

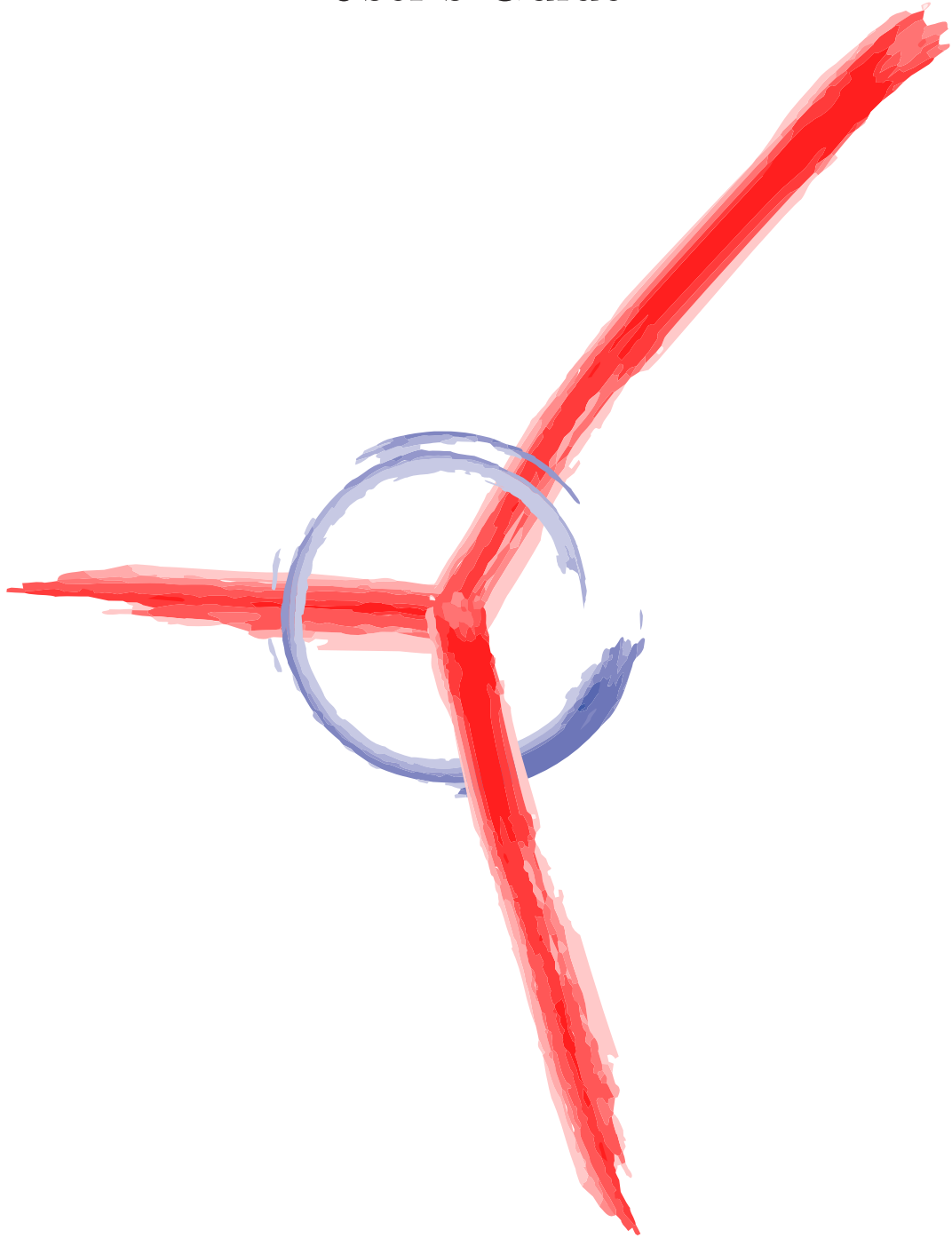
# Mie Scattering Utilities

Michael P. Schubmehl

Version 1.00

August 4, 2002

## User's Guide



# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>The mietable Program</b>	<b>4</b>
2.1	Getting Started . . . . .	4
2.2	Reference . . . . .	5
2.3	Implementation . . . . .	5
2.4	Modifying and Compiling the Code . . . . .	6
<b>3</b>	<b>The snell Program</b>	<b>7</b>
3.1	Getting Started . . . . .	7
3.2	Reference . . . . .	8
3.3	Implementation . . . . .	8
3.4	Modifying and Compiling the Code . . . . .	8
<b>4</b>	<b>The fitmie Program</b>	<b>10</b>
4.1	Getting Started . . . . .	10
4.2	Reference . . . . .	11
4.3	Implementation . . . . .	12
4.4	Modifying and Compiling the Code . . . . .	12

# Chapter 1

## Introduction

The `mietable`, `fitdist`, `snell`, and `fitmie` programs documented here were written for use in sizing droplets by Mie scattering measurements. In particular, they are geared towards analyzing data from experimental setups like the those in Figure ?? and Figure 1.1. To determine the size of a cloud or suspension of droplets, scattered light is measured as a function of angle, and the resulting data are fit to a theory curves generated by the `mietable` program. The actual fitting procedure is performed by one of two programs: in the event the droplets are believed to be entirely of a single size, the `fitmie` program can be used. If the configuration is not thought to be monodisperse, distributed fitting to a variety of distributions can be performed. Finally, it may or may not be necessary to correct for the presence of a cuvette full of some sort of solvent, depending on the exact experimental configuration. This can be done with the `snell` program, which outputs corrected Mie tables.

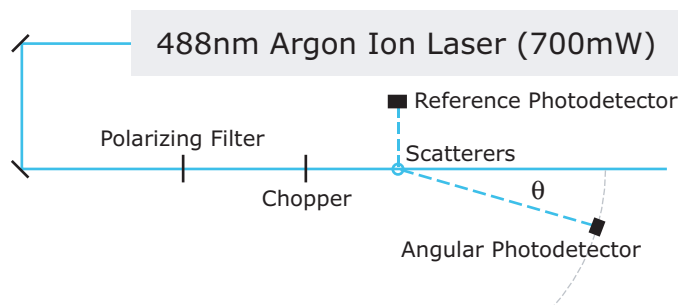


Figure 1.1: Experimental setup for Mie scattering measurements.

First, the `mietable` program is used to generate theory tables of scattered intensity as a function of angle for a given relative index of refraction and a specific range of size parameters. The dimensionless *size parameter* of a droplet is directly proportional to its diameter, and is defined by

