DRA

N84-21117

FINAL REPORT

BETA SYSTEMS ERROR ANALYSIS

January 26, 1984

Contract No. NAS8-35329

Prepared for

George C. Marshall Space Flight Center
Marshall Space Flight Center, AL 35812

Applied Research Inc.
P.O. Box 11220 • Huntsville, AL 35805 • (205) 837-3600

FINAL REPORT

BETA SYSTEMS ERROR ANALYSIS

January 26, 1984

Contract No. NAS8-35329

Prepared for

George C. Marshall Space Flight Center
Marshall Space Flight Center, AL 35812

Applied Research Inc.

P.O. Box 11220 • Huntsville, AL 35805 • (205) 837-8600

TABLE OF CONTENTS

LIST OF FIGURES

1.0 Introduction

Since 1980, personnel of Applied Research, Inc. have supported NASA at Marshall Space Flight Center in the measurement of the atmospheric backscatter coefficient, β, with an airborne CO Laser Doppler Velocimeter (LDV) system operating in a continuous wave, focussed mode. A method, called the Single Particle Mode (SPM) algorithm, has been developed from concept through analysis of an extensive amount of data obtained with the system aboard a NASA aircraft. The SPM algorithm is intended to be employed in situations where one particle at a time appears in the sensitive volume of the LDV. In addition to giving the backscatter coefficient, the SPM algorithm also produces as intermediate results the aerosol density and the aerosol backscatter cross-section distribution.

A second method, which measures only the atmospheric backscatter coefficient, is called the Volume Mode (VM) and was simultaneously employed in obtaining the aforementioned data. The results of these two methods generally differed by slightly less than an order of magnitude. The purpose of this report is to examine the measurement uncertainties or other errors in the results of the two methods.

A review of the basis of each method is given in Section 2, including a discussion of the computer programming implementation of the SPM. A discussion of error inherent in each is given in Section 3, with conclusions summarized in Section 4.

2.0 Review of the Algorithms for Determining the Backscatter Coefficient

For convenience, the basis of the VM and SPM algorithms for obtaining the atmospheric backscatter coefficient β is presented here. It will be seen that the VM method works under more general conditions than the SPM method, but that the latter gives more information, namely the aerosol density and backscatter distribution. Each method has its own special calibration requirements.

2.1 Volume Mode Algorithm

The VM algorithm will now be derived from the response of the LDV to a single particle in its sensitive volume. This expression will be compared to the well known result for the signal-to-noise, S, of a focussed, cw LDV operating in the conventional "volume mode" with very many particles in its sensitive volume: $S = \beta G_V$ where

$$\beta = \int \sigma \; n(\sigma) d\sigma$$

σ = single particle backscatter cross-section (m^2)

$n(\sigma)$ = backscatter cross-section distribution (m^{-5})

G_V = volume mode gain factor determined by calibration (m).

In this derivation, the system will be assumed to be aircraft borne with optical axis perpendicular to the aircraft velocity vector. In this case each aerosol particle passes through the sensitive volume perpendicular to the optic axis with velocity v. See Figure 1. A particle at x,y,z with cross-section σ produces a signal-to-noise at time t given by

$$S(\sigma,t) = \sigma u(x,y,z)$$

where u(x,y,z) defines the LDV sensitive volume. That is, requiring S>1 defines a sensitive volume dependent on σ. Extending this expression to many particles with cross-section σ moving in the positive x direction, each with initial position x_{oi} such that

$$x_i = vt + x_{oi} \quad \text{gives}$$

$$S(\sigma,t) = \sigma \sum_i u(vt + x_{oi}, y_i, z_i)$$

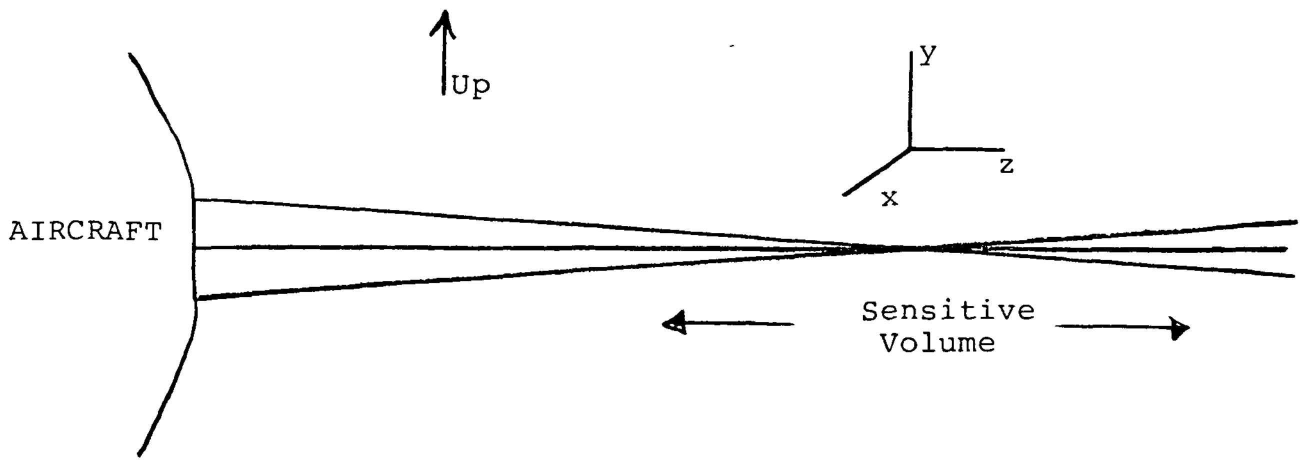
Further extending to a time average over M time intervals Δt gives

$$S(\sigma) = \sum_{j=1}^{M} S(\sigma,t_j)/M$$

$$= \sigma \sum_i u(y_i, z_i) m_i /M$$

with $$u(y_i, z_i) = \sum_j u(vt_j + x_{oi}, y_i, z_i)/m_i$$

and m_i is the number of integration points within the sensitive volume. This quantity depends on the trajectory of the particle



y is up

z is laser beam

x is along aircraft velocity vector

Particles penetrate perpendicular to the yz plane

Figure 1. Geometry and Coordinates

defined by the constants y_i and z_i through the sensitive volume and is independent of velocity. Actually

$$m_i = \Delta x(y_i, z_i)/v\Delta t$$

where $x(y_i, z_i)$ is the width of the sensitive volume at y_i and z_i. Once again extending the expression by replacing the summation over particle number by integration over volume and the uniform density $n(\sigma)$ gives

$$S(\sigma) = n(\sigma)\sigma\int u(y,z)\Delta x(y,z)dydz$$

$$= n(\sigma)\sigma G(\sigma).$$

A possible dependence of the gain factor G on σ is emphasized because of its importance in case this method is used when a single particle at a time is sensed. That is, if an effective threshold exists at S=1, the sensitive volume is defined by S>1, and the single particle sensitive volume is smaller than the many particle sensitive volume because many particles outside the single particle volume can build up a signal S>1. In order to include all particle sizes, the expression must be integrated over σ:

$$S = \int n(\sigma)\sigma G(\sigma)d\sigma.$$

Putting this in the form given at the beginning of this section such that β can be isolated requires defining

$$G_{VM} = \int n(\sigma)\,\sigma\, G(\sigma)\, d\sigma \Big/ \int n(\sigma)\,\sigma\, d\sigma$$

so that

$$S = \beta\, G_{VM} \quad .$$

Note that $G_{VM} < G_V$, the conventional many particle volume mode gain, if $G(\sigma)$ is dependent on σ. This means that, if β is calculated from a measured signal S obtained from an average of single particle signals with $\beta = S/G_V$, too small a result would be obtained.

The basis of the VM method for measuring the backscatter cross-section has been developed. However it has been shown that the gain factor may depend on whether data is taken in a many particle or single particle situation.

2.2 Single Particle Mode Algorithm

The SPM algorithm functions by recording the peak signal from each particle which transits the sensitive volume. The geometry is the same as described previously. Single particle transits are assumed not to overlap in time. From the statistics of the peak signal distribution and knowledge of sensitivity contours within the sensitive volume, one may derive the particle density, backscatter cross-section distribution, and atmospheric backscatter coefficient. The algorithm will now be described using discrete mathematics with direct application to the computational programs.

Consider an interval of the single particle backscatter cross-section axis from σ_L to σ_H which covers all particles seen, and which itself is divided into M intervals. The backscatter coefficient will be taken as

$$\beta = \sum_{j=1}^{M} n_j \sigma_j$$

where

n_j = number of particles per unit volume within the jth interval

σ_j = cross-section at the center of the jth interval.

Suppose that the total number of particles seen per unit volume is D; then

$$w_j = n_j/D$$

is a probability distribution since

$$\sum_{j=1}^{M} w_j = \sum_{j=1}^{M} n_j/D = 1$$

Since each σ_j has a particular amount of the sensitive volume V_j in which the particle would give a signal above threshold S_L, the total number of particles seen, N, is

$$N = \sum_{j=1}^{M} n_j V_j = D \sum_{j=1}^{M} w_j V_j$$

and the density D is

$$D = N / \sum_{j=1}^{M} w_j V_j$$

which says that the relevant volume is an average volume weighted by the cross-sectional distribution w_j. Also

$$\beta = \sum_{j=1}^{M} n_j \sigma_j / 4\Pi = D \sum_{j=1}^{M} w_j \sigma_j / 4\Pi$$

$$= D \langle\sigma\rangle / 4\Pi$$

so that β is given by the particle density times the average cross-section. The SPM algorithm determines the probability distribution w_j through the relationship between the peak signal distribution and the sensitivity contours within the sensitive volume. This same information also determines the density.

In order to understand the SPM algorithm, the concept of the volume V_j sensitive to a particle of cross-section σ_j must be clear. For the geometry previously described with the laser axis perpendicular to the aircraft vector, this volume is a cylinder with axis in the direction of the aircraft velocity vector and length vt, where v is the aircraft speed and t is the time of observation. This cylinder intersects a vertical plane through the optic axis in an area A_j . The definition of the volume V_j is that particles σ_j passing through the area A_j will give a peak signal S (signal-to-noise) above threshold S_L . The gain factor g(y,z) in the equation $S_L \leq g(y,z)$ determines the area A_j, where y,z are coordinates in the vertical plane of the sensitive volume. Furthermore, the notation g(A) is used to mean the gain on the contour surrounding the area A. These concepts and the relationship between the σ and S distributions are shown in Figure 2a. The line g(0) represents the maximum gain of the sensitive volume because the maximum gain occurs at a point. Three σ and S intervals are shown in this figure, with boundaries related by $S = \sigma g(0)$. Any other gain is represented by another line with slope $g(a) < g(0)$. Notice that a particle in interval $\Delta\sigma_2$ may contribute to ΔS_1 and ΔS_2 , depending on the gain, but not

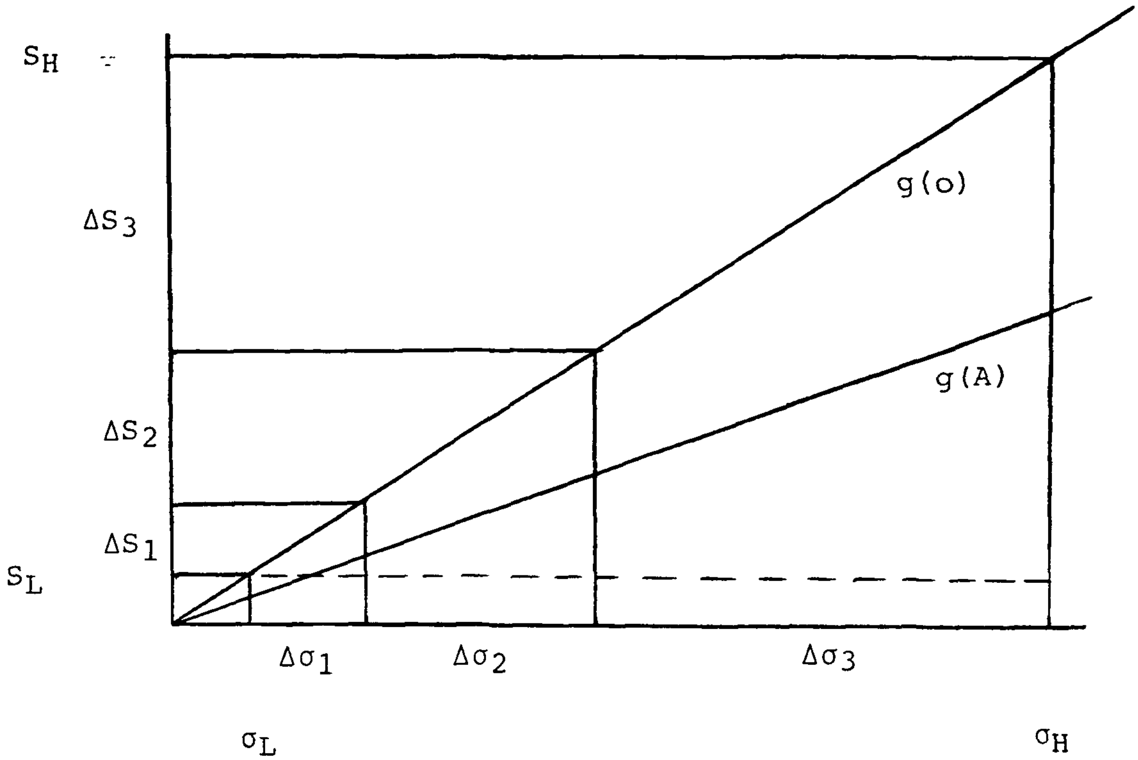


Figure 2a.

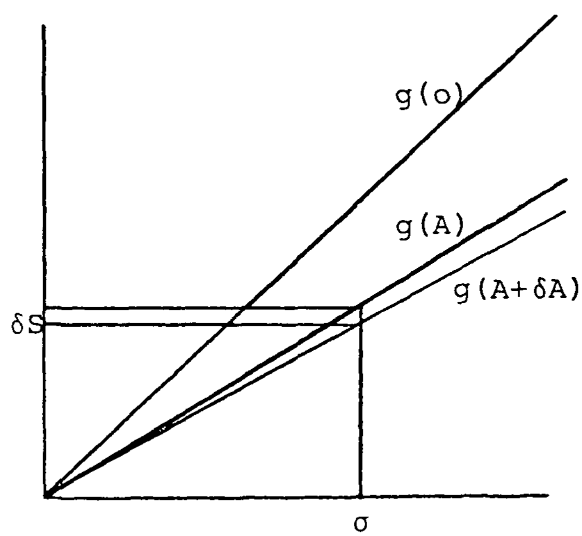


Figure 2b.

