(\theta)\sin(\beta)\cos(\varphi) + \cos(\beta)\sin(\varphi)]\end{aligned} \tag{1.5}$$

$$\begin{aligned}\mathbf{e}_z = {} & \mathbf{a}_1 \sin(\theta)\sin(\varphi) \\ & + \mathbf{a}_2[\cos(\theta)\cos(\beta)\sin(\varphi) + \sin(\beta)\cos(\varphi)] \\ & - \mathbf{a}_3[\cos(\theta)\sin(\beta)\sin(\varphi) - \cos(\beta)\cos(\varphi)]\end{aligned} \tag{1.6}$$

1.7.2 Orientation of the Incident Beam in the Target Frame

Under some circumstances, one may wish to specify the target orientation such that $\mathbf{e}_x$ (the direction of propagation of the radiation) and $\mathbf{e}_y$ (usually the first polarization direction) and $\mathbf{e}_z$ $(= \mathbf{e}_x \times \mathbf{e}_y)$ refer to certain directions in the TF. Given the definitions of the LF and TF above, this is simply an exercise in coordinate transformation. For example, one might wish to have the incident radiation propagating along the (1,1,1) direction in the TF (example 14 below). Here we provide some selected examples:

$\mathbf{e}_x = \mathbf{a}_1$	$\mathbf{e}_y = \mathbf{a}_2$	$\mathbf{e}_z = \mathbf{a}_3$	$: \theta = 0$	$\varphi - \beta = 0$	
$\mathbf{e}_x = \mathbf{a}_1$	$\mathbf{e}_y = \mathbf{a}_3$	$\mathbf{e}_z = -\mathbf{a}_2$	$: \theta = 0$	$\varphi - \beta = 90$	
$\mathbf{e}_x = \mathbf{a}_2$	$\mathbf{e}_y = \mathbf{a}_1$	$\mathbf{e}_z = -\mathbf{a}_3$	$: \theta = 90$	$\beta = 180$	$\varphi = 0$
$\mathbf{e}_x = \mathbf{a}_2$	$\mathbf{e}_y = \mathbf{a}_3$	$\mathbf{e}_z = \mathbf{a}_1$	$: \theta = 90$	$\beta = 180$	$\varphi = 90$
$\mathbf{e}_x = \mathbf{a}_3$	$\mathbf{e}_y = \mathbf{a}_1$	$\mathbf{e}_z = \mathbf{a}_2$	$: \theta = 90$	$\beta = -90$	$\varphi = 0$
$\mathbf{e}_x = \mathbf{a}_3$	$\mathbf{e}_y = \mathbf{a}_2$	$\mathbf{e}_z = -\mathbf{a}_1$	$: \theta = 90$	$\beta = -90$	$\varphi = -90$
$\mathbf{e}_x = -\mathbf{a}_1$	$\mathbf{e}_y = \mathbf{a}_2$	$\mathbf{e}_z = -\mathbf{a}_3$	$: \theta = 180$	$\beta + \varphi = 180$	
$\mathbf{e}_x = -\mathbf{a}_1$	$\mathbf{e}_y = \mathbf{a}_3$	$\mathbf{e}_z = \mathbf{a}_2$	$: \theta = 180$	$\beta + \varphi = 90$	
$\mathbf{e}_x = -\mathbf{a}_2$	$\mathbf{e}_y = \mathbf{a}_1$	$\mathbf{e}_z = \mathbf{a}_3$	$: \theta = 90$	$\beta = 0$	$\varphi = 0$
$\mathbf{e}_x = -\mathbf{a}_2$	$\mathbf{e}_y = \mathbf{a}_3$	$\mathbf{e}_z = -\mathbf{a}_1$	$: \theta = 90$	$\beta = 0$	$\varphi = -90$
$\mathbf{e}_x = -\mathbf{a}_3$	$\mathbf{e}_y = \mathbf{a}_1$	$\mathbf{e}_z = -\mathbf{a}_2$	$: \theta = 90$	$\beta = -90$	$\varphi = 0$
$\mathbf{e}_x = -\mathbf{a}_3$	$\mathbf{e}_y = \mathbf{a}_2$	$\mathbf{e}_z = \mathbf{a}_1$	$: \theta = 90$	$\beta = -90$	$\varphi = 90$

1.8 Orientational averaging

`DDSCAT` has been constructed to facilitate the computation of orientational averages. How to go about this depends on the distribution of orientations which is applicable.

1.8.1 Randomly-Oriented Targets

For randomly-oriented targets, all you need to do is edit the `ddscat.par` file so that `DDSCAT` knows what ranges of the angles beta, theta, and phi are of interest. For a randomly-oriented target with no symmetry, you would need to let beta run from 0 to 360, theta from 0 to 180, and `PHI` from 0 to 360. For targets with symmetry, on the other hand, the ranges of beta, theta, and phi may be reduced. First of all, remember that averaging over phi is "cheap", so when in doubt average over 0 to 360; most of the computational "cost" is associated with the number of different values of (beta,theta) which are used. Consider a cube, for example, with axis $\mathbf{a}_1$ normal to one of the cube faces; for this cube beta need run only from 0 to 90, since the cube has fourfold symmetry for rotations around the axis $\mathbf{a}_1$. Furthermore, the angle theta need run only from 0 to 90, since the orientation (beta,theta,phi) is indistinguishable from (beta,180-theta,360-phi). For targets with symmetry, the user is encouraged to test the significance of `BETA`, `THETA`, `PHI` on targets with small numbers of dipoles (say, of the order of 100 or so) but having the desired symmetry. It is important to remember that `DDSCAT.4c` handles even and odd values of `NTHETA` differently – see section 1.3 above! For averaging over random orientations odd values of `NTHETA` are to be preferred, as this will allow use of Simpson's rule in evaluating the "integral" over cos(`THETA`).