$$x = \frac{\pi dn}{\lambda},$$

where  $n$  is the index of refraction of the medium surrounding the droplet,  $d$  the diameter of the droplet, and  $\lambda$  the wavelength of the light used for the scattering in vacuum. These tables can be generated for any range of size parameters and angles needed for analysis.

The tables generated by the `mietable` program are not always directly useful in analyzing experimental data, however, because they fail to account for two important optical effects—the presence of a cuvette, and the fact that the medium surrounding the droplets may not be the same one in which the angle measurements are taking place. In other words, the

tables need to be adjusted using Snell's law, in the event a cuvette is present. To do this, the tables are passed to the `snell` program, which uses various aspects of the experimental setup to numerically correct the tables for Snell's law effects. In particular, the user must specify the distance  $R$  of the detector from the cuvette, the half-length of a cuvette side  $L$ , the relative index of refraction of the medium surrounding the spheres to that in which the measurements are taking place  $n$ , and the polarization of the incident light. Options are also available to adjust the accuracy of the correction and specify whether or not to correct for the light passing through the front and side faces of the cuvette.

Once the tables have been adjusted, if necessary, to correspond to the appropriate experimental setup, the experimental data can be fit to these theory tables using either the `fitmie` program or the `fitdist` program. Note that, in an experimental setup like the ones shown above, there may be other detectors used to normalize the intensity and reduce error. Any corrections used for these effects should be applied to the data prior to fitting. Once the data and theory tables are ready, both are passed to the fitting routine of the user's choice.

