CHARACTERIZATION OF SPHERES
WITH THE SUBMICRON PARTICLE ANALYZER: FEASIBILITY

DTIC ELECTED
DEC 01 1992

Jerold R. Bottiger

RESEARCH DIRECTORATE

September 1992

Approved for public release; distribution is unlimited.
Disclaimer

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorizing documents.
A method is described for inverting light scattering data to find the size parameter \( x \) and refractive index \( n \) of small dielectric spheres. The data considered is restricted to that obtainable with the U.S. Army Chemical Research, Development and Engineering Center’s Submicron Particle Analyzer (i.e., ratios of intensities detected at a number of fixed directions about the scattering sphere). The inversion process works by comparing measured flux ratios with the same ratios previously calculated over a range of \( x, n \) values, and finding those \( x, n \) pairs for which measured and calculated ratios are consistently in agreement, to within the experimental uncertainty. Using numerical simulations of measurements and estimating the experimental error to be \(+/- 10\%\), we find that about 13 ratio measurements are needed to perform satisfactory inversions.
The work described in this report was authorized under Project No. 10162622A552, Smoke and Obscurants. This work was started in October 1990 and completed in September 1991.

The use of trade names or manufacturers' names in this report does not constitute an official endorsement of any commercial products. This report may not be cited for purposes of advertisement.

Reproduction of this document in whole or in part is prohibited except with permission of the Commander, U.S. Army Chemical Research, Development and Engineering Center, ATTN: SMCCR-SPS-T, Aberdeen Proving Ground, MD 21010-5423. However, the Defense Technical Information Center and the National Technical Information Service are authorized to reproduce the document for U.S. Government purposes.

This report has been approved for release to the public.
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INTRODUCTION</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>OUTLINE OF THE INVERSION METHOD</td>
<td>8</td>
</tr>
<tr>
<td>2.1</td>
<td>N-X Plane</td>
<td>8</td>
</tr>
<tr>
<td>2.2</td>
<td>Selection of Scattering Properties</td>
<td>9</td>
</tr>
<tr>
<td>2.3</td>
<td>Pixel Acceptance Criterion</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>CALCULATION OF SCATTERED INTENSITIES</td>
<td>11</td>
</tr>
<tr>
<td>3.1</td>
<td>Scattered Intensity at a Point</td>
<td>11</td>
</tr>
<tr>
<td>3.2</td>
<td>Correction for Finite Acceptance Angle</td>
<td>15</td>
</tr>
<tr>
<td>3.3</td>
<td>Mie Computer Programs</td>
<td>19</td>
</tr>
<tr>
<td>3.4</td>
<td>Calculation of the Flux Ratio Data Sets: HILO.F</td>
<td>21</td>
</tr>
<tr>
<td>4</td>
<td>TESTING THE INVERSION METHOD</td>
<td>23</td>
</tr>
<tr>
<td>4.1</td>
<td>Program INVERT</td>
<td>23</td>
</tr>
<tr>
<td>4.2</td>
<td>Results of the Inversion Trials</td>
<td>29</td>
</tr>
<tr>
<td>5</td>
<td>CONCLUSION</td>
<td>30</td>
</tr>
<tr>
<td>LITERATURE CITED</td>
<td></td>
<td>39</td>
</tr>
<tr>
<td>APPENDIXES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>HILO.F LISTING</td>
<td>41</td>
</tr>
<tr>
<td>B</td>
<td>CALCULATED RATIO MAPS</td>
<td>53</td>
</tr>
<tr>
<td>C</td>
<td>INVERT.F LISTING</td>
<td>63</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

1 Detailed View of the H090 Flux Ratio Over a Segment of the X,N Plane ............................................ 10

2 Light Scattering Geometry of the Submicron Particle Analyzer ................................................................. 12

3 Scattered Light of Varying Intensity Falling on a SELF0C Fiber-Optic Lens .............................................. 15

4 Calculation, at One Degree Intervals, of Light Scattering in the Vicinity of a SELF0C Lens .......................... 17

5 Surface of Pixel Minimum Values of the Flux Ratio U090 Over the X,N Plane ........................................... 24

6 Surface of Pixel Maximum Values of the Flux Ratio U090 Over the X,N Plane ........................................... 25

7 Comparison, for One Flux Ratio, of Calculated (Individual Bars) and Measured (Horizontal Band) Ratio Values .......................................................... 28

8 Inversion Results for Four Spheres, Assuming +/- 0.3% Data Uncertainty ...................................................... 31

9 Inversion Results for Four Spheres, Assuming +/- 3.0% Data Uncertainty ...................................................... 33

10 Inversion Results for Four Spheres, Assuming +/- 10.0% Data Uncertainty .................................................. 33

11 Inversion Results for Twelve Spheres, Assuming +/- 30.0% Data Uncertainty .............................................. 34

12 Inversion Results for Twelve Spheres, Assuming +/- 0.3% Data Uncertainty .............................................. 35

13 Inversion Results for Twelve Spheres, Assuming +/- 3.0% Data Uncertainty .............................................. 36

14 Inversion Results for Twelve Spheres, Assuming +/- 10.0% Data Uncertainty .............................................. 37

15 Inversion Results for Twelve Spheres, Assuming +/- 30.0% Data Uncertainty .............................................. 38
1. INTRODUCTION

The Submicron Particle Analyzer (SPA) is an instrument built by Wyatt Technology Corporation (Santa Barbara, CA) for the U.S. Army Chemical Research, Development and Engineering Center (CRDEC) that is used to study light scattering by aerosol particles.\textsuperscript{1,2} The SPA comprises a spherical chamber in the center of which a dilute vertical stream of sampled aerosol particles traverses an intense horizontal laser beam, one particle at a time. Light scattered from each particle is intercepted and measured via 22 optical fibers that are distributed on the surface of the sphere and lead to 22 photomultiplier tubes and associated electronics in a separate instrument rack. The optical fibers, which are terminated on the chamber end with SELFOC gradient index lens, can be deployed among any of 72 ports on the sphere; the same nine port scattering angles are repeated along eight semi-great circles that are 45 degrees apart. The function of the SPA is to gather a set of light scattering data from each aerosol particle, from which physical characteristics of the particles (i.e., size and shape) may be inferred. The aim of our current research with this instrument is to work out the appropriate types of data to be included in the measured sets and to discover the manner in which those data sets may be manipulated to reveal desired particle characteristics.

Light scattering instruments built for sizing spherical particles typically measure scattered intensity in one or two directions and produce a "signal" proportional to either an intensity or a ratio of intensities. These instruments differ according to the scattering and acceptance angles chosen and the spectral nature of the illuminating radiation. In any case, a response curve is calculated that gives the signal level expected as a function of particle diameter; often, a calibration with a few spheres of known size is also required to match calculated and actual signals. There are several well-known problems with this approach. First, it is the nature of scattering by spheres that the response curves are not single valued -- that is, there are several sizes of spheres that can produce any given signal level. This effect may be mitigated, but never completely eliminated, by choices made in the instrument's design. Also, the response curve depends on the refractive index of the sphere; therefore, a different curve and calibration, if necessary, must be available for every different particle material. In practice, this is rarely done. Finally, these instruments provide no way to check whether each scattering particle is actually a sphere. The spherical response curve is simply applied to every particle that scatters light whether it is appropriate or not.

With the SPA and its multiplicity of light scattering channels, we may have an opportunity to design a measurement and analysis technique that will characterize spherical particles with a confidence never before attained. In this report, we will propose a direct inversion technique for processing SPA data, write the necessary computer codes to implement the technique, and then test the procedure using computer-generated, synthetic data to represent...
experimental measurements that can be collected with the SPA. We wish to
discover whether a useful characterization of spherical particles is feasible
given the nature and accuracy of SPA light scattering data.

2. OUTLINE OF THE INVERSION METHOD

In the context of the present problem, an inversion method may be
said to be successful when it produces estimates for the size and refractive
index of a sphere such that the calculated light scattering properties of that
particular sphere agree with the corresponding measured properties. A formal
mathematical inverse solution to the MIE equations, to express size parameter
(x) and refractive index (n) in terms of scattering intensities, is not
possible. Instead, we shall discover acceptable values for x and n by
considering, one pair at a time, all possible values of x and n, repeatedly
asking if the spheres specified by the x,n pairs scatter light in agreement
with the measurements, and noting the ones that do. The phrases "all possible
values" and "in agreement with the measurements" require some elaboration.

2.1 N-X Plane.

In principle, there are an infinite number of possible values of
x and n, so the range of those variables necessarily must be restricted. A
large majority of dielectric materials, both liquids and solids, have
refractive indices between 1.3 and 1.8 at visible wavelengths, and so that
range is selected. (Absorbing spheres, in contrast with dielectric ones, must
be characterized with a third parameter in addition to x and n, and are
consequently beyond the scope of this report.) The size parameters considered
will be restricted to the range 0<x<10, though with less clear justification;
inversion of scattering from small particles should be more reliable owing to
their simpler scattering patterns; whereas, a maximum size parameter of 10 --
corresponding in blue light to a physical diameter of about 1.6 μm -- is large
enough to include many standard particles available for experiments.

Therefore, as a starting point, we consider spheres represented by
their coordinates on the x-n plane in the region 0<x<10 and 1.3<n<1.8. The
region is divided into a number of much smaller rectangular areas (pixels) of
dimensions Δx and Δn, with the aim of letting the sphere described by the
central coordinates of each pixel stand for all the spheres represented within
that pixel. This scheme succeeds if the relevant scattering properties of
central spheres in adjacent pixels differ by less than the expected
experimental uncertainty. Clearly, the pixel resolution must be at least as
small as the accuracy with which we wish to recover x and n. However, the
finer the resolution, the lengthier the inversion calculation, and there is no
point in demanding a higher resolution than that which can be supported by the
experimental accuracy. None of these factors are known a priori; to get
started, we choose, rather arbitrarily, Δx = .05 and Δn = .005. This results
in an array of 20,200 pixels stacked in 200 columns centered at x = 0.05,
0.10, ..., 10.00, and along 101 rows centered at n = 1.300, 1.305, ..., 1.800.

Although the pixels represent a 2-dimensional array (the x-n
plane), it will be convenient for later purposes to name the pixels with a
single, rather than a double, index. The pixel in the lower left corner \((x,n) = (0.05, 1.300)\) is pixel number 1; that in the lower right corner \((10.00, 1.300)\) is number 200, etc., left to right and up the rows to pixel number 20,200 in the upper right corner \((10.00, 1.800)\).

2.2 Selection of Scattering Properties.

The nature of the experimentally observable quantities, which will be compared to their calculated values for each sphere, must now be specified. As noted in Section 1 of this report, the SPA allows 22 intensities to be recorded for each particle. The laser beam intensity profile is Gaussian; since the exact path of particles through the beam is uncontrollable, the incident beam intensity for any particle is unknown, and the absolute values of the scattered intensities have little meaning. The ratios among the various scattered intensities for each particle are independent of the incident beam and can serve as the light scattering properties for inversion. Since all the detector SELFOC lenses have the same apertures and are equidistant from the scattering particle, we may refer to the observable quantities interchangeably as either intensity ratios or flux ratios.

To distinguish spherical from nonspherical particles, eight detectors, without polarizers, are placed in a ring at a scattering angle of \(\theta = 55^\circ\), and the incident beam is prepared in a right circularly polarized state. For spherical particles illuminated in this way, there can be no variation of light scattering with azimuth angle, \(\psi\); uniformity of the eight detector signals confirms particle sphericity. Random experimental noise will preclude exact equality of the eight measurements even for perfect spheres. However, the standard deviation will be small and the average value, owing to the eight times sampling, provides an excellent reference figure for forming ratios. The average intensity measurement of these eight ring detectors will be the denominator in every intensity ratio.

Eight equivalent (scattering independent of \(\psi\) for spheres) detector ports are available at each of the remaining eight scattering angles: \(\theta = 10, 40, 75, 90, 105, 125, 140,\) and \(170^\circ\). At any port, the scattered light may be detected through either a linear polarizer in one of two orientations or without a polarizer at all. There are then 64 ports through which up to 24 different measurements may be made and ratioed to the \(\theta = 55^\circ\) measurement to form 24 scattering properties of the particle. Only 14 of the original 22 detectors remain available; therefore, a 14-member subset of the 24 possible ratios must be chosen. The final selection will be discussed in Section 3.4 of this report.

2.3 Pixel Acceptance Criterion.

Preliminary calculations and plots to assess the character of flux ratios over the \(x-n\) plane revealed a deficiency in the original plan that divided the plane into pixels with sides \(Ax = .05\) and \(An = .005\). There are regions that correspond to morphology-dependent resonances\(^3\) of the sphere where the flux ratio grows rapidly to several times its value in neighboring pixels. Figure 1 shows the value of one flux ratio over a line in the \(x-n\) plane defined by \(n = 1.75\) and \(9.3 \leq x \leq 9.6\). This particular ratio happens to
be the intensity received through a horizontal polarizer at \( \theta = 90° \) to (as always) the intensity received through no polarizer (unpolarized) at \( \theta = 55° \). The heavy vertical lines represent pixel boundaries in the x direction, and the dots correspond to the pixel centers, at (9.30, 1.75), (9.35, 1.75), ..., (9.60, 1.75).

The amplitude of the ratio can vary by a factor of four or more across a single pixel of this size (\( \Delta x = .05 \)) in the vicinity of a resonance; clearly, no single ratio value can adequately represent so large a domain. One obvious cure is to reduce the pixel size; however, judging from the steepest slope in Figure 1, it is estimated that the pixel dimension must be reduced to about \( \Delta x = .001 \), 50 times smaller than present, to ensure that the ratio varies by no more than 5% across the x-dimension of any pixel. A reduction in deltax by a similar factor would also be required, resulting in a network of some 50 million pixels over the area of the x-n plane now covered by about 20 thousand. The inefficiency in designing everywhere for the worst-case condition (up the slope of a resonance) would be colossal; in any event, the enormous demands on computer memory and central processing unit time prohibit this approach.