1.8.2 Nonrandomly-Oriented Targets

Some special cases (where the target orientation distribution is uniform for rotations around the x-axis = direction of propagation of the incident radiation), one may be able to use DDSCAT with appropriate choices of input parameters. More generally, however, you will need to modify subroutine ORIENT to generate a list of NBETA values of beta, NTHETA values of theta, and NPHI values of phi, plus two weighting arrays WGTA(1-NTHETA,1-NPHI) and WGTB(1-NBETA). Here WGTA gives the weights which should be attached to each (theta,phi) orientation, and WGTB gives the weight to be attached to each beta orientation. Thus each orientation of the target is to be weighted by the factor WGTA*WGTB. For the case of random orientations, DDSCAT chooses theta values which are uniformly spaced in cos(theta), and beta and phi values which are uniformly spaced, and therefore uses uniform weights

```
WGTB=1./NBETA
```

When NTHETA is even, DDSCAT sets

```
WGTA=1./(NTHETA*NPHI)
```

but when NTHETA is odd, DDSCAT uses Simpson's rule when integrating over THETA and

```
WGTA= (1/3 or 4/3 or 2/3)/(NTHETA*NPHI)
```

Note that the program structure of DDSCAT may not be ideally suited for certain highly oriented cases. If, for example, the orientation is such that for a given phi value only one theta value is possible (this situation might describe ice needles oriented with the long axis perpendicular to the vertical in the Earth's atmosphere, illuminated by the Sun at other than the zenith) then it is foolish to consider all the combinations of theta and phi which the present version of DDSCAT is set up to do. We hope to improve this in a future version of DDSCAT.

1.9 Target generation

DDSCAT contains routines to generate arrays representing targets of various geometries, including spheres, ellipsoids, rectangular solids, cylinders, hexagonal prisms, tetrahedra, two touching ellipsoids, and three touching ellipsoids. The target type is specified by variable CSHAPE (see p. 16) on line 5 of ddscat.par, and parameters PAR1, PAR2, PAR3 on line 6. Types currently, supported include:

RCTNGL to create a rectangular target of length SHPAR1, SHPAR2, SHPAR3; Unit vector $\mathbf{a}_1$ will be along longest dimension; unit vector $\mathbf{a}_2$ will be along intermediate dimension.

RCTNGL – Rectangular solid with x, y, z length/d = PAR1, PAR2, PAR3. Must set NCOMP=1.

ELLIPS to create an ellipsoidal target of extent SHPAR1, SHPAR2, SHPAR3 along the x, y, z directions (note that a sphere is special case of ELLIPS where SHPAR1, SHPAR2, SHPAR3 are identical); unit vector $\mathbf{a}_1$ will be along longest axis; $\mathbf{a}_2$ will be along intermediate axis.

ELLIPS – Ellipsoid $(x/\mathrm{PAR1}^2 + (y/\mathrm{PAR2})^2 + (z/\mathrm{PAR3})^2 = d^2/4$ (sphere is special case where PAR1=PAR2= PAR3=diameter. Must set NCOMP=1.

`CYLNDR` to create a cylinder with length `SHPAR1`, diameter `SHPAR2`, and with cylinder axis in x, y, z directions for `SHPAR3`=1, 2, 3.

`CYLNDR` – Cylinder with length/d=`PAR1`, diameter/d=`PAR2`, and `PAR3`=1,2,3 for symmetry axis along x, y, z direction. Must set `NCOMP=1`.

`HEXGON` to create hexagonal prism with length `SHPAR1`, thickness SHPAR2 ("thickness" = distance between opposite vertices of hexagon = 2*length of one side of hexagon)

`SHPAR3`	$\mathbf{a}_1$ along	$\mathbf{a}_2$ along
1	x	z (rect. face normal to y)
2	x	y (rect. face normal to z)
3	y	x (rect. face normal to z)
4	y	z (rect. face normal to x)
5	z	y (rect. face normal to x)
6	z	x (rect. face normal to y)

where $\mathbf{a}_1$=6-fold symmetry axis, $\mathbf{a}_2$=axis emerging from vertex. Thus, for example, if SHPAR3=1 the target will have the prism axis along the x direction, the two hexagonal faces normal to the x axis, and two of the six rectangular faces normal to the y axis.

`HEXGON` – Hexagonal prism with length/d = `PAR1`, hex side = `PAR2` `PAR3=1` for prism axis along x axis, one face normal to y axis (`PAR3`=2, 3, 4, 5, 6 for other orientations: see `tartet.f`). Must set `NCOMP=1`.

`TETRAH` to create a tetrahedron with edges of length `SHPAR1`, with one face parallel to $y-z$ plane. Tetrahedron volume will be approximately `SHPAR1`**3/(6.*sqrt(2))

`TETRAH` – Tetrahedron with `PAR1`=length/d of one edge, one face parallel to $y-z$ plane, opposite "vertex" in $+x$ direction, and one edge parallel to z axis. Must set `NCOMP=1`.

`TWOELL` – Two identical touching ellipsoids with same orientation

`PAR1`=x-length/d of one ellipsoid

`PAR2`=y-length/d of one ellipsoid

`PAR3`=z-length/d of one ellipsoid

Line connecting ellipsoid centers is along x-axis. Must set `NCOMP=2` and provide dielectric functions for ellipsoids. Ellipsoids are in order of increasing x.

