Princeton Observatory Preprint POPe-695

User Guide for the Discrete Dipole Approximation Code DDSCAT
(Version 5a)

Bruce T. Draine
Princeton University Observatory
Princeton NJ 08544-1001
(draine@astro.princeton.edu)
and

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

last revised: 1997 April 24

Contents

1 Introduction

DDSCAT.5a is a Fortran software package to calculate scattering and absorption of electromagnetic waves by targets with arbitrary geometries using the "discrete dipole approximation" (DDA). In this approximation the target is replaced by an array of point dipoles (or, more precisely, polarizable points); the electromagnetic scattering problem for an incident periodic wave interacting with this array of point dipoles is then solved essentially exactly. The DDA (sometimes referred to as the "coupled dipole approximation") was apparently first proposed by Purcell & Pennypacker (1973). DDA theory was reviewed and developed further by Draine (1988), Draine & Goodman (1993), and recently reviewed by Draine & Flatau (1994).

DDSCAT.5a is a Fortran implementation of the DDA developed by the authors. It is intended to be a versatile tool, suitable for a wide variety of applications ranging from interstellar dust to atmospheric aerosols. As provided, **DDSCAT.5a** should be usable for many applications without modification, but the program is written in a modular form, so that modifications, if required, should be fairly straightforward.

The authors make this code openly available to others, in the hope that it will prove a useful tool. We ask only that:

- If you publish results obtained using **DDSCAT**, please consider acknowledging the source of the code.
- If you discover any errors in the code or documentation, please promptly communicate them to the authors.
- You comply with the "copyleft" agreement (more formally, the GNU General Public License) of the Free Software Foundation: you may copy, distribute, and/or modify the software identified as coming under this agreement. If you distribute copies of this software, you must give the recipients all the rights which you have. See the file `doc/copyleft` distributed with the DDSCAT software.

We also strongly encourage you to send email to the authors identifying yourself as a user of DDSCAT; this will enable the authors to notify you of any bugs, corrections, or improvements in DDSCAT.

The current version, **DDSCAT.5a** , uses the DDA formulae from Draine (1988). The code incorporates Fast Fourier Transform (FFT) methods (Goodman, Draine, & Flatau 1991). The "lattice dispersion relation" (LDR) prescription (Draine & Goodman 1993) is used for determining dipole polarizabilities.

This User Guide assumes that you have already obtained the Fortran source code for **DDSCAT.5a** following the instructions in the `README` file.[1] We refer you to the list of references at the end of this document for discussions of the theory and accuracy of the DDA [first see the recent review by Draine and Flatau (1994)]. In §4 we describe the principal changes between **DDSCAT.5a** and the previous releases.[2] The succeeding sections contain instructions for:

- compiling and linking the code;
- running a sample calculation;
- understanding the output from the sample calculation;
- modifying the parameter file to do your desired calculations;
- specifying target orientation;
- changing the `DIMENSION`ing of the source code to accommodate 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.

Finally, this User Guide can be obtained by anonymous `ftp` from `astro.princeton.edu`, subdirectory `draine/scat/ddscat/ver5a`. For postscript, `get` either the uncompressed postscript file `UserGuide.ps` or the gzipped file `UserGuide.ps.gz` (remember to set the `binary` option in `ftp` before using the `get` command). Alternatively, you can `get` the LaTeX source `UserGuide.tex.gz` and the figures `f1.eps.gz`, `f2.eps.gz`, `f3.eps.gz`, `f4.eps.gz`, and `f5.eps.gz`.

2 Applicability of the DDA

The principal advantage of the DDA is that it is completely flexible regarding the geometry of the target, being limited only by the need to use an interdipole separation d small compared to (1) any structural lengths in the target, and (2) the wavelength λ. Numerical studies (Draine & Goodman 1993; Draine & Flatau 1994) indicate that the second criterion is adequately satisfied if

$$|m|kd < 1 \quad , \tag{1}$$

where m is the complex refractive index of the target material, and $k \equiv 2\pi/\lambda$, where λ is the wavelength *in vacuo*.

[1] To obtain the `README` file: (1) anonymous `ftp` to `astro.princeton.edu` , (2) `cd draine/scat/DDA/ver5a`, and (3) `get README`.

[2] The previous "official release" was **DDSCAT.4b**, although **DDSCAT.4c** – while never announced – was made available to a number of interested users.

Practical considerations of CPU speed and computer memory currently available on scientific workstations limit the number of dipoles employed to $N < 10^5$; this translates into a limitation to target sizes and wavelengths such that the "size parameter"

$$x \equiv \frac{2\pi a_{\rm eff}}{\lambda} < \frac{29}{|m|}\left(\frac{N}{10^5}\right)^{1/3} \quad , \tag{2}$$

or, equivalently,

$$a_{\rm eff} < 4.6\left(\frac{N}{10^5}\right)^{1/3}\frac{\lambda}{|m|} \quad , \tag{3}$$

where

$$a_{\rm eff} \equiv (3V/4\pi)^{1/3} \tag{4}$$

is defined to be the radius of a sphere of volume equal to the target volume $V = Nd^3$ (the dipoles are assumed to be located on a cubic lattice). It is therefore clear that the DDA is not suitable for very large values of the size parameter x, or very large values of the refractive index m. The primary utility of the DDA is for scattering by dielectric targets with sizes comparable to the wavelength. As discussed by Draine & Goodman (1993) and Draine & Flatau (1994), the DDA overall cross sections accurate to a few percent provided $N > 10^4$ dipoles are used, criterion (1) is satisfied, and $|m-1| < 2$. Examples illustrating the accuracy of the DDA are shown in Figs. 1-2 which show overall scattering and absorption efficiencies as a function of wavelength for different discrete dipole approximations to a sphere, with N ranging from 304 to 59728. The DDA calculations assumed radiation incident along the (1,1,1) direction in the "target frame". Figs. 3-4 show the scattering properties calculated with the DDA for $x = ka = 7$. Additional examples can be found in Draine & Flatau (1994).

3 DDSCAT.5a

DDSCAT.5a solves the problem of scattering and absorption by an array of polarizable point dipoles interacting with a monochromatic plane wave incident from infinity. **DDSCAT.5a** has the capability of automatically generating dipole array representations for a variety of target geometries (see §18) and can also accept dipole array representations of targets supplied by the user (although the dipoles must be located on a cubic lattice). The incident plane wave can have arbitrary elliptical polarization (see §20), and the target can be arbitrarily oriented relative to the incident radiation (see §16). The following quantities are calculated by **DDSCAT.5a** :

- Absorption efficiency factor $Q_{abs} \equiv C_{abs}/\pi a_{\rm eff}^2$, where C_{abs} is the absorption cross section;
- Scattering efficiency factor $Q_{sca} \equiv C_{sca}/\pi a_{\rm eff}^2$, where C_{sca} is the scattering cross section;
- Extinction efficiency factor $Q_{ext} \equiv Q_{sca} + Q_{abs}$;
- Phase lag efficiency factor Q_{pha}, defined so that the phase-lag (in radians) of a plane wave after propagating a distance L is just $n_t Q_{pha} \pi a_{\rm eff}^2 L$, where n_t is the number density of targets.

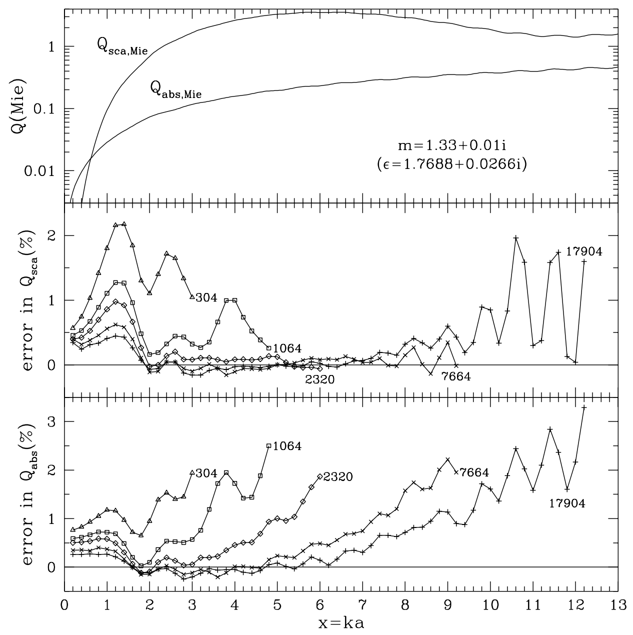

Figure 1: Scattering and absorption for a sphere with $m = 1.33 + 0.01i$. The upper panel shows the exact values of Q_{sca} and Q_{abs}, obtained with Mie theory, as functions of $x = ka$. The middle and lower panels show fractional errors in Q_{sca} and Q_{abs}, obtained using **DDSCAT** with polarizabilities obtained from the Lattice Dispersion Relation, and labelled by the number N of dipoles in each pseudosphere. After Fig. 1 of Draine & Flatau (1994).

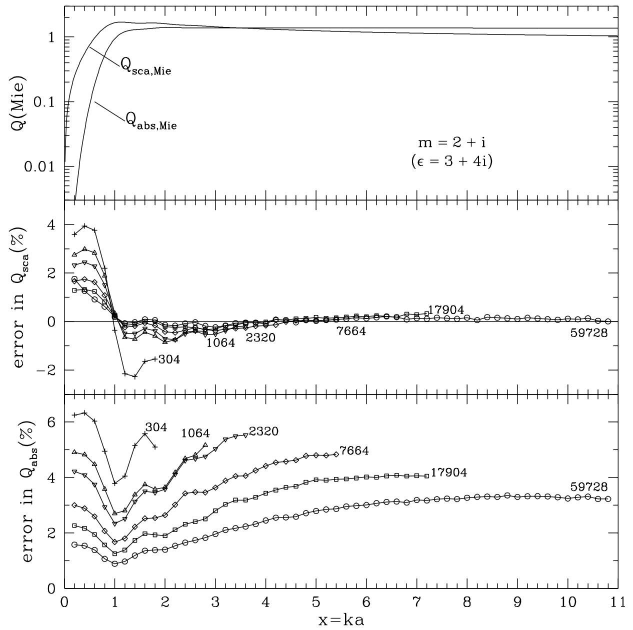

Figure 2: Same as Fig. 1, but for $m = 2 + i$. After Fig. 2 of Draine & Flatau (1994).

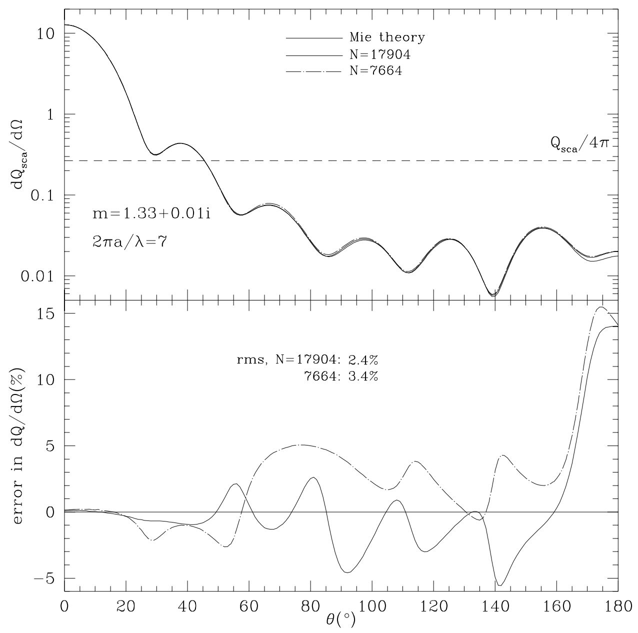

Figure 3: Differential scattering cross section for $m = 1.33+0.01i$ pseudosphere and $ka = 7$. Lower panel shows fractional error compared to exact Mie theory result. The $N = 17904$ pseudosphere has $|m|kd = 0.57$, and an rms fractional error in $d\sigma/d\Omega$ of 2.4%. After Fig. 5 of Draine & Flatau (1994).

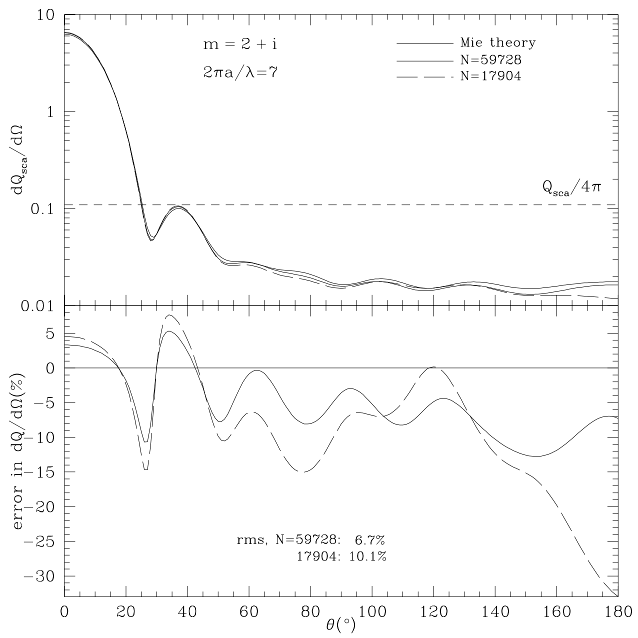

Figure 4: Same as Fig. 3 but for $m = 2 + i$. The $N = 59728$ pseudosphere has $|m|kd = 0.65$, and an rms fractional error in $d\sigma/d\Omega$ of 6.7%. After Fig. 8 of Draine & Flatau (1994).