Figure 1. Detailed View of the H090 Flux Ratio Over a Segment of the X,N Plane
Instead, we leave the pixel dimensions unchanged and record for each pixel the minimum and maximum values of the flux ratios over all the pixel, not just the ratio values at the center of the pixel. This doubles the amount of data that must be made available to the inversion program, but that is all. This method will give correct, if not necessarily useful, results with pixels of any dimensions that may be chosen later on.

Each flux ratio determined with the SPA will have an experimental uncertainty associated with it, depending on the absolute magnitude of the measured intensities. Establishing rules for estimating the uncertainty will be an important task in the experimental phase of this inversion project; however, for the present feasibility study, we may simply specify uncertainties as needed. The inversion process may be summarized as follows. Consider a single pixel and a single flux ratio: if the physical sphere’s size and refractive index are given by any point \( (x, n) \) within the pixel, then the true flux ratio must lie between the calculated minimum and maximum flux ratios. The experimentally determined flux ratio is really a range of ratios from \((\text{Measurement} - \text{Uncertainty})\) to \((\text{Measurement} + \text{Uncertainty})\). If the calculated and experimental ranges overlap, the pixel may contain the sphere’s actual size and refractive index. If there is no overlap, the pixel cannot contain the sphere’s parameters. After all the flux ratios have been checked in this manner over a pixel, that pixel may be reported as an inversion solution if agreement occurred for every flux ratio. If at least in principle even one pair of calculated and measured flux ratios failed to overlap, the pixel is to be rejected.

### 3. CALCULATION OF SCATTERED INTENSITIES

#### 3.1 Scattered Intensity at a Point

Consider a plane wave incident on an isolated single particle as shown in Figure 2 and ignore for the moment the indicated polarizer. The Stokes vector of the scattered light falling onto an infinitesimally small aperture in the direction \( \theta \) and located a distance \( R \) from the particle is given by

\[
\begin{pmatrix}
I_s \\
Q_s \\
U_s \\
V_s
\end{pmatrix} = \frac{1}{k^2 R^2} \begin{pmatrix}
S_{11} & S_{12} & S_{13} & S_{14} \\
S_{21} & S_{22} & S_{23} & S_{24} \\
S_{31} & S_{32} & S_{33} & S_{34} \\
S_{41} & S_{42} & S_{43} & S_{44}
\end{pmatrix} \begin{pmatrix}
I_i \\
Q_i \\
U_i \\
V_i
\end{pmatrix} [\text{watt/cm}^2]
\]

where \( I_i \), \( Q_i \), \( U_i \), and \( V_i \) are the Stokes parameters of the incident beam and \( k = 2\pi/\lambda \). Each Stokes parameter has the dimension of intensity (e.g., watts/cm\(^2\)), whereas, the 16 scattering matrix elements \( S_{ij} \) are dimensionless and functions of \( \theta \).
In the particular case where the particle is spherically symmetric and the incident beam is right circularly polarized and of intensity $I_0$, equation 1 simplifies to the form

$$\left( \begin{array}{c} I_s \\ Q_s \\ U_s \\ V_s \end{array} \right) = \frac{1}{k^2 R^2} \left( \begin{array}{cccc} S_{11} & S_{12} & 0 & 0 \\ S_{12} & S_{11} & 0 & 0 \\ 0 & 0 & S_{33} & S_{34} \\ 0 & 0 & -S_{34} & S_{33} \end{array} \right) \cdot I_0 \left( \begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \end{array} \right)$$

(2)
The Stokes vector of the scattered light is then

\[
\begin{pmatrix}
I_s \\
Q_s \\
U_s \\
V_s
\end{pmatrix} = \frac{I_0}{k^2 R^2} \begin{pmatrix}
S_{11} \\
S_{12} \\
S_{34} \\
S_{33}
\end{pmatrix}
\]  

(3)

and the intensity of the scattered light (i.e., the first Stokes parameter) falling upon the aperture is

\[I_s = \frac{I_0}{k^2 R^2} S_{11}\]  

(4)

Let an ideal linear polarizer be placed in front of the aperture with its transmission axis oriented at an angle \(\rho\) with respect to the scattering plane -- the plane containing the aperture and the incident beam. When the polarizer transmission axis is parallel to the scattering plane, \(\rho = 0^\circ\). The angle grows positively as the polarizer is rotated counterclockwise, as seen by an observer looking inward through the aperture. With the polarizer inserted, the Stokes vector of scattered light reaching the aperture is given by

\[
\begin{pmatrix}
I_s \\
Q_s \\
U_s \\
V_s
\end{pmatrix} = \frac{1}{2} \begin{pmatrix}
1 & \cos 2\rho & \sin 2\rho & 0 \\
\cos 2\rho & \cos^2 2\rho & \cos 2\rho \sin 2\rho & 0 \\
\sin 2\rho & \cos 2\rho \sin 2\rho & \sin^2 2\rho & 0 \\
0 & 0 & 0 & 0
\end{pmatrix} \\
\times \frac{1}{k^2 R^2} \begin{pmatrix}
S_{11} & S_{12} & 0 & 0 \\
S_{12} & S_{11} & 0 & 0 \\
0 & 0 & S_{33} & S_{34} \\
0 & 0 & -S_{34} & S_{33}
\end{pmatrix} I_0 \begin{pmatrix}
1 \\
0 \\
0 \\
1
\end{pmatrix}
\]  

(5)
After multiplying the factors in equation 5, we find the intensity falling on the small aperture as

$$I_s = \frac{I_0}{2k^2 R^2} (S_{11} + S_{12} \cos 2\rho + S_{34} \sin 2\rho)$$

(6)

Two polarizer orientations lead to simple results. When the polarizer is oriented horizontally ($\rho = 0^\circ$), the intensity is proportional to $S_{11} + S_{12}$; if it is oriented diagonally ($\rho = 45^\circ$), then the intensity is proportional to $S_{11} + S_{34}$. It is impossible to bring an explicit $S_{33}$ dependence to the scattered light in this configuration; however, the scattering matrix elements for a sphere are related by

$$S_{11}^2 = S_{12}^2 + S_{34}^2 + S_{33}^2$$

(7)

so that, except for its sign, the $S_{33}$ element is implicitly determined when the other three elements are determined.

Letting subscripts H, D, and U represent the cases of horizontal polarizer, diagonal polarizer, and no polarizer, respectively, the following equation summarizes the results of this section and gives the intensity of the scattered light reaching an infinitesimal aperture at scattering angle $\theta$

$$IU(\theta) = \frac{I_0}{2k^2 R^2} \left[ S_{11}(\theta) + S_{11}(\theta) \right]$$

$$IH(\theta) = \frac{I_0}{2k^2 R^2} \left[ S_{11}(\theta) + S_{12}(\theta) \right]$$

$$ID(\theta) = \frac{I_0}{2k^2 R^2} \left[ S_{11}(\theta) + S_{34}(\theta) \right]$$

(8)
3.2 Correction for Finite Acceptance Angle.

If the intensity of light "I," perpendicularly incident on an aperture of area "A," is constant over its surface, then the flux through the aperture is simply IA. However, if the intensity varies with position on the aperture, an integration is required to find the total flux.

Figure 3 shows a circular aperture of radius r (representing the clear opening of one of the SELFOC lens collecting scattered light in the SPA) located a distance R away from the scatterer in the direction $\theta_0$. The intensity of the scattered light reaching the aperture varies with $\theta$ (only), as is suggested by the figure's shading.

Figure 3. Scattered Light of Varying Intensity Falling on a SELFOC Fiber-Optic Lens
Since \( x = R (\theta - \theta_0) \) and \( dx = R \, d\theta \), the total flux through the circular aperture is given by

\[
F = \int_{\theta_0-\delta}^{\theta_0+\delta} I(\theta) \, 2h(\theta) \, R \, d\theta
\] (9)

The height \( h \) satisfies

\[
r^2 = x^2 + h^2 = R^2(\theta - \theta_0)^2 + h^2
\] (10)

and so

\[
F = \int_{\theta_0-\delta}^{\theta_0+\delta} 2R^2 I(\theta) \sqrt{r^2 - (\theta - \theta_0)^2} \, d\theta
\] (11)

If the expressions for \( I(\theta) \) (equation 8) are substituted into equation 11, the resulting integral cannot be evaluated analytically owing to the complexity of the \( S_{ij} \). A tractable approximation to \( I(\theta) \) can be derived by evaluating \( I(\theta) \) at a set of discrete angles in the vicinity of the aperture and interpolating linearly between the calculated values. The SELFOC lenses have a clear diameter of 1.8 mm and are located 93 mm from the center of the SPA chamber; therefore, \( \theta_\tau = .9/93 = .00968 \) radians, or 0.55°. The computer program (described Section 3.3) can calculate the \( S_{ij} \) at intervals of one degree or greater, and the lenses are centered on integral degree values. Figure 4 shows the situation approximately to scale.

From Figure 4, we see that \( I(\theta) \) to the left of \( \theta_0 \) is given by the straight line whose equation is

\[
I(\theta) = \frac{I(\theta_0) - I(\theta_0 - d)}{d} [\theta - (\theta_0 - d)] + I(\theta_0 - d)
\] (12)
while for $\theta = \theta_0$

$$I(\theta) = \frac{I(\theta_0 + d) - I(\theta_0)}{d} [\theta - (\theta_0)] + I(\theta_0)$$

(13)

Figure 4. Calculation, at One Degree Intervals, of Light Scattering in the Vicinity of a SELFOC Lens.
Substituting equations 12 and 13 into equation 11 gives the total flux through the aperture as

\[ F = 2R^2 \int_{\theta_0-\theta}^{\theta_0} \sqrt{\theta^2 - (\theta - \theta_0)^2} \left\{ \frac{I(\theta_0) - I(\theta_0 - d)}{d} [\theta - (\theta_0 - d)] + I(\theta_0 - d) \right\} d\theta \]

(14)

\[ + 2R^2 \int_{\theta_0}^{\theta_0+\theta} \sqrt{\theta^2 - (\theta - \theta_0)^2} \left\{ \frac{I(\theta_0 + d) - I(\theta_0)}{d} [\theta - (\theta_0)] + I(\theta_0) \right\} d\theta \]

Making a change of variable to \( \phi = \theta - \theta_0 \) simplifies the expression to

\[ F = 2R^2 \left\{ \frac{I(\theta_0) - I(\theta_0 - d)}{d} \int_{-\phi}^{0} \sqrt{\theta^2 - \phi^2} (\phi + d) d\phi + I(\theta_0 - d) \int_{-\phi}^{0} \sqrt{\theta^2 - \phi^2} d\phi \right\} \]

(15)

\[ + 2R^2 \left\{ \frac{I(\theta_0 + d) - I(\theta_0)}{d} \int_{0}^{\phi} \sqrt{\theta^2 - \phi^2} \phi d\phi + I(\theta_0) \int_{0}^{\phi} \sqrt{\theta^2 - \phi^2} d\phi \right\} \]

The integrals in equation 15 are readily evaluated; after some algebra, the flux equation reduces to

\[ F = \pi \theta^2 R^2 \left[ \frac{2\theta_r}{3\pi d} I(\theta_0 - d) + \left(1 - \frac{4\theta_r}{3\pi d} I(\theta_0)\right) + \frac{2\theta_r}{3\pi d} I(\theta_0 + d) \right] \]

(16)
This equation takes the simple form

\[ F = \pi \theta_r^2 R^2 I(\theta_0) \]  

(17)

as expected when \( \theta_r \) goes to zero or when all three intensities are set equal to \( I(\theta_0) \).

When the numerical values for \( \theta_r \) and \( d \) are substituted (.55 and 1°), we find

\[ F = \pi \theta_r^2 R^2 \left[ .1175 I(\theta_0-1^\circ) + .7650 I(\theta_0) + .1175 I(\theta_0+1^\circ) \right] \]  

(18)

which gives the proportions for combining the intensity at the nominal detector angle with the intensities calculated one degree before and after that angle to correct for the finite acceptance angle of the SELFOC lenses. The correction makes little difference for spheres in the size range of interest because the intensity is quite a slow varying function of scattering angle for these small particles; however, for completeness and safety, the correction will be included in the intensity calculations that follow.

3.3 Mie Computer Programs.

The expression of the quantities \( S_{i,j} \) (scattering matrix elements in equations 2 through 8) in terms of the size and refractive index of spheres is the subject of Mie theory, which is discussed in detail in the standard texts of light scattering. The Mie equations are quite complex and are evaluated by computer to yield up numerical values for the scattering matrix elements.