`TWOAEL` – Two touching ellipsoids with same size and orientation, but different and anisotropic dielectric function. `PAR1`, `PAR2`, `PAR3` have same meanings as for `TWOELL`. Must set `NCOMP=6` and provide x, y, z dielectric functions for first ellipsoid, and x, y, z dielectric functions for second ellipsoid (ellipsoids are in order of increasing x).

`UNICYL` to create cylinder of unixial material. `SHPAR1`=cylinder length; `SHPAR2`=cylinder diameter; cylinder axis in x, y, z directions for `SHPAR3`=1, 2, 3. Material 1 for E field along cyl. axis, material 2 for E field perp. to cyl. axis (i.e. "crystal axis" parallel to cylinder axis). Note that this option requires that NCOMP=2 dielectric functions be specified.

UNICYL – uniaxial cylinder. PAR1, PAR2, PAR3 have same meaning as for CYLNDR, but now diel. func. 1 is for E field parallel to cylinder axis, diel.func. 2 for E field perp. to axis. Must set NCOMP=2 and provide parallel, perpendicular dielectric functions.

ANIELL to create ellipsoid of anisotropic material. SHPAR1, SHPAR2, SHPAR3 = ellipsoid extent in x, y, z directions (just as for option ELLIPS). Target axis $\mathbf{a}_1$ will be along longest axis; target axis $\mathbf{a}_2$ along intermediate axis (just as for option ELLIPS). Material 1 for E field parallel to axis $\mathbf{a}_1$, material 2 for E field parallel to $\mathbf{a}_2$, material 3 for E field parallel to $\mathbf{a}_3 = \mathbf{a}_1 \times \mathbf{a}_2$. Note that this option requires that NCOMP=3 dielectric functions be specified.

ANIELL – anisotropic ellipsoid. PAR1, PAR2, PAR3 have same meaning as for ELLIPS, but with dielectric functions 1,2,3 for xx,yy,zz elements of (diagonalized) dielectric tensor. Must set NCOMP=3 and provide x, y, z dielectric functions.

THRELL – three touching ellipsoids of equal size and orientation PAR1, PAR2, PAR3 have same meaning as for TWOELL. Line connecting ellipsoid centers is parallel to x axis. Must set NCOMP=3 and provide (isotropic) dielectric functions for first, second, and third ellipsoid.

THRAEL – three touching ellipsoids with same size and orientation but different and anisotropic dielectric properties. PAR1, PAR2, PAR3 have same meanings as for THRELL. Must set NCOMP=9 and provide x, y, z dielectric function for first ellipsoid, x, y, z dielectric function for second ellipsoid, and x, y, z dielectric function for third ellipsoid (ellipsoids are in order of increasing x).

FRMFIL to read target description from a file shape.dat (look at subroutine REASHP to see how this file should be formatted). FRMFIL is used for targets which cannot be generated using MKRECT, ELLIPS, CYLNDR, HEXGON, GRPHIT, TETRAH, UNICYL, or ANIELL.

The user should be able to easily modify these routines, or write new routines, to generate targets with other geometries. The user should first examine the routine target.f and modify it to call any new target generation routines desired. Alternatively, targets may be generated separately, and the target description (locations of dipoles and "composition" corresponding to x, y, z dielectric properties at each dipole site) read in from a file by invoking the option FRMFIL in ddscat.par.

It is often desirable to be able to run the target generation routines without firing up DDSCAT. We have therefore provided a program calltarget which allows the user to generate targets interactively; to create this executable just type make calltarget. (Non-Unix sites: The source code for CALLTARGET is in the file MISC.FOR. You must compile and link this to ERRMSG, GETSET, REASHP, TAR2EL, TAR3EL, TARCYL, TARELL, TARGET, TARHEX, TARREC, and WRIMSG. The source code for ERRMSG is in SRC2.FOR, GETSET is in SRC5.FOR, and the rest is in SRC9.FOR.)

The program calltarget is to be run interactively; the prompts are self-explanatory. You may need to edit the code to change the device number IDVOUT as for DDSCAT (see section 1.1).

After running, `calltarget` will leave behind an ASCII file `target.out` which is a list of the occupied lattice sites in the last target generated.

1.10 Scattering directions

`DDSCAT` calculates scattering in selected directions, and elements of the scattering matrix are reported in the output files `wAArBBkCCC.sca`. The scattering direction is specified through angles θ_s and φ_s (not to be confused with the angles θ and φ which specify the orientation of the target relative to the incident radiation!).

The scattering angle θ_s is simply the angle between the incident beam (along direction $\mathbf{e}_x$) and the scattered beam ($\theta_s = 0$ for forward scattering, $\theta_s = 180$ for backscattering). The scattering angle φ_s specifies the orientation of the scattering plane. When $\varphi_s = 0$ the scattering plane is assumed to contain $\mathbf{e}_x$ and $\mathbf{e}_{01}$ (unit vector specifying incident polarization state 1). When $\varphi_s = 90$ the scattering plane is assumed to contain $\mathbf{e}_x$ and $\mathbf{e}_{02} = \mathbf{e}_x \times \mathbf{e}_{01}$. Note that if the user chooses to let $\mathbf{e}_{01} = \mathbf{e}_y$ (as will often be the case), $\varphi_s = 0$ will specify the $x - y$ plane, and $\varphi_s = 90$ will specify the $x - z$ plane. However, this will not be the case if $\mathbf{e}_{01}$ is specified to be other than $\mathbf{e}_y$.