- The 4×4 Mueller scattering intensity matrix S_{ij} describing the complete scattering properties of the target for scattering directions specified by the user.
- The radiation force efficiency vector $\mathbf{Q}_{rad}$ (see §14).
- The radiation torque efficiency vector $\mathbf{Q}_\Gamma$ (see §14).

4 What's New?

DDSCAT.5a differs from previous versions in five major respects:

1. Use of the new Generalized Prime Factor Algorithm (GPFA) developed by Clive Temperton (1992) for FFT calculations. The GPFA algorithm is generally faster than the previous algorithms, yet requires no more memory than the algorithm of Brenner (1969). See §12.
2. Availability of a new algorithm for iterative solution of the system of complex linear equations. This algorithm is often faster than the algorithm of Petravic and Kuo-Petravic (1979) which was used through **DDSCAT.4b** (and which remains available as an option in **DDSCAT.5a**). See §13.
3. Automatic calculation of the transverse components of the radiative force on the target. See §14.
4. Capability to compute the electromagnetic torque on the target, due to absorption and scattering of light from the incident beam, as described by Draine and Weingartner (1996). See §14
5. Automatic calculation of the 4×4 Mueller scattering matrix. See §22.
6. Improved output handling, including FORTRAN unformatted binary write option as well as a netCDF interface and IDL postprocessing code.
7. Finally, this much-improved and expanded User Guide!

We also call users' attention to a minor but possibly confusing change: **DDSCAT.5a** uses a different convention for specifying the target axes $\hat{\mathbf{a}}_1$ and $\hat{\mathbf{a}}_2$ for certain target choices (`RCTNGL`, `ELLIPS`, ...). We think the new convention is more straightforward. See the discussion in §18 below for details.

5 Compiling and Linking

In the discussion below, it will be assumed that the source code for **DDSCAT.5a** has been installed in a directory `DDA/src` . To compile the code on a Unix system, position yourself in the directory `DDA/src`. The `Makefile` supplied has compiler options appropriate for Sun Fortran under Solaris 2.x. If you have a different Fortran compiler, you will probably need to edit `Makefile` to provide the desired compiler options. `Makefile` contains samples of compiler options for selected operating systems, including HP AUX, IBM AIX, and SGI IRIX operating systems.

So far as we know, there are only two operating-system-dependent aspects of **DDSCAT.5a**: (1) the device number to use for "standard output", and (2) the `TIMEIT` routine. There is, in addition, one installation-dependent aspect to the code: the procedure for linking to the netCDF library (see §9.3 for discussion of the possibility of writing binary files using the machine-independent netCDF standard).

5.1 Device Numbers `IDVOUT` and `IDVERR`

The variables `IDVOUT` and `IDVERR` specify device numbers for "running output" and "error messages", respectively. 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` statements in the "main" program `DDSCAT.f` and in the output routine `WRIMSG` (file `wrimsg.f`). 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/`, and edit `wrimsg.f` to change `DATA IDVOUT/0/` to `DATA IDVOUT/6/`.

5.2 Subroutine `TIMEIT`

The only other operating system-dependent part of **DDSCAT.5a** is the single subroutine `TIMEIT`. Several versions of `TIMEIT` are provided:

- `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_osf.f` uses the DEC OSF system call `etime`
- `timeit_sgi.f` uses the IRIX system call `etime`
- `timeit_vms.f` uses the VMS system calls `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.

You *must* compile and link one of the `timeit_xxx.f` routines, possibly after modifying it to work on your system; `timeit_null.f` is the easiest choice, but it will return no timing information.[3]

5.3 Leaving the netCDF Capability Disabled

The `Makefile` supplied with the distribution of **DDSCAT.5a** is set up to link to a "dummy" subroutine `dummywritenet.f` instead of subroutine `writenet.f`, in order to minimize problems during initial compilation and linking. The "dummy" routine has no functionality, other than bailing out with a fatal error message if the user makes the mistake of trying to specify one of the netCDF options (`ALLCDF` or `ORICDF`). First-time users of **DDSCAT.5a** should *not* try to use the netCDF option – simply skip this section, and specify option `NOTCDF` in `ddscat.par`. After successfully using **DDSCAT.5a** you can return to §5.4 to try to enable the netCDF capability.

5.4 Enabling the netCDF Capability

Subroutine `WRITENET` (file `writenet.f`) provides the capability to output binary data in the netCDF standard format (see §9.3). In order to use this routine (instead of `dummywritenet.f`), it is necessary to take the following steps:[4]

1. Have netCDF already installed on your system (check with your system administrator).
2. Find out where `netcdf.inc` is located and edit the `include` statement in `writenet.f` to specify the correct path to `netcdf.inc`.
3. Find out where the `libnetcdf.a` library is located, and edit the Makefile so that the variable `LIBNETCDF` specifies the correct path to this library.
4. Edit the Makefile to "comment out" (with a `#` symbol in column 1) the line `writenet = dummywritenet` and "uncomment" (remove the `#` symbol) the line `writenet = writenet` so that `writenet.f` will be compiled instead of the dummy routine `dummywritenet.f`.

5.5 Compiling and Linking...

After suitably editing the Makefile (while still positioned in `DDA/src`) simply type[5]

```
make ddscat
```

which should create an executable file `DDA/src/ddscat` . The resulting executable will *not* have

[3]Non-Unix sites: The VMS-compatible version of `TIMEIT` is included in the file `SRC9.FOR`. For non-VMS sites, you will need to replace this version of `TIMEIT` with one of the other versions, which are to be found in the file `MISC.FOR`. If you are having trouble getting this to work, choose the "dummy" version of `TIMEIT` from `MISC.FOR`: this will return no timing information, but is platform-independent.

[4]Non-UNIX sites: You need to replace the "dummy" version of `SUBROUTINE WRITENET` in `SRC1.FOR` with the version provided in `MISC.FOR`. You will also need to consult your systems administrator to verify that the netCDF library has already been installed on your system, and to find out how to link to the library routines.

[5]Non-UNIX sites: The objective is to compile and link all of the Fortran code [except for (1) the program `CALLTARGET`, (2) the program `TSTFFT`, (3) the multiple versions of `WRITEBIN`, and (4) the multiple versions of `TIMEIT`] to create an executable with a name like `ddscat` or `DDSCAT.EXE` or whatever your local convention for Fortran executables dictates. If you are unable to use the "tarfile", you will want to compile and link all of the source code in the files "SRCn.FOR" (for n=0-9), "`CGCOMMON.FOR`", "`GPFA.FOR`", "`LAPACK.FOR`", and "`PIM.FOR`".

netCDF capability, and will contain timing instructions compatible with the Solaris 1.x or 2.x operating systems on Sun computers, as well as several other versions of Unix. To add netCDF capability, see §5.4. To replace the Sun-compatible timing routine with another, see §5.2.

6 Moving the Executable

Now reposition yourself into the directory `DDA` (e.g., type `cd ..`), and copy the executable from `src/ddscat` to the `DDA` directory by typing

```
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
```

7 The Parameter File `ddscat.par`

The directory `DDA` should now contain a sample file `ddscat.par` which provides parameters to the program `ddscat`. As provided (see AppendixA), the file `ddscat.par` is set up to calculate scattering by a 8×6×4 rectangular array of 192 dipoles, with an effective radius $a_{\text{eff}} = 1\mu$m, at a wavelength of 6.2832μm (for a "size parameter" $2\pi a_{\text{eff}}/\lambda = 1$).

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`.

The sample parameter file as supplied calls for the new GPFA FFT routine (`GPFAFT`) of Temperton (1992) to be employed and the `PBCGST` iterative method to be used for solving the system of linear equations. (See section §12 and §13 for discussion of choice of FFT algorithm and choice of equation-solving algorithm.)

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. See §10 for discussion of other options.

The sample `ddscat.par` file specifies that the calculations be done for a single wavelength (6.2832μm) and a single effective radius (1.00μm). Note that in **DDSCAT.5a** the "effective radius" a_{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 $a_{\text{eff}} = (3N/4\pi)^{1/3}d$.

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 $\hat{\mathbf{e}}_{01}$ to be along the y axis (and consequently polarization state $\hat{\mathbf{e}}_{02}$ will automatically be taken to be along the z axis). `IORTH=2` in `ddscat.par` calls for calculations to be carried out for both incident polarization states ($\hat{\mathbf{e}}_{01}$ and $\hat{\mathbf{e}}_{02}$).

The target is assumed to have two vectors $\hat{\mathbf{a}}_1$ and $\hat{\mathbf{a}}_2$ embedded in it; $\hat{\mathbf{a}}_2$ is perpendicular to $\hat{\mathbf{a}}_1$. In the case of the 8×6×4 rectangular array of the sample calculation, the vector $\hat{\mathbf{a}}_1$ is along the "long" axis of the target, and the vector $\hat{\mathbf{a}}_2$ is along the "intermediate" axis. The target orientation in the Lab Frame is set by three angles: β, Θ, and Φ, defined and discussed below in §16. Briefly, the polar angles Θ and Φ specify the direction of $\hat{\mathbf{a}}_1$ in the Lab Frame. The target is assumed to be rotated around $\hat{\mathbf{a}}_1$ by an angle β. The sample `ddscat.par` file specifies $\beta = 0$ and $\Phi = 0$ (see lines in `ddscat.par` specifying variables `BETA` and `PHI`), and calls for three values of the angle Θ (see line in `ddscat.par` specifying variable `THETA`). **DDSCAT.5a** chooses Θ values uniformly spaced in $\cos\Theta$; thus, asking for three values of Θ between 0 and 90° yields $\Theta = 0$, 60°, and 90°.

Appendix A provides a detailed description of the file `ddscat.par`.[6]

8 Running DDSCAT.5a Using the Sample `ddscat.par` File

To execute the program on a UNIX system (running either `sh` or `csh`), 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 (8x6x4=192 dipole target, 3 orientations, two incident polarizations, with scattering calculated for 14 distinct scattering directions), requires 1.0 cpu seconds on a Sun Ultra-170 workstation.

9 Output Files

9.1 ASCII files

If you run DDSCAT using the command

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

you will have various types of ASCII files when the computation is complete:

- a file `ddscat.out`;
- a file `mtable`;
- a file `qtable`;
- a file `qtable2`;
- files w*xx*r*yy*`ori.avg` (one, `w00r00ori.avg`, for the sample calculation);
- if `ddscat.par` specified `IWRKSC`=1, there will also be files w*xx*r*yy*k*zzz*`.sca` (3 for the sample calculation: `w00r00k000.sca`, `w00r00k001.sca`, `w00r00k002.sca`).

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 iterative calculations, Q_{ext}, Q_{abs}, and Q_{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.

[6]Note that the structure of `ddscat.par` depends on the version of **DDSCAT** being used. Make sure you update old parameter files before using them with **DDSCAT.5a** !

The file `qtable` contains a summary of the orientationally-averaged values of Q_{ext}, Q_{abs}, Q_{sca}, $g(1) = \langle\cos(\theta_s)\rangle$, and Q_{bk}. Here Q_{ext}, Q_{abs}, and Q_{sca} are the extinction, absorption, and scattering cross sections divided by $\pi a_{\rm eff}^2$. Q_{bk} is the differential cross section for backscattering (area per sr) divided by $\pi a_{\rm eff}^2$.

The file `qtable2` contains a summary of the orientationally-averaged values of Q_{pha}, Q_{pol}, and Q_{cpol}. Here Q_{pha} is the "phase shift" cross section divided by $\pi a_{\rm eff}^2$ (see definition in Draine 1988). Q_{pol} is the "polarization efficiency factor", equal to the difference between Q_{ext} for the two orthogonal polarization states. We define a "circular polarization efficiency factor" $Q_{cpol} \equiv Q_{pol}Q_{pha}$, 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_{cpol}.

For each wavelength and size, **DDSCAT.5a** produces a file with a name of the form w*xx*r*yy*ori.avg, where index *xx* (=00, 01, 02....) designates the wavelength and index *yy* (=00, 01, 02...) designates the "radius"; this file contains Q values and scattering information averaged over however many target orientations have been specified (see §16. The file `w00r00ori.avg` produced by the sample calculation is provided below in Appendix B.

In addition, if `ddscat.par` has specified `IWRKSC`=1 (as for the sample calculation), **DDSCAT.5a** will generate files with names of the form w*xx*r*yy*k*zzz*.avg, where *xx* and *yy* are as before, and index *zzz* =(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 w*xx*r*yy*ori.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.5a** with the sample `ddscat.par` file will therefore generate files `w00r00k000.sca`, `w00r00k001.sca`, and `w00r00k002.sca` . To understand the information contained in one of these files, please consult Appendix C, which contains an example of the file `w00r00k000.sca` produced in the sample calculation.

9.2 Binary Option

It is possible to output an "unformatted" or "binary" file (`dd.bin`) with fairly complete information, including header and data sections. This is accomplished by specifying either `ALLBIN` or `ORIBIN` in `ddscat.par` .

Subroutine `writebin.f` provides an example of how this can be done. The "header" section contains dimensioning and other variables which do not change with wavelength, particle geometry, and target orientation. The header section contains data defining the particle shape, wavelengths, particle sizes, and target orientations. If `ALLBIN` has been specified, the "data" section contains, for each orientation, Mueller matrix results for each scattering direction. The data output is limited to actual dimensions of arrays; e.g. `nscat,4,4` elements of Muller matrix are written rather than `mxscat,4,4`. This is an important consideration when writing postprocessing codes.

A skeletal example of a postprocessing code was written by us and is provided in subdirectory `DDA/IDL`. If you do plan to use the Interactive Data Language (IDL) for postprocessing, you may consider using the netCDF binary file option which offers substantial advantages over the FORTRAN unformatted write. More information about IDL is available at `http://www.rsinc.com/idl`.