The computer program adapted for this project was distributed several years ago by Peter Barber, now at Clarkson University (Pottsdam, NY), and is named, appropriately, MIE. MIE calculates various light scattering functions for homogeneous spheres and is based on the original program by Dave, later expanded upon by Wiscombe. MIE requests as input the size parameter and real and imaginary parts of the sphere’s refractive index and the desired angular increment (1, 2, 3, 5, and 10° are allowed) at which calculations are to be performed. The output is written to a file named SPHERE.DAT, and comprises a few lines repeating the input data and displaying calculated efficiencies, followed by seven columns of data headed with the following titles: ANGLE, M SUB 2, M SUB 1, S SUB 21, D SUB 21, INTENSITY, and POLAR.
The scattering angles run from 0 to 180°, inclusively, in steps of the specified increment. The next four columns relate to the scattering matrix elements as

\[
\frac{1}{2}(M_{SUB_2} + M_{SUB_1}) = S_{11}
\]
\[
\frac{1}{2}(M_{SUB_2} - M_{SUB_1}) = S_{12}
\]
\[
S_{SUB_21} = S_{33}
\]
\[
D_{SUB_21} = S_{34}
\]

The sixth and seventh columns are useful combinations of previous ones, namely

\[
\text{INTENSITY} = \frac{1}{2} (M_{SUB_2} + M_{SUB_1}) = S_{11}
\]
\[
\text{POLAR} = \frac{(M_{SUB_2} - M_{SUB_1})}{(M_{SUB_2} + M_{SUB_1})} = \frac{S_{12}}{\text{INTENSITY}}
\]

The heart of Barber’s MIE program\(^3\) is the subroutine SMIE, which does the actual evaluation of the quantities M\(_{SUB_2}\), etc. The input parameters, or functions derived from them, are passed to SMIE, which then returns a 3-dimensional array called ELTRMX(I, J, K) to the calling program.

The index I (\(= 1,4\)) specifies the type of scattering function in each element of the array, according to

\[
1 \sim M_{SUB_2}
\]
\[
2 \sim M_{SUB_1}
\]
\[
3 \sim S_{SUB_21}
\]
\[
4 \sim D_{SUB_21}
\]
The indices $J = 1, (90/D)+1$ and $K = 1, 2$, together, specify the scattering angle appropriate to each element of the array, given $D$, the increment between scattering angles. The relations between $\theta$, $D$, $J$, and $K$ are

\begin{align*}
\theta &= D(J - 1) \quad \text{when } K = 1, \quad (0 \leq \theta \leq 90^\circ) \\
\theta &= 180^\circ - D(J - 1) \quad \text{when } K = 2, \quad (91^\circ \leq \theta \leq 180^\circ)
\end{align*}

This subroutine SMIE is used, unaltered, in my program HILO.F for calculating the data sets of flux ratios.

3.4 Calculation of the Flux Ratio Data Sets: HILO.F.

All elements have now been assembled to write a program to calculate the minimum and maximum values of each flux ratio over each pixel in the $x,n$ plane. Looking ahead to the results of the calculation, when a 3-D surface plot is made of any of these flux ratios over the $x,n$ plane (Figures 5 and 6 of the main body of this report and Appendix B), a landscape is revealed of nearly parallel valleys and sharply rimmed ridges. There are no pits or peaks in the flux ratio values -- apparent spikes along the tops of ridges are artifacts of the plotting program -- therefore, in virtually every case, the extreme of the flux ratio for a pixel will occur on the perimeter of the pixel. Occasionally, when an absolute valley bottom or ridge crest is attained on the plane, the slope at that point is very gentle in at least one direction; therefore, there is no significant difference between that interior minimum (or maximum) value and the minimum (or maximum) around the pixel’s perimeter.

So, instead of calculating flux ratios at a high density of $x,n$ points over the pixels, it suffices to perform the calculations only on the perimeter around each pixel. More time is saved by noting that each interior pixel edge is common to two pixels and, in most cases, only two edges per pixel need be calculated. By avoiding calculations at points internal to the pixel, we save computation time by a factor of nearly $NSAMPS/2$, where $NSAMPS$ is the number of sampling points along one edge of a pixel. In the program HILO.F, $NSAMPS$ is set equal to 6 in the left half of the plane and equal to 20 in the more steeply corrugated right half of the plane.

The source code for HILO.F is listed in Appendix A and is reasonably well self documented; an overview of its operation follows. The main program HILO selects pixels in order from left to right, one row at a time, beginning with the bottom row ($n = 1.300$), and on each pixel selects which of the four edges to evaluate. HILO calls subroutine SIDE once for each edge and specifies $NSAMPS$. SIDE calculates the values of $x$ and $n$ at the
sampling sites along the edge, and at each sampling site calls the subroutine SMIE. SMIE returns (to SIDE) the array ELTRMX, which holds scattering functions as a function of scattering angle (every one degree) for the site. SIDE then calls subroutine UDINT, which computes the three intensities, IH, ID, and IU, for each of the nine SPA scattering angles (correcting for detector acceptance angle) and writes them to one column of the array IO. When IO is finally filled, it contains 27 rows and NSAMPS columns of intensities.

The intensities in IO are converted by SIDE to flux ratios by dividing each element in every row by the corresponding element in the row representing the reference intensity, namely the intensity scattered at 55° through no polarizer. Another array, MINMAX, is filled with values taken from IO; the smallest of the NSAMPS values in the kth row of IO is put into the first column of the kth row of MINMAX and the largest value of the same row goes in the second column. At the end, MINMAX is returned to HILO and contains the minimum and maximum values of each flux ratio over whichever pixel edge HILO originally requested.

Back in HILO, the contents of MINMAX are entered in a much larger 3-D array called ACCUM, which accumulates this edge data for all of the 200 pixels in the row. HILO keeps track of which edge data can be copied from an earlier calculation and which must be computed freshly. When ACCUM is filled, the four edges of each pixel are compared and the very smallest and largest flux ratio values for each pixel are appended to the output files. HILO then moves up to the next row of pixels and the process begins again.

Each output file comprises 20,200 lines, one for each pixel and in pixel order. A single real number specifying the minimum (or maximum) value attained by a particular flux ratio on the corresponding pixel is on each line. The files are named by describing the numerator flux with four characters. The first letter, one of H, D, or U indicates that the light either passed through a horizontally oriented polarizer, a diagonally oriented polarizer, or no polarizer, respectively, while the last three digits indicate the scattering angle. The extensions " .min" and " .max" are added to specify files of minimum and maximum flux ratios. There are 54 data files in all -- minimums and maximums for each of 3 polarizations at 9 scattering angles -- but of course two files, U055.MIN and U055.MAX, contain only 1.0’s.

HILO was compiled and run on a Stardent minicomputer (Ardent Computer Corporation, Dobbs Ferry, NY) and required just over 5 hr to complete. The data sets were transferred to a personal computer, where the graphing program "Surfer" was used to prepare plots for visualizing the flux ratios. Figures 5 and 6 show typical data sets; the surfaces of minimum (Figure 5) and maximum (Figure 6) values of the flux ratio U090 were plotted over the x-n plane in both surface and contour formats. The true surface of U090 lies on or above the surface of U090.MIN and on or below the surface of U090.MAX. The regularly spaced spikes along mountain crests, especially evident in Figure 5, are plotting artifacts, which result because the sampling and plotting mesh of pixels is too coarse to represent the true knife-edge ridges. Actually, there are no points of relative maxima of U090 over the plotted part of the x-n plane, and only two relative minima I can find --
shallow valley bottoms indicated by closed contour loops at about \((x,n) = (6.6,1.5)\) and \((8.3,1.4)\).

Plots of flux ratio maximum values, such as those in Figure 6, present a better picture of the true flux ratio than do plots of flux ratio minimum values. One can imagine that the upper plot of Figure 6 represents an opaque paint, covering - perhaps too thick at places - the flux ratio surface.

Surface plots of the maximum values of all the flux ratios are cataloged in Appendix B of this report. At scattering angles of 10 and 170°, there are few differences among the H, D, and U flux ratios (except for a factor of 2), which was expected. At 0 and 180°, the distinctions vanish altogether because \(S_{12} = S_{34} = 0\) (equation 8). Therefore, we would deploy detectors to measure, at the most, one of the three polarizations at those two extreme angles. Polarization-dependent differences among the flux ratios at more central scattering angles are much more apparent.

Graphs in Appendix B of this report allow us to envision the nature of the light scattering flux ratios, but they offer little guidance in selecting the optimum ratios to measure with the 14 available detectors. The problem is that contour lines for all the flux ratios follow the same patterns, which lay in similar directions along lines of \(n,x = \) constant. There are no data sets whose contours run perpendicular to the others, which, because of their orthogonality, would be especially important to include among the 14 measurements. Instead, we settled on data sets according to the following experimentally pragmatic criteria. Too much background light is picked up by detectors at 10 and 170°; until that can be corrected by redesigning the scattering chamber, no data will be collected at those angles. Generally, more light is available without a polarizer than through one; because higher intensity implies better signal to noise ratio, we take all the remaining unpolarized data sets (e.g., U040, U075, U090, U105, U125, and U140). At most scattering angles, the scattering through a horizontal polarizer seems to differ more from the unpolarized scattering than the scattering through a diagonal polarizer; therefore, we collect horizontal data sets at the same six scattering angles. This would leave room for two diagonally polarized data sets, which we take at 40 and 90°.

4. TESTING THE INVERSION METHOD

4.1 Program INVERT.

The FORTRAN program "INVERT" was written to explore and test the inversion procedure. INVERT first reads in a number of files, which includes the computed min/max values over the \(x-n\) plane for the 14 selected flux ratios, a row of experimental detector calibration coefficients (used in this study to apply controlled errors to the synthetic input data), and the synthetic input data (an \(N\) by 24 array of numbers \(\text{EXPDAT??}.TST\)) generated in a separate program (DATGEN) and simulating SPA measurements on a run of \(N\) particles). When inverting real data, the experimental uncertainty associated with each flux ratio measurement will be individually determined, based on the
Figure 5. Surface of Pixel Minimum Values of the Flux Ratio U090 Over the X,N Plane
Figure 6. Surface of Pixel Maximum Values of the Flux Ratio U090 Over the X,N Plane
measured absolute scattering intensity. However, for this feasibility study, we have assigned a number of constant uncertainties to the flux ratios to observe their effect.

Taking one particle at a time (i.e., one row of EXPDAT??, TST), INVERT computes the 55°-ring average intensity and the 14 flux ratios, and then expands the 14 flux ratios into measured flux ratio ranges (nominal value plus and minus the assigned uncertainty). On each pixel, INVERT must check for agreement (overlap) between the particle's "measured" range of ratio values and the previously calculated range of ratio values for each of the 14 flux ratios. This is accomplished with an algorithm (described in the following paragraph) that is much faster than literally asking 20,200 times for each of 14 flux ratios if an overlap occurs.

The 28 files of minimum and maximum flux ratios read in by INVERT are not exactly the files outputted by HILO -- U040.MIN and others. HILO files are first acted on by the program ORDERALL, which sorts each file in ascending order of flux ratio and simultaneously writes for each file an ancillary integer file that relates the original pixel order to the newly sorted order. The file U040.MIN, written by HILO, is renamed ORD_U040.MIN when its data are sorted, and the corresponding integer file is named NDX_U040.MIN; other files are analogously named. The files ORD*.* and NDX*.* (a total of 56) are read by INVERT during its initialization.

The method of determining if overlap occurs between a pixel's calculated range and the experimental range for a flux ratio and the motive for sorting the data files into ascending order are illustrated in Figure 7. In the upper half of Figure 7, the short horizontal bars represent the calculated minimum values of one of the 14 flux ratios; the shaded patches above the bars are reminders that the calculated flux ratio range is at or above the minimum value. There are 20,200 short horizontal bars, but only a few are shown in Figure 7. The pixel numbers, written above the horizontal axis, indicate which pixels on the x-n plane the horizontal bars correspond to (corresponding ORD_ and NDX_ files list, in order from left to right, the indicated flux ratio values and pixel numbers, respectively). The lower half of Figure 7 is analogous to the upper half but represents the maximum calculated values of the same flux ratio.

The measured value of the same flux ratio for one particular particle is indicated in Figure 7 by a horizontal band whose top edge has the flux ratio value (Measurement + Uncertainty) and whose bottom edge is at (Measurement - Uncertainty). This horizontal band is the same in both the minima and maxima halves of the figure. Notice that the bars are calculated values and fixed in place for all time; a different horizontal band is superimposed for each measured particle. Which pixels have an overlap between the measurement band and the calculated ranges? Both plots divide naturally into three regions of answers. In the upper plot, the pixels from the leftmost one out to the marker, A, may or may not overlap, depending on the pixels' maximum values (information not available is in the upper plot). Between markers A and B, the calculated ranges must overlap with the measured range because the minima are embedded in the measured range. Beyond marker B,
no pixels may overlap because the minimum calculated value exceeds the
greatest value allowed by measurement.

Similarly, in the lower part of the Figure 7 where calculated
maxima are plotted, pixels from No. 1 to marker A cannot overlap, pixels from
A to B must overlap, and pixels to the right of marker B might overlap with
the measurement band, depending on how low the calculated minima are. A bit
of reflection shows that if a pixel is in the "cannot overlap" region of one
plot, it must be in the "might overlap" region of the other. Therefore, the
aggregate of all the "cannot overlap" pixels from both plots includes, exactly
once, every pixel that does not overlap with the measured flux ratio range.
The remaining pixels then do overlap (i.e., agree with the measured data for
this particular flux ratio). So to find the pixels that do not overlap, we
must find point B of the upper plot and point A of the lower plot. More
precisely, we must search the file of ordered minimum values to find the
number of the first entry that is larger than (Measurement + Uncertainty).
For example, if the 12,313th entry were the first entry to exceed (Measure-
ment + Uncertainty), then lines 12,313 through 20,200 of the corresponding
NDX_file would contain the pixel numbers of pixels, which do not agree with
the data. We then search the file of ordered maximum calculated values to
find the last entry that is less than (Measurement - Uncertainty) -- perhaps
entry number 4002. Then, lines 1 through 4,002 of its corresponding NDX_file
contain all the remaining pixels that do not satisfy the measurement for this
particular flux ratio. The appropriate ranges of the two NDX_files can be
copied to an array that keeps track of how many times each pixel fails to
agree with flux ratio measurements for each particle.