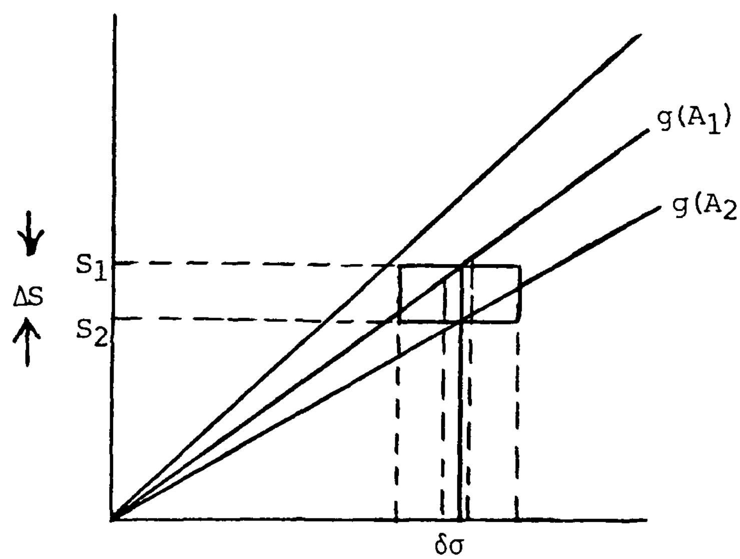


Figure 2c.

Figure 2. Peak Signal versus Backscatter Cross-section Diagrams

to ΔS_3 since this would require a gain larger than g(0).

In order to relate the σ and S distributions, consider Figure 2b. Particles with a number per unit area density $n(\sigma)$ at σ contribute signals ΔN within δS according to $\Delta N = \delta \dot{A} n(\sigma)$ To determine the contribution of a macroscopic interval $\Delta\sigma$ to a macroscopic interval ΔS, as shown in Figure 2c, the intergral

$$\Delta N = \int_{\Delta\sigma} n(\sigma)\, \Delta A(\sigma)\, \delta\sigma$$

is evaluated where $n(\sigma)$ is the number of particles per unit volume per σ increment at σ, $\Delta A(\sigma) = A_2 - A_1$, and $\delta\sigma$ is a subinterval in σ. Notice that the areas can be considered functions of the gain, and therefore $A_2 = A(S_2/\sigma)$ and $A_1 = A(S_1/\sigma)$. This function has been determined by calibration and is included in the computational program (Section 2.3, 2.4 and Appendix in tabular form in subroutine DATANAL. The method of obtaining these data is described in Reference 1. In order to perform the above integral, the probabilities behind the $n(\sigma)$ are assumed (w_j in previous discussion) and the probability of obtaining ΔN particles is calculated in subroutine OPTIM. The actual numbers of particles are obtained by multiplying by the total number of particles observed. Contributions to each ΔS interval from each possible $\Delta\sigma$ interval are evaluated. The assumed distribution $n(\sigma)$ is then varied until the best fit to the peak signal distribution is obtained in the least squares sense. With this "best fit" distribution the density, average cross-section $\langle\sigma\rangle$, and backscatter coefficient β are determined.

2.3 Computer Implementation

Implementation of the β prediction algorithms involved setting several parameters and making certain assumptions. These parameters/assumptions have little precedence, and therefore Applied Research will identify these variables/assumptions as clearly as possible.

1. The number of signal bins = 6. Six bins were processed and used for the inversion. A seventh bin was identified and used to store all large particle signals, but not considered in the inversion.
2. The number of processor bins which made up each of the six signal bins were:

   Signal bin 1 = processor bins 5-8

   Signal bin 2 = processor bins 9-16

   Signal bin 3 = processor bins 17-32

   Signal bin 4 = processor bins 33-64

   Signal bin 5 = processor bins 65-128

   Signal bin 6 = processor bins 129-255

   Signal bin 7 = processor bin 256

   Note that signal bins 1-6 contain progressively twice as many processor bins as the previous signal bin. This type of consolidation was done because the lower order processor bins get many more signals/bin than the higher order bins.

   Note also that processor bins 1-4 are not used because the threshold was set at 4.

3. The signal processor threshold is equal to 4.

4. The volume channel bandwidth is 860 Khz.

5. The single particle processor bandwidth is 1.5 Mhz.

6. The LDV is always focussed at 10 meters.

7. The number of sigma bins = 6.

The above mentioned parameters/assumptions were employed throughout the data prediction results.

In addition to the pre-set parameters, several other pieces of information are very important to the data predictions. This information is all related to calibration of either the single particle mode or the volume mode. The volume mode calibration and method of prediction will be discussed first, followed by the single particle calibration and method of prediction.

For the VM calculations, two sets of calibration data were used. The first is identified as the "count" calibration and this data is read in at the beginning of the main program and stored in the array CAL (J,I). The first index J is over IF gain values from 45 to 70. The second index I is over DBSM values from -71 to -20. The proper calibration curve was picked depending upon the IF gain parameter. The volume "count" calibration data also had to be divided by 500 in order to scale the counts to one average count value.* The volume "count" calibration data relates "counts" in the volume channel

---

*Volume mode calibration data was performed by W.Jones of NASA. This information was relayed verbally to Applied Research, Inc.

to DBSM values. The second set of calibration data required is a measurement of the signal response of the LDV as a function of longitudinal distance while focussed at 10 meters. This data is shown in Figure 3. This data was used to calculate L_{eff} at 10 meters. (Reference 1) Longitudinal distance means along the optical axis of the LDV.

The volume β was calculated using the following equation:

$$\beta = (V_S / V_n) f_r / (S_d L_{eff})$$

V_S - integrated volume signal

V_n - integrated volume noise

S_d - signal-to-noise from a sandpaper disk at 10 meters focus with noise scaled from the calibration bandwidth of 100kHz to 860kHz, the processor bandwidth, giving 1.84×10^6.

f_r - bidirectional reflectance of a sandpaper disk equal to .016.

L_{eff} - effective length of the LDV while focussed at 10 meters, found to be 64.1 cm.

L_{eff}, S_d and f_r were all determined prior to beginning the data predictions and remain fixed throughout. The volume signal and noise were calculated for each data point. The LDV data had alternating samples of noise data throughout each of the flights. This was accomplished by dithering a mirror in the optical train thereby effectively "losing" the signal. The raw output of the

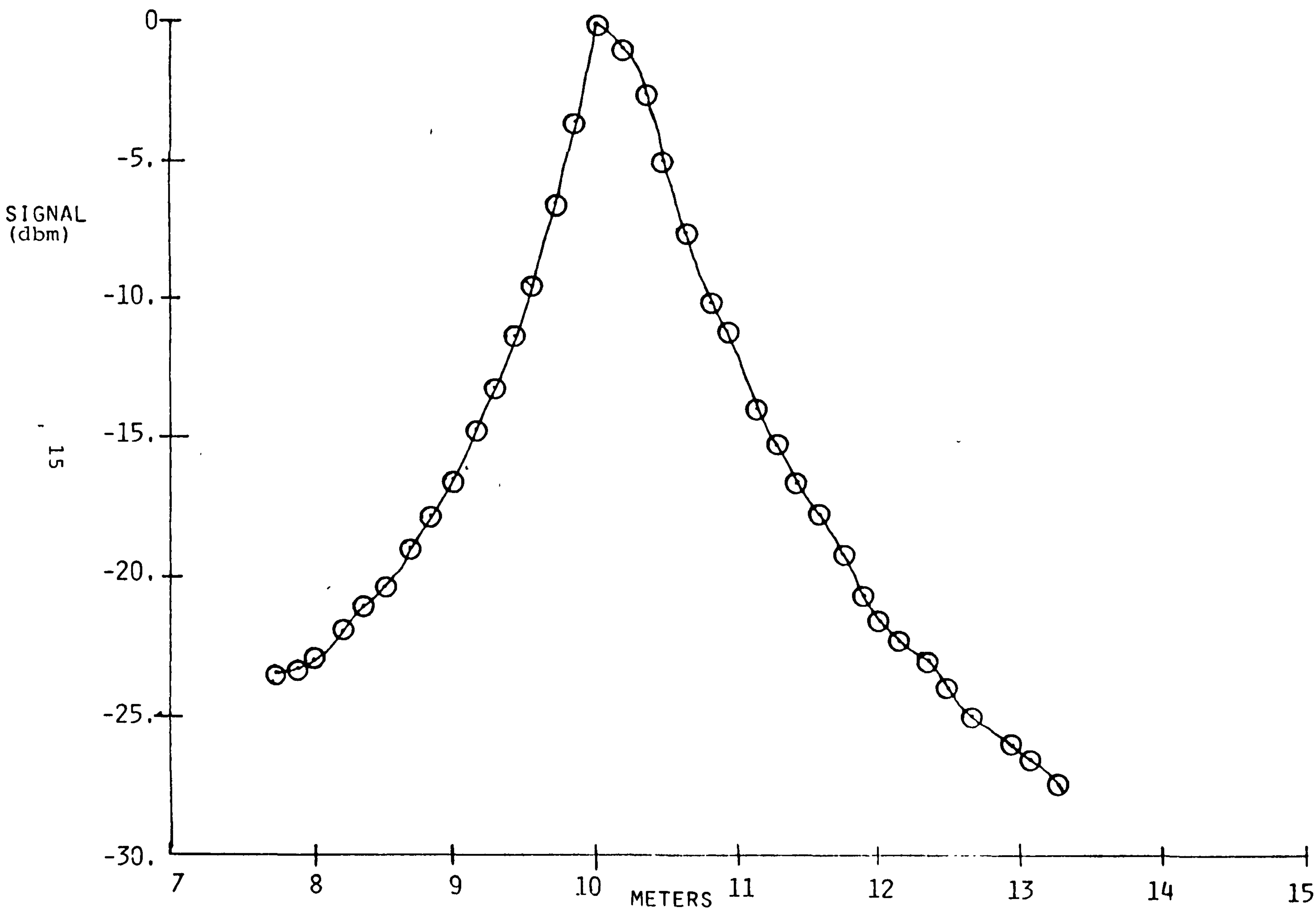


Figure 3. Longitudinal Profile at 10 Meter Focus

volume β channel on the LDV processor is called "counts". This count has to be divided by the number of times the channel was scanned in order to provide the average count for the time interval. The average count is then compared to the count calibration data (using the correct IF gain) to obtain a DBSM signal. The assumption was made that the volume channel count (data record) actually was signal plus noise while the noise particle count was noise only. Therefore to obtain the volume signal (V_S) the noise signal (V_n) was subtracted from the signal plus noise. This provided the V_S and V_n required to complete the volume β prediction.

Two pieces of calibration data are required to complete calibration for the single particle β calculation. These data are the correlation between signal value and channel number on the signal processor and the gain (equals signal/noise/cross-section) versus area curve.

The correlation between channel number and signal value is important to establishing what size particles are being seen by the processor. The "signals" here are tied to the calibration of the area curve discussed next. It is not known how this calibration data may depend upon the IF gain. All predictions were obtained using one calibration curve, Figure 4, which has an IF gain of 57.

The gain versus area curve is crucial to making the correct single particle β predictions. This data was taken by Applied

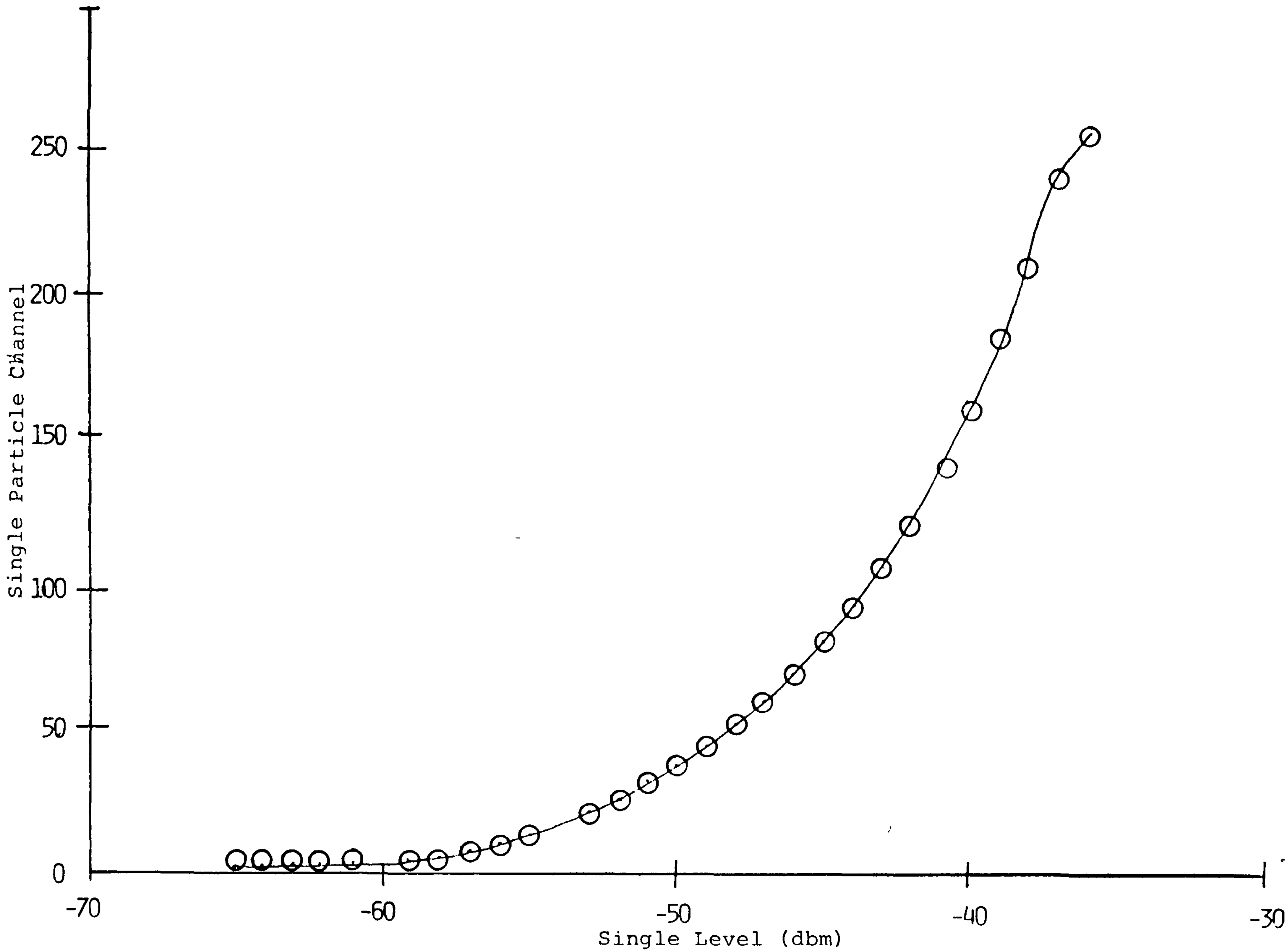


Figure 4. Signal Processor Channel vs. Signal Level

Research, Inc. and presented in Reference 1. This data resulted in a relative mapping of the sensitive focal volume of the LDV in a vertical plane containing the optical axis. "Relative" means that the scale of the map (signal/noise/sigma) was undefined. The remaining procedure then was to define the scale of the gain versus area map. This was done by using the following equation:

$$S_d/f_r = \int C g'(A)\,dA$$

where S_d is the signal return from a sandpaper disk at 10 meters, f_r is the bidirectional reflectance of a sandpaper disk, and g' is the relatively scaled gain data. The values used were:

$$S_d = 1.053 \times 10^6$$

$$f_r = .016$$

$$\int g' dA = 1.44 \times 10^6$$

The S_d value is the calibration value of $10^{7.2}$ scaled from 100kHz to 1.5MHz. The value of C determined was : C = 45.99. This value is used in subroutine DATANAL in a data statement as variable TRANS to scale the area curve to the proper values. Also as a result of this determination of a scale factor C, the single particle gain at an area of 0 can be used from the equation $S/\sigma/_{A=0} = g(0)$. to relate a S/N value S to a σ value; i.e. $\sigma = S/g(0)$. This relationship is used in the main program about lines 218-220 with a value of $1.1037 \times 10^7/cm^2$ for g(0).