1.11 Miscellanea

Additional source code, refractive index files, etc., contributed by users will be located in the directory `misc`. These routines and files should be considered to be NOT SUPPORTED by Draine and Flatau – caveat receptor! These routines and files should be accompanied by enough information (e.g., comments in source code) to explain their use.

At the moment (December 18, 1994) directory misc is empty.

1.12 Finale

The structure of the `ddscat.par` file is intended to be simple and suggestive such that, after reading the above notes once, the user may not have to refer to them again.

The file `rel_notes` in `DDA/doc` lists known bugs in `DDSCAT.4c` . An up-to-date version will be maintained in the directory where the publicly available source code is located (see the `README file`). Users are encouraged to provide B. T. Draine or P. J. Flatau with their email address; bug reports, and any new releases of `DDSCAT`, will be made known to those who do!

Concrete suggestions for improving `DDSCAT` (and these instructions) would be welcomed. The authors would also greatly appreciate reprints of any papers which make use of `DDSCAT`.

Chapter 2

Understanding and modifying “ddscat.par”

2.1 Listing of ddscat.par

```
' =================== Parameter file ===================='
'**** PRELIMINARIES ****'
'BRENNR'= CMETHD*6 (BRENNR, TMPRTN, CONVEX)
'LATTDR'= CALPHA*6 (LATTDR, LDRISO, GOBR88, DRAI88)
'RCTNGL'= CSHAPE*6 (FRMFIL,ELLIPS,CYLNDR,RCTNGL,HEXGON,TETRAH,UNICYL,ANIELL,...)
8 6 4 = shape parameters PAR1, PAR2, PAR3
1         = NCOMP = number of dielectric materials
'TABLES'= CDIEL*6 (H2OICE,H2OLIQ,TABLES; if TABLES, then filenames follow...)
'diel.tab'
'**** CONJUGATE GRADIENT DEFINITIONS ****'
0        = INIT (TO BEGIN WITH |X0> = 0)
1.00e-5 = ERR = MAX ALLOWED (NORM OF |G>=AC|E>-ACA|X>)/(NORM OF AC|E>)
'**** ANGLES FOR CALCULATION OF CSCA, G'
33       = ICTHM (number of theta values for evaluation of Csca and g)
12       = IPHM (number of phi values for evaluation of Csca and g)
'**** Wavelengths (micron) ****'
6.283185 6.283185 1 'LIN' = wavelengths (first,last,how many,how=LIN,INV,LOG)
'**** Effective Radii (micron) **** '
1. 1. 1  'LIN' = eff. radii (first, last, how many, how=LIN,INV,LOG)
'**** Define Incident Polarizations ****'
(0,0) (1.,0.) (0.,0.) = Polarization state e01 (k along x axis)
2 = IORTH  (=1 to do only pol. state e01; =2 to also do orth. pol. state)
1 = IWRKSC (=0 to suppress, =1 to write ".sca" file for each target orient.
'**** Prescribe Target Rotations ****'
 0.   0.  1  = BETAMI, BETAMX, NBETA (beta=rotation around a1)
 0.  90.  3  = THETMI, THETMX, NTHETA (theta=angle between a1 and k)
 0.   0.  1  = PHIMIN, PHIMAX, NPHI (phi=rotation angle of a1 around k)
```

```
'**** Specify Scattered Directions ****'
0.  0. 180. 30 = phi_s, theta_smin, theta_smax, dtheta_s (in deg.) for plane A
90. 0. 180. 30 = phi_s, ... for plane B
```

In order to use DDSCAT to perform the specific calculations of interest to you, it will be necessary to modify to `ddscat.par` file. Here are line-by-line instructions on how to do this. All numerical data in DDSCAT is read with free-format READ statements. Therefore you do not need to worry about the precise format in which integer or floating point numbers are entered on a line. The crucial thing is that lines in `ddscat.par` containing numerical data have the correct number of data entries, with informational comments on the line appearing *after* all numerical data.

2.2 Description of variables

CMETHD specifies the FFT routine.

- BRENNR specifies the coding due to Brenner, which works well for any NX, NY, NZ, and which minimizes memory requirements.
- TMPRTN specifies the routine due to Temperton, which requires that NX, NY, NZ each be of form $2^a 3^b 5^c$, with a, b, c integers; if the target dimensions NX, NY, NZ do not satisfy this condition, the target is artificially expanded by adding vacant sites. On vector machines, Temperton's method may be considerably faster than Brenner's. One disadvantage is that memory requirements are somewhat greater.
- CONVEX specifies the native FFT routine in the Convex vector library.

CALPHA specifies the prescription used for determining dipole polarizabilities for a given refractive index m and value of kd (where k is the propagation vector and d the lattice spacing).

- LATTDR specifies the "Lattice Dispersion Relation" result of Draine and Goodman (1993)
- LDRISO specifies the "Isotropized Lattice Dispersion Relation" of Draine and Goodman (1993)
- GOBR88 specifies the approximation used by Goedecke and Obrien (1988) and Hage and Greenberg (1990)
- DRAI88 specifies the approximation used by Draine (1988)

LATTDR *is recommended* for general use; the other options are included mainly for comparison purposes.

CSHAPE is used to specify the target shape.

The six character string specifies the shape, and the three variables SHPAR1, SHPAR2, SHPAR3 specify the size and orientation. See section 1.9 for detailed description of CSHAPE.