The reason for doing all this is that finding the points A and B is
computationally very fast; a simple sequence of binary decisions homes right
in on them. For example, to find marker B in the array of minima values, we
start at entry No. 10,100 (the midpoint between 1 and 20,200) and ask if the
entry value is greater than or less than the target value of (Measurement +
Uncertainty). Everything to the left or right of 10,100 (depending on the
answer) can be eliminated. We then go to the midpoint of the surviving range
and again compare the value found there to the target value, which results in
eliminating half that range. After only 15 repetitions ($2^{15} = 32768$
> 20200), we find the desired point in the array. Similarly, 15 comparisons
between (Measurement - Uncertainty) and entries in the file of maxima values
find the critical point there, point A, and, as a result, the remainder of the
nonoverlapping pixels. Thirty yes/no questions asked on the sets of sorted
data provide exactly the same information as 20,200 pairs of yes/no questions
asked on the original data in a brute force manner.

A separate 20,200 element array accumulates for each pixel the
number of times (from 0 to 14) that the calculated and measured flux ratios
failed to overlap for each particle. Ideally, only pixels with a score of
zero (never failed to overlap) should be counted as solutions for the particle
in question. However, because actual measurements of flux ratios may
occasionally be in error by more than our best estimate of the experimental
uncertainty, it may be desirable to admit solutions that do not necessarily
satisfy all 14 of the available measurements. In this investigation, we
wanted to see how the number of false returns grew as we pared the number of
Figure 7. Comparison, for One Flux Ratio, of Calculated (Individual Bars) and Measured (Horizontal Band) Ratio Values
ratios with which agreement was required, and how the domain of solutions varied with different levels of experimental uncertainty. A few of the many tests done are shown in Figures 8-15.

Figures 8-15 are drawn by the page formatting program "PageGarden," by Bloc Publishing Corporation (Coral Gables, FL). The actual output file written by INVERT is a set of statements that instruct PageGarden where to draw pixels in the x-n plane and how dark to shade them, which depends on the number of overlaps. Simple changes to INVERT can alter the information related by the pixel print density.

4.2 Results of the Inversion Trials.

Four pairs of x,n coordinates (indicated by crosses in Figure 8) were selected and used to calculate the four rows of scattering measurements that would be recorded by the SPA instrument sampling the corresponding four spheres. These data were then entered to INVERT and the uncertainty level was set to +/− 0.3%. In Figure 8, the solidly shaded pixels are those on which flux ratios formed from the entered as "measured" data agree with previously calculated minimum and maximum flux ratios for all 14 cases. Partially shaded pixels agreed with 12 or 13 of the flux ratio "measurements," and open (nonshaded) pixels agreed with 10 or 11 of the flux ratio "measurements." An uncertainty of only 0.3% is completely unrealistic. Figure 8 only confirms that INVERT returns the correct solution when given essentially perfect data. However, notice that some spurious returns are already appearing in the upper right of the Figure 8.

Figure 9 shows the return when the stated experimental uncertainty is increased to +/− 3.0%. We believe 3% is about the upper limit for the SPA accuracy; under the most favorable conditions, that accuracy might be approached by one or two of the detectors. There are more returns further from the true solutions, but still the pattern is quite compact.

An accuracy of +/− 10% is more typical of the expected SPA performance. This is the uncertainty level assumed in Figure 10, which shows results for the same four spheres. We still see returns that are tightly clustered, except when as few as 10 agreements (overlaps) are accepted.

At +/− 30% uncertainty (Figure 11), the number of returns with only 10-11 agreements is becoming quite large. Even worse than the high number of these returns is the way they are distributed in disconnected patches all over the x-n plane; there is no hint of where the right answer might lie. The pattern looks much better for 12-13 agreements and is quite good for 14 agreements. We expect 30% to be near the lower limit of SPA measurement accuracy.

We have looked at many plots such as those in Figures 8-11, including plots in which the input data was corrupted with random errors, although always within the limits set by the assumed experimental uncertainty. There is surprisingly little difference if the data is actually distorted or not; the nature of the returns is pretty much completely established by the level of experimental uncertainty allowed. Taking 10% as an average
uncertainty value for the SPA, we concluded that requiring 13 or more agreements of a pixel to return it as a possible inversion solution should produce useful results.

Figures 12 through 15 show the inversion result for 12 spheres, with uncertainty levels of 0.3, 3.0, 10.0, and 30.0% assumed in the measurements. A pixel is shaded if it agrees with at least 13 of the 14 measurements. The outcome is encouraging. We see mostly compact connected patches of returns whose size parameter spread is about 0.3, corresponding to roughly 0.05 \( \mu \text{m} \) for blue light. The refractive index spread is not so useful (about 0.1), but the product \( n_x \) is very accurately determined for each particle.

5. CONCLUSION

We conclude that it is feasible to characterize small dielectric spheres with data measured by the Submicron Particle Analyzer (SPA). The data sets available are not as independent as one could wish for; each additional measurement contributes relatively little to the process of winnowing away unsuitable \((x, n)\) pairs so that about a dozen measurements are required. Although the method is inefficient, it does apparently work. Matters could be improved by including information not directly related to the flux ratios, such as the number of relative minima in the angular scattering from 0 to 180°. These measurements are suggested by Quist and Wyatt in their paper,11 which describes an inversion method very similar to this one; however, the SPA does not currently support any types of measurements other than flux ratios.

In the future, we will undertake to demonstrate the inversion method using actual experimental data taken with the SPA. The first step must be an accurate assessment of the instrument’s error characteristics as functions of received intensity. If we are successful in the inversion of experimental data, we will attempt to extend the method to spheres of larger size and/or made of absorbing materials. At the same time, we believe improvements can be made to the inversion code itself. It may be possible to replace each of the ordered minimum and maximum calculated flux ratio files with a fairly simple equation that can be solved for the critical points in Figure 7, thereby greatly reducing the size of computer random access memory required to run the inversion program.
Figure 8. Inversion Results for Four Spheres, Assuming +/- 0.3% Data Uncertainty
Figure 9. Inversion Results for Four Spheres, Assuming +/- 3.0% Data Uncertainty
Figure 10. Inversion Results for Four Spheres, Assuming +/- 10.0% Data Uncertainty
Figure 12. Inversion Results for Twelve Spheres, Assuming +/- 0.3% Data Uncertainty.
Figure 13. Inversion Results for Twelve Spheres, Assuming +/- 3.0% Data Uncertainty
Figure 14. Inversion Results for Twelve Spheres, Assuming +/- 10.0% Data Uncertainty.
Figure 15. Inversion Results for Twelve Spheres, Assuming +/- 30.0% Data Uncertainty
LITERATURE CITED