The result of the inversion is to return an average sensitive volume cross-sectional area (WV) and average particle cross-section (WS). The equation

$$\beta = (ISUM/4\Pi)(WS/FLTL/WV)$$

was used to compute the single particle β , were ISUM is the number of particles in the histogram. The flight length, FLTL, is computed by using the data recorded in each record (or records) for the elapsed time and flight speed.

The results of the computer implementation were presented in Reference 2. The results presented were all obtained using a power law solution for the σ distribution in the inversion subroutine OPTIM. This was done because early test runs indicated that the power law solution was nearly always the best fit. Since run time was becoming a problem for processing the many flights, the exponential and log-normal solutions were bypassed. This "bypass" can be eliminated by removing line 603.07 - IF(IP.EQ.2) RETURN in the subroutine OPTIM.

Figures 5 - 7 contain a flowchart of the main program and various subprograms which constitute the computer implementation of the β prediction algorithms. The Appendix has a short description of each program of the β prediction algorithms and also contains a listing of the main program and associated subprograms.

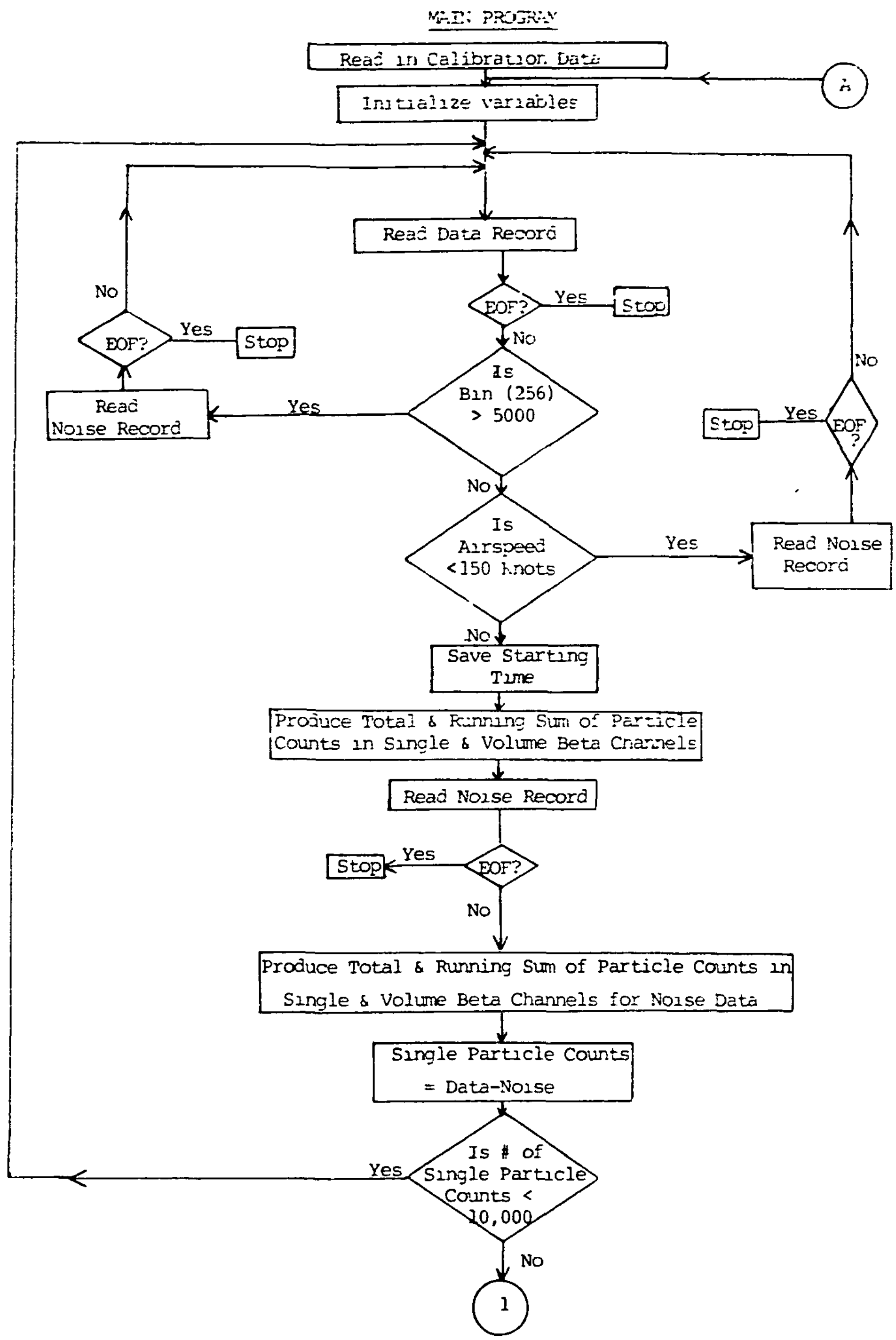


Figure 5a. Flow Chart of Computer Implementation

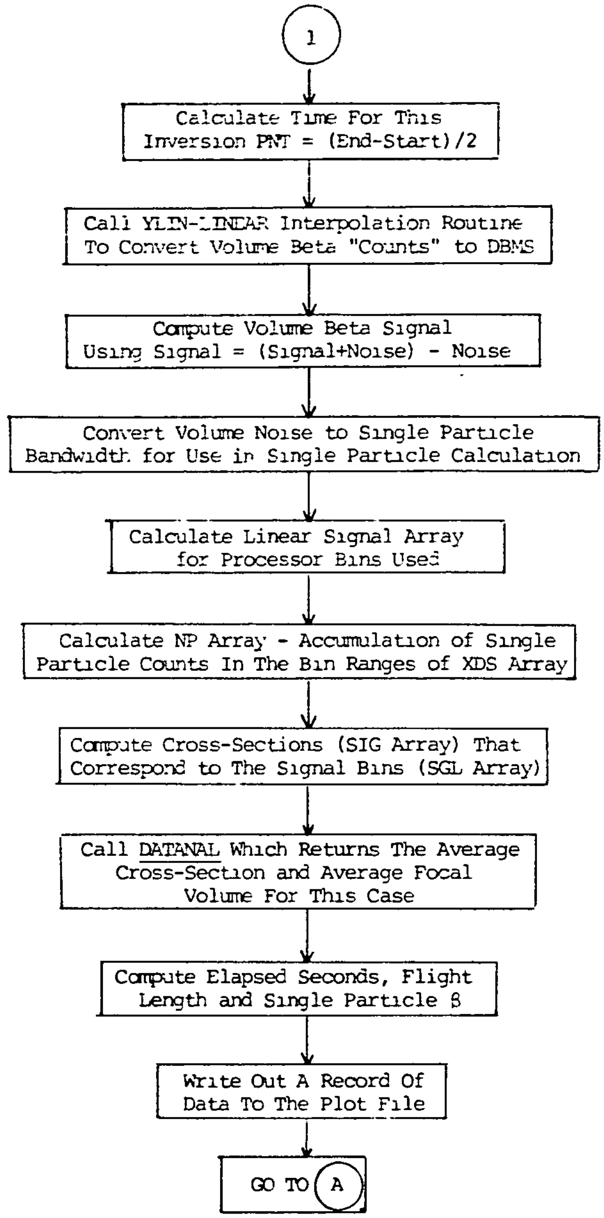


Figure 5b. Flow Chart of Computer Implementation (continued)

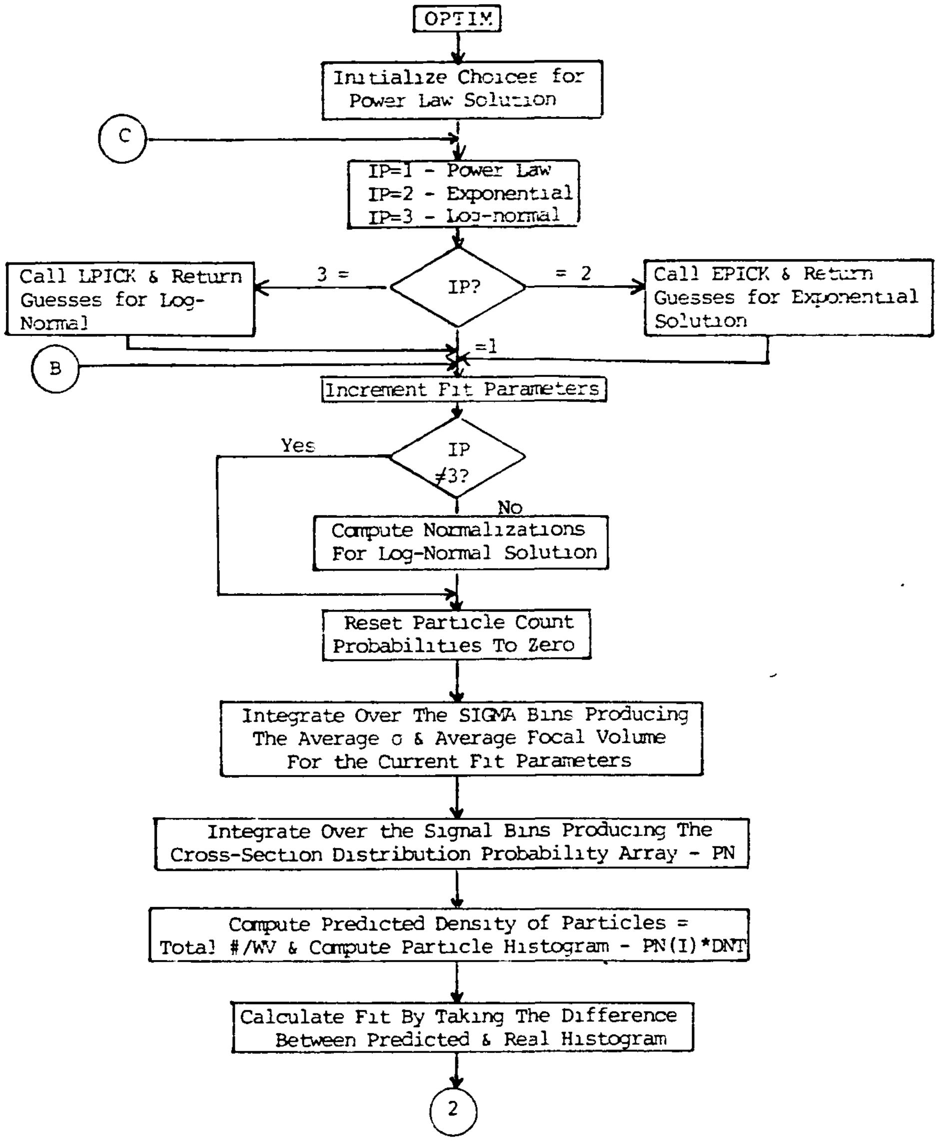


Figure 6a. Flow Chart of Computer Implementation of Single Particle Inversion Subprogram OPTIM

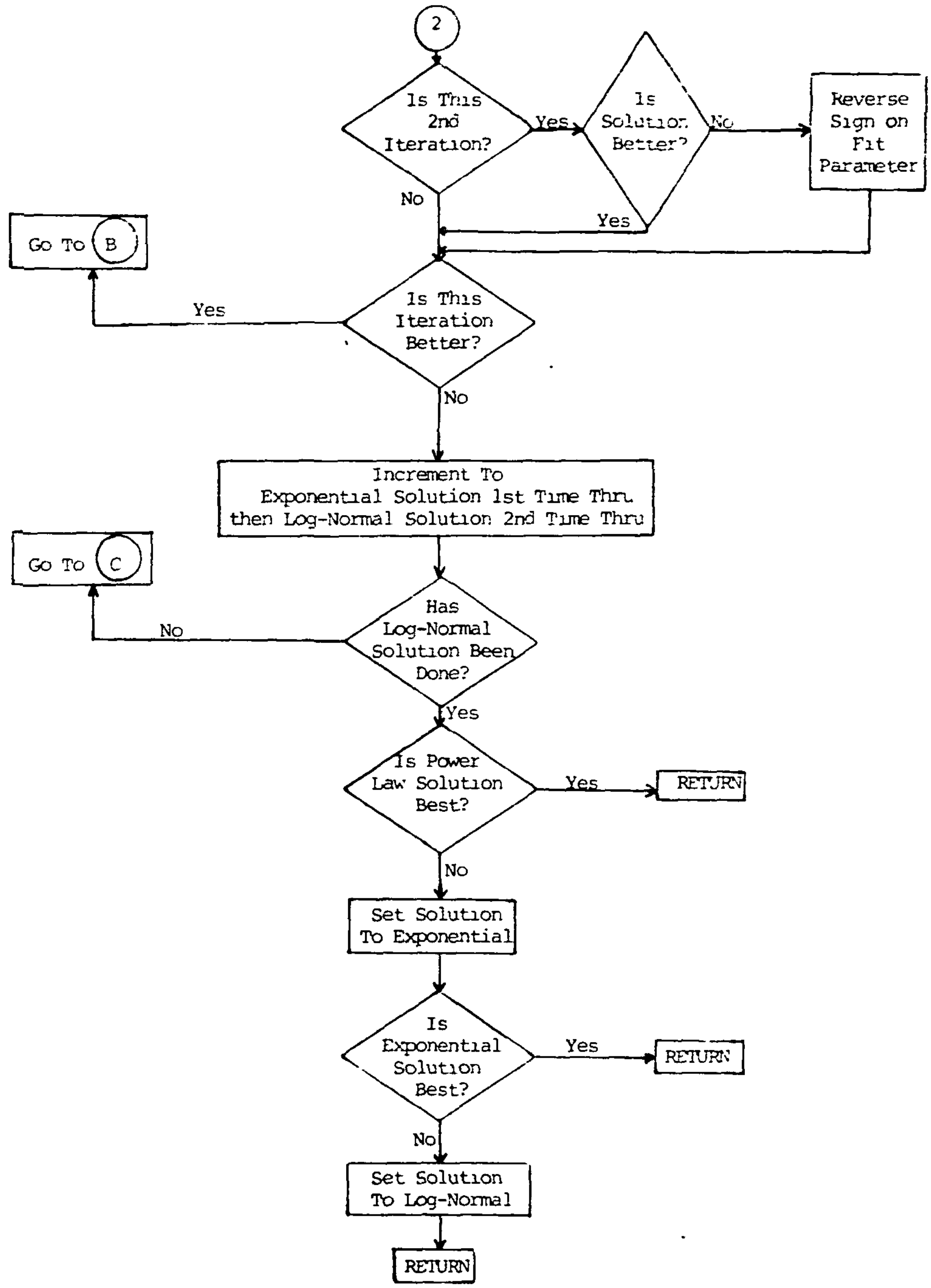


Figure 6b. Flow Chart of Computer Implementation of Single Particle Inversion Subprogram OPTIM (continued)