NCOMP specifies the number NCOMP of dielectric functions needed (=1 for homogeneous isotropic target). For anisotropic materials you will need to use NCOMP=2 or 3 dielectric functions. Note that the present version of DDSCAT assumes that for the unrotated target, the dielectric tensor has principal axes in the x, y, and z directions (i.e., the dielectric tensor is diagonalized). The only anisotropic targets which DDSCAT is set up to handle automatically are the UNICYL and ANIELL cases (see above). If you are interested in a different anisotropic target geometry, you will need to either modify DDSCAT (have a look at the source code for subroutine TARGET and, say, subroutine TARCYL) or else prepare separately a file named shape.dat describing the target dipole array in a format suitable for reading using the FRMFIL option.

CDIEL If NCOMP=1 *and* line 8 specified either H2OICE or H2OLIQ, then no other information on composition is given. However, this is not the typical use. Normally, line 8 specified TABLES (as in this example). In this case, the next NCOMP lines list the names of files, one for each composition.

Lines 9...(8+NCOMP) specify the names of files in which the dielectric functions are tabulated. Note that if you specify NCOMP ¿ 1 you *must* enter the dielectric functions via tables. In this example, the file name is just diel.tab

INIT specifies the value of INIT. Set

- INIT=0 to begin with |X0>=0 for CCG method for (this works well and is recommended);
- INIT=1 to begin with |X0> for CCG method from 4th order expansion in dipole polarizability.

ERR is the value of ERR=convergence criterion in CCG method. ERR=1.e-5 is a conservative value (high accuracy); a larger value of ERR will allow "convergence" in fewer iterations. WARNING: DDSCAT as provided has most calculations carried out in "single precision". On most computers this means 32 bits for real variables (64 bits for complex variables) and an accuracy of order 10^{-7}. *Avoid using valuse of* ERR *which are comparable to the machine accuracy.* Do not set ERR to less than 1.e-6 unless you are on a machine with more than 32 bit words (e.g., a Cray).

ICTHM, IPHM set the values of ICTHM and IPHM, which control the number of scattering directions used in evaluation of scattering properties $Q_{\rm sca}$ and $g =< \cos(\theta) >$. ICTHM must be an odd number (DDSCAT uses Simpson's rule to integrate over θ). Note that scattering directions must be spaced closely enough to sample angular structure in the scattering phase function. ICTHM, IPHM=33,12 seems to work well for $x = ka_{\rm eff} < 5$, , but for large values of the scattering parameter $x = ka_{\rm eff}$ it may be desirable to verify that using larger values of ICTHM and IPHM does not change the computed values of Qsca and g. Note that the scattered intensity must be evaluated ICTHM*IPHM times; and each evaluation involves roughly $10N$ complex operations (where N is the number of dipoles). Therefore one should not use really large values of ICTHM*IPHM unless they are actually needed.

Wavelengths first, last, how many, how (LIN, INV, LOG) specifies the first wavelength, the last wavelength, how many wavelengths, and the rule for choosing wavelengths between first and last:

- LIN for wavelengths uniformly spaced in wavelength
- INV for wavelengths uniformly spaced in frequency
- LOG for wavelengths uniformly spaced in log(wavelength)

In the event that only 1 wavelength is desired, the first and last wavelength should be the same (as in the example above). In the event that either 1 or 2 wavelengths is desired, it does not matter which rule (LIN, INV, LOG) is specified.

Effective radii first, last, how many, how(LIN, INV, LOG) specifies the first and last effective radius, the number of effective radii, and the rule for choosing effective radii between first and last:

- LIN for eff. radii uniformly spaced
- INV for eff. radii uniformly spaced in 1/radius
- LOG for eff. radii uniformly spaced in log(radius)

In the event that only 1 size is desired, the first and last eff. radius should be the same. In the event that either 1 or 2 sizes are used, it does not matter which rule (LIN, INV, LOG) is specified.

Incident polarization. Provides a complex vector to specify the direction of the incident polarization state 1. The vector need not be normalized. Generally this will be taken to be linear polarization in the y direction: (0.,0.) (1.,0.) (0.,0.). If desired, however, polarization state JO=1 could be specified to be a more general state of elliptical polarization (which is why complex numbers are needed to specify the polarization state). Given polarization state 1, DDSCAT will automatically construct the (orthogonal) polarization state 2.

IORTH Set IORTH=1 if desiring to study only a single incident polarization. Set IORTH=2 to compute for both incident polarization states. (Note that if NPHI>1 is specified below, DDSCAT will go ahead and do both polarization states anyway even if IORTH=1, since no additional work is required.)

IWRKSC controls the amount of output generated.

- IWRKSC=0 to only generate one output file per wavelength and radius
- IWRKSC=1 to generate one .sca file for every wavelength, radius, and target orientation.

BETAMI,BETAMX,NBETA BETAMI = minimum value of rotation angle beta. BETAMX = maximum value of rotation angle beta. NBETA = number of different rotation angles beta to be used. Note that the angle beta specifies rotation of the target arount target axis $\mathbf{a}_1$. If the discrete dipole array is symmetric under rotation through, say, 90 degrees, then one should use only beta values between 0 and 90 – doing computations for, say, both 45 and 135 would be unnecessary duplication. The actual beta values used for computations will be located at the midpoints of `NBETA` uniform intervals extending from `BETAMI` to `BETAMX`. Thus, for example, if one set `BETAMI=0`, `BETAMX=90`, and `NBETA=2`, the beta values used would be 22.5 and 67.5 deg.