APPENDIX A
HILO.F LISTING

PROGRAM HILO
VERSION
Tue Sep 10 09:18:02 1991

LINE # | SOURCE CODE | PAGE 1
---|---|---
1 | PROGRAM HILO | 
2 | C NO INPUT DATA. A REGION OF THE X-N PLANE, 0<X<=10 AND | 
3 | C 1.3<=N<=1.8, IS DIVIDED INTO RECTANGLES (PIXELS) WITH DELTA X = | 
4 | C .05 AND DELTA N = .005 : 200 COLUMNS AND 101 ROWS. HILO | 
5 | C CALCULATES THE MINIMUM AND MAXIMUM VALUES OF EACH OF 27 FLUX | 
6 | C RATIOS FOR EACH PIXEL. OUTPUT FILENAMES SPECIFY FLUX NUMERATOR | 
7 | C POLARIZATION AND ANGLE; DENOMINATOR ALWAYS U055. ONLY X,N PAIRS | 
8 | C ON THE PERIMETERS OF PIXELS CONSIDERED. | 
9 | INTEGER NROW, NSAMPS, NSQR, DIRECT, K, I | 
10 | REAL MINMAX(27,2), ACCUM(8,200,27) | 
11 | CHARACTER*1 DUMMY | 
12 | CHARACTER FILNAM(54)*8 | 
13 | 
14 | FILNAM(1) = 'U010.MIN' | 
15 | FILNAM(2) = 'U010.MAX' | 
16 | FILNAM(3) = 'H010.MIN' | 
17 | FILNAM(4) = 'H010.MAX' | 
18 | FILNAM(5) = 'D010.MIN' | 
19 | FILNAM(6) = 'D010.MAX' | 
20 | FILNAM(7) = 'U040.MIN' | 
21 | FILNAM(8) = 'U040.MAX' | 
22 | FILNAM(9) = 'H040.MIN' | 
23 | FILNAM(10) = 'H040.MAX' | 
24 | FILNAM(11) = 'D040.MIN' | 
25 | FILNAM(12) = 'D040.MAX' | 
26 | FILNAM(13) = 'U055.MIN' | 
27 | FILNAM(14) = 'U055.MAX' | 
28 | FILNAM(15) = 'H055.MIN' | 
29 | FILNAM(16) = 'H055.MAX' | 
30 | FILNAM(17) = 'D055.MIN' | 
31 | FILNAM(18) = 'D055.MAX' | 
32 | FILNAM(19) = 'U075.MIN' | 
33 | FILNAM(20) = 'U075.MAX' | 
34 | FILNAM(21) = 'H075.MIN' | 
35 | FILNAM(22) = 'H075.MAX' | 
36 | FILNAM(23) = 'D075.MIN' | 
37 | FILNAM(24) = 'D075.MAX' | 
38 | FILNAM(25) = 'U090.MIN' | 
39 | FILNAM(26) = 'U090.MAX' | 
40 | FILNAM(27) = 'H090.MIN' | 
41 | FILNAM(28) = 'H090.MAX' | 
42 | FILNAM(29) = 'D090.MIN' | 
43 | FILNAM(30) = 'D090.MAX' | 
44 | FILNAM(31) = 'U105.MIN' | 
45 | FILNAM(32) = 'U105.MAX' | 
46 | FILNAM(33) = 'H105.MIN' | 
47 | FILNAM(34) = 'H105.MAX' | 
48 | FILNAM(35) = 'D105.MIN' | 
49 | FILNAM(36) = 'D105.MAX' | 
50 | FILNAM(37) = 'U125.MIN' | 
51 | FILNAM(38) = 'U125.MAX' | 
52 | FILNAM(39) = 'H125.MIN' | 
53 | FILNAM(40) = 'H125.MAX' | 
54 | FILNAM(41) = 'D125.MIN' | 
55 | FILNAM(42) = 'D125.MAX' | 
56 | FILNAM(43) = 'U140.MIN' | 
57 | FILNAM(44) = 'U140.MAX' | 
58 | FILNAM(45) = 'H140.MIN' | 
59 | FILNAM(46) = 'H140.MAX' |
<table>
<thead>
<tr>
<th>LINE #</th>
<th>SOURCE CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>61</td>
<td>FILNAM(47) = 'D140.MIN'</td>
</tr>
<tr>
<td>62</td>
<td>FILNAM(48) = 'D140.MAX'</td>
</tr>
<tr>
<td>63</td>
<td>FILNAM(49) = 'U170.MIN'</td>
</tr>
<tr>
<td>64</td>
<td>FILNAM(50) = 'U170.MAX'</td>
</tr>
<tr>
<td>65</td>
<td>FILNAM(51) = 'H170.MIN'</td>
</tr>
<tr>
<td>66</td>
<td>FILNAM(52) = 'H170.MAX'</td>
</tr>
<tr>
<td>67</td>
<td>FILNAM(53) = 'D170.MIN'</td>
</tr>
<tr>
<td>68</td>
<td>FILNAM(54) = 'D170.MAX'</td>
</tr>
<tr>
<td>69</td>
<td>THE FIRST ROW (N=1,300) AND THE REMAINING 100 ROWS ARE TREATED SEPARATELY</td>
</tr>
<tr>
<td>70</td>
<td>************* FIRST ROW *************</td>
</tr>
<tr>
<td>71</td>
<td>GET MINS AND MAXS ON ALL FOUR SIDES (DIRECTIONS) OF FIRST SQUARE</td>
</tr>
<tr>
<td>72</td>
<td>NSQR = 1</td>
</tr>
<tr>
<td>73</td>
<td>WHenever X &lt; 5, HAVE &quot;SIDE&quot; LOOK AT 6 POINTS ALONG EACH EDGE OF THE PERIMETER</td>
</tr>
<tr>
<td>74</td>
<td>NSAMPS = 6</td>
</tr>
<tr>
<td>75</td>
<td>DO 20 DIRECT=1,4</td>
</tr>
<tr>
<td>76</td>
<td>CALL SIDE(DIRECT, NROW, NSQR, NSAMPS, MINMAX)</td>
</tr>
<tr>
<td>77</td>
<td>DO 22 K=1,27</td>
</tr>
<tr>
<td>78</td>
<td>ACCUM(7,NSQR,K) = ACCUM(3,NSQR-1,K)</td>
</tr>
<tr>
<td>79</td>
<td>ACCUM(8,NSQR,K) = ACCUM(4,NSQR-1,K)</td>
</tr>
<tr>
<td>80</td>
<td>CONTINUE</td>
</tr>
<tr>
<td>81</td>
<td>CONTINUE</td>
</tr>
<tr>
<td>82</td>
<td>20 CONTINUE</td>
</tr>
<tr>
<td>83</td>
<td>22 CONTINUE</td>
</tr>
<tr>
<td>84</td>
<td>REMAINING 199 SQUARES IN THIS FIRST ROW REQUIRE THE CALLING OF &quot;SIDE&quot; FOR EACH OF THREE DIRECTIONS</td>
</tr>
<tr>
<td>85</td>
<td>DO 30 NSQR=2,100</td>
</tr>
<tr>
<td>86</td>
<td>WRITE(*,400) NSQR</td>
</tr>
<tr>
<td>87</td>
<td>400 FORMAT(1H 'WORKING ON ROW 1 SQUARE ',I3)</td>
</tr>
<tr>
<td>88</td>
<td>WRITE(*,401) NSQR</td>
</tr>
<tr>
<td>89</td>
<td>401 FORMAT(1H 'WORKING ON ROW 1 SQUARE ',I3)</td>
</tr>
<tr>
<td>90</td>
<td>WEST SIDE OF CURRENT SQUARE = EAST SIDE OF PREVIOUS SQUARE:</td>
</tr>
<tr>
<td>91</td>
<td>DO 32 K=1,27</td>
</tr>
<tr>
<td>92</td>
<td>ACCUM(7,NSQR,K) = ACCUM(3,NSQR-1,K)</td>
</tr>
<tr>
<td>93</td>
<td>ACCUM(8,NSQR,K) = ACCUM(4,NSQR-1,K)</td>
</tr>
<tr>
<td>94</td>
<td>32 CONTINUE</td>
</tr>
<tr>
<td>95</td>
<td>30 CONTINUE</td>
</tr>
<tr>
<td>96</td>
<td>34 CONTINUE</td>
</tr>
<tr>
<td>97</td>
<td>FOR THE RIGHT HALF OF THE ROW (X&gt;5) INCREASE NSAMPS TO 20</td>
</tr>
<tr>
<td>98</td>
<td>NSAMPS = 20</td>
</tr>
<tr>
<td>99</td>
<td>DO 40 NSQR=101,200</td>
</tr>
<tr>
<td>100</td>
<td>WRITE(*,401) NSQR</td>
</tr>
<tr>
<td>101</td>
<td>401 FORMAT(1H 'WORKING ON ROW 1 SQUARE ',I3)</td>
</tr>
<tr>
<td>102</td>
<td>WEST SIDE OF CURRENT SQUARE = EAST SIDE OF PREVIOUS SQUARE:</td>
</tr>
<tr>
<td>103</td>
<td>DO 42 K=1,27</td>
</tr>
<tr>
<td>104</td>
<td>ACCUM(7,NSQR,K) = ACCUM(3,NSQR-1,K)</td>
</tr>
</tbody>
</table>
DO 50 K=1,27
OPEN(UNIT=3, FILE=FILENAME(2*K-1), ACCESS='SEQUENTIAL',
& FORM='FORMATTED', STATUS='NEW')
OPEN(UNIT=4, FILE=FILENAME(2*K), ACCESS='SEQUENTIAL',
& FORM='FORMATTED', STATUS='NEW')
DO 52 NSQR=1,200
WRITE (3,'(E10.4)') MIN(ACCUM(1,NSQR,K),ACCUM(3,NSQR,K),
& ACCUM(5,NSQR,K),ACCUM(7,NSQR,K))
52 CONTINUE
DO 54 NSQR=1,200
WRITE (4,'(E10.4)') MAX(ACCUM(2,NSQR,K),ACCUM(4,NSQR,K),
& ACCUM(6,NSQR,K),ACCUM(8,NSQR,K))
54 CONTINUE
CLOSE(3)
CLOSE(4)
50 CONTINUE
*************** THIS ENDS THE FIRST ROW **********************
*************** *************** *************** ***************
54 C
THE REMAINING 100 ROWS ARE ALL HANDLED THE SAME SO WE HAVE ONE
GIANT LOOP FROM HERE TO THE END OF THE PROGRAM:
DO 300 NROW=2,101
52 C
FOR WHATEVER THE CURRENT ROW IS, PUT THE NORTH MIN AND MAX VALUES
OF THE PREVIOUS ROW INTO THE SOUTH MIN AND MAX VALUES OF THE
CURRENT ROW FOR EACH OF THE 27 PAGES:
DO 60 K=1,27
DO 62 NSQR=1,200
ACCUM(5,NSQR,K) = ACCUM(1,NSQR,K)
ACCUM(6,NSQR,K) = ACCUM(2,NSQR,K)
62 CONTINUE
60 CONTINUE
THE FIRST SQUARE MUST BE TREATED SEPARATELY FROM THE REST:
NSAMPS=6
NSQR=1
CALL SIDE(1,NROW,NSQR,NSAMPS,MINMAX)
DO 70 K=1,27
ACCUM(1,1,K) = MINMAX(K,1)
70 CONTINUE
<table>
<thead>
<tr>
<th>LINE #</th>
<th>SOURCE CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>181</td>
<td>ACCUM(2,1,K) = MINMAX(K,2)</td>
</tr>
<tr>
<td>182</td>
<td>70 CONTINUE</td>
</tr>
<tr>
<td>183</td>
<td>CALL SIDE(2,NROW,NSQR,NSAMPS,MINMAX)</td>
</tr>
<tr>
<td>184</td>
<td>DO 72 K=1,27</td>
</tr>
<tr>
<td>185</td>
<td>ACCUM(3,1,K) = MINMAX(K,1)</td>
</tr>
<tr>
<td>186</td>
<td>ACCUM(4,1,K) = MINMAX(K,2)</td>
</tr>
<tr>
<td>187</td>
<td>72 CONTINUE</td>
</tr>
<tr>
<td>188</td>
<td>CALL SIDE(4,NROW,NSQR,NSAMPS,MINMAX)</td>
</tr>
<tr>
<td>189</td>
<td>DO 74 K=1,27</td>
</tr>
<tr>
<td>190</td>
<td>ACCUM(7,1,K) = MINMAX(K,1)</td>
</tr>
<tr>
<td>191</td>
<td>ACCUM(8,1,K) = MINMAX(K,2)</td>
</tr>
<tr>
<td>192</td>
<td>74 CONTINUE</td>
</tr>
<tr>
<td>193</td>
<td>THIS COMPLETES THE FIRST SQUARE ONLY. THE REMAINDER OF THE</td>
</tr>
<tr>
<td>194</td>
<td>ROW IS DONE IN TWO PARTS; THE LEFT HALF AT LOW RESOLUTION</td>
</tr>
<tr>
<td>195</td>
<td>(NSAMPS=6) AND THE RIGHT HALF AT HIGH RESOLUTION (NSAMPS=20):</td>
</tr>
<tr>
<td>196</td>
<td>&quot;SIDE&quot; MUST NOW ONLY BE CALLED TWICE FOR EACH PIXEL, SINCE EACH</td>
</tr>
<tr>
<td>197</td>
<td>PIXEL'S BOTTOM AND LEFT EDGES HAVE ALREADY BEEN CONSIDERED (AND</td>
</tr>
<tr>
<td>198</td>
<td>REMEMBERED IN &quot;ACCUM&quot;) AS PARTS OF PREVIOUSLY EVALUATED ADJACENT</td>
</tr>
<tr>
<td>199</td>
<td>PIXELS.</td>
</tr>
<tr>
<td>200</td>
<td>*************** LEFT HALF OF ROW ******************</td>
</tr>
<tr>
<td>201</td>
<td>NSAMPS = 6</td>
</tr>
<tr>
<td>202</td>
<td>DO 80 NSQR=2,100</td>
</tr>
<tr>
<td>203</td>
<td>WRITE(*,402) NROW, NSQR</td>
</tr>
<tr>
<td>204</td>
<td>402 FORMAT(1H , 'WORKING ON ROW ',I3,2X,'SQUARE ',I3)</td>
</tr>
<tr>
<td>205</td>
<td>GET CURRENT SQUARE'S WEST SIDE FROM PREVIOUS SQUARE'S EAST SIDE:</td>
</tr>
<tr>
<td>206</td>
<td>DO 82 K=1,27</td>
</tr>
<tr>
<td>207</td>
<td>ACCUM(7,NSQR,K) = ACCUM(3,NSQR-1,K)</td>
</tr>
<tr>
<td>208</td>
<td>ACCUM(8,NSQR,K) = ACCUM(4,NSQR-1,K)</td>
</tr>
<tr>
<td>209</td>
<td>82 CONTINUE</td>
</tr>
<tr>
<td>210</td>
<td>CALL SIDE(DIRECT,NROW,NSQR,NSAMPS,MINMAX)</td>
</tr>
<tr>
<td>211</td>
<td>DO 86 K=1,27</td>
</tr>
<tr>
<td>212</td>
<td>ACCUM(2*DIRECT-1,NSQR,K) = MINMAX(K,1)</td>
</tr>
<tr>
<td>213</td>
<td>ACCUM(2*DIRECT,NSQR,K) = MINMAX(K,2)</td>
</tr>
<tr>
<td>214</td>
<td>86 CONTINUE</td>
</tr>
<tr>
<td>215</td>
<td>84 CONTINUE</td>
</tr>
<tr>
<td>216</td>
<td>80 CONTINUE</td>
</tr>
<tr>
<td>217</td>
<td>*************** RIGHT HALF OF ROW *********************</td>
</tr>
<tr>
<td>218</td>
<td>NSAMPS = 20</td>
</tr>
<tr>
<td>219</td>
<td>DO 90 NSQR=101,200</td>
</tr>
<tr>
<td>220</td>
<td>WRITE(*,403) NROW, NSQR</td>
</tr>
<tr>
<td>221</td>
<td>403 FORMAT(1H , 'WORKING ON ROW ',I3,2X,'SQUARE ',I3)</td>
</tr>
<tr>
<td>222</td>
<td>GET CURRENT SQUARE'S WEST SIDE FROM PREVIOUS SQUARE'S EAST SIDE:</td>
</tr>
<tr>
<td>223</td>
<td>DO 92 K=1,27</td>
</tr>
<tr>
<td>224</td>
<td>ACCUM(7,NSQR,K) = ACCUM(3,NSQR-1,K)</td>
</tr>
<tr>
<td>225</td>
<td>ACCUM(8,NSQR,K) = ACCUM(4,NSQR-1,K)</td>
</tr>
<tr>
<td>226</td>
<td>92 CONTINUE</td>
</tr>
</tbody>
</table>