Unfortunately IDL requires a license and hence is not distributed with **DDSCAT**.

9.3 Machine-Independent Binary File Option: netCDF

The "unformatted" binary file is compact, fairly complete, and (in comparison to ASCII output files) is efficiently written from FORTRAN. However, binary files are not compatible between different computer architectures if written by "unformatted" `WRITE` from FORTRAN. Thus, you have to process the data on the same computer architecture that was used for the **DDSCAT** calculations. We provide an option of using netCDF with DDSCAT. The netCDF library defines a machine-independent format for representing scientific data. Together, the interface, library, and format support the creation, access, and sharing of scientific data. For more information see `http://www.unidata.ucar.edu/packages/netcdf`.

Several major graphics packages (for example IDL) have adopted netCDF as a standard for data transport. For this reason, and because of strong and free support of netCDF over the network by UNIDATA, we have implemented a netCDF interface in **DDSCAT**. This may become the method of choice for binary file storage of output from **DDSCAT**.

After the initial "learning curve" netCDF offers many advantages:

- It is easy to examine the contents of the file (using tools such as `ncdump`).
- Binary files are portable - they can be created on a supercomputer and processed on a workstation.
- Major graphics packages now offer netCDF interfaces.
- Data input and output are an order of magnitude faster than for ASCII files.
- Output data files are compact.

The disadvantages include:

- Need to have netCDF installed on your system.
- Lack of support of complex numbers.
- Nontrivial learning curve for using netCDF.
- Lack of portability of netCDF libraries.

The public-domain netCDF software is available for many operating systems from `http://www.unidata.ucar.edu/packages/netcdf` . The steps necessary for enabling the netCDF capability in **DDSCAT.5a** are listed above in §5.4. Once these have been successfully accomplished, and you are ready to run **DDSCAT** to produce netCDF output, you must choose either the `ALLCDF` or `ORICDF` option in `ddscat.par`; `ALLCDF` will result in a file which will include the Muller matrix for every wavelength, particle size, and orientation, whereas `ORICDF` will result in a file limited to just the orientational averages for each wavelength and target size.

10 Dipole Polarizabilities

Option `LATTDR` specifies 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 & O'Brien 1988 and Hage & Greenberg 1990). In the limit $|m|kd \ll 1$ (where $k = 2\pi/\lambda$ 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 & Goodman (1993) for discussion and comparison of these three prescriptions. **Option `LATTDR` is recommended for general use**, based upon the tests presented by Draine & Goodman (1993).

11 Dielectric Functions

In order to assign the appropriate dipole polarizabilities, **DDSCAT.5a** must be given the dielectric constant of the material (or materials) of which the target of interest is composed. Unless the user wishes to use the dielectric function of either solid or liquid H_2O (see below), his information is supplied to **DDSCAT** through a table (or tables), read by subroutine `DIELEC` in file `dielec.f`, and providing either the complex refractive index $m = n + ik$ or complex dielectric function $\epsilon = \epsilon_1 + i\epsilon_2$ as a function of wavelength λ. Since $m = \epsilon^{1/2}$, or $\epsilon = m^2$, the user must supply either m or ϵ, but not both.

The table formatting is intended to be quite flexible. The first line of the table consists of text, up to 80 characters of which will be read and included in the output to identify the choice of dielectric function. (For the sample problem, it consists of simply the statement `m = 1.33 + 0.01i`.) The second line consists of 5 integers; either the second and third *or* the fourth and fifth should be zero.

- The first integer specifies which column the wavelength is stored in.
- The second integer specifies which column $\mathrm{Re}(m)$ is stored in.
- The third integer specifies which column $\mathrm{Im}(m)$ is stored in.
- The fourth integer specifies which column $\mathrm{Re}(\epsilon)$ is stored in.
- The fifth integer specifies which column $\mathrm{Im}(\epsilon)$ is stored in.

If the second and third integers are zeros, then `DIELEC` will read $\mathrm{Re}(\epsilon)$ and $\mathrm{Im}(\epsilon)$ from the file; if the fourth and fifth integers are zeros, then $\mathrm{Re}(m)$ and $\mathrm{Im}(m)$ will be read from the file.

The third line of the file is used for column headers, and the data begins in line 4. *There must be at least 3 lines of data:* even if ϵ is required at only one wavelength, please supply two additional "dummy" wavelength entries in the table so that the interpolation apparatus will not be confused.

In the event that the user wishes to compute scattering by targets with the refractive index of either solid or liquid H_2O, we have included two "built-in" dielectric functions. If `H2OICE` is specified on line 10 of `ddscat.par`, **DDSCAT** will use the dielectric function of H_2O ice at T=250K as compiled by Warren (1984). If `H2OLIQ` is specified on line 10 of `ddscat.par`, **DDSCAT** will use the dielectric function for liquid H_2O at T=280K using subroutine `REFWAT` written by Eric A. Smith. For more information see `http://atol.ucsd.edu/~pflatau/scatlib/refr.html` .

12 Choice of FFT Algorithm

One major change in going from **DDSCAT.4b** to **4c** and **5a** was modification of the code to permit use of the GPFA FFT algorithm developed by Dr. Clive Temperton. DDSCAT continues to offer both the Brenner code as well as the “old” Temperton code as alternative FFT implementations. The “old” Temperton code requires about 11% more memory than either the Brenner or GPFA codes (to use the “old” Temperton algorithm, `DDSCAT.f` must be compiled with `MXMEM`=1 rather than `MXMEM`=0).

To help persuade the user that the GPFA code is an important step forward, we provide a program `TSTFFT` to compare the performance of different 3-D FFT implementations. To compile, link, and run this program on a Unix system,[7] position yourself in the `DDA/src` directory and type

```
make tstfft
tstfft
```

Output results will be written into a file `res.dat`. Here is a copy of the `res.dat` file created when run on a Sun Ultrasparc 170:

```
 CPU time (sec) for different 3-D FFT methods
 Machine= Sun Ultrasparc-170
 parameter LVR = 64
 LVR="length of vector registers" in gpfa2f,gpfa3f,gpfa5f
=============================================================
                         Brenner       Temperton      Temperton
   NX   NY   NZ          (FOURX)         (Old)          (GPFA)
    2    2    2         0.000068       0.000065       0.000262
    3    3    3         0.000200       0.000077       0.000096
    4    4    4         0.000064       0.000086       0.000111
    5    5    5         0.000651       0.000292       0.000137
    6    6    6         0.001171       0.000170       0.000223
    8    8    8         0.000350       0.000250       0.000323
    9    9    9         0.004654       0.000300       0.000505
   10   10   10         0.005218       0.000457       0.000575
   12   12   12         0.008664       0.000814       0.000751
   15   15   15         0.023387       0.001583       0.001798
   16   16   16         0.003137       0.001359       0.002386
   18   18   18         0.042196       0.003908       0.003479
   20   20   20         0.043754       0.003881       0.004010
   24   24   24         0.075672       0.010722       0.007198
   25   25   25         0.103492       0.008142       0.011232
   27   27   27         0.160173       0.011126       0.012293
   30   30   30         0.185177       0.017111       0.014631
   32   32   32         0.042211       0.034032       0.023245
   36   36   36         0.322581       0.071013       0.027502
   40   40   40         0.379551       0.133391       0.044182
```

[7] Non-Unix systems: the `TSTFFT` Fortran source code is in the file `MISC.FOR`.

45	45	45	0.797107	0.199537	0.069740
48	48	48	0.668408	0.255849	0.094520
50	50	50	0.983722	0.287208	0.115833
54	54	54	1.530319	0.427839	0.142388
60	60	60	1.804154	0.607189	0.181221
64	64	64	1.005013	0.806311	0.342762
72	72	72	3.812469	1.219364	0.362328
75	75	75	5.364067	1.234528	0.437496
80	80	80	4.509048	1.576118	0.563839
81	81	81	8.017456	1.798195	0.560326
90	90	90	10.201881	2.455399	0.779869

It is clear that the GPFA code is generally MUCH faster than the Brenner FFT (by a factor of 13 for the 90×90×90 case) and significantly faster than the "old" Temperton FFT code (by a factor of 3 for the 90×90×90 case), although for some cases the differences are not large (e.g., 32×32×32). Since the GPFA code is memory-efficient as well, it appears to be the method of choice on scalar machines. It appears to also be best on vector machines.

The GPFA code contains a parameter `LVR` which is set in `data` statements in the routines `gpfa2f`, `gpfa3f`, and `gpfa5f`. `LVR` is supposed to be optimized to correspond to the "length of a vector register" on vector machines. As delivered, this parameter is set to 64, which is supposed to be appropriate for Crays other than the C90 (for the C90, 128 is supposed to be preferable).[8] The value of `LVR` is not critical for scalar machines, as long as it is fairly large. We found little difference between `LVR`=64 and 128 on a Sparc 10/51 and on an Ultrasparc 170. You may wish to experiment with different `LVR` values on your computer architecture. To change `LVR`, you need to edit `gpfa.f` and change the three `data` statements where `LVR` is set.

The choice of FFT implementation is obtained by specifying one of:

- `GPFAFT` to use the GPFA algorithm (Temperton 1992) – **this is recommended!**;
- `TMPRTN` to obtain the "old" Temperton (1983) implementation;
- `BRENNR` to obtain the `FOURX` implementation of the Brenner (1969) algorithm;
- `CONVEX` to use the native FFT routine on a Convex system

13 Choice of Iterative Algorithm

As discussed elsewhere (e.g., Draine 1988), the problem of electromagnetic scattering of an incident wave $\mathbf{E}_{inc}$ by an array of N point dipoles can be cast in the form

$$\mathbf{AP} = \mathbf{E} \tag{5}$$

where $\mathbf{E}$ is a $3N$-dimensional (complex) vector of the incident electric field $\mathbf{E}_{inc}$ at the N lattice sites, $\mathbf{P}$ is a $3N$-dimensional (complex) vector of the (unknown) dipole polarizations, and $\mathbf{A}$ is a $3N \times 3N$ complex matrix.

[8] In place of "preferable" users are encouraged to read "necessary" – there is some basis for fearing that results computed on a C90 with `LVR` other than 128 run the risk of being incorrect! Please use `LVR=128` if running on a C90!

Because $3N$ is a large number, direct methods for solving this system of equations for the unknown vector $\mathbf{P}$ are impractical, but iterative methods are useful: we begin with a guess (typically, $\mathbf{P} = 0$) for the unknown polarization vector, and then iteratively improve the estimate for $\mathbf{P}$ until equation (5) is solved to some error criterion. The error tolerance may be specified as

$$\frac{|\mathbf{A}^\dagger \mathbf{A}\mathbf{P} - \mathbf{A}^\dagger \mathbf{E}|}{|\mathbf{A}^\dagger \mathbf{E}|} < h \quad , \tag{6}$$

where $\mathbf{A}^\dagger$ is the Hermitian conjugate of $\mathbf{A}$ $[(A^\dagger)_{ij} \equiv (A_{ji})^*]$, and h is the error tolerance. We typically use $h = 10^{-5}$ in order to satisfy eq.(5) to high accuracy. The error tolerance h can be specified by the user (see Appendix A).

A major change in going from **DDSCAT.4b** to **5a** is the implementation of several different algorithms for iterative solution of the system of complex linear equations. **DDSCAT.5a** has been modified to permit solution algorithms to be treated in a fairly "modular" fashion, facilitating the testing of different algorithms. Many algorithms were compared by Flatau (1997)[9]; two of them performed well and are made available to the user in **DDSCAT.5a**. The choice of algorithm is made by specifying one of the options:

- `PBCGST` – Preconditioned BiConjugate Gradient with STabilitization method from the Parallel Iterative Methods (PIM) package created by R. Dias da Cunha and T. Hopkins.
- `PETRKP` – the complex conjugate gradient algorithm of Petravic & Kuo-Petravic (1979), as coded in the Complex Conjugate Gradient package (CCGPACK) created by P.J. Flatau. This is the algorithm discussed by Draine (1988) and used in previous versions of **DDSCAT**.

Both methods work well. Our experience suggests that `PBCGST` is often faster than `PETRKP`, by perhaps a factor of two. We therefore recommend it, but include the `PETRKP` method as an alternative.

The Parallel Iterative Methods (PIM) by Rudnei Dias da Cunha (`rdd@ukc.ac.uk`) and Tim Hopkins (`trh@ukc.ac.uk`) is a collection of Fortran 77 routines designed to solve systems of linear equations on parallel and scalar computers using a variety of iterative methods (available at `http://www.mat.ufrgs.br/pim-e.html`). PIM offers a number of iterative methods, including

- Conjugate-Gradients, CG (Hestenes 1952),
- Bi-Conjugate-Gradients, BICG (Fletcher 1976),
- Conjugate-Gradients squared, CGS (Sonneveld 1989),
- the stabilised version of Bi-Conjugate-Gradients, BICGSTAB (van der Vorst 1991),
- the restarted version of BICGSTAB, RBICGSTAB (Sleijpen & Fokkema 1992)
- the restarted generalized minimal residual, RGMRES (Saad 1986),
- the restarted generalized conjugate residual, RGCR (Eisenstat 1983),

[9] A postscript copy of this report – file `cg.ps` – is distributed with the **DDSCAT.5a** documentation.

- the normal equation solvers, CGNR (Hestenes 1952 and CGNE (Craig 1955),
- the quasi-minimal residual, QMR (highly parallel version due to Bucker & Sauren 1996),
- transpose-free quasi-minimal residual, TFQMR (Freund 1992),
- the Chebyshev acceleration, CHEBYSHEV (Young 1981).