THETMI,THETMX,NTHETA `THETMI` = minimum value of rotation angle θ `THETMX` = maximum value of rotation angle θ `NTHETA` = number of different rotation angles theta to be used. Here theta is the angle between $\mathbf{a}_1$ and the x-axis = direction of propagation of the incident radiation. If the discrete dipole array is symmetric under, say, θ - (180 - θ), then doing calculations for, say, both 45 and 135 would be unnecessary duplication. In this case one would set `THETMI`=0 and `THETMX`=90. If `NTHETA` is odd (recommended when doing averages over random orientation), then the actual theta values used for computations will be uniformly spaced in cos(theta) and will include the endpoints THETMI and `THETMX`. Simpson's rule will be used for orientational averaging. if `NTHETA` is even, then the actual theta values used for computations will be located at the "midpoints" of intervals which are uniform in cos(θ), and extend from `THETMI` to `THETMX`. Thus, for example, if one set `THETMI`=0, `THETMX`=90, and `NTHETA`=2, the θ values used would be 41.4 and 75.5 [arccos(0.75)and arccos(0.25)].

PHIMIN,PHIMAX,NPHI `PHIMIN` = minimum value of rotation angle φ `PHIMAX` = maximum value of rotation angle φ `NPHI` = number of different values of rotation angle φ to be used. Here phi is the angle between the $x - \mathbf{a}_1$ plane and the $x - y$ plane. It is important to appreciate that obtaining results for new values of phi is relatively inexpensive – `DDSCAT` computes the scattering and absorption properties from stored results for two orthogonal polarization states computed for the first phi value (for a given beta and theta).

phi_s, theta_smin, theta_smax, dtheta_s selected scattering directions in the Nth scattering plane. Each "scattering plane" is assumed to contain the x-axis. φ_s = angle (in degrees) between scattering plane and plane containing incident pol. vector $\mathbf{e}_{01}$ $\theta_{\rm smin}$ = first value of scattering angle θ_s (in degrees) $\theta_{\rm smax}$ = last value of scattering angle θ_s (in degrees) $dtheta_s$ = increments in θ_s (in degrees). Thus, for $N = 1$ the sample file `ddscat.par` specifies 0. 0. 180. 30., so that the first scattering plane will have $\varphi_s = 0$ (therefore containing $\mathbf{e}_x$ and the incident polarization vector $\mathbf{e}_{01}$; since the sample `ddscat.par` specified $\mathbf{e}_{01} = \mathbf{e}_y$, the first scattering plane will be the $x - y$ plane), and will be represented by 7 scattering directions with $\theta_s = 0, 30, 60, \cdots, 150, 180$. For $N = 2$, the sample file `ddscat.par` specifies 90. 180. 0. 30., so that the second scattering plane will have $\varphi_s = 90$ (i.e., will contain $\mathbf{e}_x$ and $\mathbf{e}_{02}$; since the sample `ddscat.par` specified $\mathbf{e}_{01} = \mathbf{e}_y$, $\mathbf{e}_{02} = \mathbf{e}_x$ and this second scattering plane will be the $x - z$ plane), and will be represented by 7 scattering directions with $\theta_s = 180, 150, \cdots, 30, 0$. Note that this particular prescription for the first two scattering was planes has the scattering direction varying continuously as one completes the directions in the first scattering plane and begins the directions for the second scattering plane. `DDSCAT` will continue to read lines for additional scattering planes until it reaches the end of the `ddscat.par` file. The example given calls for just two scattering planes; this may often be all that one can afford to display results for. In practice one may wish to decrease the value of $\Delta\theta_s$ from the present value of 10 (degrees) to, say, 5 or 3 or 2 or 1, depending on how much fine structure one expects to see in the scattering phase function.