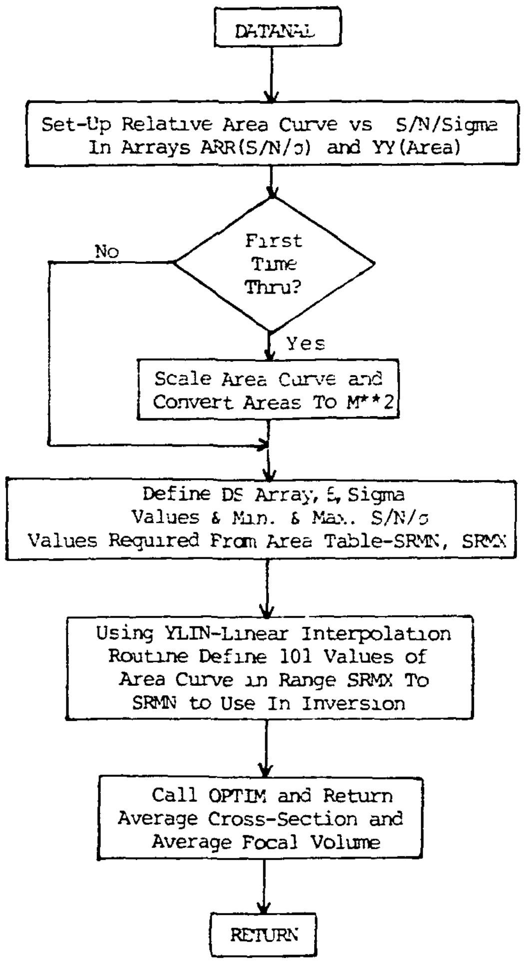


Figure 7. Flowchart of Computer Implementation of Single Particle Inversion Subprogram Datanal

Figure 8. DATA RECORD

Ø-	TIMH:	12	CURRENT TIME
	TIMM:	Ø	
	TIMS:	Ø	
	TIMF:	10°°	
4-	TIMH1:	12	TIME AT START OF RECORD
	TIMM1:	Ø	
	TIMS1:	Ø	
	TIMF1:	Ø	
8-	DATEM:	6	CURRENT DATE
	DATED:	1	
	DATEY:	32	
11-	STEPR1:	Ø	STEPPER POSITIONS (NOT PRESENTLY USED)
	STEPR2:	Ø	
	STEPR3:	Ø	
14-	VOLBH:	Ø	VOLUME BETA INTEGRATORS
	VOLBL:	Ø	
	VOLBN:	Ø	
19-	PARMS:	171356	CONTROL REGISTER PARAMETERS
	CLOCKP:	11.	DIGITAL FILTER CLOCK PERIOD
	FILTER:	2	IF FILTER WIDTH
20-	TIMEON:	3	VIDEO TIME CONSTANT
	VCOFRQ:	140.	VCO FREQUENCY
	DOFSET:	Ø 11.	A/D FILTER BIAS
23-	DISCRM:	Ø	DISCRIMINATOR THRESHOLD
	IFGAIN:	45.	IF GAIN
25-	FWDID:	Ø	FORWARD/AFT IDENTIFIER
	NDSETS:	1	# SCANS PER WRITE
	INTSEC:	1ØØØ.	MS PER INTEGRATION PERIOD
	ADDASV:	Ø	UP IF ADDAS LINK ON
	ADDASI:	1ØØ.	# MS PER ADDAS OUTPUT
30-	SCAN:	Ø	UP IF SCANNER ENABLED
	TASADD:	3ØØ	TAS FROM ADDAS
	TASKB:	3ØØ	TAS FROM KEYBOARD
	THETA:	5ØØ.	ANGLE THETA IN .Ø1-DEG UNITS
34-	VCOOFS:	Ø	VCO OFFSET IN 53-KHZ UNITS
88	IDENT:	BLHB 8Ø.	IDENTIFICATION STRING
128			
383	DATA		

2.4 Running The Data Prediction Algorithms

The β prediction algorithms are stored on disk on the Sigma V computer in account BILBRO, EB23175. The file name is DATANAL. A batch file which is used with the code is stored under BXD. Two input files are required, they are unit numbers 5 and 12. File unit number 5 must be LDV processor data in the format shown in Figure 8. This defines the data for which predictions are to be made. File unit number 12 is volume β calibration data and is stored under the file name CALF1L4 in account BILBRO, EB23175. Units 6 and 10 are used as output unit numbers to obtain the printed results of the algorithm. Unit 13 is also an output unit but is used to store results for plotting and should therefore be given a file name by the user. Unit 13 produces one record for each β prediction during the flight in the following format:

```
Write(13) LH,LM,LS,LF,SINGLEB,VOLB,WS,FLTL,ELS,WV.
```

The definition of these variables is:

- LH - Hours for this prediction
- LM - Minutes for this prediction
- LS - Seconds for this prediction
- LF - Fractional seconds for this prediction
- SINGLEB - Single particle β prediction
- VOLB - Volume channel β prediction
- WS - Predicted average cross-section
- FLTL - Estimated flight length
- ELS - Number seconds for which data-recording occurred for this prediction
- WV - Predicted average focal "volume" cross-section

3.0 Measurement Errors

In the application of the previously discussed VM and SPM algorithms to the processing of the NASA flight data, two general classes of error are considered. The first class, systematic errors, involves errors of concept, principle, or application. The second class concerns statistical errors or uncertainties which result from lack of knowledge or from fluctuating random variables. Systematic error is the most difficult to handle since even its presence is not always obvious. In the case of the data under consideration, an average offset of the results from the two algorithms of about one order of magnitude indicates systematic error. However, more than one such error could be present. To uncover these errors, a reconsideration of concepts and procedures is necessary.

Statistical error is more easily treated. With estimates of the statistical uncertainty in independent variables, the uncertainty in the final result may be obtained. In the present case, also the number of particles processed per output β value is expected to govern statistical fluctuation in β . This result has been analyzed.

3.1 Volume Mode Errors

The VM algorithm has been derived in Section 2.1. Possible systematic errors in the backscatter coefficient calculation may reside in

a) the processor output

b) the interpretation of the processor output

c) the concept for the volume mode gain when used with single particles (as discussed in Section 2.1)

d) a systematic offset in the parameters used to calculate the backscatter coefficient value

Possible errors in the processor output or the curves supplied by NASA which translate output into signal level are not considered here, but must be investigated in the laboratory.

The interpretation of the processor output has been discussed in Section 2.3 and no error has been discovered.

As discussed in Section 2.1, if the gain factor $G(\sigma)$ is dependent on σ as a result of a decreased sensitivity volume for smaller σ , calibration of the "single particle volume mode" should not be treated as the "conventional volume mode." As mentioned in Section 2.1, $G_{VM}<G_V$ implies that β values calculated with the gain factor G_V for the conventional volume mode give too small a value by the ratio G_{VM}/G_V. Evaluation of G_{VM} requires laboratory determination of the sensitive volume for each value of σ , integration of the single particle gain over this volume, and then integration of this result over the backscatter cross-section distribution. Therefore, in measurement situations, this effect would require determination of the σ distribution. An

effect very similar to this occurs in the SPM where the aerosol density is calculated by dividing the number of particles seen by the average volume, calculated by averaging the sensitive volume elements for the SPM over the σ probability distribution. It is not clear whether the VM sensitive volume elements would be the same as those for the SPM since signal acquisition requirements are different for the two modes. One has the feeling that this effect would contribute errors of less than an order of magnitude. This question might be answered with computer modeling.

The backscatter coefficient is calculated from the expression

$$\beta = S/G_V$$

where S is the signal to noise found from the data as explained in Section 2.3, and G_V is found by calibration with

$$G_V = S_d \bar{L}_{eff}/f_r$$

where

S_d = signal to noise return from a calibration disk of bidirectional reflectivity f_r

$$L_{eff} = \int g(x,y,z)dxdydz/\int g(x,y,0)dxdy$$

(see Figure 1)

L_{eff} = the effective length of the sensitive volume

Errors or uncertainties in β may be evaluated as

$$\delta\beta/\beta = \delta S/S + \delta S_d/S_d + \delta L_{eff}/L_{eff} + \delta f_r/f_r .$$

The change $\Delta\beta$ in units of orders of magnitude is defined as

$$\Delta\beta = \log(\beta + \delta\beta) - \log \beta$$

$$= \log(1 + \delta\beta / \beta)$$

Each of the above error sources is considered separately.

Error due to S:

The signal out of the processor and corrected by noise subtraction is assumed here as exact. A possible interpolation error of less than 1dBm occurs in obtaining the dBm level from the volume channel counts. This yields a $\Delta\beta$ found as

$$\Delta\beta = \log(1 + \delta S/S) = \log(S+\delta S) - \log S$$

$$= [(S+\delta S)_{dBm} - S_{dBm}]/10$$

$$= .1$$

Error due to S_d :

The calibration signal error from the rotating sandpaper disk at the sensitive volume waist would not be likely to have a reading error of more than 2 dBm, from inspection of the variation of the calibration data. Thus

$$\Delta\beta = .2$$

Error due to f_r:

The bidirectional reflectance of the aluminum oxide target for incidence and reflection at 45 degrees was taken to be .016/sr. An uncertainty of ±25% has been quoted by workers in this field (Reference 3). This gives

$$\Delta\beta = .1$$

Error due to L_{eff}:

The error in the determination of L_{eff} is less than 10% from consideration of the methods used. This gives

$$\Delta\beta = .04 \quad .$$

In summary, the possible errors due to uncertainty in the quantities used to calculate the backscatter signal in the volume mode are given in the following table:

Source	$\Delta\beta$ (orders of magnitude)
S	.1
S_d	.2
f_r	.1
L_{eff}	.04

These uncertainties may be positive or negative, and combine to less than half an order of magnitude at maximum. The error given for S considers only the uncertainty in interpolating the processor output. Systematic errors are not included in these numbers, but once identified, their effects would be calculated with the above formulae.

3.2 Single Particle Mode Errors

As discussed in Section 2.2, the backscatter coefficient for the SPM may be expressed as

$$\beta = D\langle\sigma\rangle/4\Pi$$

where D is the particle density and $\langle\sigma\rangle$ is the average single particle backscatter cross-section. These quantities are determined from the peak signal distribution and the gain contours versus enclosed area function resulting from mapping the sensitive focal volume area in the vertical plane of maximum gain - the yz plane of Figure 1.

Possible systematic errors for this mode concern

a) the processor output
b) the interpretation of the processor output
c) the single particle per transit time condition on the applicability of the algorithm
d) the gain versus area function determined by calibration
e) operation of the SPM algorithm

Random errors and measurement uncertainties will be discussed following discussion of these topics.

The processor output will be assumed valid, subject to laboratory tests, except possibly for point c). The interpretation of this output is described in Section 2.3 and questions are raised under point d). Some implications regarding conditions of validity for the SPM algorithms also follow from the discussion of point d) below. The SPM algorithm has been found valid for predicting β, subject to random errors, as previously documented.

In the course of the present work, considerable effort has been spent in an attempt to improve the application of calibration procedures which result in the single particle gain versus area function, with the result that a significant shift in the previously processed backscatter coefficient values is indicated. Originally, calibration plans for this program included a special calibration device which would emit single particles of known cross-section. Time constraints from the flight plans forced this effort to be interrupted and an alternate calibration procedure was sought involving the spinning disk. Under this contract this procedure has been reconsidered and improved.

Contours of constant single particle gain, $g(x,y,z)$, were mapped out in the $x=0$ (vertical) laser beam plane within the sensitive volume, in a previous contract. Only relative values of these contours were determined by this mapping procedure, with overall normalization to be determined by another method. A more accurate way of obtaining this normalization follows.

The conventional volume mode gain G_V is defined as the integral over the single particle mode gain $g(x,y,z)$ and determined by calibration with a spinning disk at the waist as

$$G_V = \int g(x,y,z)dxdydz$$

$$= S_d L_{eff}/f_r \quad .$$

The desired integral to be evaluated for normalization of the gain versus area function is

$$I = \int g(o,y,z)dydz$$

$$= \int g(A)dA$$

where a change of variables transforming the integral from a two dimensional to a one dimensional integral is implied. The variable A is the total area within closed contours of g. With this integral evaluated, g is related to its unnormalized version g' by

$$g = ag'$$

where

$$a = I/\int g'(A)dA.$$

The value I can be determined from G_V if the sensitive volume is assumed cylindrically symmetric about the optical axis of the beam. Representing the integral for G_V by transforming to cylindrical coordinates gives

$$G_V = \int g(\rho,z)\rho d\rho \, d\Theta dz$$

$$= 2\Pi \int g(\rho,z)\rho d\rho dz$$

$$= 2\Pi \quad \rho_{av} \int g(\rho,z) d\rho dz.$$

This last integral is one-half of the desired integral I. Also

$$\rho_{av} = \int g(\rho,z)\rho d\rho dz / \int g(\rho,z) d\rho dz$$

is an average radius of the sensitive volume. Hence

$$G_V = \Pi \ \rho_{av} \ I = S \ L_{eff}/f_r$$

and from

$$I = a\int g'(A) dA = aI'$$

one gets for the normalization factor

$$a = [S_d/(f_r I')] L_{eff}/(\Pi \rho_{av})$$

$$= a_o \ L_{eff}/(\Pi \ \rho_{av})$$

where a_o was the normalization used in the processing of the flight data. The integrals for ρ_{av} were numerically evaluated to give ρ_{av} = .0265cm. (Compare this with Figure 9.21 of

Reference 1.) This gives

$$a = [(64.1cm)/(.0265\Pi\ cm)]a_o$$

$$= 770a_o$$

Any adjustment in the normalization of the gain versus area function affects the sizes of the particle distribution which can be detected and associated with a particular peak signal distribution. For the power law backscatter cross-section distribution function, which best fits the processed flight data, an adjustment "a" divides the average cross-section, but does not change the density. (See subroutine OPTIM). Thus

$$\beta = \langle\sigma\rangle\ D/4\Pi$$

$$= \beta_o/770$$

and indicates that all calculated single particle β values should be decreased by about 2.9 orders of magnitude. A calibration factor of the same order of magnitude is indicated by using the incoherent beam profiles of Figures 8.3 and 8.4 of Reference 1. This implies that the system is sensitive to cross-sections as small as .00166 square microns since threshold peak signal levels required cross-sections of 1.28 square microns at the original calibration.