The source code for these methods is distributed with `DDSCAT` but only `PBCGST` and `PETRKP` can be called directly via `ddscat.par`. It is possible (and was done) to add other options by changing the code in `getfml.f` . A helpful introduction to conjugate gradient methods is provided by the report "Conjugate Gradient Method Without Agonizing Pain" by Jonathan R. Shewchuk, available as a postscript file: `ftp://REPORTS.ADM.CS.CMU.EDU/usr0/anon/1994/CMU-CS-94-125.ps`.

14 Calculation of Radiative Force and Torque

In addition to solving the scattering problem for a dipole array, **DDSCAT.5a** can compute the three-dimensional force $\mathbf{F}_{rad}$ and torque $\mathbf{\Gamma}_{rad}$ exerted on this array by the incident and scattered radiation fields. This calculation is carried out, after solving the scattering problem, provided `DOTORQ` has been specified in `ddscat.par`. For each incident polarization mode, the results are given in terms of dimensionless efficiency vectors $\mathbf{Q}_{pr}$ and $\mathbf{Q}_{\Gamma}$, defined by

$$\mathbf{Q}_{pr} \equiv \frac{\mathbf{F}_{rad}}{\pi a_{\rm eff}^2 u_{rad}} \quad , \tag{7}$$

$$\mathbf{Q}_{\Gamma} \equiv \frac{k\mathbf{\Gamma}_{rad}}{\pi a_{\rm aeff}^2 u_{rad}} \quad , \tag{8}$$

where $\mathbf{F}_{rad}$ and $\mathbf{\Gamma}_{rad}$ are the time-averaged force and torque on the dipole array, $k = 2\pi/\lambda$ is the wavenumber *in vacuo*, and $u_{rad} = E_0^2/8\pi$ is the time-averaged energy density for an incident plane wave with amplitude $E_0\cos(\omega t + \phi)$. The radiation pressure efficiency vector can be written

$$\mathbf{Q}_{rad} = Q_{ext}\hat{\mathbf{k}} - Q_{sca}\mathbf{g} \quad , \tag{9}$$

where $\hat{\mathbf{k}}$ is the direction of propagation of the incident radiation, and the vector $\mathbf{g}$ is the mean direction of propagation of the scattered radiation:

$$\mathbf{g} = \frac{1}{C_{sca}} \int d\Omega \frac{dC_{sca}(\hat{\mathbf{n}},\hat{\mathbf{k}})}{d\Omega}\hat{\mathbf{n}} \quad , \tag{10}$$

where $d\Omega$ is the element of solid angle in scattering direction $\hat{\mathbf{n}}$, and $dC_{sca}/d\Omega$ is the differential scattering cross section.

Equations for the evaluation of the radiative force and torque are derived by Draine & Weingartner (1996). It is important to note that evaluation of $\mathbf{Q}_{pr}$ and $\mathbf{Q}_{\Gamma}$ involves averaging over scattering directions to evaluate the linear and angular momentum transport by the scattered wave. This evaluation requires appropriate choices of the parameters `ICTHM` and `IPHM` – see §21.

15 Memory Requirements

Since Fortran-77 does not allow for dynamic memory allocation, 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, the target must fit within an $8\times8\times8$=512 cube) and involve at most 9 different dielectric functions (`MXCOMP=9`). With this dimensioning, the executable requires about 1.3 MB of memory to run on an Ultrasparc system; memory requirements on other hardware and with other compilers should be similar. To run larger problems, you will need to edit the code 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. Set 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 for **DDSCAT.5a** (with the netCDF option disabled) is approximately (1059+0.623`MXNX`$\times$`MXNY`$\times$`MXNZ`) kbytes when compiled with `MXMEM=0`, or (1059+0.686`MXNX`$\times$ `MXNY` $\times$ `MXNZ`) kbytes when compiled with `MXMEM=1`. Thus with `MXMEM=0` a $32\times32\times32$ calculation requires 21.0 MBytes, while a $48\times48\times48$ calculation requires 68.4 MBytes. These values are for an Ultrasparc system using the Sun FORTRAN compiler.

16 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 $\hat{\mathbf{e}}_{01}$ (which obviously must lie in the y,z plane in the LF); **DDSCAT** automatically constructs a second polarization state $\hat{\mathbf{e}}_{02} = \hat{\mathbf{x}} \times \hat{\mathbf{e}}_{01}$ orthogonal to $\hat{\mathbf{e}}_{01}$ (here $\hat{\mathbf{x}}$ is the unit vector in the $+x$ direction of the LF. For purposes of discussion we will always let unit vectors $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, $\hat{\mathbf{z}} = \hat{\mathbf{x}} \times \hat{\mathbf{y}}$ be the three coordinate axes of the LF. Users will often find it convenient to let polarization vectors $\hat{\mathbf{e}}_{01} = \hat{\mathbf{y}}$, $\hat{\mathbf{e}}_{02} = \hat{\mathbf{z}}$ (although this is not mandatory – see §20).

Recall that definition of a target involves specifying two unit vectors, $\hat{\mathbf{a}}_1$ and $\hat{\mathbf{a}}_2$, which are imagined to be "frozen" into the target. We require $\hat{\mathbf{a}}_2$ to be orthogonal to $\hat{\mathbf{a}}_1$. Therefore we may define a "Target Frame" (TF) defined by the three unit vectors $\hat{\mathbf{a}}_1$, $\hat{\mathbf{a}}_2$, and $\hat{\mathbf{a}}_3 = \hat{\mathbf{a}}_1 \times \hat{\mathbf{a}}_2$.

For example, when **DDSCAT** creates a $8\times6\times4$ rectangular solid, it fixes $\hat{\mathbf{a}}_1$ to be along the longest dimension of the solid, and $\hat{\mathbf{a}}_2$ to be along the next-longest dimension.

Orientation of the target relative to the incident radiation can in principle be determined two ways:

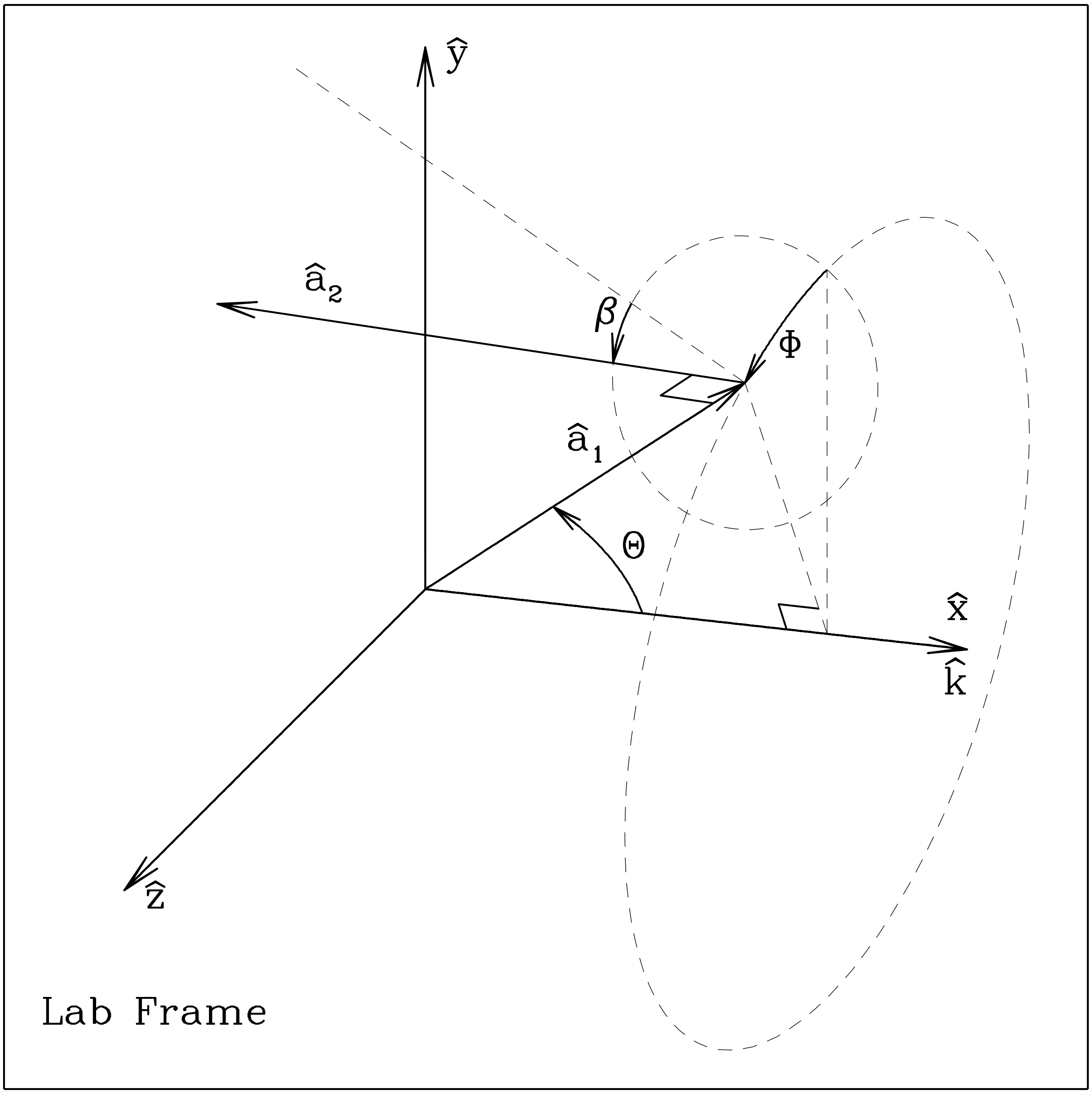

Figure 5: Target orientation in the Lab Frame. $\hat{\mathbf{x}}$ is the direction of propagation of the incident radiation, and $\hat{\mathbf{y}}$ is the direction of the "real" component of the first incident polarization mode. In this coordinate system, the orientation of target axis $\hat{\mathbf{a}}_1$ is specified by angles Θ and Φ. With target axis $\hat{\mathbf{a}}_1$ fixed, the orientation of target axis $\hat{\mathbf{a}}_2$ is then determined by angle β specifying rotation of the target around $\hat{\mathbf{a}}_1$. When $\beta = 0$, $\hat{\mathbf{a}}_2$ lies in the $\hat{\mathbf{a}}_1$,$\hat{\mathbf{x}}$ plane.

1. specifying the direction of $\hat{\mathbf{a}}_1$ and $\hat{\mathbf{a}}_2$ in the LF, or

2. specifying the directions of $\hat{\mathbf{x}}$ (incidence direction) and $\hat{\mathbf{y}}$ in the TF.

DDSCAT.5a uses method 1.: the angles Θ, Φ, and β are specified in the file `ddscat.par`. The target is oriented such that the polar angles Θ and Φ specify the direction of $\hat{\mathbf{a}}_1$ relative to the incident direction $\hat{\mathbf{x}}$, where the $\hat{\mathbf{x}}$,$\hat{\mathbf{y}}$ plane has $\Phi = 0$. Once the direction of $\hat{\mathbf{a}}_1$ is specified, the angle β than specifies how the target is to rotated around the axis $\hat{\mathbf{a}}_1$ to fully specify its orientation. A more extended and precise explanation follows:

16.1 Orientation of the Target in the Lab Frame

DDSCAT uses three angles, Θ, Φ, and β, to specify the directions of unit vectors $\hat{\mathbf{a}}_1$ and $\hat{\mathbf{a}}_2$ in the LF (see Fig. 5).

Θ is the angle between $\hat{\mathbf{a}}_1$ and $\hat{\mathbf{x}}$.

When $\Phi = 0$, $\hat{\mathbf{a}}_1$ will lie in the $\hat{\mathbf{x}}, \hat{\mathbf{y}}$ plane. When Φ is nonzero, it will refer to the rotation of $\hat{\mathbf{a}}_1$ around $\hat{\mathbf{x}}$: e.g., $\Phi = 90°$ puts $\hat{\mathbf{a}}_1$ in the $\hat{\mathbf{x}}, \hat{\mathbf{z}}$ plane.

When $\beta = 0$, $\hat{\mathbf{a}}_2$ will lie in the $\hat{\mathbf{x}}, \hat{\mathbf{a}}_1$ plane, in such a way that when $\Theta = 0$ and $\Phi = 0$, $\hat{\mathbf{a}}_2$ is in the $\hat{\mathbf{y}}$ direction: e.g, $\Theta = 90°$, $\Phi = 0$, $\beta = 0$ has $\hat{\mathbf{a}}_1 = \hat{\mathbf{y}}$ and $\hat{\mathbf{a}}_2 = -\hat{\mathbf{x}}$. Nonzero β introduces an additional rotation of $\hat{\mathbf{a}}_2$ around $\hat{\mathbf{a}}_1$: e.g., $\Theta = 90°$, $\Phi = 0$, $\beta = 90°$ has $\hat{\mathbf{a}}_1 = \hat{\mathbf{y}}$ and $\hat{\mathbf{a}}_2 = \hat{\mathbf{z}}$.