The distributed fitting code, `fitdist`, reads in experimental data and theory tables, along with a fit file specifying the ranges of parameters to test. It then uses one of two fitting methods—a simple exploration of parameter space, or the more complicated, more efficient Levenberg-Marquardt method—to determine which of the size distributions produces a Mie theory composite that minimizes the chi-squared per degree of freedom. Detailed results of the fit, including the data, parameter ranges, best fit, size distribution, and Mie theory, can be output to file.

The `fitmie` program looks through the theory tables and attempts to minimize one of several error quantities that can be specified. The program can perform a least-squares fit with or without uncertainty information, a divide-by-point fit as described in [BH], or a simple front-to-back fit of the ratio of forward- to backward-scattered intensity. The user can also specify a fitting range, in order to exclude points that may contain too much error to be useful. The output of the `fitmie` program is used to determine the size parameter of the droplets. By default, it returns only the best fit and an estimate of the error in that fit. An option is available, however, to output tables of chi-squared values for the data relative to each theory table, in a form suitable for graphing. The user can also output scaled theory tables at a particular size parameter, for use in creating a data-theory overlay plot.

The remaining chapters of this document explain in detail how to set up and use each of the programs, as well as how they work and how to modify them.

## Chapter 2

# The mietable Program

### 2.1 Getting Started

The `mietable` program generates tables of intensity as a function of angle for three different polarizations of incident light—perpendicular to the scattering plane, parallel to the scattering plane, and unpolarized. It outputs tables for angles ranging from 0 degrees out to some user-specified maximum angle, at all size parameters in a given interval. The user must specify the index of refraction of the scatterers relative to the surrounding medium.

To install the program, just copy the file `mietable.exe` to the desired directory. It is recommended that the install directory be added to the system's `PATH`, or that all data files are kept in the same directory as the program. Consult your operating system's documentation for information on adding directories to the path.

Once the program has been installed, it can be run from the command line. On a PC running Microsoft Windows 9x, a command line can be accessed by running `command.com`. In Windows NT/2000/XP, run `cmd.exe` instead. Change to the directory where you placed the `mietable` program and use the following command to run the program:

```
mietable xmin xmax xstep nr [maxoutputangle] [> outputfile]
```

The required arguments are summarized in the following list.

<code>xmin</code>	The smallest size parameter to include in the table. Must be a real value greater than zero, and in practice greater than around 0.1.
<code>xmax</code>	The largest size parameter to include in the table. Must be a real value greater than or equal to <code>xmin</code> , and in practice less than around 10,000.
<code>xstep</code>	Step size between consecutive size parameters in the table. Must be a positive real number.
<code>nr</code>	Relative index of refraction of scatterers to surrounding medium. For example, latex spheres in water have $1.59/1.33 \approx 1.2$ , and water droplets in air have $1.33/1.0 = 1.33$ .
<code>[maxoutputangle]</code>	Optional argument giving the maximum angle to include in output. Default (and maximum) is 180 degrees.

The output of the program goes to standard output, and can thus be redirected to an output file with the command `> outputfile` in a UNIX or DOS environment. The format of the output is described in detail in the next section.

Typical uses of `mietable` might be

```
mietable 8 14 0.1 1.33 > mietables.txt
```

```
mietable 1 40 0.2 1.33 70 > tables
```

which generates a table of three intensities as a function of angle (one for each polarization), for angles going from 0 to 180° in the first case, and from 0 to 70° in the second case. The first table would start at a size parameter of 8.0, then include size parameters 8.1, 8.2, and so on up to a size parameter of 14.0. The second tables would include size parameters 1, 1.2, and so forth. The index of refraction is that of water relative to air, so these tables could both be used to fit water droplets suspended in air. The redirect commands `> mietables.txt` and `> tables` are *not* part of the program's syntax, but rather DOS and UNIX commands to write the program's output into the files `mietables.txt` and `tables`.