Appendix A 44
DO 94 DIRECT = 1, 2
CALL SIDE DIRECT, NROW, NSQR, NSAMPS, MINMAX
DO 96 K = 1, 27
  ACCUM(2*DIRECT-1, NSQR, K) = MINMAX(K, 1)
  ACCUM(2*DIRECT, NSQR, K) = MINMAX(K, 2)
CONTINUE
94 CONTINUE
96 CONTINUE
THE 27 PAGES OF ACCUM ARE ALL FILLED NOW FOR THE CURRENT ROW.
WRITE RESULTS OUT TO FILES, APPENDING EACH TIME A FILE IS RE-OPENED. (THERE ARE TOO MANY FILES - 54 - TO HAVE THEM ALL OPENED AT ONCE. WE MUST OPEN AND CLOSE AS NEEDED FOR EACH ROW.)
DO 100 K = 1, 27
OPEN(UNIT=3, FILE=FILENAM(2*K-1), ACCESS='SEQUENTIAL',
     & FORM='FORMATTED', STATUS='OLD')
OPEN(UNIT=4, FILE=FILENAM(2*K), ACCESS='SEQUENTIAL',
     & FORM='FORMATTED', STATUS='OLD')
IN ORDER TO APPEND DATA TO AN 'OLD' FILE, REOPEN FILE AND READ A DUMMY CHARACTER REPEATEDLY TO PLACE THE FILE POINTER TO JUST PAST EOF, THEN BACKSPACE TO 'ERASE' EOF MARKER, LEAVING THE POINTER POINTING TO WHERE EOF USED TO BE. START APPENDING NEW DATA AT THAT LOCATION.
DO 103 I = 1, 21000
  READ(3, '(A)', END=105) DUMMY
  BACKSPACE(3)
  DO 102 NSQR = 1, 200
    WRITE (3, '(E10.4)') MIN(ACCUM(1, NSQR, K), ACCUM(3, NSQR, K),
    & ACCUM(5, NSQR, K), ACCUM(7, NSQR, K))
  CONTINUE
102 NSQR = 1, 200
103 CONTINUE
DO 107 I = 1, 21000
  READ(4, '(A)', END=109) DUMMY
  BACKSPACE(4)
  DO 104 NSQR = 1, 200
    WRITE (4, '(E10.4)') MAX(ACCUM(2, NSQR, K), ACCUM(4, NSQR, K),
    & ACCUM(6, NSQR, K), ACCUM(8, NSQR, K))
  CONTINUE
104 CONTINUE
107 CONTINUE
CLOSE(3)
CLOSE(4)
100 CONTINUE
105 CONTINUE
FINISHED APPENDING TO MIN & MAX DATA FILES FOR THIS ROW
STOP
END
SUBROUTINE SIDE DIRECT, NROW, NSQR, NSAMPS, MINMAX)
GIVEN THE LOCATION OF A PIXEL (VIA NROW & NSQR), WHICH OF THE FOUR EDGES TO EVALUATE (VIA DIRECT), AND THE NUMBER OF EVALUATION POINTS ALONG THAT EDGE (VIA NSAMPS), RETURNS
2-D ARRAY MINMAX, WHOSE ROWS CORRESPOND TO THE 27 FLUX RATIOS, AND WHICH GIVES IN COL 1 THE MIN VALUE OF THE FLUX RATIO.
```
301 C ALONG GIVEN EDGE AND IN COL 2 THE MAX VALUE.
302
303 IMPLICIT DOUBLE PRECISION (A-H,O-Z)
304 DIMENSION THETD(91), ELTR4X(4091,2)
305 INTEGER NROW, NSQR, NSAMPS, SAMP, NOLT, NANG, DIRECT, JX, NN
306 REAL MINMAX(27,2), IO(28,20), IU, IN, ID
307
308 NDLT=1
309 RFI=0.0
310 JX=90/NDLT + 1
311 NANG=0
312 DO 10 J=1,JX
313 THETO(J) = DBLE(NANG)
314 NANG = NANG + NDLT
315 10 CONTINUE
316
317 DELTAX = .05/NSAMPS
318 DELTAN = .005/NSAMPS
319 CENTRN = 1.300 + .005*(NROW -1)
320 CENTRX = 0.05 * NSQR
321 AT EACH SAMPLING POINT ALONG EDGE, CALC X AND N
322 DO 20 SAMP=1,NSAMPS
323 IF (DIRECT .EQ. 1) THEN
324 RFR = CENTRN + .0025
325 X = CENTRX -.025 -0.5*DELTAX + SAMP*DELTAX
326 ELSE IF (DIRECT .EQ. 2) THEN
327 RFR = CENTRN + .0025 -.5*DELTAN + SAMP*DELTAN
328 X = CENTRX +.025
329 ELSE IF (DIRECT .EQ. 3) THEN
330 RFR = CENTRN -.0025
331 X = CENTRX -.025 -0.5*DELTAX + SAMP*DELTAX
332 ELSE IF (DIRECT .EQ. 4) THEN
333 RFR = CENTRN -.0025 -.5*DELTAN + SAMP*DELTAN
334 X = CENTRX -.025
335 ENDIF
336 20 CONTINUE
337 GET SCATT FUNCTIONS VS ANGLE AT SAMPLING POINT, VIA ELTRMX
338 CALL SHIE (X, RFR, RFI, QSCAT, QEXT, CTBRQS, THETO, ELTRMX, JX)
339 FORM UNPOL, HORIZ POL, AND DIAG POL INTENSITIES AT SPA SCATTERING
340 ANGLES FROM ELTRMX RETURNS, CORRECTING FOR DETECTOR ACCEPTANCE
341 ANGLE. PUT THEM INTO ARRAY IO (27+1 ROWS, ONE FOR EACH FLUX
342 RATIO PLUS THE REFERENCE INTENSITY (IO55) REPEATED IN LAST Row),
343 AND ONE COL FOR EACH SAMPLING POINT (6 OR 20).
344 CALL UDHINT(11,1,1U,1H,1D, ELTRMX)
345 10(1,SAMP) = IU
346 10(2,SAMP) = IH
347 10(3,SAMP) = ID
348 CALL UDHINT(41,1,1U,1H,1D, ELTRMX)
349 10(4,SAMP) = IU
350 10(5,SAMP) = IH
351 10(6,SAMP) = ID
352 CALL UDHINT(56,1,1U,1H,1D,ELTRMX)
353 10(7,SAMP) = IU
354 10(8,SAMP) = IH
355 10(9,SAMP) = ID
```
<table>
<thead>
<tr>
<th>LINE #</th>
<th>SOURCE CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>361</td>
<td>CALL UDHINT(76,1,IU,IH,ID,ELTRMX)</td>
</tr>
<tr>
<td>362</td>
<td>IO(10,SAMP) = IU</td>
</tr>
<tr>
<td>363</td>
<td>IO(11,SAMP) = IH</td>
</tr>
<tr>
<td>364</td>
<td>IO(12,SAMP) = ID</td>
</tr>
<tr>
<td>365</td>
<td></td>
</tr>
<tr>
<td>366</td>
<td>IU = 0.05825*(ELTRMX(2,90,1)+ELTRMX(1,90,1))</td>
</tr>
<tr>
<td>367</td>
<td>1 +0.3835*(ELTRMX(2,91,1)+ELTRMX(1,91,1))</td>
</tr>
<tr>
<td>368</td>
<td>2 +0.05825*(ELTRMX(2,90,2)+ELTRMX(1,90,2))</td>
</tr>
<tr>
<td>369</td>
<td>IN = 0.05825*ELTRMX(1,90,1)</td>
</tr>
<tr>
<td>370</td>
<td>1 +0.3835*ELTRMX(1,91,1)</td>
</tr>
<tr>
<td>371</td>
<td>2 +0.05825*ELTRMX(1,90,2)</td>
</tr>
<tr>
<td>372</td>
<td>ID = 0.029125*(ELTRMX(2,90,1)+ELTRMX(1,90,1))</td>
</tr>
<tr>
<td>373</td>
<td>1 +0.05825*ELTRMX(4,90,1)</td>
</tr>
<tr>
<td>374</td>
<td>2 +0.19175*(ELTRMX(2,91,1)+ELTRMX(1,91,1))</td>
</tr>
<tr>
<td>375</td>
<td>3 +0.3835*ELTRMX(4,91,1)</td>
</tr>
<tr>
<td>376</td>
<td>4 +0.029125*(ELTRMX(2,90,2)+ELTRMX(1,90,2))</td>
</tr>
<tr>
<td>377</td>
<td>5 +0.05825*ELTRMX(4,90,2)</td>
</tr>
<tr>
<td>378</td>
<td>IO(13,SAMP) = IU</td>
</tr>
<tr>
<td>379</td>
<td>IO(14,SAMP) = IH</td>
</tr>
<tr>
<td>380</td>
<td>IO(15,SAMP) = ID</td>
</tr>
<tr>
<td>381</td>
<td></td>
</tr>
<tr>
<td>382</td>
<td>CALL UDHINT(76,2,IU,IH,ID,ELTRMX)</td>
</tr>
<tr>
<td>383</td>
<td>IO(16,SAMP) = IU</td>
</tr>
<tr>
<td>384</td>
<td>IO(17,SAMP) = IH</td>
</tr>
<tr>
<td>385</td>
<td>IO(18,SAMP) = ID</td>
</tr>
<tr>
<td>386</td>
<td></td>
</tr>
<tr>
<td>387</td>
<td>CALL UDHINT(56,2,IU,IH,ID,ELTRMX)</td>
</tr>
<tr>
<td>388</td>
<td>IO(19,SAMP) = IU</td>
</tr>
<tr>
<td>389</td>
<td>IO(20,SAMP) = IH</td>
</tr>
<tr>
<td>390</td>
<td>IO(21,SAMP) = ID</td>
</tr>
<tr>
<td>391</td>
<td></td>
</tr>
<tr>
<td>392</td>
<td>CALL UDHINT(41,2,IU,IH,ID,ELTRMX)</td>
</tr>
<tr>
<td>393</td>
<td>IO(22,SAMP) = IU</td>
</tr>
<tr>
<td>394</td>
<td>IO(23,SAMP) = IH</td>
</tr>
<tr>
<td>395</td>
<td>IO(24,SAMP) = ID</td>
</tr>
<tr>
<td>396</td>
<td></td>
</tr>
<tr>
<td>397</td>
<td>CALL UDHINT(11,2,IU,IH,ID,ELTRMX)</td>
</tr>
<tr>
<td>398</td>
<td>IO(25,SAMP) = IU</td>
</tr>
<tr>
<td>399</td>
<td>IO(26,SAMP) = IH</td>
</tr>
<tr>
<td>400</td>
<td>IO(27,SAMP) = ID</td>
</tr>
<tr>
<td>401</td>
<td></td>
</tr>
<tr>
<td>402</td>
<td>IO(28,SAMP) = IO(7,SAMP)</td>
</tr>
<tr>
<td>403</td>
<td>20 CONTINUE</td>
</tr>
<tr>
<td>404</td>
<td></td>
</tr>
<tr>
<td>405</td>
<td>C THE ARRAY IO IS NOW FILLED UP WITH INTENSITIES.</td>
</tr>
<tr>
<td>406</td>
<td>C NEXT DIVIDE BY IU055 (IN 28TH ROW) TO GET FLUX RATIOS IN IO</td>
</tr>
<tr>
<td>407</td>
<td></td>
</tr>
<tr>
<td>408</td>
<td>DO 30 IROW=1,27</td>
</tr>
<tr>
<td>409</td>
<td>DO 31 ICOL=1,NSSAMPS</td>
</tr>
<tr>
<td>410</td>
<td>IO(IROW,ICOL) = IO(IROW,ICOL) / IO(28,ICOL)</td>
</tr>
<tr>
<td>411</td>
<td>31 CONTINUE</td>
</tr>
<tr>
<td>412</td>
<td>30 CONTINUE</td>
</tr>
<tr>
<td>413</td>
<td></td>
</tr>
<tr>
<td>414</td>
<td>C FILL THE ARRAY MINMAX WITH THE MINIMUM AND MAXIMUM VALUES</td>
</tr>
<tr>
<td>415</td>
<td>C FOR EACH FLUX RATIO</td>
</tr>
<tr>
<td>416</td>
<td></td>
</tr>
<tr>
<td>417</td>
<td>DO 34 K=1,27</td>
</tr>
<tr>
<td>418</td>
<td>MINMAX(K,1) = IO(K,1)</td>
</tr>
<tr>
<td>419</td>
<td>MINMAX(K,2) = IO(K,1)</td>
</tr>
<tr>
<td>420</td>
<td>DO 35 SAMP=2,NSSAMPS</td>
</tr>
</tbody>
</table>

Appendix A
SUBROUTINE SMIE(X.RFR.RFI,QSCAT,QEXT,CTBQS,THETO,ELTROM.JX)
C STRAIGHT FROM PETER BARBER'S PROGRAM
C COMMENT C'S ADDED NEAR END TO SUPPRESS PRINTING NO. PARTIAL WAVES
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION T(4),WFNR(2),WFNI(2),ELTROM(4),PI,TAU(3),THETO(91),COSTH(91),SINTH(91)
DIMENSION ACAPR(1050),ACAPI(1050)
C DTR = .01745329250
C RFR**2+RFI**2
RRFR = RFR/C
RRFI = RFI/C
RX = 1.0/X
RRFXR = RRFR*RX
RRFXI = RRFI*RX
NMX1 = IFIX(SNLG(X+4.05D0*X**.333333330+2.0D0))
IF(NMX1.LE.100) GO TO 6
WRITE(*,2)
:2 FORMAT( 'KA IS TOO LARGE')
STOP
IF(NMX1.LT.150) NMX1 = 150
CNAR = 0.000
CNAI = 0.000
DO 7 M = NMX1,NMX2,-1
CN = DBLE(FLOAT(N+1))
CNR = CN*RRFXR
CNI = CN*RRFXI
DR = CNR+CNR
DI = CNI+CNAI
D = DR**2+DI**2
CNAR = CNR-DR/D
CNAI = CNI+DI/D
7 CONTINUE
ACAPR(NMX2) = CNAR
ACAPI(NMX2) = CNAI
NM = NMX2-1
DO 8 N = NM,NM,1,-1
CN = DBLE(FLOAT(N+1))
CNR = CN*RRFXR
CNI = CN*RRFXI
DR = CNR+ACAPR(N+1)
DI = CNI+ACAPI(N+1)
8 D = DR**2+DI**2
ACAPR(N) = CNR-DR/D
ACAPI(N) = CNI+DI/D
481 8 CONTINUE
482 DO 9 J = 1, JX
483 TH = DTR*THETD(J)
484 COSTH(J) = DCOS(TH)
485 SINTH2(J) = DSIN(TH)**2
486 PI(1,J) = 0.000
487 PI(2,J) = 1.000
488 TAU(1,J) = 0.000
489 TAU(2,J) = COSTH(J)
490 9 CONTINUE
491 WMIR = DCOS(X)
492 WMII = -DSIN(X)
493 WFNR(1) = -WMII
494 WFNI(1) = WMIR
495 WFNR(2) = WFNR(1)*RX-WMIR
496 WFNI(2) = WFNI(1)*RX-WMII
497 TCIR = ACAPR(1)*RFRACAPR(1)-RFI+RX
498 TCII = ACAPR(1)*RFRACAPR(1)-RFI
499 TC2R = ACAPR(1)*RFRACAPR(1)-RFI+RX
500 TC2I = ACAPR(1)*RFRACAPR(1)-RFI
501 A = TC1R*WFNR(2)-WFNR(1)
502 B = TC1II*WFNR(2)
503 C = TC1R*WFNR(2)-TC1II*WFNI(2)-WFNR(1)
504 D = TC1II*WFNR(2)+TC1R*WFNI(2)-WFNI(1)
505 CD2 = C**2+D**2
506 FNAR = (A*C+B*D)/CD2
507 FNAI = (B*C-A*D)/CD2
508 A = TC2R*WFNR(2)-WFNR(1)
509 B = TC2II*WFNR(2)
510 C = TC2R*WFNR(2)+TC2II*WFNI(2)-WFNR(1)
511 D = TC2II*WFNR(2)+TC2R*WFNI(2)-WFNI(1)
512 CD2 = C**2+D**2
513 FNBR = (A*C+B*D)/CD2
514 FNB1 = (B*C-A*D)/CD2
515 FNAPR = FNAR
516 FNAPI = FNAI
517 FNBPR = FNBR
518 FNBPI = FNB1
519 QSCAT = 3.000*(FNAR**2+FNBR**2+FNBR**2+FNBI**2)
520 QEXT = 3.000*(FNAR+FNBR)
521 CTBROS = 0.000
522 FNAR = 1.500*FNAR
523 FNAI = 1.500*FNAI
524 FNBR = 1.500*FNBR
525 FNB1 = 1.500*FNB1
526 10 CONTINUE
527 ELTR4X(1,J,1) = FNAR*PI(2,J)+FNBR*TAU(2,J)
528 ELTR4X(2,J,1) = FNAI*PI(2,J)+FNBI*TAU(2,J)
529 ELTR4X(3,J,1) = FNBR*PI(2,J)+FNAR*TAU(2,J)
530 ELTR4X(4,J,1) = FNBI*PI(2,J)+FNAI*TAU(2,J)
531 ELTR4X(1,J,2) = FNAR*PI(2,J)-FNBR*TAU(2,J)
532 ELTR4X(2,J,2) = FNAI*PI(2,J)-FNBI*TAU(2,J)
533 ELTR4X(3,J,2) = FNBR*PI(2,J)-FNAR*TAU(2,J)
534 ELTR4X(4,J,2) = FNBI*PI(2,J)-FNAI*TAU(2,J)
535 10 CONTINUE
536 N = 2
537 DO 15 I = 1, MM02
538 FN = DBLE(FLOAT(M))
539 T(I) = 2.000*FN-1.000
540 T(2) = FN-1.000