Mathematically:

$$\begin{aligned}
\hat{\mathbf{a}}_1 &= \hat{\mathbf{x}}\cos\Theta + \hat{\mathbf{y}}\sin\Theta\cos\Phi + \hat{\mathbf{z}}\sin\Theta\sin\Phi && (11)\\
\hat{\mathbf{a}}_2 &= -\hat{\mathbf{x}}\sin\Theta\cos\beta + \hat{\mathbf{y}}[\cos\Theta\cos\beta\cos\Phi - \sin\beta\sin\Phi] \\
&\quad +\hat{\mathbf{z}}[\cos\Theta\cos\beta\sin\Phi + \sin\beta\cos\Phi] && (12)\\
\hat{\mathbf{a}}_3 &= \hat{\mathbf{x}}\sin\Theta\sin\beta - \hat{\mathbf{y}}[\cos\Theta\sin\beta\cos\Phi + \cos\beta\sin\Phi] \\
&\quad -\hat{\mathbf{z}}[\cos\Theta\sin\beta\sin\Phi - \cos\beta\cos\Phi] && (13)
\end{aligned}$$

or, equivalently:

$$\begin{aligned}
\hat{\mathbf{x}} &= \hat{\mathbf{a}}_1\cos\Theta - \hat{\mathbf{a}}_2\sin\Theta\cos\beta + \hat{\mathbf{a}}_3\sin\Theta\sin\beta && (14)\\
\hat{\mathbf{y}} &= \hat{\mathbf{a}}_1\sin\Theta\cos\Phi + \hat{\mathbf{a}}_2[\cos\Theta\cos\beta\cos\Phi - \sin\beta\sin\Phi] \\
&\quad -\hat{\mathbf{a}}_3[\cos\Theta\sin\beta\cos\Phi + \cos\beta\sin\Phi] && (15)\\
\hat{\mathbf{z}} &= \hat{\mathbf{a}}_1\sin\Theta\sin\Phi + \hat{\mathbf{a}}_2[\cos\Theta\cos\beta\sin\Phi + \sin\beta\cos\Phi] \\
&\quad -\hat{\mathbf{a}}_3[\cos\Theta\sin\beta\sin\Phi - \cos\beta\cos\Phi] && (16)
\end{aligned}$$

16.2 Orientation of the Incident Beam in the Target Frame

Under some circumstances, one may wish to specify the target orientation such that $\hat{\mathbf{x}}$ (the direction of propagation of the radiation) and $\hat{\mathbf{y}}$ (usually the first polarization direction) and $\hat{\mathbf{z}}$ ($= \hat{\mathbf{x}} \times \hat{\mathbf{y}}$) 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:

1. $\hat{\mathbf{x}} = \hat{\mathbf{a}}_1$, $\hat{\mathbf{y}} = \hat{\mathbf{a}}_2$, $\hat{\mathbf{z}} = \hat{\mathbf{a}}_3$: $\Theta = 0$, $\Phi + \beta = 0$
2. $\hat{\mathbf{x}} = \hat{\mathbf{a}}_1$, $\hat{\mathbf{y}} = \hat{\mathbf{a}}_3$, $\hat{\mathbf{z}} = -\hat{\mathbf{a}}_2$: $\Theta = 0$, $\Phi + \beta = 90°$
3. $\hat{\mathbf{x}} = \hat{\mathbf{a}}_2$, $\hat{\mathbf{y}} = \hat{\mathbf{a}}_1$, $\hat{\mathbf{z}} = -\hat{\mathbf{a}}_3$: $\Theta = 90°$, $\beta = 180°$, $\Phi = 0$
4. $\hat{\mathbf{x}} = \hat{\mathbf{a}}_2$, $\hat{\mathbf{y}} = \hat{\mathbf{a}}_3$, $\hat{\mathbf{z}} = \hat{\mathbf{a}}_1$: $\Theta = 90°$, $\beta = 180°$, $\Phi = 90°$
5. $\hat{\mathbf{x}} = \hat{\mathbf{a}}_3$, $\hat{\mathbf{y}} = \hat{\mathbf{a}}_1$, $\hat{\mathbf{z}} = \hat{\mathbf{a}}_2$: $\Theta = 90°$, $\beta = -90°$, $\Phi = 0$
6. $\hat{\mathbf{x}} = \hat{\mathbf{a}}_3$, $\hat{\mathbf{y}} = \hat{\mathbf{a}}_2$, $\hat{\mathbf{z}} = -\hat{\mathbf{a}}_1$: $\Theta = 90°$, $\beta = -90°$, $\Phi = -90°$
7. $\hat{\mathbf{x}} = -\hat{\mathbf{a}}_1$, $\hat{\mathbf{y}} = \hat{\mathbf{a}}_2$, $\hat{\mathbf{z}} = -\hat{\mathbf{a}}_3$: $\Theta = 180°$, $\beta + \Phi = 180°$
8. $\hat{\mathbf{x}} = -\hat{\mathbf{a}}_1$, $\hat{\mathbf{y}} = \hat{\mathbf{a}}_3$, $\hat{\mathbf{z}} = \hat{\mathbf{a}}_2$: $\Theta = 180°$, $\beta + \Phi = 90°$
9. $\hat{\mathbf{x}} = -\hat{\mathbf{a}}_2$, $\hat{\mathbf{y}} = \hat{\mathbf{a}}_1$, $\hat{\mathbf{z}} = \hat{\mathbf{a}}_3$: $\Theta = 90°$, $\beta = 0$, $\Phi = 0$
10. $\hat{\mathbf{x}} = -\hat{\mathbf{a}}_2$, $\hat{\mathbf{y}} = \hat{\mathbf{a}}_3$, $\hat{\mathbf{z}} = -\hat{\mathbf{a}}_1$: $\Theta = 90°$, $\beta = 0$, $\Phi = -90°$
11. $\hat{\mathbf{x}} = -\hat{\mathbf{a}}_3$, $\hat{\mathbf{y}} = \hat{\mathbf{a}}_1$, $\hat{\mathbf{z}} = -\hat{\mathbf{a}}_2$: $\Theta = 90°$, $\beta = -90°$, $\Phi = 0$
12. $\hat{\mathbf{x}} = -\hat{\mathbf{a}}_3$, $\hat{\mathbf{y}} = \hat{\mathbf{a}}_2$, $\hat{\mathbf{z}} = \hat{\mathbf{a}}_1$: $\Theta = 90°$, $\beta = -90°$, $\Phi = 90°$
13. $\hat{\mathbf{x}} = (\hat{\mathbf{a}}_1 + \hat{\mathbf{a}}_2)/\sqrt{2}$, $\hat{\mathbf{y}} = \hat{\mathbf{a}}_3$, $\hat{\mathbf{z}} = (\hat{\mathbf{a}}_1 - \hat{\mathbf{a}}_2)/\sqrt{2}$: $\Theta = 45°$, $\beta = 180°$, $\Phi = 90°$
14. $\hat{\mathbf{x}} = (\hat{\mathbf{a}}_1 + \hat{\mathbf{a}}_2 + \hat{\mathbf{a}}_3)/\sqrt{3}$, $\hat{\mathbf{y}} = (\hat{\mathbf{a}}_1 - \hat{\mathbf{a}}_2)/\sqrt{2}$, $\hat{\mathbf{z}} = (\hat{\mathbf{a}}_1 + \hat{\mathbf{a}}_2 - 2\hat{\mathbf{a}}_3)/\sqrt{6}$: $\Theta = 54.7356°$, $\beta = 135°$, $\Phi = 30°$.

16.3 Sampling in Θ, Φ, and β

The present version, **DDSCAT.5a**, chooses the angles β, Θ, and Φ 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 β;
- uniformly sample in Φ;
- uniformly sample in $\cos\Theta$.

This prescription is appropriate for random orientation of the target, within the specified limits of β, Φ, and Θ.

Note that when **DDSCAT.5a** chooses angles it handles β and Φ differently from Θ.[10] The range for β 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 get $\Phi = 45°$ and $135°$.

Sampling in Θ is done quite differently from sampling in β and Φ. First, as already mentioned above, **DDSCAT.5a** samples uniformly in $\cos\Theta$, not Θ. Secondly, the sampling depends on whether `NTHETA` is even or odd.

[10]This is a change from **DDSCAT.4a**!!.

- If NTHETA is odd, then the values of Θ selected include the extreme values THETMI and THETMX; thus, THETMI=0, THETMX=90, NTHETA=3 will give you $\Theta = 0, 60^\circ, 90^\circ$.
- If NTHETA is even, 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 $\Theta = 41.41^\circ$ 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 Θ values.

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

17 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.

17.1 Randomly-Oriented Targets

For randomly-oriented targets, we wish to compute the orientational average of a quantity $Q(\beta,\Theta,\Phi)$:

$$\langle Q\rangle = \frac{1}{8\pi^2}\int_0^{2\pi} d\beta \int_{-1}^{1} d\cos\Theta \int_0^{2\pi} d\Phi \; Q(\beta,\Theta,\Phi) \quad . \tag{17}$$

To compute such averages, all you need to do is edit the file `ddscat.par` so that DDSCAT knows what ranges of the angles β, Θ, and Φ are of interest. For a randomly-oriented target with no symmetry, you would need to let β run from 0 to 360°, Θ from 0 to 180°, and Φ from 0 to 360°.

For targets with symmetry, on the other hand, the ranges of β, Θ, and Φ may be reduced. First of all, remember that averaging over Φ is relatively "inexpensive", so when in doubt average over 0 to 360°; most of the computational "cost" is associated with the number of different values of (β,Θ) which are used. Consider a cube, for example, with axis $\hat{\mathbf{a}}_1$ normal to one of the cube faces; for this cube β need run only from 0 to 90°, since the cube has fourfold symmetry for rotations around the axis $\hat{\mathbf{a}}_1$. Furthermore, the angle Θ need run only from 0 to 90°, since the orientation (β,Θ,Φ) is indistinguishable from (β, $180^\circ - \Theta$, $360^\circ - \Phi$).

For targets with symmetry, the user is encouraged to test the significance of β,Θ,Φ 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.4b** handled even and odd values of NTHETA differently – see §7 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 $\cos\Theta$.

17.2 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.5a** with appropriate choices of input parameters. More generally, however, you will need to modify subroutine `ORIENT` to generate a list of `NBETA` values of β, `NTHETA` values of Θ, and `NPHI` values of Φ, plus two weighting arrays `WGTA(1-NTHETA,1-NPHI)` and `WGTB(1-NBETA)`. Here `WGTA` gives the weights which should be attached to each (Θ,Φ) orientation, and `WGTB` gives the weight to be attached to each β orientation. Thus each orientation of the target is to be weighted by the factor `WGTA`$\times$`WGTB`. For the case of random orientations, **DDSCAT.5a** chooses Θ values which are uniformly spaced in $\cos\Theta$, and β and Φ values which are uniformly spaced, and therefore uses uniform weights

`WGTB`=1./`NBETA`

When `NTHETA` is even, **DDSCAT** sets

`WGTA`=1./(`NTHETA`$\times$`NPHI`)

but when `NTHETA` is odd, **DDSCAT** uses Simpson's rule when integrating over Θ and

`WGTA`= (1/3 or 4/3 or 2/3)/(`NTHETA`$\times$`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 Φ value only one Θ 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 Θ and Φ which the present version of **DDSCAT** is set up to do. We hope to improve this in a future version of **DDSCAT**.

18 Target Generation

DDSCAT contains routines to generate dipole 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` on line 7 of `ddscat.par`, and up to 6 parameters (`SHPAR1`, `SHPAR2`, `SHPAR3`, ...) on line 8. The target geometry is most conveniently described in a coordinate system attached to the target which we refer to as the "Target Frame" (TF), so in this section *only* we will let x,y,z be coordinates in the Target Frame. Once the target is generated, the orientation of the target in the Lab Frame is accomplished as described in §16.

Target geometries currently supported include:

- `ELLIPS` – Homogeneous ellipsoid with "lengths" `SHPAR1`, `SHPAR2`, `SHPAR3` in the x, y, z directions in the TF. $(x/\texttt{SHPAR1})^2 + (y/\texttt{SHPAR2})^2 + (z/\texttt{SHPAR3})^2 = d^2/4$, where d is the interdipole spacing. A sphere is obtained by setting `SHPAR1` = `SHPAR2` = `SHPAR3` = diameter/d. The target axes are set to $\hat{\mathbf{a}}_1 = (1,0,0)$ and $\hat{\mathbf{a}}_2 = (0,1,0)$ in the TF. The user must set `NCOMP`=1 on line 9 of `ddscat.par`.

- `CYLNDR` – Homogeneous cylinder with length/d=`SHPAR1`, diameter/d=`SHPAR2`, with cylinder axis=$\hat{\mathbf{a}}_1 = (1,0,0)$ and $\hat{\mathbf{a}}_2 = (0,1,0)$ in the TF. User must set `NCOMP`=1.

- `RCTNGL` – Homogeneous rectangular solid with x, y, z lengths/d = `SHPAR1`, `SHPAR2`, `SHPAR3`. Target axes $\hat{\mathbf{a}}_1 = (1,0,0)$ and $\hat{\mathbf{a}}_2 = (0,1,0)$ in the TF. User must set `NCOMP`=1.

- `HEXGON` – Homogeneous hexagonal prism with length/d = `SHPAR1`, hexagon side/d = `SHPAR2`, with prism axis $\hat{\mathbf{a}}_1 = (1,0,0)$ in the TF, and prism axis $\hat{\mathbf{a}}_2 = (0,1,0)$ in the TF normal to one of the rectangular sides of the hexagonal prism. User must set `NCOMP`=1.

- `TETRAH` – Homogeneous tetrahedron with `SHPAR1`=length/d of one edge, one face parallel to y,z plane (in the TF), opposite "vertex" in $+x$ direction, and one edge parallel to z axis (in the TF). Target axes $\hat{\mathbf{a}}_1 = (1,0,0)$ [emerging from one vertex] and $\hat{\mathbf{a}}_2 = (0,1,0)$ [emerging from an edge] in the TF. User must set `NCOMP`=1.