If the program encounters a problem processing the arguments, it will exit with error status and print an abbreviated version of the argument list above.

## 2.2 Reference

Installation instructions and syntax for the `mietable` program are described in the Getting Started section above. This section describes the program's output format and discusses some of its limitations.

The `mietable` program's output is plain text containing the desired intensity data. The specific format is geared towards correction by the `snell` program and/or fitting with the `fitdist` and `fitmie` program. Output is produced in columns in the following format:

```
*    0.500
    0.00      0.0002690      0.0002690      0.0002690
    1.00      0.0002690      0.0002689      0.0002690
    2.00      0.0002690      0.0002687      0.0002688
    <angle>   <perp intens>   <par intens>   <unp intens>
    .         .           .           .
    .         .           .           .
    .         .           .           .
    <maxangle> 0.0000000    0.0000000    0.0000000
* <size parameter>
    .         .           .           .
    .         .           .           .
    .         .           .           .
```

The intensities are in arbitrary units, and the angles are in degrees. The asterisk sentinel flags where the table for a new size parameter begins. Note that, in general, the perpendicular polarization intensity will decrease very slowly as a function of angle and the parallel intensity will decrease very rapidly, while the unpolarized intensity is the average of the other two.

The program's limitations are inherited fairly directly from Warren Wiscombe's MIEVO program, which does all of the underlying computation needed to generate the tables. A detailed discussion of the limitations of the MIEVO code in different domains is given in [WW]. In practice, the program's accuracy should be adequate in a reasonable range of size parameters (say from around 0.5 to 1000). This range should be more than adequate to perform particle sizing of micron-sized droplets.

## 2.3 Implementation

The `mietable` program is written almost entirely in ANSI Fortran 77. Only the `INCLUDE` command used to import Wiscombe's `miev0.f` and `ErrPack.f` code is non-standard. The main body of the program is simple; it begins by processing its arguments and checking for errors. If any are found, it write a usage message to standard output, then exits. Note that if there is an error and the output has been redirected to a file, the usage message will be redirected to this file and *not* printed to the screen.

After performing all necessary error checking, the program calculates a table of cosines of angles to be passed to the `MIEVO` subroutine. Then it steps through the specified size parameters and passes the cosine table, size parameter and index of refraction to the `MIEVO` subroutine. The subroutine returns a variety of pieces of information, including intensity as a function of angle for the three polarizations. The `mietable` program prints these data to standard output in the format described above. After it has finished the last size parameter, the program exits.

For details on the inner workings of the `MIEVO` routines, see [WW] and a good optics book with a discussion of Mie scattering.

## 2.4 Modifying and Compiling the Code

It would be fairly easy to modify the program to output other pieces of information provided by the `MIEVO` routines. In the output section of the program, one would only need to change which variables are examined and how they are printed out. It would be similarly easy to change the program's output format, although caution should be exercised in doing so, because the `snell`, `fitdist` and `fitmie` programs expect the current format.

The easiest way to compile the code is with a freeware compiler like GNU's `g77`. With `g77`, simply place the files `mietable.f`, `miev0.f`, and `ErrPack.f` in the same directory, and type

```
g77 mietable.f -o mietable.exe
```

to compile the program. This should generate no errors or warnings. Note that some compilers may not support the use of the command `INCLUDE`, which is used to incorporate Wiscombe's `MIEVO` routines into the program. If successful, this procedure will generate an executable called `mietable.exe`, which can be used as described above to generate tables of intensity as a function of angle.

## Chapter 3

# The `snell` Program

### 3.1 Getting Started

The `snell` program reads in tables of intensity as a function of angle for three different polarizations of incident light, in the format produced by the `mietable` program, discussed in section 2.2. The user specifies a polarization, and the program takes this information, as well as information about the geometry of the setup and adjusts the theory tables for the specified polarization for Snell's law and foreshortening effects. It outputs corrected tables for angles ranging from 0 degrees to some user-specified maximum angle, at all size parameters in the original tables. It outputs these new theory values to the output file in the same format as the `mietable` program, with the two unused columns (polarizations) zeroed out. The user must specify the index of refraction of the medium surrounding the scatterers relative to the medium in which measurements are being made.