The signal-to-noise for a given cross-section is unchanged by this rescaling since its effect on g and σ cancel for a power law σ distribution. For example, around record 6 of flight number 9, processing of 10,000 particles gives a signal-to-noise of about .14 for the threshold in this case. To obtain this number one must use the calibration curve supplied by NASA for IF gain of 57. Flight 9 had an IF gain of 63. It is clearly these low signal-to-noise numbers combined with the higher single particle gain values which are responsible for such low β values.

This adjustment in the normalization of the gain means that the system is sensitive to smaller particles than previously indicated, without undergoing an adjustment in the required signal-to-noise threshold. The backscatter coefficient is consequently smaller because no corresponding adjustment in density occurs. This gives a result which is about two orders of magnitude lower than the VM measurement. Nominal densities calculated with the SMP algorithm give about 1 particle per 100 cubic centimeters, which seems rather low for distributions with an average cross-section of .007 square microns, as in the flight 9 case above. This density would be more appropriate for particles of greater than 1 micron diameter. Therefore, since the gain profile of the sensitive volume is fairly well established (coherent and incoherent methods give the same order of magnitude for the scaling), there is reason to question the signal-to-noise values assigned to the signal bins of the processor output. The method of calculating these has been rechecked with no mistake found.

The actual calibration curve provided by NASA which assigns signal level to a bin should be rechecked. Also the variation of these assignments with IF Gain should be considered. An increase in signal values through recalibration of the processor gain could bring SPM and VM results into agreement.

The following consideration suggests another possibility. With the same particle distribution, the VM result predicts densities of about two orders of magnitude higher. For this case of one particle per cubic centimeter, the system may see more than one particle per transit time since the sensitive cross-sectional area is 42 square centimeters. A coincidence rejection mechanism could give erroneous results since no compensation is made for rejected cases; or a coincidence rejection mechanism which is not properly responding could interpret clusters of particles as single particles, thereby decreasing the density. Hence it may be possible that the SPM was used outside the conditions of validity without proper compensation being made.

Random errors, or errors of uncertainty of measurement, in the single particle prediction can result from the gain versus area function determination, the interpretation of the peak signal bins, the operation of the algorithm, and the number of particles seen. As before, the uncertainty in β in units of orders of magnitude due to a component uncertainty $\Delta\beta$ is

$$\Delta\beta = \log (1 + \delta\beta/\beta) \quad .$$

As documented in an informal memo, the algorithms produce an uncertainty of less than 5%.

Error due to algorithm:

$$\Delta\beta = .02$$

Random error due to interpolation of signal bin calibration is 1 dBm. (This does not include possible systematic error due to this calibration data discussed earlier). For small errors this has an approximately linear affect on the σ interval, and therefore on $\langle\sigma\rangle$. Hence

$$\delta\beta = 26\%$$

Error due to interpolation of peak signal calibration:

$$\Delta\beta = .1$$

The gain versus area function was previously discussed in terms of systematic error. Any alteration of this function by a factor "a" goes directly through to $\langle\sigma\rangle$ and then to $\delta\beta$. It is helpful to derive an estimate for the uncertainty in "a" due to measurement errors by referring to the previous expression

$$a = a_o L_{eff}/\rho_{av}\Pi$$

where

$$a_o = S_d/f_r I'$$

and I' is the integration over the unnormalized contours. The uncertainty in I' is difficult to estimate precisely but is within 10%. Using the previous values for S_d and f_r, and taking ρ_{av} to have the same uncertainty as I' and L_{eff} gives the following results for random errors due to uncertainties in measurements used to calculate the backscatter coefficient in the single particle mode.

Source	$\Delta\beta$ (orders of magnitude)
Algorithm	.02
Peak signal interpolation	.1
S_d	.2
f_r	.1
L_{eff}	.04
ρ_{av}	.04
I'	.04

Errors due to particle number fluctuations are not treated in this format but considered in the following paragraph.

Errors resulting from fluctuations due to small particle samples have been examined by comparing with the results from the case when a 10,000 particle minimum was used. For processing runs utilizing various sample sizes less than 10,000, a running time-weighted average of the results sufficient to include 10,000 particles was developed, and considered a "local average". This result was used in two ways. The difference of this local average from the 10,000 particle case was considered the error in the local average. (The local average value nearest in time to the 10,000 particle value was used). Also, the fluctuation from the local average was calculated from the differences of the small particle number data from the local average. This method was found to be reasonable if too sharp variations in the data do not occur. Figure 9 shows these results for a region of flight 16 with the sharp β spike at time 878 removed. These results show rms deviations in units of orders of magnitude. They indicate that the local error can be maintained with .1 orders of magnitude with particle samples of 1000 minimum. Fluctuations of the "instantaneous" data from the local average do not approach this value until around 5000 particles minimum are processed. In this case the SPM has lower deviations and fluctuations than the VM data. (The VM data was also compared with itself in the manner described above). Figure 10 shows the same flight with the β spike included, with results which do not settle down as well. Figure 9 shows the deviations of the local average to be smaller than the fluctuations of the data from the local average. Figure 10 shows some of the SPM versus VM behavior.

Figure 11 shows the same data from a region of flight 9. Here the fluctuations are lower than the deviation of the local mean, and the results do not settle down as well as the 10,000 particle case is approached. For the SPM the .1 order of magnitude error is approached at 1000 particles, while 3,000 particles seem to be required for the VM result.

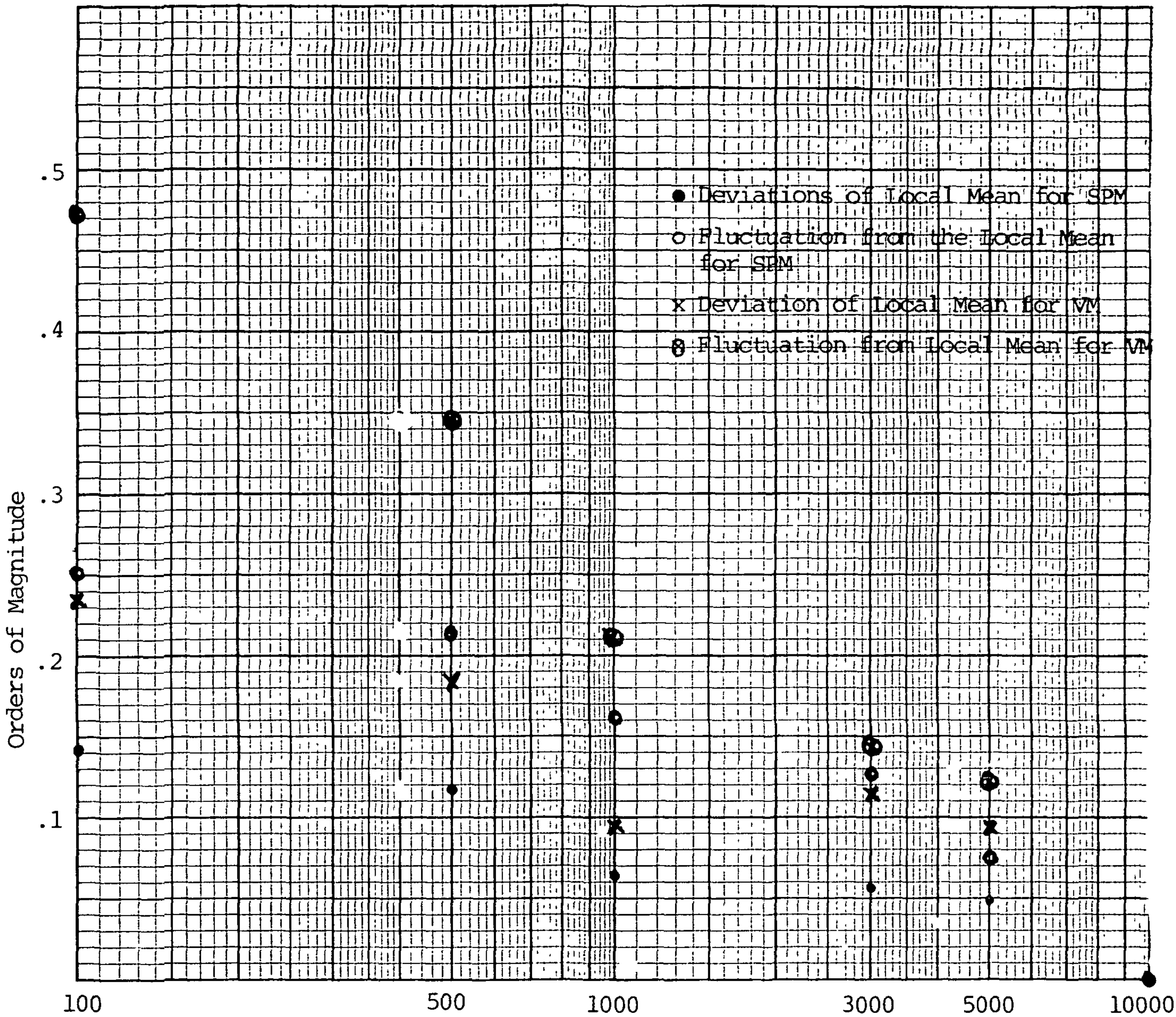


Figure 9. Statistical Error in Backscatter Coefficient as a Function of Number of Particles Processed for Flight 16 (Spike Removed)

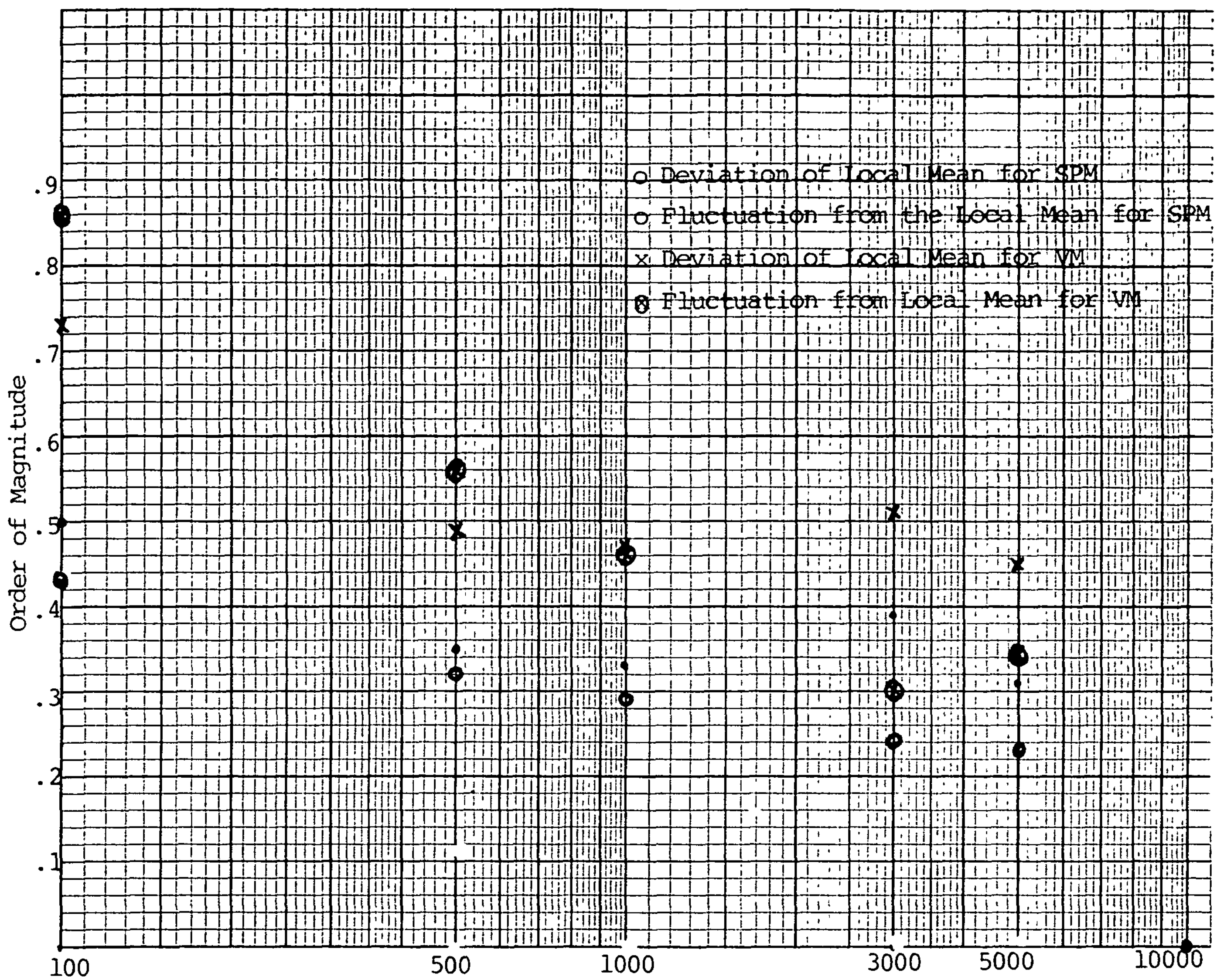


Figure 10. Statistical Error in Backscatter Coefficient as a Function of Number of Particles Processed for Flight 16 (with Spike)