Appendix A

wAArBBori.avg **files**

The file `w00r00ori.avg` generated by the sample calculation should look like the following

```
DDSCAT --- DDSCAT.4b.1                         [designates the version of the code]
TARGET --- Rectangular prism; NX,NY,NZ=   8   6   4        [8x6x4 rect. target]
BRENNR --- method of solution                  [used Brenner FFT implementation]
LATTDR --- prescription for polarizabilities  [used Lattice Dispersion Relation]
RCTNGL --- shape                                  [generated rectangular target]
NAT0 =     192 = number of dipoles                       [192 dipoles in target]
 AEFF=   1.00000 = effective radius (physical units)   [eff. radius of target]
 WAVE=   6.28319 = wavelength (physical units)            [wavelength in vacuo]
  K*A=   1.00000 = 2*pi*aeff/lambda               [eff. scatt. param. x=k*a_eff]
n= ( 1.3300 ,  0.0100),  epsilon= (  1.7688 ,  0.0266)  for material 1
            [target refractive index=1.33+.01i, diel.const= 1.7688+.0266i]
  TOL= 1.000E-05 = error tolerance for CCG method
ICTHM= 33 = theta_s values used in comp. of Qsca,g [number of scattering angles
                                theta_s used in eval. of Q_sca, g=<cos(theta)>]
IPHIM= 12 = phi_s values used in comp. of Qsca,g   [number of scattering angles
                                  phi_s used in eval. of Q_sca, g=<cos(theta)>]
( 1.0000 0.0000 0.0000) = target axis A1 in Target Frame
( 0.0000 1.0000 0.0000) = target axis A2 in Target Frame
                   [target orientation is defined by directions of axes A1, A2
                    embedded in target.  These are the directions of A1 and A2
                    in the so-called "Target Frame"=TF, or the directions of
                    A1 and A2 in the "Lab Frame"=LF prior to target rotation.]
( 0.2794 0.0000 0.0000) = k vector (latt. units) in Lab Frame
                                [the k vector of the incident radiation in TF]
( 0.0000, 0.0000)( 1.0000, 0.0000)( 0.0000, 0.0000) inc. pol. vec. 1 in LF
( 0.0000, 0.0000)( 0.0000, 0.0000)( 1.0000, 0.0000) inc. pol. vec. 2 in LF
  0.000   0.000 = beta_min, beta_max ;  NBETA = 1   [one value, BETA=0]
  0.000  90.000 = theta_min, theta_max; NTHETA= 3 [avg. THETA=0-90, 3 values]
  0.000   0.000 = phi_min, phi_max   ;   NPHI = 1   [one value, PHI=0]
Results averaged over   3 target orientations
```

```
                   and   2 incident polarizations
>DDSCAT   Qext      Qabs      Qsca     g=<cos>    Qbk       Qpha
  JO=1: 1.330E-01 3.269E-02 1.003E-01 2.345E-01 5.552E-03 4.836E-01
  JO=2: 9.063E-02 2.397E-02 6.666E-02 2.672E-01 3.431E-03 4.159E-01
  mean: 1.118E-01 2.833E-02 8.347E-02 2.476E-01 4.491E-03 4.497E-01
  Qpol= 4.234E-02                                  Qcpol= 6.769E-02
    [Selected scattering directions specified by
     THETA=angle between incident and scattered beam
     PHI=angle of scattered beam (PHI=0 for scatt. beam in xy plane)
                               (PHI=90 for scatt. beam in xz plane)
     <|fji|2>= scattering intensity (cf. D88 eq. 3.12)
               for scattering from incident pol. state i
                                 to scattered pol. state j
               (where j=1 for E in scattering plane,
                      j=2 for E perp. to scatt. plane)
      DEPOLR=depolarization ratio (measure of degree to
               which fully polarized incident beam produces
               depolarized scattered light; cf. BH83, p. 403)
      PPOL=degree of polarization of scattered light for
               unpolarized incident light; cf. BH83, p. 382)]
***************** Selected scattering directions *****************
ND THETA   PHI <|f11|^2> <|f21|^2> <|f12|^2> <|f22|^2>  DEPOLR    PPOL
 1   0.0   0.0 5.959E-02 2.339E-19 1.589E-19 4.375E-02 0.00000 -0.15321
 2  30.0   0.0 4.091E-02 1.208E-19 2.715E-19 4.001E-02 0.00000 -0.01118
 3  60.0   0.0 1.146E-02 1.161E-19 1.487E-19 3.192E-02 0.00000  0.47183
 4  90.0   0.0 1.335E-05 3.174E-19 2.025E-18 2.385E-02 0.00000  0.99888
 5 120.0   0.0 5.996E-03 4.814E-19 1.334E-18 1.740E-02 0.00000  0.48743
 6 150.0   0.0 1.500E-02 2.866E-19 1.880E-19 1.305E-02 0.00000 -0.06955
 7 180.0   0.0 1.744E-02 5.543E-20 8.116E-20 1.078E-02 0.00000 -0.23618
 8   0.0  90.0 4.375E-02 2.154E-18 2.850E-18 5.959E-02 0.00000  0.15321
 9  30.0  90.0 3.206E-02 2.256E-08 6.191E-06 5.767E-02 0.00014  0.28528
10  60.0  90.0 9.770E-03 4.668E-08 1.846E-05 5.102E-02 0.00061  0.67806
11  90.0  90.0 7.506E-06 2.886E-08 2.093E-05 3.980E-02 0.00105  0.99857
12 120.0  90.0 4.505E-03 6.440E-09 1.092E-05 2.801E-02 0.00067  0.72233
13 150.0  90.0 9.485E-03 5.219E-10 2.342E-06 2.007E-02 0.00016  0.35811
14 180.0  90.0 1.078E-02 1.474E-18 1.933E-18 1.744E-02 0.00000  0.23618
```