To install the program, copy `snell.exe` to the desired directory. This can be the same directory as any or all of the other Mie programs. It is recommended that the install directory be added to the system's PATH, or that all data files are kept in the same directory as the program. Consult your operating system's documentation for information on adding directories to the path.

Once the program has been installed, it can be run from the command line. Change to the directory where you placed the `snell` program and use the following command to run the program:

```
snell inputfile outputfile [-R int] [-L int] [-n double]
    [-pol {1,2,3}] [-w double] [-N int] [-max double]
    [-front {1,0}] [-side {1,0}] [-help | -?]
```

The required arguments and optional switches are summarized in the following list.

<code>inputfile</code>	Uncorrected Mie tables in <code>mietable</code> format.
<code>outputfile</code>	File to output corrected Mie tables to.
<code>[-R int]</code>	Radius from detector to cuvette in mm (default = 245).
<code>[-L int]</code>	Half the length of cuvette side in mm (default = 6).
<code>[-n double]</code>	Relative index of refraction of medium surrounding scatterers to medium in which measurements are being made (e.g., water to air = 1.33 when using particles suspended in water) (default = 1.33).
<code>[-pol 1,2,3]</code>	Polarization. 1 = perp, 2 = par, 3 = unp (default).
<code>[-w double]</code>	Angular resolution of detector, in degrees (default = 0.3).
<code>[-N int]</code>	Number of scatterers to use in correction (default = 20).
<code>[-max double]</code>	Maximum angle to output corrected data at (default = 70.0).
<code>[-front 1,0]</code>	Use 0 to disable front correction, 1 to enable (default).
<code>[-side 1,0]</code>	Use 0 to disable side correction, 1 to enable (default).
<code>[-help   -?]</code>	Print this message.



The tables output by the program are written directly to the file specified by the argument `outputfile`, so there is no need to redirect output as there is with the `mietable` program. The only information printed to standard output is a status indicator that shows the program's progress in correcting the original tables.

A typical use of `snell` might be

```
snell mietables.txt snelltables.txt -pol 2 -n 1.33
```

which corrects the parallel-polarized column of the file `mietables.txt` using a relative index of refraction of 1.33, which is appropriate for an experiment where scatterers are suspended in water and measurements take place in the surrounding air. The results are output in a table where the first and third columns are zeroed out and the second column contains the corrected parallel-polarized theory. The output produced on standard output looks like:

```
C:\Mie>snell mietables.txt snelltables.txt
Performing correction on file 'mietables.txt': 60%
```

If the program encounters a problem processing the arguments, it will exit with error status and print an abbreviated version of the list above.

## 3.2 Reference

The output format of the program is identical to that of the `mietable` program, except that `snell` truncates the tables at the angle specified by the `-max` switch. Although the user can specify an angle as high as 180, the default of 70 is sufficient for the setup shown in the introduction, since data is hard to collect at higher angles.

Most of the optional parameters are fairly self-explanatory. The angular resolution of the detector, specified by the `-w` switch, is not central to the computation, and can be a rough approximation. The `-front` and `-side` correction switches are present to give experimenters the option of blocking the side of the cuvette with an opaque material, then disabling the side correction, for example.

## 3.3 Implementation

The `snell` program is implemented in ANSI C++, and has been successfully compiled using Microsoft Visual Studio 6.0. It relies on the `dataSet` class defined in `dataset.hh` and related files, which are included with the source code for the program.