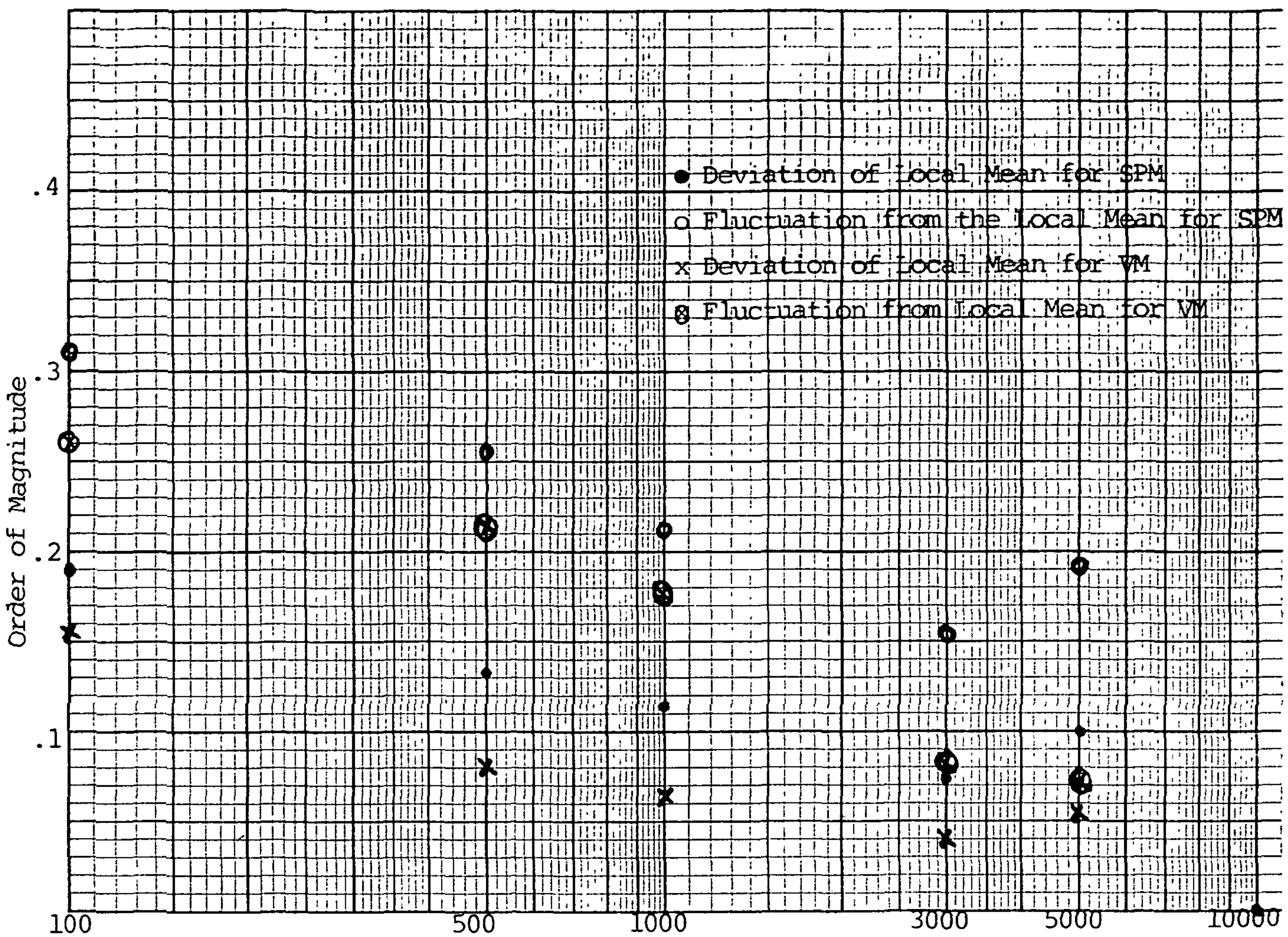


Figure 11. Statistical Error in Backscatter Coefficient as a Function of Number of Particles Processed for Flight 9

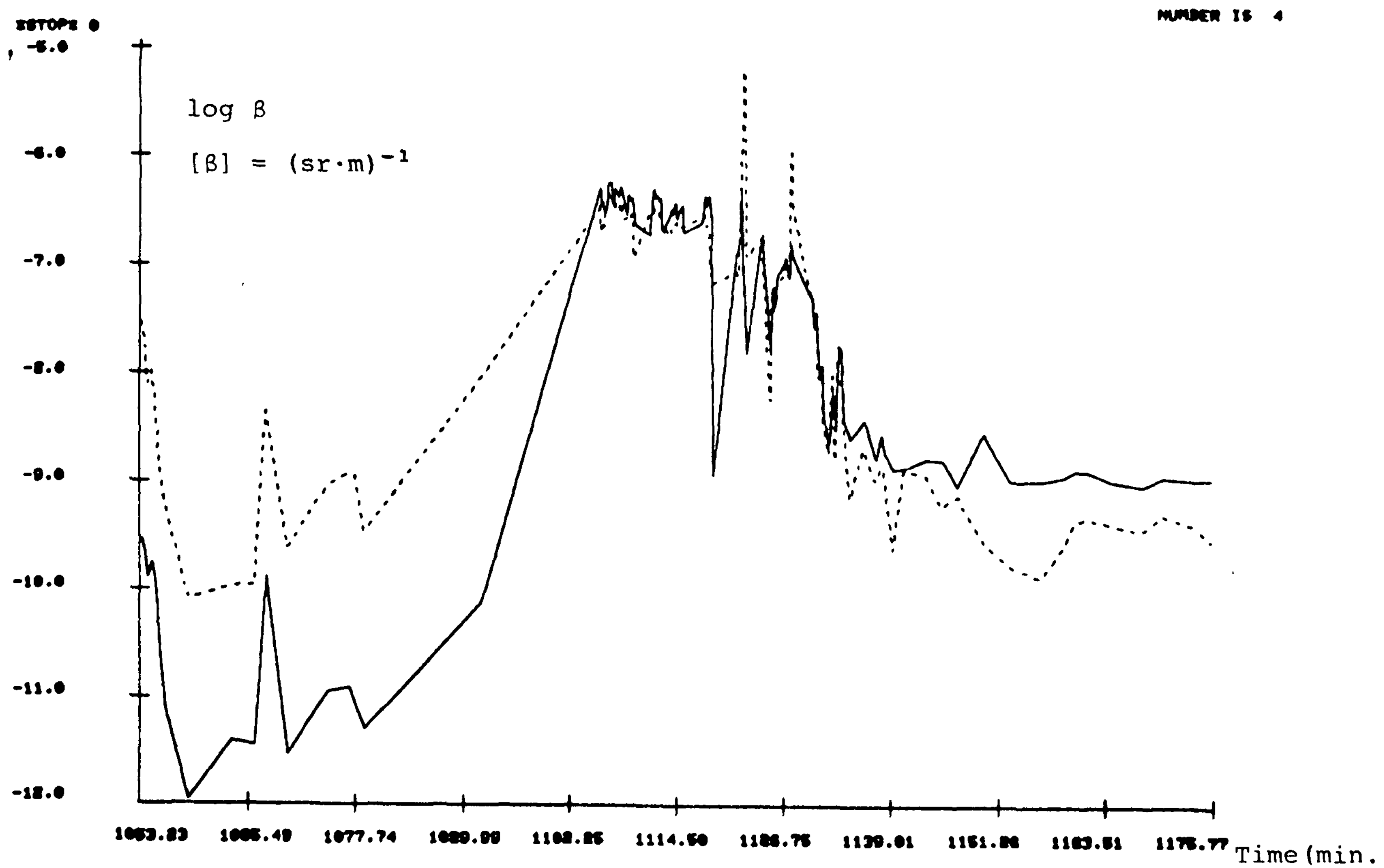


Figure 12. Backscatter Coefficient β for Flight 4

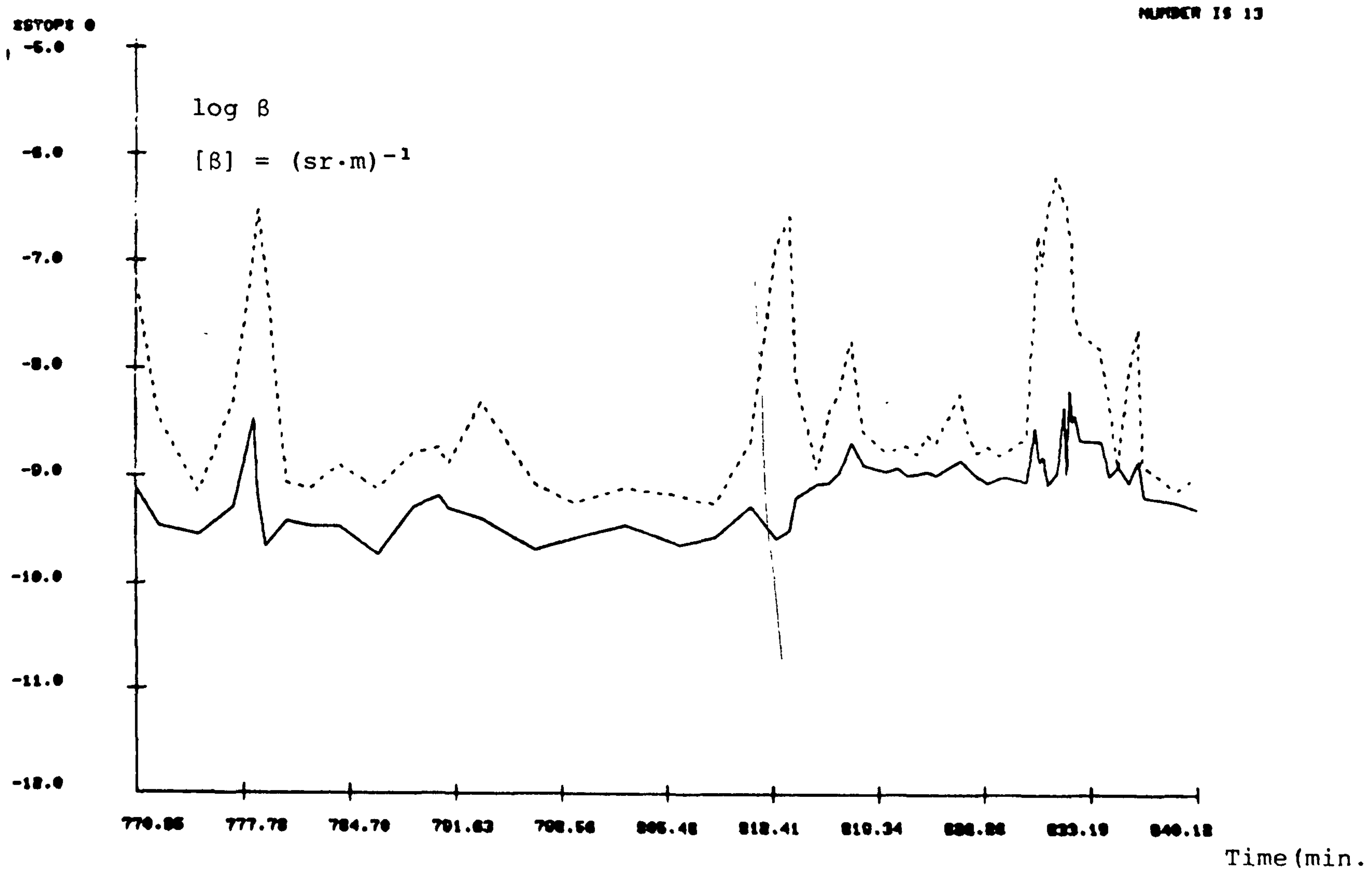


Figure 13. Backscatter Coefficient β for Flight 13

3.3 Anomalous Cases From Processed Data

Two pathological data prediction cases can be identified in Reference 2 and are shown in Figures 12 and 13. These cases are Flights 4 and 13. Flight 4 is unusual in that initially the single β predictions were much lower than the volume β and then about halfway through the flight the situation reversed. It was found that this was the case because the airspeed value was incorrect. Since airspeed is used to calculate flight length and then single particle β, the predicted results were much too low. Later on in the flight the problem was corrected and the predictions reversed the earlier trend.

Flight 13 also showed the trend of single β predictions less than the volume β predictions. A closer look at the computer run showed that the single β inversion program could not fit the histograms that were coming from the signal processor. The algorithm could not produce a fit because the lower signal bins were empty. The assumption made for all flights was that the threshold was set at 4 and therefore there would be counts in bin 5 and above. For flight 13 the counts did not pick up until around bin 30. This left a hole in the particle cross-section distribution which the inversion algorithm could not fit with a power law, and spurious predictions resulted.

4.0 Conclusions

The basis of the volume mode and single particle mode algorithms has been discussed. Programming and data handling techniques have been described. Some sources of systematic error in the previously processed flight data have been examined. Statistical uncertainty in the processed flight data has been evaluated.

A rescaling of the single particle gain function as a result of a more exact handling of the calibration has resulted in a decrease by 2.9 orders of magnitude of the backscatter coefficient in the single particle mode. This leaves a differential of about two orders of magnitude with the prediction of the volume mode method. Two possible explanations were offered for this difference: a) The single particle mode signal versus bin calibration for the processor is not correct. b) The particle density conditions during data collection did not conform to one particle per transit time. A further possibility is that the volume channel signal is either incorrect, or incorrectly interpreted.

The statistical errors for the two methods produced a maximum of ±.44 orders of magnitude uncertainty for the volume mode and ±.54 orders of magnitude uncertainty for the single particle mode. The processed results do not agree within these errors.

Studies of the effects of particle number fluctuations in the processing of both modes show that, in order to reduce deviations of the local mean from a mean calculated with 10000 particles, and to reduce fluctuations of the data from the local mean, care must be exercised regarding the number of particles processed. To keep these errors near .1 orders of magnitude, this number of particles must be 1000 for the single particle mode, and 3000 for the volume mode. This requires a very short flight length (<1km) for particles with densities as low as one particle per 1000 cubic centimeters.

In order to better understand the behavior of LDV systems operating in these two modes, it is suggested that laboratory measurements on aerosol standards be accomplished with the inflight processor. The calibration techniques which were interrupted by the flight schedule could be continued, or perhaps the input or exhaust from a Knollenberg counter could be used. A comparison of the two methods needs to be accomplished and their limits of applicability determined in the laboratory. The methods are independent assessments of aerosol distribution characteristics and, operating together, offer a unique means of converging on more precise measurements.

REFERENCES

1. Final Report on Contract No. NAS8-34337, Beta Experiment, June 1982, Applied Research, Inc.

2. Final Report on "Added Scope" to Contract NAS8-34337, Beta Experiment, April 30, 1983, Applied Research, Inc.

3. M.J. Post, et. al., "Calibration of Coherent Lidar Targets", Applied Optics, 19, 15 August 1983, p.2828.

APPENDIX

Program Descriptions

Main Program - Sets up calibration data (except for gain versus area table); controls reading of data and number of particles/inversion; contains volume β computations and writes the output data file; computes flight length and single β after calling DATANAL.