- `TWOELL` – Two touching ellipsoids with same orientation. `SHPAR1`, `SHPAR2`, `SHPAR3`=x-length/d, y-length/d, z-length/d of one ellipsoid. The two ellipsoids have identical shape and orientation, but distinct dielectric functions. The line connecting ellipsoid centers is along the x-axis in the TF. Target axes $\hat{\mathbf{a}}_1 = (1,0,0)$ [along line connecting ellipsoids] and $\hat{\mathbf{a}}_2 = (0,1,0)$. User must set `NCOMP`=2 and provide dielectric function file names for both ellipsoids. Ellipsoids are in order of increasing x.

- `TWOAEL` – Two touching ellipsoids as for `TWOELL`, but with anisotropic dielectric functions. `SHPAR1`, `SHPAR2`, `SHPAR3` have same meanings as for `TWOELL`. Target axes $\hat{\mathbf{a}}_1 = (1,0,0)$ and $\hat{\mathbf{a}}_2 = (0,1,0)$ in the TF. User 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).

- `ANIELL` – Anisotropic ellipsoid. `SHPAR1`, `SHPAR2`, `SHPAR3` have same meaning as for `ELLIPS`, but with dielectric functions 1,2,3 for xx, yy, zz elements of the (diagonal) dielectric tensor. Target axes $\hat{\mathbf{a}}_1 = (1,0,0)$ and $\hat{\mathbf{a}}_2 = (0,1,0)$ in the TF. User must set `NCOMP`=3 and provide xx, yy, zz elements of the (diagonal)dielectric tensor.

- `UNICYL` – Uniaxial cylinder. `SHPAR1`, `SHPAR2` have same meaning as for `CYLNDR`, but now diel. func. 1 is for **E** parallel to cylinder axis, diel.func. 2 for **E** perp. to axis. Cylinder axis $=\hat{\mathbf{a}}_1 = (1,0,0)$, $\hat{\mathbf{a}}_2 = (0,1,0)$. User must set `NCOMP`=2 and provide parallel, perpendicular dielectric functions.

- `THRELL` – Three touching ellipsoids of equal size and orientation `SHPAR1`, `SHPAR2`, `SHPAR3` have same meaning as for `TWOELL`. Line connecting ellipsoid centers is parallel to x axis. Target axis $\hat{\mathbf{a}}_1 = (1,0,0)$ along line of ellipsoid centers, $\hat{\mathbf{a}}_2 = (0,1,0)$. User 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. `SHPAR1`, `SHPAR2`, `SHPAR3` have same meanings as for `THRELL`. Target axis $\hat{\mathbf{a}}_1 = (1,0,0)$ along line of ellipsoid centers, $\hat{\mathbf{a}}_2 = (0,1,0)$. User must set `NCOMP`=9 and provide x, y, z dielectric function for first ellipsoid, x, y, z diel. func. for second ellipsoid, and x, y, z diel. func. for third ellipsoid (ellipsoids are in order of increasing x).

- `CONELL` – Two concentric ellipsoids, `SHPAR1`, `SHPAR2`, `SHPAR3` specify the lengths along x, y, z axes (in the TF) of the *outer* ellipsoid; `SHPAR4`, `SHPAR5`, `SHPAR6` are the lengths, along the x, y, z axes (in the TF) of the *inner* ellipsoid. The "core" within the inner ellipsoid is composed of material 1; the "mantle" between inner and outer ellipsoids is composed of material 2. Target axes $\hat{\mathbf{a}}_1 = (1,0,0)$, $\hat{\mathbf{a}}_2 = (0,1,0)$ in TF. User must set NCOMP=2 and provide delectric functions for "core" and "mantle".

- TWOSPH – Two touching spheroids. First spheroid has length SHPAR1 along symmetry axis, diameter SHPAR2 perpendicular to symmetry axis. Second spheroid has length SHPAR3 along symmetry axis, diameter SHPAR4 perpendicular to symmetry axis. Contact point is on line connecting centroids. Line connecting centroids is in x direction. Symmetry axis of first spheroid is in y direction. Symmetry axis of second spheroid is in direction $\hat{\mathbf{y}}\cos(\mathtt{SHPAR5}) + \hat{\mathbf{z}}\sin(\mathtt{SHPAR5})$, where $\hat{\mathbf{y}}$ and $\hat{\mathbf{z}}$ are basis vectors in TF, and SHPAR5 is in degrees. If SHPAR6=0., then target axes $\hat{\mathbf{a}}_1 = (1,0,0)$, $\hat{\mathbf{a}}2 = (0,1,0)$. If SHPAR6=1., then axes $\hat{\mathbf{a}}_1$ and $\hat{\mathbf{a}}_2$ are set to principal axes with largest and 2nd largest moments of inertia assuming spheroids to be of uniform density. User must set NCOMP=2 and provide dielectric function files for each spheroid.

- BLOCKS – Homogeneous target constructed from cubic "blocks". Number and location of blocks are specified in separate file blocks.par with following structure:
  - one line of comments
  - PRIN = 0 or 1 (see below)
  - N = number of blocks
  - B = width/d of one block,
  - $x\ y\ z$ = position of 1st block (in units of Bd)
  - $x\ y\ z$ = position of 2nd block ...
  - ...
  - $x\ y\ z$ = position of Nth block ...

  If PRIN=0, then $\hat{\mathbf{a}}_1 = (1,0,0)$, $\hat{\mathbf{a}}_2 = (0,1,0)$. If PRIN=1, then $\hat{\mathbf{a}}_1$ and $\hat{\mathbf{a}}_2$ are set to principal axes with largest and second largest moments of inertia, assuming target to be of uniform density. User must set NCOMP=1.

- DW1996 – Target constructed from 13 blocks used by Draine & Weingartner (1996) in study of radiative torques on irregular grains. $\hat{\mathbf{a}}_1$ and $\hat{\mathbf{a}}_2$ are principal axes with largest and second-largest moments of inertia. User must set NCOMP=1.

- FRMFIL – This option causes **DDSCAT** to read the target geometry information from a file shape.dat instead of automatically generating one of the geometries listed above. The shape.dat file is read by routine REASHP (file reashp.f). The user can customize REASHP as needed to conform to the manner in which the target geometry is stored in file shape.dat.

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.f.

It is often desirable to be able to run the target generation routines without running the entire **DDSCAT** code. We have therefore provided a program CALLTARGET which allows the user to generate targets interactively; to create this executable just type[11]

```
make calltarget .
```

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 §5.1 above).

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.

19 Scattering Directions

DDSCAT calculates scattering in selected directions, and elements of the scattering matrix are reported in the output files w*xx*r*yy*k*zzz*.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 $\hat{\mathbf{x}}$) 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" relative to the $\hat{\mathbf{x}}, \hat{\mathbf{y}}$ plane in the Lab Frame. When $\phi_s = 0$ the scattering plane is assumed to coincide with the $\hat{\mathbf{x}}, \hat{\mathbf{y}}$ plane. When $\phi_s = 90°$ the scattering plane is assumed to coincide with the $\hat{\mathbf{x}}, \hat{\mathbf{z}}$ plane. Within the scattering plane the scattering directions are specified by $0 \leq \theta_s \leq 180°$.

Scattering directions for which the scattering properties are to be calculated are set in the file `ddscat.par` by specifying one or more scattering planes (determined by the value of ϕ_s) and for each scattering plane, the number and range of θ_s values. The only limitation is that the number of scattering directions not exceed the parameter `MXSCA` in `DDSCAT.f` (in the code as distributed it is set to `MXSCA=1000`).

20 Incident Polarization State

Recall that the "Lab Frame" is defined such that the incident radiation is propagating along the $\hat{\mathbf{x}}$ axis. **DDSCAT.5a** allows the user to specify a general elliptical polarization for the incident radiation, by specifying the (complex) polarization vector $\hat{\mathbf{e}}_{01}$. The orthonormal polarization state $\hat{\mathbf{e}}_{02} = \hat{\mathbf{x}} \times \hat{\mathbf{e}}_{01}$ is generated automatically if `ddscat.par` specifies `IORTH=2`.

For incident linear polarization, one can simply set $\hat{\mathbf{e}}_{01} = \hat{\mathbf{y}}$ by specifying `(0,0) (1,0) (0,0)` in `ddscat.par`; then $\hat{\mathbf{e}}_{02} = \hat{\mathbf{z}}$ For polarization mode $\hat{\mathbf{e}}_{01}$ to correspond to right-handed circular polarization, set $\hat{\mathbf{e}}_{01} = (\hat{\mathbf{y}} + i\hat{\mathbf{z}})/\sqrt{2}$ by specifying `(0,0) (1,0) (0,1)` in `ddscat.par` (**DDSCAT.5a** automatically takes care of the normalization of $\hat{\mathbf{e}}_{01}$); then $\hat{\mathbf{e}}_{02} = (-i\hat{\mathbf{y}} + \hat{\mathbf{z}})/\sqrt{2}$, corresponding to left-handed circular polarization.

[11]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`, `TAR2SP`, `TAR3EL`, `TARBLOCKS`, `TARCEL`, `TARCYL`, `TARELL`, `TARGET`, `TARHEX`, `TARREC`, `TARTET`, and `WRIMSG`. The source code for `ERRMSG` is in `SRC2.FOR`, `GETSET` is in `SRC5.FOR`, and the rest are in `SRC8.FOR` and `SRC9.FOR` .

21 Averaging over Scattering: $g(1) = \langle \cos\theta_s \rangle$, etc.

DDSCAT automatically carries out numerical integration of various scattering properties. In particular, it calculates the mean value $g(1) = \langle \cos\theta_s \rangle$ for the scattered intensity for each incident polarization state. This is accomplished by evaluating the scattered intensity for `ICTHM` different values of θ_s (including $\theta_s = 0$ and $\theta_s = \pi$), and taking a weighted sum. For each value of θ_s except 0 and π, the scattering intensity is evaluated for `IPHM` different values of the scattering angle ϕ_s. The angular integration over θ_s is accomplished using Simpson's rule (with uniform intervals in $\cos\theta_s$), so `ICTHM` should be an *odd* number. The angular integration over ϕ_s is accomplished by sampling uniformly in ϕ_s with uniform weighting; `IPHM` can be either even or odd.

The following quantities are evaluated by this angular integration:

- $\mathbf{g} = \langle \cos\theta_s \rangle \hat{\mathbf{x}} + \langle \sin\theta_s \cos\phi_s \rangle \hat{\mathbf{y}} + \langle \sin\theta_s \sin\phi_s \rangle \hat{\mathbf{z}}$ (see §14);
- $\mathbf{Q}_\Gamma$ (see §14).

It is important that the user recognize that accurate evaluation of these angular averages requires adequate sampling over scattering angles. For small values of the size parameter $x = 2\pi a_{\rm eff}/\lambda$, the angular distribution of scattered radiation has a dipolar character and the sampling in θ_s and ϕ_s does not need to be very fine, so `ICTHM` and `IPHM` need not be large. For larger values of the size parameter x, however, higher multipoles in the scattered radiation field become important, and finer sampling in θ_s and ϕ_s is required. We do not have any foolproof prescription to offer, since the scattering pattern will depend upon the target geometry and dielectric constant in addition to overall size parameter. However, as a very rough guide, we suggest that the user specify values of `ICTHM` and `IPHM` satisfying

$$\begin{aligned} \texttt{ICTHM} &> 5(1+x) \quad , \qquad (18) \\ \texttt{IPHM} &> 2(1+x) \quad . \qquad (19) \end{aligned}$$

The sample `ddscat.par` file supplied has `ICTHM` $=$ `33` and `IPHM` $= 12$; the above criteria would suggest that this would be suitable for $x < 5$.

The cpu time required for evaluation of these angular averages is proportional to [2+`IPHM`(`ICTHM`$-$2)]. Since the computational time spent in evaluating these angular integrals can be a significant part of the total, it is important to choose values of `ICTHM` and `IPHM` which will provide a suitable balance between accuracy (in this part of the overall calculation) and cpu time.

Within one scattering plane, the scattered intensity tends to have approximately $(1+x)$ peaks for $0 \leq \theta_s \leq \pi$, so that the above prescription for `ICTHM` would have at least 5 sampling points per maximum. The angular distribution over ϕ_s is usually not as structured as that over θ_s so we suggest that `IPHM` need not be as large as `ICTHM`. We have refrained from "hard-wiring" the values of `ICTHM` and `IPHM` because we are not confident of the reliability of the recommended criteria (18,19) – it is up to the user to specify appropriate values of `ICTHM` and `IPHM` according to the requirements of the problem being addressed.

22 Mueller Matrix for Scattering in Selected Directions

22.1 Two Orthogonal Incident Polarizations (IORTH=2)

DDSCAT.5a internally computes the scattering properties of the dipole array in terms of a complex scattering matrix $f_{ml}(\theta_s,\phi_s)$ (Draine 1988), where index $l = 1,2$ denotes the incident polarization state, $m = 1,2$ denotes the scattered polarization state, and θ_s,ϕ_s specify the scattering direction. Normally **DDSCAT** is used with `IORTH=2` in `ddscat.par`, so that the scattering problem will be solved for both incident polarization states ($l = 1$ and 2); in this subsection it will be assumed that this is the case.