The Snell's law and foreshortening corrections are both performed numerically, since closed-form solutions are difficult to work with. The program reads in the tables from the input file sequentially, selects the appropriate polarization, and stores the data to a `dataSet` object. It then approximates the angle shifts upon leaving the front and side faces of the cuvette and corrects for the intensity being significantly higher at small angles due to the foreshortening effect described in [IW]. The interested reader is directed to this paper as a reference for further details. The corrected tables are then output in the same format they were read in, but with the unused polarization columns zeroed out. The tables produced by the `snell` program are in a format suitable for use by the `fitmie` program.

## 3.4 Modifying and Compiling the Code

Probably the most useful modification to the `snell` program is changing the default values to match a new experimental setup. This can be accomplished with relative ease by modifying `snell.cpp`, which is well-documented with comments. Care should be taken to change the default values printed in the help table if the actual defaults are to be modified.

As noted above, the program uses `dataset.hh` for the `dataSet` class in which the theory data is stored. The key issue in compiling the code is to properly include this file and the `polynomial.hh` file. This can be accomplished through a makefile in a UNIX environment. To compile under Visual Studio, carefully follow the How To Compile instructions included with the `snell` source code.

## Chapter 4

# The `fitmie` Program

### 4.1 Getting Started

The `fitmie` program reads in two tables — one containing experimental data in the format described in the Reference section, and one containing Mie theory tables in the format produced by the `mietable` and `snell` programs. In the default fitting mode, it then examines the  $\chi^2$  between the data and the theory at each size parameter in the theory table. The theory table is scaled by a constant multiplicative factor to account for the fact that the units of the intensity data are arbitrary. This is called an  $\alpha$  regression. The program can output the results of this fit in a simple form, or dump the  $\chi^2$  as a function of size parameter. It can also output the scaled fit tables in a form suitable for graphing along with the original. The program has an option to restrict the range of the fit and an option to disregard uncertainty data in case some part of the experimental data is deemed unreliable. The fit types supported by the program are:

1. An alpha regression. In this method, the theory curve is first scaled so that it has the same amplitude as the data, taking into account the uncertainty at each point in the data set. This new theory curve is then compared to the data set by computing a  $\chi^2$  (i.e., the sum of the squares of the residuals at each point in the data set, divided by the number of points in the set.) A lower  $\chi^2$  indicates a better fit.
2. An alpha regression disregarding uncertainties. Same as above, but without taking into account uncertainty. This could be useful for data without uncertainties, or data with suspect uncertainty.
3. A divide-by-point normalization. This method, suggested by Bohren and Huffman [BH], uses a scaling factor for both the data and theory. Both scaling factors are computed by dividing all of the intensities present by the intensity at some chosen angle. A  $\chi^2$  is then computed as above, taking into account uncertainties.
4. A front-back fit. This fitting method takes the average of a given number of points from the front and back of the fitting range in the data and in the theory. It then compares the ratio of these to averages in the data and theory, and uses the squared difference between the theory ratio and data ratio to measure fit quality.

Each of these fit methods can be selected on the command line, as described below.

To install the program, copy `fitmie.exe` to the desired directory. This can be the same directory as any or all of the other Mie programs. It is recommended that the install directory be added to the system's `PATH`, or that all data files are kept in the same directory as the program.

Once the program has been installed, change to the directory where you placed it and use the following command to run the program:

```
fitmie datafile theoryfile [-pol {1,2,3}] [-nouncert]
      [-dividebypoint double] [-frontback int] [-dump {data, chi, double}]
      [-range double double] [-help | -?]
```

The required arguments and optional switches are summarized in the following list.

<code>datafile</code>	Experimental data (angle-intensity-sigma table).
<code>theoryfile</code>	Theoretical Mie tables in <code>mietable</code> format.
<code>[-pol 1,2,3]</code>	Polarization. 1 = perp, 2 = par, 3 = unp.
<code>[-nouncert]</code>	Fit by alpha regression without uncertainties.
<code>[-dividebypoint double]</code>	Fit by Bohren and Huffman's divide by point method, using the specified angle.
<code>[-frontback int]</code>	Fit by back-to-front ratio using averages of a number of points from front and back of data.
<code>[-dump data,chi,double]</code>	Dump input data, chi-square, or the fit curve at the size parameter nearest to the double.
<code>[-range double double]</code>	Fit only data in the range from the first angle specified to the second angle specified.
<code>[-help   -?]</code>	Print help message.