Subroutine GETDAT - Reads the flight data from disk as 192 words (4 bytes long) and converts to 384 words (unpacks the data).

- Inputs - None
- Outputs- BUF= 384 word array containing all data.
  DATA= 256 word array containing the counts in the signal processor for the single particle.

Subroutine DATANAL - Sets up the gain versus area table and then calls OPTIM to do the actual inversion.

- Inputs - NT= Total # of particle counts
  M = # of signal bin segments
  MS= # of cross-section segments
- Outputs- WS= average cross-section
  WV= average focal volume cross-section area

Subroutine OPTIM - does the inversion using three different fits for the cross-section distribution: (1) power law (2) exponential (3) and log-normal. This routine iterates using each distribution type in turn to find the best match between the real single particle count histogram and the predicted count histogram.

- Inputs - B= first power law exponent trial
  - N= total # of particle counts
  - M= # of signal bin segments
  - MS= # of sigma (cross-section) segments
  - NI= # of integration steps
  - YY= array that contains scaled gain values for the "area" table
  - ARR= array that contains areas for corresponding array YY
- Outputs- WSS= predicted average cross-section
  - WVS= predicted average focal volume cross-sectional area

Subroutine COEF2 - calculates the difference between two area values in an integration step.

- Inputs - I= index in SGL array (signal bin)
  - J= index in SIG array (cross-section)
  - K= integration step
  - DJ= Δ sigma for index J
  - M= # of cross-section segments
- Outputs - C= area difference

Subroutine LPICK - generate a good starting point for the log-normal solution.

- Inputs - SIG= cross-section array
  MS= # of cross-section segments
- Outputs - BP= selected mean value of log
  DB= mean value of log increment selected
  ALP= selected standard deviation of log
  DALP= standard deviation of log increment selected

Subroutine EPICK - generate a good starting point for the exponential solution.

- Inputs - SIG= cross-section array
  BP1= selected power law fit from power law solution
  MS= # of cross-section segments
- Outputs - BP= selected exponential power
  DP= exponential power increment

Subroutine PROW - computes normalized probabilities for the power law, exponential and log-normal distribution depending upon the value of IP.

- Inputs - BP= fit parameter for solution:
  - power law - exponent
  - exponential - exponential coefficient
  - log-normal - mean value of log

IP= 1=power law

2=exponential

3=log-normal

SIP= selected cross-section for which probability will be calculated

SO= (required only for log-normal) selected standard deviation of log value

A3= (required only for log-normal) normalization factor for the selected log-normal parameters

Function YLIN - linear interpolation function.