Appendix A 49
<table>
<thead>
<tr>
<th>LINE #</th>
<th>SOURCE CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>541</td>
<td>T(3) = 2.000*FN+1.000</td>
</tr>
<tr>
<td>542</td>
<td>DO 11 J = 1, JX</td>
</tr>
<tr>
<td>543</td>
<td>PI(3,J) = (T(1)*PI(2,J)<em>COSTH(J)-FN</em>PI(1,J))/T(2)</td>
</tr>
<tr>
<td>544</td>
<td>TAU(3,J) = COSTH(J)*[PI(3,J)-PI(1,J)]-T(1)*SINH2(J)*PI(2,J)+TAU(1)</td>
</tr>
<tr>
<td>545</td>
<td>1,J</td>
</tr>
<tr>
<td>546</td>
<td>11 CONTINUE</td>
</tr>
<tr>
<td>547</td>
<td>WMIR = WFNR(1)</td>
</tr>
<tr>
<td>548</td>
<td>WMII = WFN(1)</td>
</tr>
<tr>
<td>549</td>
<td>WFNR(1) = WFNR(2)</td>
</tr>
<tr>
<td>550</td>
<td>WFN(1) = WFN(2)</td>
</tr>
<tr>
<td>551</td>
<td>WFNR(2) = T(1)<em>RX</em>WFNR(1)-WMIR</td>
</tr>
<tr>
<td>552</td>
<td>WFN(2) = T(1)<em>RX</em>WFNR(1)-WMII</td>
</tr>
<tr>
<td>553</td>
<td>TCIR = ACAPR(N)*RFR+ACAPI(N)<em>RFI+FN</em>RX</td>
</tr>
<tr>
<td>554</td>
<td>TCII = ACAPI(N)*RFR+ACAPR(N)*RFI</td>
</tr>
<tr>
<td>555</td>
<td>TC2R = ACAPR(N)*RFR+ACAPI(N)<em>RFI+FN</em>RX</td>
</tr>
<tr>
<td>556</td>
<td>TC21 = ACAPI(N)*RFR-ACAPR(N)*RFI</td>
</tr>
<tr>
<td>557</td>
<td>A = TCIR*WFNR(2)-WFNR(1)</td>
</tr>
<tr>
<td>558</td>
<td>B = TCII*WFNR(2)</td>
</tr>
<tr>
<td>559</td>
<td>C = TC1R<em>WFNR(2)-TC1I</em>WFNR(2)-WFNR(1)</td>
</tr>
<tr>
<td>560</td>
<td>D = TCII<em>WFNR(2)+TCIR</em>WFNR(2)-WFNR(1)</td>
</tr>
<tr>
<td>561</td>
<td>CD2 = C<strong>2+D</strong>2</td>
</tr>
<tr>
<td>562</td>
<td>FNAR = (A<em>C+B</em>D)/CD2</td>
</tr>
<tr>
<td>563</td>
<td>FNAI = (B<em>C-A</em>D)/CD2</td>
</tr>
<tr>
<td>564</td>
<td>A = TC2R*WFNR(2)-WFNR(1)</td>
</tr>
<tr>
<td>565</td>
<td>B = TC2I*WFNR(2)</td>
</tr>
<tr>
<td>566</td>
<td>C = TC2R<em>WFNR(2)-TC2I</em>WFNR(2)-WFNR(1)</td>
</tr>
<tr>
<td>567</td>
<td>D = TC2I<em>WFNR(2)+TC2R</em>WFNR(2)-WFNR(1)</td>
</tr>
<tr>
<td>568</td>
<td>CD2 = C<strong>2+D</strong>2</td>
</tr>
<tr>
<td>569</td>
<td>FNBR = (A<em>C+B</em>D)/CD2</td>
</tr>
<tr>
<td>570</td>
<td>FNBI = (B<em>C-A</em>D)/CD2</td>
</tr>
<tr>
<td>571</td>
<td>T(4) = T(1)/(FN*T(2))</td>
</tr>
<tr>
<td>572</td>
<td>T(2) = T(2)*(FN+1.000)/FN</td>
</tr>
<tr>
<td>573</td>
<td>QMSQ = QMSQ+T(2)<em>(FNAPR</em>FNAR+FNAPI<em>FNAI+FNBR</em>FNBP+FNBPI*FNBI)</td>
</tr>
<tr>
<td>574</td>
<td>1+T(4)<em>(FNAPR</em>FNBP+FNAPI*FNBPI)</td>
</tr>
<tr>
<td>575</td>
<td>T(4) = FNAR<strong>2+FNBI</strong>2+FNBP<strong>2+FNBR</strong>2</td>
</tr>
<tr>
<td>576</td>
<td>QEXT = QEXT+T(3)*T(4)</td>
</tr>
<tr>
<td>577</td>
<td>T(2) = FN*(FN+1.000)</td>
</tr>
<tr>
<td>578</td>
<td>T(1) = T(3)/T(2)</td>
</tr>
<tr>
<td>579</td>
<td>K = (N/2)-2</td>
</tr>
<tr>
<td>580</td>
<td>DO 13 J = 1, JX</td>
</tr>
<tr>
<td>581</td>
<td>ELTRMR(1,J,1) = ELTRMR(1,J,1)+T(1)<em>(FNMP</em>PI(3,J)+FNMR*TAU(3,J))</td>
</tr>
<tr>
<td>582</td>
<td>ELTRMR(2,J,1) = ELTRMR(2,J,1)+T(1)<em>(FNAPR</em>PI(1,J)+FNBR*TAU(1,J))</td>
</tr>
<tr>
<td>583</td>
<td>ELTRMR(3,J,1) = ELTRMR(3,J,1)+T(1)<em>(FNAPR</em>PI(3,J)+FNBR*TAU(3,J))</td>
</tr>
<tr>
<td>584</td>
<td>ELTRMR(4,J,1) = ELTRMR(4,J,1)+T(1)<em>(FNAPR</em>PI(3,J)+FNBR*TAU(3,J))</td>
</tr>
<tr>
<td>585</td>
<td>IF(EQ.N) GO TO 12</td>
</tr>
<tr>
<td>586</td>
<td>ELTRMR(1,J,2) = ELTRMR(1,J,2)+T(1)<em>(FNAPR</em>PI(3,J)-FNMR*TAU(3,J))</td>
</tr>
<tr>
<td>587</td>
<td>ELTRMR(2,J,2) = ELTRMR(2,J,2)+T(1)<em>(FNAPR</em>PI(3,J)-FNMR*TAU(3,J))</td>
</tr>
<tr>
<td>588</td>
<td>ELTRMR(3,J,2) = ELTRMR(3,J,2)+T(1)<em>(FNAPR</em>PI(3,J)-FNMR*TAU(3,J))</td>
</tr>
<tr>
<td>589</td>
<td>ELTRMR(4,J,2) = ELTRMR(4,J,2)+T(1)<em>(FNAPR</em>PI(3,J)-FNMR*TAU(3,J))</td>
</tr>
<tr>
<td>590</td>
<td>GO TO 13</td>
</tr>
<tr>
<td>591</td>
<td>13 CONTINUE</td>
</tr>
<tr>
<td>592</td>
<td>IF(T(4).LT.1.00-14) GO TO 20</td>
</tr>
<tr>
<td>593</td>
<td>N = N+1</td>
</tr>
<tr>
<td>594</td>
<td>DO 14 J = 1, JX</td>
</tr>
<tr>
<td>595</td>
<td>PI(1,J) = PI(2,J)</td>
</tr>
</tbody>
</table>

Appendix A

50
### Appendix A

#### 1. Calculation of Matrix Elements

The program `HILF.O` defines several matrix elements, including:

- **Element 1:**
  \[ T(1,J) = T(3,J) \]

- **Element 14:**
  \[ FNAPR = FNAR \]

- **Element 25:**
  \[ QSCAT = 0.000 \]

- **Element 30:**
  \[ C = 2.000*RX**2 \]

#### 2. Subroutine Calculation

The subroutine `UDHINT(JJ,KK,IU,IH,ID,ELTRMX)` is defined to calculate matrix elements for a given angle, with correction for finite detector acceptance angle accomplished by adding in bits from scattering angles one degree higher and lower.

#### 3. Format Output

The program uses format `(16)` for output:

\[ WRITE(6,200) N \]

#### 4. Example Calculation

For given integers `JJ` and `KK`, the subroutine returns combinations of the scattering matrix elements with correction for finite detector acceptance angle.

**Example Calculation:**

- For `IU`, `IH`, and `ID` as defined:
  - `IU = 0.05825*(ELTRMX(2,JJ-1,KK)+ELTRMX(1,JJ-1,KK))`
  - `IH = 0.05825*ELTRMX(1,JJ,KK)`
  - `ID = 0.029125*(ELTRMX(2,JJ-1,KK)+ELTRMX(1,JJ-1,KK))`

#### 5. Double Precision

The program uses double precision for calculations.

**Example Calculation:**

- `IU = 0.3835*(ELTRMX(2,JJ,KK)+ELTRMX(1,JJ,KK))`
- `IH = 0.19175*(ELTRMX(2,JJ,KK)+ELTRMX(1,JJ,KK))`
- `ID = 0.029125*(ELTRMX(2,JJ+1,KK)+ELTRMX(1,JJ+1,KK))`
APPENDIX C
INVERT.F LISTING

<table>
<thead>
<tr>
<th>LINE #</th>
<th>SOURCE CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PROGRAM INVERT</td>
</tr>
<tr>
<td>2</td>
<td>REAL ORDMIN(14,20200), ORDMAX(14,20200), EXPDAT(1000,25)</td>
</tr>
<tr>
<td>3</td>
<td>REAL CALDAT(25), AVG, STD, SPX, SPHLIM, UNCAVG, UNCFIX</td>
</tr>
<tr>
<td>4</td>
<td>REAL FLUX(14), UNC(14), HIVAL(14), LOVAL(14)</td>
</tr>
<tr>
<td>5</td>
<td>INTEGER NDMIN(14,20200), NDMAX(14,20200), NPARTS</td>
</tr>
<tr>
<td>6</td>
<td>INTEGER I, J, K, NUMCHN, HITS(20200), MISSES(20200)</td>
</tr>
<tr>
<td>7</td>
<td>INTEGER MISLIM, ITRY, ILO, IHI, COUNT, DENS</td>
</tr>
<tr>
<td>8</td>
<td>CHARACTER*12 ORNFIL(14), ORXFIL(14), NDNFIL(14), NDXFIL(14)</td>
</tr>
<tr>
<td>9</td>
<td>CHARACTER*24 EXPNAM, OUTNAI4, CALNAI4, OLDNN4</td>
</tr>
<tr>
<td>10</td>
<td>CHARACTER*7 DATE</td>
</tr>
<tr>
<td>11</td>
<td>Below, left side expressions are character variables used</td>
</tr>
<tr>
<td>12</td>
<td>only in do loops to read in the previously calculated data.</td>
</tr>
<tr>
<td>13</td>
<td>Right side strings are the file names containing the data.</td>
</tr>
<tr>
<td>14</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>ORNFIL(1) = 'ORD_U040.MIN'</td>
</tr>
<tr>
<td>16</td>
<td>ORXFIL(1) = 'ORD_U040.MAX'</td>
</tr>
<tr>
<td>17</td>
<td>NDNFIL(1) = 'NDX_U040.MIN'</td>
</tr>
<tr>
<td>18</td>
<td>NDXFIL(1) = 'NDX_U040.MAX'</td>
</tr>
<tr>
<td>19</td>
<td>ORNFIL(2) = 'ORD_U075.MIN'</td>
</tr>
<tr>
<td>20</td>
<td>ORXFIL(2) = 'ORD_U075.MAX'</td>
</tr>
<tr>
<td>21</td>
<td>NDNFIL(2) = 'NDX_U075.MIN'</td>
</tr>
<tr>
<td>22</td>
<td>NDXFIL(2) = 'NDX_U075.MAX'</td>
</tr>
<tr>
<td>23</td>
<td>ORNFIL(3) = 'ORD_U090.MIN'</td>
</tr>
<tr>
<td>24</td>
<td>ORXFIL(3) = 'ORD_U090.MAX'</td>
</tr>
<tr>
<td>25</td>
<td>NDNFIL(3) = 'NDX_U090.MIN'</td>
</tr>
<tr>
<td>26</td>
<td>NDXFIL(3) = 'NDX_U090.MAX'</td>
</tr>
<tr>
<td>27</td>
<td>ORNFIL(4) = 'ORD_U105.MIN'</td>
</tr>
<tr>
<td>28</td>
<td>ORXFIL(4) = 'ORD_U105.MAX'</td>
</tr>
<tr>
<td>29</td>
<td>NDNFIL(4) = 'NDX_U105.MIN'</td>
</tr>
<tr>
<td>30</td>
<td>NDXFIL(4) = 'NDX_U105.MAX'</td>
</tr>
<tr>
<td>31</td>
<td>ORNFIL(5) = 'ORD_U125.MIN'</td>
</tr>
<tr>
<td>32</td>
<td>ORXFIL(5) = 'ORD_U125.MAX'</td>
</tr>
<tr>
<td>33</td>
<td>NDNFIL(5) = 'NDX_U125.MIN'</td>
</tr>
<tr>
<td>34</td>
<td>NDXFIL(5) = 'NDX_U125.MAX'</td>
</tr>
<tr>
<td>35</td>
<td>ORNFIL(6) = 'ORD_U140.MIN'</td>
</tr>
<tr>
<td>36</td>
<td>ORXFIL(6) = 'ORD_U140.MAX'</td>
</tr>
<tr>
<td>37</td>
<td>NDNFIL(6) = 'NDX_U140.MIN'</td>
</tr>
<tr>
<td>38</td>
<td>NDXFIL(6) = 'NDX_U140.MAX'</td>
</tr>
<tr>
<td>39</td>
<td>ORNFIL(7) = 'ORD_H040.MIN'</td>
</tr>
<tr>
<td>40</td>
<td>ORXFIL(7) = 'ORD_H040.MAX'</td>
</tr>
<tr>
<td>41</td>
<td>NDNFIL(7) = 'NDX_H040.MIN'</td>
</tr>
<tr>
<td>42</td>
<td>NDXFIL(7) = 'NDX_H040.MAX'</td>
</tr>
<tr>
<td>43</td>
<td>ORNFIL(8) = 'ORD_H075.MIN'</td>
</tr>
<tr>
<td>44</td>
<td>ORXFIL(8) = 'ORD_H075.MAX'</td>
</tr>
<tr>
<td>45</td>
<td>NDNFIL(8) = 'NDX_H075.MIN'</td>
</tr>
<tr>
<td>46</td>
<td>NDXFIL(8) = 'NDX_H075.MAX'</td>
</tr>
<tr>
<td>47</td>
<td>ORNFIL(9) = 'ORD_H090.MIN'</td>
</tr>
<tr>
<td>48</td>
<td>ORXFIL(9) = 'ORD_H090.MAX'</td>
</tr>
<tr>
<td>49</td>
<td>NDNFIL(9) = 'NDX_H090.MIN'</td>
</tr>
<tr>
<td>50</td>
<td>NDXFIL(9) = 'NDX_H090.MAX'</td>
</tr>
</tbody>
</table>
The previously calculated data is now all set up in RAM. Read in the experimental data from a group of ~500 particles, i.e.,
file outputted from TRANS.EXE such as E118A.PRN. The length
of the filename cannot be known ahead of time, so it must be
determined during runtime. The maximum length of the filename
should not exceed 24 characters; I will also assume that the
number of particles represented by the data does not exceed
1000.