The output of the program goes to standard output, and can thus be redirected to an output file with the command `> outputfile` in a UNIX or DOS environment. This is useful once a reasonably good fit has been obtained, and the user wishes to output a scaled theory curve to overlay on the data. It is also useful for producing tables of  $\chi^2$  as a function of size parameter to visually depict the quality of the fit. The format of this output is described in the next section.

A typical use of `fitmie` might be

```
fitmie data.txt snelltables.txt -pol 2
```

or, once a good fit has been found and the user wishes to produce a  $\chi^2$  table,

```
fitmie data.txt snelltables.txt -pol 2 -range 12 48 -dump chi > chidist.txt
```

Both of these fit the data to parallel-polarized tables generated by the `snell` program (or tables produced directly by the `mietable` program if no `snell` correction is needed for some reason). Again, note that the redirect command `> chidist.txt` is *not* part of the program's syntax, but rather a DOS and UNIX command to write the program's output (in this case a table of  $\chi^2$  values) to the file `chidist.txt`.

If the program encounters any errors processing the arguments, it will exit with error status and print an abbreviated version of the list above.

## 4.2 Reference

The program takes as input a data table in the format:

```

4      24782.229965156796    1526.0627957332767
6      4436.597110754414     575.72747321778
8      1076.4872521246461    158.36175478043836
<angle>    <intensity>        <uncertainty>
.          .                .
.          .                .
.          .                .
```

where the intensity units are arbitrary, but the same as those of the uncertainties. It also takes a theory table in the `mietable` format described in section 1.2. Note that if the tables have been corrected by the `snell` program, two of the three columns will be zeroed out.

### 4.3 Implementation

The `fitmie` program is implemented in ANSI C++, and has been successfully compiled using Microsoft Visual Studio 6.0. It relies on the `dataSet` class defined in `dataset.hh` and the polynomial class defined in `polynomial.hh`.

The program reads in the theory tables one at a time, computes the appropriate scaling factor according to the fit method, and then computes a  $\chi^2$  (or other measure of fit quality). It keeps track of the minimum  $\chi^2$  encountered, the size parameter that produced it, and the corresponding alpha (scaling) factor. It then either prints the results of the fit, dumps the data, dumps the  $\chi^2$  at each point, or dumps the scaled theory at a particular intensity, depending on the command line switches specified. For details on each of the fitting methods, see [IW].

### 4.4 Modifying and Compiling the Code

The most useful modification to this program would be to make it a stand-alone. That is, allow the user to pass in a data file, and generate any needed theory on the fly. Implemented like this, the program would be able to refine the accuracy of the fit on the fly, producing better size parameter approximations. Because of the uncertainty present in the data and the vastly greater development time required to implement this method, the current system was more than sufficient in the experiments performed.

To compile the code, keep in mind that the program uses `dataset.hh` for the `dataSet` class in which the theory data is stored. The key issue in compiling it is to properly include this file and the `polynomial.hh` file. This can be accomplished through a makefile in a UNIX environment. To compile under Visual Studio, carefully follow the How To Compile instructions included with the `fitmie` source code. Note that the program also requires the standard library files `stdlib.h`, `iostream.h`, `fstream.h`, and `string.h`.

# Bibliography

- [IW] I. Weiner, M. Rust, and T.D. Donnelly. “Particle Size Determination: An Undergraduate Lab in Mie Scattering.” *American Journal of Physics* February 2001: 129-136.
- [WW] W. J. Wiscombe. “Mie Scattering Calculations: Advances in Technique and Fast, Vector-Speed Computer Codes.” NCAR Technical Note. June 1996.
- [BH] Craig F. Bohren and Donald R. Huffman. *Absorption and Scattering of Light by Small Particles*. New York: Wiley, 1983.