Incident polarization states $l = 1,2$ correspond to polarization states $\hat{\mathbf{e}}_{01}$, $\hat{\mathbf{e}}_{02}$. Scattered polarization state $m = 1$ corresponds to linear polarization of the scattered wave parallel to the scattering plane ($\hat{\mathbf{e}}_1 = \hat{\mathbf{e}}_{\| s} = \hat{\theta}_s$) and $m = 2$ corresponds to linear polarization perpendicular to the scattering plane (in the $+\hat{\phi}_s$ direction). The scattered electric field $\mathbf{E}_s$ is related to the incident electric field $\mathbf{E}_i(0)$ at the origin (where the target is assumed to be located) by

$$\begin{pmatrix} \mathbf{E}_s\cdot\hat{\theta}_s \\ \mathbf{E}_s\cdot\hat{\phi}_s \end{pmatrix} = \frac{\exp(i\mathbf{k}\cdot\mathbf{r})}{kr} \begin{pmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{pmatrix} \begin{pmatrix} \mathbf{E}_i(0)\cdot\hat{\mathbf{e}}_{01} \\ \mathbf{E}_i(0)\cdot\hat{\mathbf{e}}_{02} \end{pmatrix} \quad . \tag{20}$$

The 2×2 complex *amplitude scattering matrix* (with elements S_1, S_2, S_3, and S_4) is defined so that (see Bohren & Huffman 1983)

$$\begin{pmatrix} \mathbf{E}_s\cdot\hat{\theta}_s \\ -\mathbf{E}_s\cdot\hat{\phi}_s \end{pmatrix} = \frac{\exp(i\mathbf{k}\cdot\mathbf{r})}{-ikr} \begin{pmatrix} S_2 & S_3 \\ S_4 & S_1 \end{pmatrix} \begin{pmatrix} \mathbf{E}_i(0)\cdot\hat{\mathbf{e}}_{i\|} \\ \mathbf{E}_i(0)\cdot\hat{\mathbf{e}}_{i\perp} \end{pmatrix} \quad , \tag{21}$$

where $\hat{\mathbf{e}}_{i\|}$, $\hat{\mathbf{e}}_{i\perp}$ are unit vectors for incident polarization parallel and perpendicular to the scattering plane (with the customary definition of $\hat{\mathbf{e}}_{i\perp} = \hat{\mathbf{e}}_{i\|}\times\hat{\mathbf{x}}$).

The amplitude scattering matrix elements can be obtained from the f_{ij}:

$$\begin{pmatrix} S_2 & S_3 \\ S_4 & S_1 \end{pmatrix} = \frac{-i}{ad-bc} \begin{pmatrix} f_{11} & f_{12} \\ -f_{21} & -f_{22} \end{pmatrix} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix} \begin{pmatrix} \cos\phi_s & \sin\phi_s \\ \sin\phi_s & -\cos\phi_s \end{pmatrix} \quad , \tag{22}$$

where

$$a \equiv \hat{\mathbf{e}}_{01}\cdot\hat{\mathbf{y}} \quad , \tag{23}$$
$$b \equiv \hat{\mathbf{e}}_{01}\cdot\hat{\mathbf{z}} \quad , \tag{24}$$
$$c \equiv \hat{\mathbf{e}}_{02}\cdot\hat{\mathbf{y}} \quad , \tag{25}$$
$$d \equiv \hat{\mathbf{e}}_{02}\cdot\hat{\mathbf{z}} \quad . \tag{26}$$

It is convenient to describe the scattering properties in terms of the Mueller matrix relating the Stokes parameters (I_i,Q_i,U_i,V_i) and (I_s,Q_s,U_s,V_s) of the incident and scattered radiation:

$$\begin{pmatrix} I_s \\ Q_s \\ U_s \\ V_s \end{pmatrix} = \frac{1}{k^2r^2} \begin{pmatrix} S_{11} & S_{12} & S_{13} & S_{14} \\ S_{21} & S_{22} & S_{23} & S_{24} \\ S_{31} & S_{32} & S_{33} & S_{34} \\ S_{41} & S_{42} & S_{43} & S_{44} \end{pmatrix} \begin{pmatrix} I_i \\ Q_i \\ U_i \\ V_i \end{pmatrix} \quad . \tag{27}$$

Once the amplitude scattering matrix elements are obtained, the Mueller matrix elements can be computed (Bohren & Huffman 1983):

$$
\begin{aligned}
S_{11} &= \left(|S_1|^2 + |S_2|^2 + |S_3|^2 + |S_4|^2\right)/2 \quad , \\
S_{12} &= \left(|S_2|^2 - |S_1|^2 + |S_4|^2 - |S_3|^2\right)/2 \quad , \\
S_{13} &= \mathrm{Re}\left(S_2 S_3^* + S_1 S_4^*\right) \quad , \\
S_{14} &= \mathrm{Im}\left(S_2 S_3^* - S_1 S_4^*\right) \quad , \\
S_{21} &= \left(|S_2|^2 - |S_1|^2 + |S_3|^2 - |S_4|^2\right)/2 \quad , \\
S_{22} &= \left(|S_1|^2 + |S_2|^2 - |S_3|^2 - |S_4|^2\right)/2 \quad , \\
S_{23} &= \mathrm{Re}\left(S_2 S_3^* - S_1 S_4^*\right) \quad , \\
S_{24} &= \mathrm{Im}\left(S_2 S_3^* + S_1 S_4^*\right) \quad , \\
S_{31} &= \mathrm{Re}\left(S_2 S_4^* + S_1 S_3^*\right) \quad , \\
S_{32} &= \mathrm{Re}\left(S_2 S_4^* - S_1 S_3^*\right) \quad , \\
S_{33} &= \mathrm{Re}\left(S_1 S_2^* + S_3 S_4^*\right) \quad , \\
S_{34} &= \mathrm{Im}\left(S_2 S_1^* + S_4 S_3^*\right) \quad , \\
S_{41} &= \mathrm{Im}\left(S_4 S_2^* + S_1 S_3^*\right) \quad , \\
S_{42} &= \mathrm{Im}\left(S_4 S_2^* - S_1 S_3^*\right) \quad , \\
S_{43} &= \mathrm{Im}\left(S_1 S_2^* - S_3 S_4^*\right) \quad , \\
S_{44} &= \mathrm{Re}\left(S_1 S_2^* - S_3 S_4^*\right) \quad .
\end{aligned} \tag{28}
$$

These matrix elements are computed in `DDSCAT` and passed to subroutine `WRITESCA` which handles output of scattering properties. As delivered, `WRITESCA` writes out 6 selected elements: S_{11}, S_{21}, S_{31}, S_{41} (these 4 elements describe the intensity and polarization state for scattering of unpolarized incident radiation), S_{12}, and S_{13}. In addition, `WRITESCA` writes out the linear polarization P of the scattered light for incident unpolarized light:

$$P = \frac{(S_{21}^2 + S_{31}^2)^{1/2}}{S_{11}} \quad . \tag{29}$$

Of course, other elements S_{ij} may be of interest. It is relatively straightforward for the user to modify subroutine `WRITESCA` to write out whatever elements of the Mueller matrix (or the scattering amplitude matrix) are desired.

22.2 One Incident Polarization State Only (`IORTH=1`)

In some cases it may be desirable to limit the calculations to a single incident polarization state – for example, when each solution is very time-consuming, and the target is known to have some symmetry so that solving for a single incident polarization state may be sufficient for the required purpose. In this case, set `IORTH=1` in `ddscat.par`.

When `IORTH=1`, only f_{11} and f_{21} are available; hence, **DDSCAT** cannot automatically generate the Mueller matrix elements. In this case, the output routine `WRITESCA` writes out the quantities $|f_{11}|^2$, $|f_{21}|^2$, $\mathrm{Re}(f_{11}f_{21}^*)$, and $\mathrm{Im}(f_{11}f_{21}^*)$ for each of the scattering directions.

Note, however, that if IPHI is greater than 1, **DDSCAT** will automatically set IORTH=2 even if ddscat.par specified IORTH=1: this is because when more than one value of the target orientation angle Φ is required, there is no additional "cost" to solve the scattering problem for the second incident polarization state, since when solutions are available for two orthogonal states for some particular target orientation, the solution may be obtained for another target orientation differing only in the value of Φ by appropriate linear combinations of these solutions. Hence we may as well solve the "complete" scattering problem so that we can compute the complete Mueller matrix.

23 Graphics and Postprocessing

23.1 IDL

At present, we do not offer a comprehensive package for **DDSCAT** data postprocessing and graphical display in IDL. However, there are several developments worth mentioning: First, we offer several output capabilities from within **DDSCAT**: ASCII (see §9.1), FORTRAN unformatted binary (see §9.2), and netCDF portable binary (see §9.3). Second, we offer several skeleton IDL utilities:

- bhmie.pro is our translation to IDL of the popular Bohren-Huffman code which calculates efficiencies for spherical particles using Mie theory.
- readbin.pro reads the FORTRAN unformatted binary file written by routine writebin.f. The variables are stored in a common block.
- readnet.pro reads NetCDF portable binary file and should be the method of choice for IDL users. It offers random data access.
- mie.pro is an example of an interface to binary files and the Bohren-Huffman code. It plots a comparison of **DDSCAT** results with scattering by equivalent radius spheres.

At present the IDL code is experimental.

24 Miscellanea

Additional source code, refractive index files, etc., contributed by users will be located in the directory DDA/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.

25 Finale

This User Guide is somewhat inelegant, but we hope that it will prove useful. The structure of the ddscat.par file is intended to be simple and suggestive so 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.5a** . 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 (`draine@astro.princeton.edu`) with their email address; bug reports, and any new releases of **DDSCAT**, will be made known to those who do!

P. J. Flatau maintains the WWW page "SCATTERLIB - Light Scattering Codes Library" with URL `http://atol.ucsd.edu/~pflatau`. The SCATTERLIB Internet site is a library of light scattering codes. Emphasis is on providing source codes (mostly FORTRAN). However, other information related to scattering on spherical and non-spherical particles is collected: an extensive list of references to light scattering methods, refractive index, etc. This URL page contains section on the discrete dipole approximation.

Concrete suggestions for improving **DDSCAT** (and this User Guide) are welcomed. If you wish to cite this User Guide, we suggest the following citation:

Draine, B.T., & Flatau, P.J. 1996, "User Guide for the Discrete Dipole Approximation Code DDSCAT (Version 5a)"

Princeton Observatory Preprint POPe-695

(URL `ftp://astro.princeton.edu/draine/scat/ddscat/ver5a/UserGuide.ps.gz`).

Finally, the authors have one special request: We would very much appreciate preprints and (especially!) reprints of any papers which make use of **DDSCAT**!

26 Acknowledgments

The routine `ESELF` making use of the FFT was originally written by Jeremy Goodman, Princeton University Observatory. The FFT routine `FOURX` is based on a FFT routine written by Norman Brenner (Brenner 1969). The routine `REFICE` was written by Steven B. Warren, based on Warren (1984). The routine `REFWAT` was written by Eric A. Smith. The `GPFAPACK` package was written by Clive Temperton, and generously made available by him for use with `DDSCAT`. Subroutine `CXFFT3` is based on a FFT routine written by Clive Temperton (Temperton 1983). We make use of routines from the LAPACK package (Anderson *et al.* 1995), the result of work by Jack Dongarra and others at the Univ. of Tennessee, Univ. of California Berkeley, NAG Ltd., Courant Institute, Argonne National Lab, and Rice University. 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, AST-9319283 to BTD, in part by support by the Office of Naval Research Young Investigator Program to PJF, and in part by DuPont Corporate Educational Assistance to PJF.

References

[1] Anderson, E., *et al.* 1995, *LAPACK Users' Guide*, (Philadelphia, SIAM).

[2] Bohren, C.F., and Huffman, D.R. 1983, *Absorption and Scattering of Light by Small Particles* (New York: Wiley-Interscience).

[3] Brenner, N.M. 1969, *IEEE Trans. Audio Electroacoust.* **AU-17**, 128.

[4] Draine, B.T. 1988, "The Discrete-Dipole Approximation and Its Application to Interstellar Graphite Grains", *Astrophysical J.*, **333**, 848-872.

[5] Draine, B.T., and Flatau, P.J. 1994, "Discrete-dipole approximation for scattering calculations", *J. Opt. Soc. Am. A*, **11**, 1491-1499.

[6] Draine, B.T., and Goodman, J.J. 1993, "Beyond Clausius-Mossotti: Wave Propagation on a Polarizable Point Lattice and the Discrete Dipole Approximation", *Astrophysical J.*, **405**, 685-697.

[7] Draine, B.T., and Weingartner, J.C. 1996, "Radiative Torques on Interstellar Grains: I. Superthermal Rotation", *Astrophysical J.*, **470**, 551-565.

[8] Draine, B.T., and Weingartner, J.C. 1997, "Radiative Torques on Interstellar Grains: II. Alignment with the Magnetic Field", *Astrophysical J.*, in press.

[9] Flatau, P.J. 1997, "Improvements of the Discrete Dipole Approximation Method", preprint.

[10] Flatau, P.J., Stephens, G.L., and Draine, B.T. 1990, "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.

[11] Goedecke, G.H., and O'Brien, S.G. 1988, "Scattering by irregular inhomogeneous particles via the digitized Green's function algorithm", *Appl. Opt.*, **28**, 2431-2438.

[12] Goodman, J.J., Draine, B.T., and Flatau, P.J. 1991, "Application of FFT Techniques to the Discrete Dipole Approximation, *Opt. Lett.*, **16**, 1198-1200.

[13] Hage, J.I., and Greenberg, J.M. 1990, "A Model for the Optical Properties of Porous Grains", *Astrophysical J.*, **361**, 251-259.

[14] Petravic, M., & Kuo-Petravic, G. 1979, *J. Computational Phys.*, 32, 263.

[15] Purcell, E.M., and Pennypacker, C.R. 1973, "Scattering and Absorption of Light by Nonspherical Dielectric Grains", Astrophysical J., **186**, 705-714.

[16] Temperton, C.J. 1983, "Self-sorting mixed-radix Fast Fourier Transforms", *J. Comp. Phys.*, **52**, 1-23.