Appendix B

wAArBBkCCC.sca **files**

This wAArBBkCC.sca file should look like the following (the w00r00k000.sca file from the sample calculation)

```
DDSCAT --- DDSCAT.4b.1                   [designates the version of the code]
TARGET --- Rectangular prism; NX,NY,NZ=   8   6   4  [8x6x4 rectangular solid]
BRENNR --- method of solution               [used Brenner FFT implementation]
LATTDR --- prescription for polarizabilies  [used Lattice Dispersion Relation]
RCTNGL --- shape                               [generated rectangular target]
NAT0 =     192 = number of dipoles                    [192 dipoles in target]
  AEFF=   1.00000 = effective radius (physical units)   [eff. radius of target]
  WAVE=   6.28319 = wavelength (physical units)           [wavelength in vacuo]
   K*A=   1.00000 = 2*pi*aeff/lambda             [eff. scatt. param. x=k*a_eff]
n= ( 1.3300 ,  0.0100),  epsilon= (  1.7688 ,  0.0266)  for material 1
              [target refractive index=1.33+.01i, diel.const= 1.7688+.0266i]
   TOL= 1.000E-05 = error tolerance for CCG method
ICTHM= 33 = theta_s values used in comp. of Qsca,g [number of scattering angles
                                theta_s used in eval. of Q_sca, g=<cos(theta)>]
IPHIM= 12 = phi_s values used in comp. of Qsca,g   [number of scattering angles
                                  phi_s used in eval. of Q_sca, g=<cos(theta)>]
( 1.0000 0.0000 0.0000) = target axis A1 in Target Frame
( 0.0000 1.0000 0.0000) = target axis A2 in Target Frame
                   [target orientation is defined by directions of axes A1, A2
                    embedded in target.  These are the directions of A1 and A2
                    in the so-called "Target Frame"=TF, or the directions of
                    A1 and A2 in the "Lab Frame"=LF prior to target rotation.]
( 0.2794 0.0000 0.0000) = k vector (latt. units) in TF
( 0.0000, 0.0000)( 1.0000, 0.0000)( 0.0000, 0.0000) inc. pol. vec. 1 in TF
( 0.0000, 0.0000)( 0.0000, 0.0000)( 1.0000, 0.0000) inc. pol. vec. 2 in TF
BETA =  0.000 = rotation of target around A1               [beta for this case]
THETA=  0.000 = angle between A1 and k                    [theta for this case]
 PHI =  0.000 = rotation of A1 around k                     [phi for this case]
>DDSCAT   Qext      Qabs      Qsca     g=<cos>    Qbk       Qpha
```

```
  JO=1: 1.111E-01 3.028E-02 8.077E-02 3.503E-01 1.859E-03 4.668E-01
  JO=2: 8.651E-02 2.441E-02 6.210E-02 3.650E-01 1.362E-03 4.197E-01
  mean: 9.878E-02 2.735E-02 7.143E-02 3.567E-01 1.611E-03 4.432E-01
  Qpol= 2.454E-02                                  Qcpol= 4.711E-02
    [Selected scattering directions specified by
     THETA=angle between incident and scattered beam
     PHI=angle of scattered beam (PHI=0 for scatt. beam in xy plane)
                                (PHI=90 for scatt. beam in xz plane)
     <|fji|2>= scattering intensity (cf. D88 eq. 3.12)
               for scattering from incident pol. state i
                               to scattered pol. state j
               (where j=1 for E in scattering plane,
                      j=2 for E perp. to scatt. plane)
      DEPOLR=depolarization ratio (measure of degree to
               which fully polarized incident beam produces
               depolarized scattered light; cf. BH83, p. 40)
      PPOL=degree of polarization of scattered light for
               unpolarized incident light; cf. BH83, p. 382)]
 ***************** Selected scattering directions *****************
 ND thetas phis <|f11|^2> <|f21|^2> <|f12|^2> <|f22|^2>  DEPOLR    PPOL
  1   0.0   0.0 5.525E-02 1.704E-19 2.742E-19 4.450E-02 0.00000 -0.10772
  2  30.0   0.0 3.948E-02 1.623E-20 1.360E-19 4.133E-02 0.00000  0.02296
  3  60.0   0.0 1.131E-02 2.788E-19 1.307E-19 3.251E-02 0.00000  0.48385
  4  90.0   0.0 2.177E-05 3.966E-20 1.612E-18 2.114E-02 0.00000  0.99794
  5 120.0   0.0 3.513E-03 1.803E-19 3.297E-20 1.150E-02 0.00000  0.53207
  6 150.0   0.0 5.914E-03 1.527E-19 9.988E-20 5.930E-03 0.00000  0.00138
  7 180.0   0.0 5.841E-03 1.032E-20 2.572E-20 4.279E-03 0.00000 -0.15437
  8   0.0  90.0 4.450E-02 5.862E-19 1.495E-18 5.525E-02 0.00000  0.10772
  9  30.0  90.0 3.253E-02 1.947E-18 1.418E-18 5.317E-02 0.00000  0.24082
 10  60.0  90.0 9.632E-03 1.136E-18 1.362E-18 4.509E-02 0.00000  0.64796
 11  90.0  90.0 9.007E-06 8.042E-19 2.692E-19 3.079E-02 0.00000  0.99941
 12 120.0  90.0 3.042E-03 6.883E-19 2.765E-21 1.653E-02 0.00000  0.68926
 13 150.0  90.0 4.568E-03 6.023E-20 1.103E-19 8.213E-03 0.00000  0.28516
 14 180.0  90.0 4.279E-03 5.545E-20 2.100E-19 5.841E-03 0.00000  0.15437
```

Bibliography

Draine, B. T., 1988: "the discrete-dipole approximation and its application to interstellar graphite grains". *Astrophys. J.*, **333**, 848—872.

Draine, B. T. and P. J. Flatau, 1994: Discrete dipole approximation for scattering calculations. *J. Opt. Soc. Am. A*, **11**, 1491–1499.

Draine, B. T. and J. Goodman, 1993: "beyond Clausius-Mossotti: wave propagation on a polarizable point lattice and the discrete dipole approximation". *Astrophys. J.*, **405**, 685—697.

Goedecke, G. H. and S. G. O'Brien, 1988: "scattering by irregular inhomogeneous particles via the digitized Green's function algorithm". *Appl. Opt.*, **27**, 2431—2438.

Hage, J. I. and J. M. Greenberg, 1990: "a model for the optical properties of porous grains". *Astrophys. J.*, **361**, 251—259.

Rader, C. and N. Brenner, 1976: A new principle for fast Fourier transformation. *IEEE Trans. Acoust. Speech Signal Process.*, **ASSP-24**(3), 268—70.

Temperton, C., 1983: Self-sorting mixed-radix fast Fourier transforms (numerical weather prediction). *J. Comput. Phys.*, **52**(1), 1—23.

Warren, S., 1984: Optical constants of ice from the ultraviolet to the microwave. *Appl. Opt.*, **23**(8), 1206—25.

Index