50 WRITE (*.*) ' ENTER PATH/FI G LE NAME OF DATA SET TO
PROCESS'
WRITE (*.*) ' EXAMPLES: 1183.PRN or C:\WYATT\V107B.PRN'
WRITE (*.*) ' or enter x to quit program'
READ (*.A) EXPNAM
IF ((EXPNAM(1:1) .EQ. 'X') .OR. (EXPNAM(1:1) .EQ. 'x')) THEN
GO TO 270
ENDIF
OPEN (UNIT=3, FILE=EXPNAM, ACCESS='SEQUENTIAL',
& FORM='FORMATTED', STATUS='OLD')

Pause here to construct OUTNAM, the name of the output file
which will be written at the end of this entire program. It
will, by the construction of its name, go to whatever sub-
directory the data set (EXPNAM) came from. In effect we will
just substitute .MAP for .PRN in the string held by EXPNAM.

IF (I .GE. 22) GO TO 50
IF (EXPNAM(I:I) .NE. '.') GO TO 60
OUTNAM = EXPNAM(1:I) // 'MAP'
The '.' is in the position corresponding to the current
value of I. The total length of OUTNAM (and EXPNAM) is
I+3 characters.

Back to reading in the data:

DO 70 I=1,1000
READ (3.*,END=80) (EXPDAT(I,J), J=1,25)
70 CONTINUE
Control should jump from the do-loop to statement 80 when EOF
is reached, presumably before I=1000. When it exits the
current value of I-I should be the
number of rows of data, i.e., the number of particles in the
data set (nparts).

80 CLOSE(3)
NPARTS = I-1

Somewhere on disk I must have a file of correction factors
previously determined in a calibration procedure and applicable
to the experimental data which is here being analyzed. Those
factors must be read in and multiplied against the corresponding
columns of EXPDAT. Assume the format of the calibration file is

Appendix C 65
First line gives date of calibration and the data set (of spheres) used to perform the calibration. Thereafter the left column gives the fiber optic channel number and the right column the corresponding multiplicative correction factor. The channels labeled 0 and 15 are the bundles, and we'll just set the correction factor to 1. 25 lines in all.

```fortran
WRITE (*,*) 'ENTER THE PATH/FILE NAME OF THE APPLICABLE'
WRITE (*,*) 'CALIBRATION FILE. Enter L to just re-use the Latest'
WRITE (*,*) 'filename used this session.'
READ (*,*) CALNAM
IF ((CALME(I:1) .EQ. 'L') .OR. (CALNAM(1:1) .EQ. '1')) GO TO 85
OLDNAM = CALNAM
OPEN (UNIT=3, FILE=CALNAM, ACCESS='SEQUENTIAL',
      FORM='FORMATTED', STATUS='OLD')
READ (3,*) (A)
READ(3,*), NUNCHN, CALDAT(K)
CONTINUE
CLOSE (3)
```

Now make the calibration correction:

```fortran
DO 100 I=1, NPARTS
   EXPDAT(I,K) = EXPDAT(I,K)*CALDAT(K)
100 CONTINUE
```

Now the array EXPDAT contains the corrected data values, ready for use.

```fortran
DO 120 J=1, 20200
   HITS(J) = 0
   MISSES(J) = 0
120 CONTINUE
```

Now begins the particle do-loop; i.e., we look at particles one at a time (the Ith particle)

```fortran
DO 200 I=1, NPARTS
```
PRGA
INVERT.F
VERSION
Thu Jan 16 12:05:06 1992

LINE # SOURCE CODE PAGE 5

241 AVG = (EXPDAT(I,13) + EXPDAT(I,14) + EXPDAT(I,15) + EXPDAT(I,16))
242 & + EXPDAT(I,22) + EXPDAT(I,23) + EXPDAT(I,24) + EXPDAT(I,25))
243 & /8.0
244 STD =SQR((EXPDAT(I,13)-AVG)**2 + (EXPDAT(I,14)-AVG)**2
245 & + (EXPDAT(I,15)-AVG)**2 + (EXPDAT(I,16)-AVG)**2
246 & + (EXPDAT(I,22)-AVG)**2 + (EXPDAT(I,23)-AVG)**2
247 & + (EXPDAT(I,24)-AVG)**2 + (EXPDAT(I,25)-AVG)**2)/8.0)
248 SPX = 1.0 - (STD/(2.6458*AVG))
249
250 C If this particle isn't a good sphere, stop right now
251 SPHLIM = 0.9
252 IF (SPX .LT. SPHLIM) GO TO 200
253
254 C The uncertainty to be associated with each data point (in
255 C EXPDAT) depends upon the magnitude of the data, generally
256 C getting worse as the intensity dims. I don't yet know how to
257 C calculate the uncertainty, but I will assume that it will be by
258 C exactly the same formula for all channels. Give the observables
259 C numbers (1-14) for identification. See page 14 hand notes.
260 C UNCAVG = f(avg) not yet known
261
262 UNCAVG = f(avg) not yet known
263 FLUX(1) = EXPDAT(I,11)
264 FLUX(2) = EXPDAT(I,18)
265 FLUX(3) = EXPDAT(I,19)
266 FLUX(4) = EXPDAT(I,20)
267 FLUX(5) = EXPDAT(I,15)
268 FLUX(6) = EXPDAT(I,16)
269 FLUX(7) = EXPDAT(I,10)
270 FLUX(8) = EXPDAT(I,17)
271 FLUX(9) = EXPDAT(I,19)
272 FLUX(10) = EXPDAT(I,21)
273 FLUX(11) = EXPDAT(I,3)
274 FLUX(12) = EXPDAT(I,4)
275 FLUX(13) = EXPDAT(I,12)
276 FLUX(14) = EXPDAT(I,18)
277
278 DO 124 J=1,14
279 C function of flux(j) still unknown
280 HIVAL(J) = (FLUX(J)/AVG) * (1.0 + UNCAVG(J))
281 LOVAL(J) = (FLUX(J)/AVG) * (1.0 - UNCAVG(J))
282
283 124 CONTINUE
284
285 C Now we do a do-loop over the observables, writing to MISSES
286 C the result of each observable trial
287
288 DO 150 K=1,14
289 C ILO = 1
290 IHI = 20200
291 ITRY = 10100
292 DO 130 COUNT = 1,15
293 IF (ORDMIN(K,ITRY) .LT. HIVAL(K)) THEN
294 ILO = ITRY
295 ITRY = INT((ITRY+IHI)/2)
296 ELSE
297 IHI = ITRY
298 ITRY = INT((ITRY+ILO)/2)
299 END IF
300 130 CONTINUE

Appendix C 67
### PROGRAM INVERT.F
**VERSION**
Thu Jan 16 12:05:06 1992

#### SOURCE CODE PAGE 6

<table>
<thead>
<tr>
<th>LINE #</th>
<th>SOURCE CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>301</td>
<td>130</td>
</tr>
<tr>
<td>302</td>
<td>CONTINUE</td>
</tr>
<tr>
<td>303</td>
<td></td>
</tr>
<tr>
<td>304 C</td>
<td>Now ITRY is the element of ORDMIN whose value equals HIVAL(K)</td>
</tr>
<tr>
<td>305</td>
<td>DO 132 L=ITRY,20200</td>
</tr>
<tr>
<td>306</td>
<td>MISSES(NDXMIN(K,L)) = MISSES(NDXMIN(K,L)) + 1</td>
</tr>
<tr>
<td>307</td>
<td>132</td>
</tr>
<tr>
<td>308</td>
<td>CONTINUE</td>
</tr>
</tbody>
</table>
| 309 C  | Repeat the above for ORDMAX:
| 310    | ILO = 1     |
| 311    | INI = 20200 |
| 312    | ITRY = 10100|
| 313    | DO 134 COUNT=1,15 |
| 314    | IF (ORDMAX(K,ITRY) .LT. LOVAL(K)) THEN |
| 315    | ILO = ITRY |
| 316    | ITRY = INT((ITRY+IHI)/2) |
| 317    | ELSE       |
| 318    | INI = ITRY |
| 319    | ITRY = INT((ITRY+ILO)/2) |
| 320    | END IF     |
| 321    | 134         |
| 322    | CONTINUE    |
| 323 C  | This time at the end of the loop ITRY is the element of ORDMAX whose value equals (closely enough) LOVAL. The elements of NDXMAX from 1 up to ITRY contain the numbers (ie, names) of some more missed pixels. Increment MISSES accordingly. |
| 324    | DO 136 L=1,ITRY |
| 325    | MISSES(NDXMAX(K,L)) = MISSES(NDXMAX(K,L)) + 1 |
| 326    | 136         |
| 327    | CONTINUE    |
| 328    | This completes the incrementing of MISSES for this one (kth) observable. Loop back for the next observable. |
| 329    | 150         |
| 330    | CONTINUE    |
| 331 C  | The array MISSES is now filled, for this one particle. Each of its 20200 elements contains an integer from 0 to 14, which counts the number of times the corresponding pixel FAILED to be in agreement with the data. |
| 332    |             |
| 333 C  | Next step for this particle is to increment the array HITS. For testing purposes, will tolerate three different levels of hitting: eg. no misses, 1-2 misses, and 3-4 misses. |
| 334 C  | (Later, define only two levels of hitting: zero or 1 misses and 2 or more misses) |
| 335 C  | Whenever HITS is incremented, reset MISSES to zeros. |
| 336    |             |
| 337    |             |
| 338    |             |
| 339 C  | DO 160 J=1,20200 |
| 340    | IF (MISSES(J) .LE. 1) THEN |
| 341    | HITS(J) = HITS(J) + 1 |
| 342    | END IF     |
| 343    | MISSES(J) = 0 |
| 344    | 160         |
| 345    | CONTINUE    |
| 346    |             |
| 347    |             |
| 348    |             |
| 349    |             |
| 350    |             |
| 351    |             |
| 352    |             |
| 353    |             |
| 354 C  | That does it for this particle, go back for the next one. |
| 355    | 200         |
| 356    | CONTINUE    |
| 357 C  | Now all the particles have been analyzed, and HITS contains the distribution over all particles (ie, each element of HITS contains an integer which is the number of particles whose r and n could have been the same as the pixel's r and n). Output the |

Appendix C 68
```plaintext
C array HITS - but sensibly; if there are no hits in a pixel, just
C skip that pixel, don't write out a zero. DENS is a parameter
C that will control how darkly a pixel square is printed under
C PageGarden. First include EXPNAM and UNCFIX on picture
C
C OPEN (UNIT=4, FILE=OUTNAM, ACCESS='SEQUENTIAL',
& FORM='FORMATTED', STATUS='NEW')
C
C WRITE(4,*) ' disp 4.5* 1.5* "TEST CASE:" '
C WRITE(4,201) EXPNAM
C 201 FORMAT (' dispc ',A,'"')
C WRITE(4,*) ' disp 4.5* 1.75* "UNCERTAINTY: +/- " '
C WRITE(4,202) 100.0*UNCFIX
C 202 FORMAT (' dispc F4.1,'"")
C
DO 220 J-1,20200
IF (HITS(J) .EQ. 0) THEN
GO TO 220
ELSE
DENS = 60
END IF
NPOSX = 525 + 12*MOO((J-1)/200)
NPOSY = 1870 - 12*INT((J-1)/200)
WRITE(4,250) NPOSX, NPOSY, DENS
250 FORMAT(1X, 'area ',I4, 'd ',I4, 'd', ' 12d 12d ',I3,/',
& 1X, 'box +0 +0 12d 12d 1d')
220 CONTINUE
CLOSE(4)
```

Appendix C

69