[17] Temperton, C. 1992, "A Generalized Prime Factor FFT Algorithm for Any $N = 2^p3^q5^r$", *SIAM Journal of Scientific and Statistical Computing*, **13**, 676-686

[18] Warren, S.G. 1984, "Optical constants of ice from the ultraviolet to the microwave", *Appl. Opt.*, **23**, 1206-1225.

A Understanding and Modifying ddscat.par

In order to use DDSCAT to perform the specific calculations of interest to you, it will be necessary to modify the `ddscat.par` file. Here we list the sample `ddscat.par` file, followed by a discussion of how to modify this file as needed. Note that all numerical input data in DDSCAT is read with free-format `READ(IDEV,*)...` 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 any informational comments on the line appearing *after* the numerical data.

```
' ================== Parameter file ==================='
'**** PRELIMINARIES ****'
'NOTORQ'= CMTORQ*6 (DOTORQ, NOTORQ) -- either do or skip torque calculations
'PBCGST'= CMDSOL*6 (PBCGST, PETRKP) -- select solution method
'GPFAFT'= CMETHD*6 (GPFAFT, BRENNR, TMPRTN, CONVEX)
'LATTDR'= CALPHA*6 (LATTDR, LDRISO, GOBR88, DRAI88)
'ALLBIN'= CBINFLAG (ALLBIN, ORIBIN, NOTBIN)
'NOTCDF'= CNETFLAG (ALLCDF, ORICDF, NOTCDF)
'RCTNGL'= CSHAPE*6 (FRMFIL,ELLIPS,CYLNDR,RCTNGL,HEXGON,TETRAH,UNICYL,UNIELL)
8 6 4 = shape parameters PAR1, PAR2, PAR3
1        = NCOMP = number of dielectric materials
'TABLES'= CDIEL*6 (TABLES,H2OICE,H2OLIQ; 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 'INV' = 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, thetan_min, thetan_max, dtheta (in degrees) for plane A
90. 0. 180. 30 = phi, ... for plane B
```

Lines	Comments
1-2	comment lines – no need to change.
3	`NOTORQ` if torque calculation is not required; `DOTORQ` if torque calculation is required.
4	`PBCGST` is recommended; other option is `PETRKP` (see §13).
5	`GPFAFT` recommended; other options are `BRENNR`, `TMPRTN`, `CONVEX` (§12).
6	`LATTDR` (LATTice Dispersion Relation) is recommended (§10).
7	`ALLBIN` for full unformatted binary dump (§9.2; `ORIBIN` for unformatted binary dump of orientational averages only; `NOTBIN` for no unformatted binary output.
8	`ALLCDF` for output in netCDF format (must have netCDF option enabled; cf. §9.3); `ORICDF` for orientational averages in netCDF format (must have netCDF option enabled). `NOTCDF` for no output in netCDF format;
9	specify choice of target shape (see §18 for description of options `RCTNGL`, `ELLIPS`, `TETRAH`, ...)
10	shape parameters SHPAR1, SHPAR2, SHPAR3, ... (see §18).
11	number of different dielectric constant tables (§11).
12	name(s) of dielectric constant table(s) (one per line).
13	comment line – no need to change.
14	`0` is recommended value of parameter `INIT`.
15	`ERR` = error tolerance h: maximum allowed value of $\|A^{\dagger}E - A^{\dagger}AP\|/\|A^{\dagger}E\|$ [see eq.(6)].
16	comment line – no need to change.
17	`ICTHM` – number of θ_s values for angular averages (§21).
18	`IPHM` – number of ϕ_s values for angular averages.
19	comment line – no need to change.
20	λ – first, last, how many, how chosen.
21	comment line – no need to change.
22	$a_{\rm eff}$ – first, last, how many, how chosen.
23	comment line – no need to change.
24	specify x,y,z components of (complex) incident polarization $\hat{\mathbf{e}}_{01}$ (§20)
25	`IORTH` = 1 to do one polarization state only; 2 to do second (orthogonal) incident polarization as well.
26	`IWRKSC` = 0 to suppress writing of ".sca" files; 2 to enable writing of ".sca" files.
27	comment line – no need to change.
28	β (see §16) – first, last, how many .
29	Θ – first, last, how many.
30	Φ – first, last, how many.
31	comment line – no need to change.
32	ϕ_s for first scattering plane, $\theta_{s,min}$, $\theta_{s,max}$, how many θ_s values;
33,...	ϕ_s for 2nd,... scattering plane, ...

B wxxryyori.avg Files

The file `w00r00ori.avg` contains the results for the first wavelength (`w00`) and first target radius (`r00`) averaged over orientations (`ori.avg`). The `w00r00ori.avg` file generated by the sample calculation should look like the following:

```
-------------------------------------------------------------------------------
 DDSCAT --- DDSCAT.5a.7 [96.11.21]
 TARGET --- Rectangular prism; NX,NY,NZ=   8   6   4
 GPFAFT --- method of solution
 LATTDR --- prescription for polarizabilies
 RCTNGL --- shape
 NAT0 =      192 = number of dipoles
  AEFF=   1.00000 = effective radius (physical units)
  WAVE=   6.28319 = wavelength (physical units)
   K*A=   1.00000 = 2*pi*aeff/lambda
 n= ( 1.3300 ,  0.0100),  epsilon= (  1.7688 ,  0.0266)  for material 1
   TOL= 1.000E-05 = error tolerance for CCG method
 ICTHM= 33 = theta values used in comp. of Qsca,g
 IPHIM= 12 = phi values used in comp. of Qsca,g
 ( 1.00000  0.00000  0.00000) = target axis A1 in Target Frame
 ( 0.00000  1.00000  0.00000) = target axis A2 in Target Frame
 ( 0.27942  0.00000  0.00000) = k vector (latt. units) in Lab Frame
 ( 0.00000, 0.00000)( 1.00000, 0.00000)( 0.00000, 0.00000)=inc.pol.vec. 1 in LF
 ( 0.00000, 0.00000)( 0.00000, 0.00000)( 1.00000, 0.00000)=inc.pol.vec. 2 in LF
   0.000   0.000 = beta_min, beta_max ;  NBETA = 1
   0.000  90.000 = theta_min, theta_max; NTHETA= 3
   0.000   0.000 = phi_min, phi_max   ;   NPHI = 1
 Results averaged over   3 target orientations
                   and   2 incident polarizations
          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  2.673E-01 3.430E-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                                  dQpha= 6.768E-02
        Qsca*g(1)  Qsca*g(2)  Qsca*g(3)
 JO=1:  2.352E-02  1.126E-03 -2.792E-10
 JO=2:  1.781E-02  2.768E-03  8.290E-10
 mean:  2.066E-02  1.947E-03  2.749E-10
            ** Mueller matrix elements for selected scattering directions **
theta   phi   S_11       S_21       S_31       S_41       S_12       S_13      Pol.
  0.0   0.0 5.167E-02  7.916E-03  6.070E-11 -9.234E-12  7.916E-03  5.061E-11  0.15320
 30.0   0.0 4.046E-02  4.518E-04  1.927E-11 -3.656E-11  4.518E-04  1.893E-11  0.01117
 60.0   0.0 2.169E-02 -1.023E-02  2.814E-11  5.079E-11 -1.023E-02  4.559E-11  0.47184
```

```
 90.0   0.0 1.193E-02 -1.192E-02  8.390E-13  8.937E-11 -1.192E-02  1.400E-11  0.99888
120.0   0.0 1.170E-02 -5.702E-03 -3.435E-11  8.106E-11 -5.702E-03  3.526E-12  0.48743
150.0   0.0 1.403E-02  9.758E-04 -5.853E-11  2.137E-11  9.758E-04  5.080E-11  0.06956
180.0   0.0 1.411E-02  3.333E-03 -1.497E-11 -3.060E-11  3.333E-03 -1.408E-12  0.23621
  0.0  90.0 5.167E-02 -7.916E-03 -7.013E-10  5.923E-12 -7.916E-03 -7.618E-10  0.15320
 30.0  90.0 4.487E-02 -1.280E-02 -3.765E-04  8.979E-05 -1.281E-02 -3.843E-04  0.28540
 60.0  90.0 3.040E-02 -2.062E-02 -3.544E-04  1.643E-04 -2.063E-02 -3.773E-04  0.67816
 90.0  90.0 1.991E-02 -1.989E-02 -1.001E-05  1.827E-04 -1.991E-02 -3.807E-05  0.99857
120.0  90.0 1.626E-02 -1.175E-02  1.796E-04  1.359E-04 -1.176E-02  1.634E-04  0.72242
150.0  90.0 1.478E-02 -5.293E-03  1.180E-04  6.554E-05 -5.295E-03  1.131E-04  0.35823
180.0  90.0 1.411E-02 -3.333E-03 -1.511E-11 -3.091E-11 -3.333E-03 -3.189E-10  0.23621
--------------------------------------------------------------------------------
```

C wxxryykzzz.sca Files

The w00r00k000.sca file contains the results for the first wavelength (w00), first target radius (r00), and first orientation (k000). The w00r00k000.sca file created by the sample calculation should look like the following:

```
---------------------------------------------------------------------------------
 DDSCAT --- DDSCAT.5a.7 [96.11.21]
 TARGET --- Rectangular prism; NX,NY,NZ=   8   6   4
 GPFAFT --- method of solution
 LATTDR --- prescription for polarizabilies
 RCTNGL --- shape
 NAT0 =      192 = number of dipoles
  AEFF=   1.00000 = effective radius (physical units)
  WAVE=   6.28319 = wavelength (physical units)
   K*A=   1.00000 = 2*pi*aeff/lambda
 n= ( 1.3300 ,  0.0100),  epsilon= (  1.7688 ,  0.0266)  for material 1
   TOL= 1.000E-05 = error tolerance for CCG method
 ICTHM= 33 = theta values used in comp. of Qsca,g
 IPHIM= 12 = phi values used in comp. of Qsca,g
 ( 1.00000  0.00000  0.00000) = target axis A1 in Target Frame
 ( 0.00000  1.00000  0.00000) = target axis A2 in Target Frame
 ( 0.27942  0.00000  0.00000) = k vector (latt. units) in TF
 ( 0.00000, 0.00000)( 1.00000, 0.00000)( 0.00000, 0.00000)=inc.pol.vec. 1 in TF
 ( 0.00000, 0.00000)( 0.00000, 0.00000)( 1.00000, 0.00000)=inc.pol.vec. 2 in TF
 BETA =  0.000 = rotation of target around A1
 THETA=  0.000 = angle between A1 and k
  PHI =  0.000 = rotation of A1 around k
         Qext      Qabs       Qsca     g=<cos>    Qbk       Qpha
 JO=1:  1.110E-01 3.028E-02 8.077E-02  3.504E-01 1.858E-03 4.668E-01
 JO=2:  8.651E-02 2.441E-02 6.209E-02  3.650E-01 1.362E-03 4.197E-01
```

```
 mean:  9.878E-02 2.735E-02 7.143E-02  3.567E-01 1.610E-03 4.432E-01
 Qpol=  2.454E-02                                   dQpha= 4.711E-02
        Qsca*g(1)  Qsca*g(2)  Qsca*g(3)
 JO=1:  2.830E-02 -1.059E-09 -2.190E-09
 JO=2:  2.266E-02  2.300E-09  1.293E-09
 mean:  2.548E-02  6.208E-10 -4.484E-10
            ** Mueller matrix elements for selected scattering directions **
theta   phi   S_11        S_21        S_31        S_41        S_12        S_13       Pol.
  0.0   0.0 4.987E-02  5.372E-03 -1.184E-11 -6.738E-12  5.372E-03 -1.225E-11  0.10772
 30.0   0.0 4.040E-02 -9.275E-04 -2.702E-11 -8.238E-11 -9.275E-04 -2.584E-11  0.02296
 60.0   0.0 2.191E-02 -1.060E-02  1.928E-11  5.637E-12 -1.060E-02  5.200E-12  0.48383
 90.0   0.0 1.058E-02 -1.056E-02 -9.754E-13 -1.038E-11 -1.056E-02 -2.142E-11  0.99794
120.0   0.0 7.506E-03 -3.995E-03  6.801E-12  3.866E-11 -3.995E-03 -2.491E-11  0.53222
150.0   0.0 5.921E-03 -9.862E-06  2.480E-11  2.154E-12 -9.862E-06  3.311E-11  0.00167
180.0   0.0 5.058E-03  7.791E-04 -1.873E-11  8.492E-12  7.791E-04  3.865E-12  0.15403
  0.0  90.0 4.987E-02 -5.372E-03 -4.352E-10  3.510E-11 -5.372E-03 -4.724E-10  0.10772
 30.0  90.0 4.285E-02 -1.032E-02 -2.709E-10 -2.025E-11 -1.032E-02 -2.915E-10  0.24082
 60.0  90.0 2.736E-02 -1.773E-02 -1.044E-10  6.915E-11 -1.773E-02 -1.856E-10  0.64798
 90.0  90.0 1.540E-02 -1.539E-02 -2.644E-12 -1.035E-11 -1.539E-02 -1.325E-10  0.99942
120.0  90.0 9.786E-03 -6.745E-03  2.369E-11 -2.483E-13 -6.745E-03 -8.581E-11  0.68918
150.0  90.0 6.389E-03 -1.820E-03 -9.220E-12  5.426E-13 -1.820E-03 -5.744E-11  0.28491
180.0  90.0 5.058E-03 -7.791E-04 -1.914E-11 -2.743E-11 -7.791E-04 -6.465E-11  0.15403
--------------------------------------------------------------------------------
```