- Inputs - N= # of data parts in array

XX= dependent variable that results is derived to be interpolated for

X= array of dependent variable values

Y= array of independent variable values for which interpolations are performed

- Outputs -YLIN= interpolated value

07 JAN 31,'84 DC/DATANALF2.BILBRO

```
C ******************************************************C
C
C  THESE FILES MUST ALL BE ASSIGNED PRIOR TO STARTING DATANAL.
C
C  1.  SET F:5/M##;IN    ( M## = DISK FILE CONTAINING FLIGHT DATA )
C  2.  SET F:12/CALFIL4;IN    ( CALIBRATION DATA DISK FILE )
C  3.  SET F:10=ME;OUT    ( CRT TERMINAL OUTPUT )
C  4.  SET F:12=ME;OUT    ( CRT TERMINAL OUTPUT )
C  5.  SET F:13/P##;OUT    ( P## = DISK FILE CONTAINING PLOT DATA )
C
C ******************************************************C
C
      COMMON/OPT/DS(22),NP(22),FSI,FN
      COMMON/COE2/SIG(22),SGL(22),WI
         INTEGER BUF, DATA
      DIMENSION DATA(256),ISBIN(256),BUF(384),CAL(26,51)
      DIMENSION IXBUF(26),XNOISE(51),YBUF(51),XSGL(22)
      DIMENSION ISN(256), XDS(22)
      DATA XSGL/-57.75,-56.0,-54.0,-50.25,-46.125,-41.5,-35.75/
      DATA XDS/4,8,16,32,64,127,1/,ITHRESH/4/,M/6/,MS/6/
C
C     XSGL IS THE SIGNAL ARRAY AND CONTAINS THE DBSM VALUES
C          FOR THE SELECTED SIGNAL BINS ON THE PROCESSOR.
C
C     XDS IS THE SIGNAL BIN ARRAY AND CONTAINS THE SIGNAL BIN RANGES
C         ( 1-4 , 5-12 , 13-28 , ETC. ).
C
C     IREC   IS THE DATA RECORD COUNTER
C
C     INPUT PROCESSING PARAMETERS / ASSUMPTIONS :
C
C     M IS THE NUMBER OF SIGNAL BINS. ( M = 6 )
C
C     MS IS THE NUMBER OF SIGMA BINS. ( MS = 6 )
C
C     ITHRESH IS THE THRESHOLD SETTING ON THE PROCESSOR. ( ITHRESH = 4 )
C
C     XDS = SIGNAL BIN RANGES.
C
C     # 1 = 4    BINS
C     # 2 = 8    BINS
C     # 3 = 16   BINS
C     # 4 = 32   BINS
C     # 5 = 64   BINS
C     # 6 = 127  BINS
C     # 7 = 1    BIN  = LAST SIGNAL BIN CONTAINS ALL PARTICLES WITH VALUES .GE.
C                CORRESPONDING SIGNAL.
C
C     XSGL = SIGNAL BIN VALUES THAT CORRESPOND TO SIGNAL BINS.
C
```

```
C
C      OTHER ASSUMPTIONS :
C
C      1. L_EFF AT 10 METERS = 0.6407 METERS.
C
C      2. WAIST RETURN FOR A SANDPAPER DISK AT 10 METERS = 1.053 E 6.
C
C      3. SINGLE PARTICLE GAIN FOR A ZERO AREA IS 1.10376 E 7 METERS ** -2.
       IREC=0
       DO 100 I =1,51
       CALL  BUFFERIN(12,1,IXBUF,26,ISTST,NWRD,IAB)
C      WRITE(6,137) (IXBUF(IJ),IJ=1,26)
137    FORMAT(1H ,'CAL  DATA ',3(1X,9I8,/))
       DO 110 J=1,26
C
C      CAL IS THE CALIBRATION DATA ARRAY.
C         IT IS USED TO CONVERT COUNTS TO DBMS FOR THE VOLUME
C         BETA CHANNEL AND TO OBTAIN NOISE FIGURES FOR THE
C         SINGLE BETA.
C             1ST INDEX IS OVER IF GAIN - 26 VALUES
C            2ND INDEX IS OVER DBSM VALUES RANGING FROM -71 TO -20
C
       CAL(J,I)  = IXBUF(J)
110    CONTINUE
100    CONTINUE
       DO 122 J=1,51
       XNOISE(J)=-71+J
122    CONTINUE
       KBUF = 0
       KH = 0
       KS = 0
       KF = 0
       KKK = 0
       NNSUM = 0
       SSVOL = 0
       NSUM = 0
       SVOL = 0
       ISUM =0
       DO 40 I=1,256
       ISN(I) = 0
40     ISBIN(I)=0
C
C      GETDAT READS THE DATA AND UNPACKS THE DATA
C      ARRAY TO CONFORM TO ORIGINAL DATA-TAKING
C

C        DROP OUT DATA IF AIRSPEED IS LESS THAN 150,
C        BECAUSE LDV USES THIS VALUE TO TUNE TO
C           THEREFORE DATA MAY BE BAD
C        DROP   OUT DATA IF THE COUNT IN THE
C           LAST BIN IS GREATER THAN 5000
C           THIS MEANS THE INVERSION ALGORITH
```

```
C          CANNOT HANDLE THESE CASES
C
5      CALL  GETDAT(BUF,DATA)
        IF(DATA(256).GT.5000) GO TO 6
       IF(DATA(1).EQ.1E6) STOP
         IREC = IREC + 1
         IF(BUF(32) .GT. 150) GO TO 175
6       CONTINUE
         CALL GETDAT(BUF,DATA)
       IF(DATA(1).EQ.1E6) STOP
       IREC=IREC+1
         GO TO 5
175      CONTINUE
       IF (KKK .GT. 0) GO TO 173
        KSH = BUF(5)
        KSM = BUF(6)
         KSS = BUF(7)
         KSF = BUF(8)
173       CONTINUE
         IF (BUF(8) .LE.BUF(4)) GO TO 69
         BUF(3) = BUF(3) - 1
         BUF(4) = BUF(4) + 1000
69       NF = BUF(4) - BUF(8)
         IF (BUF(7) .LE. BUF(3)) GO TO 71
         BUF(2) = BUF(2) - 1
         BUF(3) = BUF(3) + 60
71       NS = BUF(3) - BUF(7)
         IF (BUF(6) .LE. BUF(2) ) GO TO 73
         BUF(1) = BUF(1) - 1
         BUF(2) = BUF(2) + 60
73       NM = BUF(2) - BUF(6)
         NH = BUF(1) - BUF(5)
         KS = KS + (3600 * NH) + (60 * NM) + NS
         KF = KF + NF
         IF (KF .LT. 1000) GO TO 74
         KF = KF - 1000
         KS = KS + 1
74       CONTINUE
         KBUF = KBUF + BUF(32)
         KKK = KKK + 1
C
C      ISUM IS THE TOTAL NUMBER OF PARTICLES FOR ALL BINS.
C           LATER, THE NOISE IS SUBTRACTED FROM ISUM.
C
C      DATA IS AN INTEGER ARRAY CONTAINING A RUNNING TOTAL OF PARTICLE
C           COUNTS FOR EACH BIN.
C
C      VB IS THE VOLUME BETA COUNT FOR THIS RECORD.
C
C      IFG IS THE IF GAIN FOR THIS RECORD.
C
C      SVOL IS THE RUNNING SUM OF THE VOLUME BETA COUNT.
```

```
C
C
C     ISN IS AN INTEGER ARRAY CONTAINING THE RUNNING TOTAL OF PARTICLE
C         COUNTS FOR THE NOISE DATA IN EACH BIN.
C
C     INOISE IS THE TOTAL NUMBER OF PARTICLE COUNTS FOR THE NOISE DATA.
C             LATER, INOISE IS SUBTRACTED FROM ISUM.
C
      DO 50 I=1,256
      ISUM=ISUM  + DATA(I)
   50 ISBIN(I) = ISBIN(I) +DATA(I)
      ITR = ISL(BUF(15),16)
      VB = IOR(ITR,BUF(16))
      IFG = BUF(25)
      NSUM = NSUM + BUF(17)
      SVOL = SVOL + VB
C
C     CALL SUBROUTINE GETDAT TO READ THE NOISE DATA RECORD.
C
       INOISE = 0
      CALL  GETDAT(BUF,DATA)
       IF(DATA(1).EQ.1E6) STOP
       IREC=IREC+1
      DO 60 L=1,256
      ISN(L) = ISN(L) + DATA(L)
   60 INOISE = INOISE + DATA(L)
       ISUM=ISUM-INOISE
      ITR = ISL (BUF(15),16)
      VB = IOR (ITR,BUF(16))
      IFG = BUF(25)
      NNSUM = NNSUM + BUF(17)
      SSVOL = SSVOL + VB
C      OUTPUT ISUM
C     OUTPUT ISUM,INOISE,ISBIN(256)
      IF (ISUM .LE. 1000)  GO TO 5
C
C      10000 PARTICLE COUNTS OR ABOVE MUST
C      BE OBTAINED BEFORE DATA IS SENT TO
C      INVERSION ALGORITHM
C
      KCH = BUF(1)
      KCM = BUF(2)
       KCS = BUF(3)
      KCF = BUF(4)
       IF(KSF .LE. KCF) GO TO 269
       KCS = KCS - 1
       KCF = KCF + 1000
  269  IFRAC = (KCF-KSF)/2
       IF (KSS .LE. KCS) GO TO 270
       KCM = KCM - 1
       KCS = KCS + 60
```

```
270   NSECS = (KCS -KSS)/2
       IF (KSM .LE. KCM) GO TO 271
       KCH = KCH - 1
       KCM = KCM + 60
271    NMN = (KCM - KSM)/2
       NHR = KCH - KSH
      LF = KSF +IFRAC
     IF (LF. LT. 1000) GO TO 331
     LF = LF - 1000
     KSS = KSS + 1
331  LS = KSS + NSECS
     IF (LS .LT. 60) GO TO 332
     LS = LS - 60
     KSM = KSM + 1
332  LM = KSM + NMN
     IF (LM .LT. 60) GO TO 334
     LM = LM - 60
     KSH = KSH +1
334  LH = KSH + NHR
     OUTPUT IREC
     WRITE(6,138) VB,IFG,BUF(17)
138     FORMAT(1X,"VB,IFG,BUF(17)= ",F10.1,2I10)
    DO 200 I=1,51
C
C   A SCALE FACTOR OF 500 CAME FROM A CONVERSATION WITH WILLIAM JONES.
C
    YBUF(I) = CAL(71-IFG,I)/500
200 CONTINUE
C
C   VB IS THE ACCUMULATED VOLUME BETA COUNT FOR THE
C   NOISE DATA.
C
C    VSIGL IS THE VOLUME BETA LINEAR SIGNAL
C           VSIGL = DATA - NOISE.
C
C    RVB IS THE AVERAGE VOLUME BETA COUNT FOR THE DATA.
C
C    VNS IS THE VOLUME BETA LINEAR NOISE SIGNAL.
C
C   FINALB IS THE FINAL VOLUME BETA FOR THIS DATA SET.
C
C   THE INTEGRATED VOLUME BETA(RVB) IS MULTIPLIED BY A
C   BIDIRECTIONAL REFLECTANCE PARAMETER(  = .016)
C   THEN DIVIDED BY THE SIGNAL RETURN FOR A
C   SANDPAPER DISK AT 10 METERS AT THE WAIST
C   ( = 1.8428924E6) AND THEN DIVIDED BY THE L-EFF
C   AT 10 METERS ( = 0.6407). SEE APPLIED
C   RESEARCH REPORTS FOR COMPLETE DOCUMENTATION.
C
C
    VB = SSVOL/NNSUM
C
```

```
C      CONVERT VB TO DBMS ( NOISE DATA ).
C
       DBMNS = YLIN(51,VB,YBUF,XNOISE)
        RVB=SVOL/NSUM
C
C      CONVERT RVB TO DBMS ( DATA ).
C
        SDBMN=YLIN(51,RVB,YBUF,XNOISE)
        VSIGL=10**(SDBMN/10.)-10**(DBMNS/10.)
        VNS=10**(DBMNS/10.)
        FINALB=VSIGL/VNS*.016/1.8428924E6/.6407
C       OUTPUT DBMNS
         OUTPUT FINALB,VSIGL,VNS,SSVOL,SVOL,NSUM
C
C      VB IS USED TO EXTRACT A NOISE VALUE FOR
C       THIS SET OF SINGLE PARTICLE DATA
C       BY INTERPOLATING
C      THE CALIBRATION DATA (CAL(I,J))
C
C      DBMNS - THE NOISE IN DBMS
C      ITHRESH - THE BIN WHERE THE SIGNALS ARE
C                THRESHOLDED - ABOVE ITHRESH CONTAINS
C                DATA
C     NP - THE NUMBER OF PARTICLES ARRAY -
C          CONTAINS NUMBER OF PARTICLE HITS IN THE
C          SELECTED BINS
C
C     THE FACTOR 1.5/.86 APPEARS IN THE NOISE
C     CALCULATION(DBMNN) BECAUSE THIS IS THE
C     DIFFERENCE IN BANDWIDTH BETWEEN THE SINGLE
C     PARTICLE DATA AND THE VOLUME CHANNEL DATA
C
C      DBMNN IS THE SINGLE BETA NOISE FACTOR ( DBMS ).
C
C      SGL (I) IS THE LINEAR SIGNAL VALUE THAT EACH PROCESSOR BIN
C               USED CORRESPONDS TO.
C
C      NP (J) IS THE ACCUMULATION OF SINGLE PARTICLE
C              COUNTS IN THE BIN RANGES SELECTED FROM
C              ARRAY XDS.
C
        CNV=10**(DBMNS/10)
        DBMNN=10*LOG10(CNV*1.5/.86)
       DO 210 I = 1,M+1
       SGL(I) = 10**((XSGL(I)-DBMNN)/10)
210    CONTINUE
C       OUTPUT SGL
       K =ITHRESH + 1
       DO 66 JJ=1,22
       NP(JJ) = 0
66     CONTINUE
        DO 55 J=1,M+1
```

```
      KK=K+XDS(J)-1
      DO 10 I = K,KK
      NP(J) = NP(J) + ISBIN(I)-ISN(I)
10      CONTINUE
      K=KK+1
55    CONTINUE
C      OUTPUT NP,ISBIN,ISN
C
C
C     DETERMINE SIGMAS THAT CORRESPOND TO SIGNAL BINS
C     1.E8- CONVERTS FROM CM**2      TO MICRON**2
C     1.10376E7  IS THE SINGLE PARTICL GAIN AT AN AREA OF ZERO
      DO 300 J = 1,M  + 1
       SIG(I) = SGL(I)*1.E8/1.10376E7
300   CONTINUE
C
C     SIG IS THE CROSS SECTION ARRAY.
C
C     SIL IS THE SMALLEST CROSS SECTION.
C
C     SIH IS THE LARGEST CROSS SECTION.
C
      SIL = SIG(1)
      SIH = SIG(M  + 1)
C      OUTPUT SIG
        OUTPUT SIL,SIH
       OUTPUT ISUM
C
C     CALL SUBROUTINE DATANAL TO DO ACTUAL INVERSION.
C
C
C     ELS - THE ELAPSED TIME IN SECS FOR WHICH DATA
C           WAS TAKEN
C     FLTL - THE FLIGHT LENGTH IN METERS
C     SINGLEB - THE SINGLE PARTICLE BETA
C      WS - THE AVERAGE CROSS-SECTION OF PARTICLE
C           SQUARE MICRONS
C       WV - THE AVERAGE SIZE OF THE TRANSVERSE
C            CROSS-SECTION OF FOCAL VOLUME
C            PREDICTED FROM INVERSION ALGORITHM
C
      CALL DATANAL(ISUM,M,MS,WS,WV)
C
      ELS = KS + KF/1000.
      FLTL = KBUF/KKK*.5144*ELS
        IF(WV.EQ.0) WV=1.E-30
      SINGLEB = ISUM/4/3.14159265359/FLTL/WV*WS
      OUTPUT SINGLEB,FLTL,ELS
      WRITE(6,702) BUF(9),BUF(10), BUF(11)
702    FORMAT(1X,'DATE ',I2,'-',I2'-',I2)
      WRITE (6,703) LH,LM,LS,LF
703   FORMAT(1X,'TIME',I2,':',I2,':',I2,'.',I3)
```

```
      WRITE(6,704)
704   FORMAT(/////)
       WRITE(13) LH,LM,LS,LF,SINGLEB,FINALB,WS,FLTL,ELS,WV
      ISUM = 0
      INOISE=0
      GO TO 122
878      STOP
      END
C
C
C
        SUBROUTINE GETDAT(BUF,DATA)
C
C       THIS SUBROUTINE READS THE DATA FROM DISK AS 192 WORDS.
C       EACH WORD READ IN CONTAINS 4 BYTES.
C       THE DATA IS THEN CONVERTED SO THAT THERE ARE 384 WORDS.
C
        DIMENSION DUM(7),DUMM(31),ID(20),DATA(256),BUFF(192),BUF(384)
        IMPLICIT INTEGER(C,S,D,F,V,B)
          III=0
  10       CONTINUE
          III=III+1
        II=0
 301    FORMAT(1X,6HBUFF= ,48(1X,8I8,/))
        CALL BUFFER IN(5,1,BUFF,192,ISTAT,NWRD,IAB)
C       OUTPUT ISTAT,NWRD,IAB
C        WRITE(6,301)  (BUFF(I),I=1,384)
         IF(ISTAT.EQ.3) DATA(1)=1E6
         IF(ISTAT.EQ.3) RETURN
        DO 300 I=1,384,2
        II=II+1
        I1=IAND(BUFF(II),8ZFF000000)
        I2=IAND(BUFF(II),8Z00FF0000)
        I3=IAND(BUFF(II),8Z0000FF00)
        I4=IAND(BUFF(II),8Z000000FF)
        I1=ISL(I1,-24)
        I2=ISL(I2,-8)
        I3=ISL(I3,-8)
        I4=ISL(I4,8)
        BUF(I)=IOR(I2,I1)
        BUF(I+1)=IOR(I4,I3)
 300      CONTINUE
        DO 100 K=1,20
        ID(K)=BUF(K+48)
100     CONTINUE
        DO 200 L=1,256
        DATA(L)=BUF(L+128)
200     CONTINUE
C       ************************************************************
C       CH = CURRENT TIME HOURS
C       CM = CURRENT TIME MINUTES
C       CS = CURRENT TIME SECONDS
```

```
C      CF = CURRENT TIME FRACTIONS
C      SH = START OF RECORD HOUR
C      SM = START OF RECORD MINUTE
C      SS = START OF RECORD SECOND
C      SF = START OF RECORD FRACTION
C      DBT = NUMBER BEFORE DIS. THRESHOLD
C      DT  = DISCRIMINATOR THRESHOLD
C      FBI = NUMBER BEFORE FORSARD/AFT IDENTIFIER
C      FI  = FORWARD/AFT IDENTIFIER
C      DATA = DATA
C      ID =  IDENTIFICTION STRING
C      *******************************************************
        CH=BUF(1)
        CM=BUF(2)
        CS=BUF(3)
        CF=BUF(4)
        SH=BUF(5)
        SM=BUF(6)
        SS=BUF(7)
        SF=BUF(8)
        DBT=BUF(23)
        DT=BUF(24)
        FBI=BUF(25)
        FI=BUF(26)
C      WRITE(6,401) SH,SM,SS,SF
401     FORMAT(1X,'START TIME: ',4I10)
C      WRITE(6,402) CH,CM,CS,CF
C      WRITE(6,405) BUF(15),BUF(16),BUF(17)
405    FORMAT (1X,'VOLUME BETA INT',3I10)
402       FORMAT(1X,'END TIME: ',4I10)
C      WRITE(6,403) DT
403     FORMAT(1X,'DISCRIMINATOR THRESHOLD ',I10)
C      WRITE(6,406) BUF(25)
406    FORMAT (1X,'IF GAIN',I10)
C      WRITE(6,404) FI
404     FORMAT(1X,'FORWARD/AFT ID ',I10)
C      WRITE(10) (ID(I),I=1,20)
C      WRITE(6,120) (DATA(I),I=1,128)
120    FORMAT(1X,16(1X,8I8,/))
       IF (III.GT.500) RETURN
       RETURN
       END
C
C
       SUBROUTINE DATANAL(NT,M,MS,WS,WV)
C
C      THE PRIMARY FUNCTION OF THIS SUBROUTINE IS TO DEFINE
C      THE ARFA CURVE PROPERLY FOR THE CALL TO SUBROUTINE
C      OPTIM. SOME OF THE CALCULATIONS DONE HERE ARE NOW
C      UNNECESSARY.
C
C
```

```
C     ARR IS AN ARRAY CONTAINING THE UNSCALED LINEAR SIGNAL / NOISE / SIGMA
C           VALUES THAT CORRESPOND TO THE AREAS UNDER THE AREA CURVE.
C
C     ARR VS. SIGMA / Y IS THE UNSCALED AREA CURVE.
C
C     DS IS AN ARRAY CONTAINING THE RANGE OF SIGMA VALUES IN THE SIG ARRAY.
C
C     AR IS THE FINAL ARRAY OF AREA VALUES THAT CORRESPOND TO
C           AXX, THE SIGNAL / NOISE / SIGMA VALUES.
C
C     SRMN IS THE MINIMUM SIGNAL / NOISE / SIGMA VALUE REQUIRED FROM
C           THE AREA TABLE.
C
C     SRMX IS THE MAXIMUM SIGNAL / NOISE / SIGMA VALUE REQUIRED FROM
C           THE AREA TABLE.
C
C     TRANS IS THE MULTIPLICATIVE FACTOR THAT CORRECTS
C      THE AREA CURVE
C       SO THAT THE AREAS CORRESPOND TO THE PROPER S/N/SIGMA
C
      DIMENSION DAND(101,2)
      DIMENSION YY(125),ARR(125)
      COMMON/OPT/DS(22),NP(22),FSI,FN
      COMMON/ARE/F(2),AL,RD,R1,A
      COMMON/ARE1/AR(101),DSR,SRMX,AXX(101)
      COMMON/COE2/SIG(22),SGL(22),WI
      COMMON/SS/X(2),Y(2),Z(2)
      DATA PW,ETA,BW,RO,F,AL,AH,NI,FSI/5.,.05,
     1 1.E+6,.046,10.,10.,1.,3.1622777,30,1.973/
     DATA PI,HP,CL,FLDA
     1/3.14159,6.62562E-34,3.E+8,10.6E-6/
      DATA B,IP/1.5,1/
      DATA ARR/31*2.4E5,2.39E5,2.38E5,2.37E5,2.36E5,
     1 2.35E5,2.34E5,2.32E5,2.31E5,2.30E5,2.30E5,2*2.29E5,
     1 2.28E5,2.27E5,2.25E5,2.22E5,2.20E5,2.19E5,2.16E5,
     1 2.12E5,2.11E5,2.10E5,2.1E5,2.05E5,2.01E5,1.98E5,1.92E5,
     1 1.89E5,1.85E5,1.80E5,1.76E5,1.72E5,1.69E5,1.65E5,
     1 1.60E5,1.57E5,1.53E5,1.50E5,1.46E5,1.43E5,1.38E5,1.35E5,
     1 1.32E5,1.25E5,1.19E5,1.13E5,1.07E5,1.03E5,9.7E4,
     1 9.3E4,8.9E4,8.6E4,8.3E4,7.7E4,7.5E4,7.3E4,6.9E4,
     1 6.6E4,6.2E4,5.6E4,5.2E4,4.8E4,4.5E4,4.2E4,3.9E4,3.7E4,
     1 3.3E4,2.9E4,2.6E4,2.35E4,2.15E4,1.95E4,1.8E4,1.75E4,
     1 1.52E4,1.40E4,1.3E4,1.2E4,1.11E4,1.03E4,9.7E3,9.2E3,
     1 8.6E3,8.1E3,7.5E3,7.2E3,6.3E3,5.6E3,5.0E3,4.5E3,
     1 4.0E3,3.5E3/
      DATA YY/0,0.01,0.02,0.03,0.04,0.05,0.06,0.07,0.08,0.09,
     A .10,.11,.12,.13,.14,.15,.16,.17,.18,.19,.20,
     A .21,.22,.23,.24,.25,.26,.27,.28,.29,.30,
     A .32,.34,.36,.38,.40,.42,.44,.46,.48,.50,.52,.54,.56,
     A .58,.60,.65,.70,.75,.80,.85,.90,.95,1.0,1.1,1.2,1.3,
     A 1.4,1.5,1.6,1.7,1.8,1.9,2.0,2.1,2.2,2.3,2.4,2.5,
     A 2.6,2.7,2.8,2.9,3.0,3.2,3.4,3.6,3.8,4.0,4.2,4.4,4.6,
```

```
     A 4.8,5.0,5.2,5.4,5.6,5.8,6.0,6.5,7.0,7.5,8.0,8.5,9.0,
     A 9.5,10.0,11.,12.,13.,14.,15.,16.,17.,18.,19.,20.,21.,
     A 22.,23.,24.,25.,26.,27.,28.,29.,30.,32.,34.,36.,38.,
     A 40.,42./
      DATA TRANS/45.99/
      DATA N/3E6/
      DATA ITEST/1/
      IF (ITEST .EQ. 0) GO TO 744
      DO 999 I=1,123
      ARR(I) = (ARR(I) * TRANS)   * 1E-8
999   CONTINUE
C
C    1.E-4 IS USED TO CONVERT THE AREAS FROM CM**2 TO M**2
C
      DO 998 I = 1,123
      YY(I) = YY(I) * 1E-4
998   CONTINUE
      DO 77 I = 1,123
      J = 124 - I
      IF(I.GT.J) GO TO 78
      ATEMP = ARR(I)
      ARR(I) = ARR(J)
      ARR(J) = ATEMP
      YTEMP = YY(I)
      YY(I) = YY(J)
      YY(J) = YTEMP
77    CONTINUE
78      CONTINUE
      ITEST = 0
744   A1 = AX*FLDA*FLDA*ETA/BW
     A=A1/(16.*PI*PI*PI)*1.E-12
     RD=PI*RO*RO/FLDA
     ALP=RD/F(1)
     R1=RO/2./RD
     REO1=F(1)*R1
     REO2=F(2)*R1
     G1=A/(REO1*REO1*REO1*REO1)
      G2=A/(REO2*REO2*REO2*REO2)
     DO 4 I = 1,M
     DS(I) = SIG(I+1) - SIG(I)
4    CONTINUE
     X(1)=0.
     Y(1)=0.
     Z(1)=0.
     SIL = SIG(1)
     SIH=SIG(M+1)
     SRMN=SGL(1)/SIH
     SRMX=SGL(1)/SIL
     ALS=1.+ALP*ALP
     G=SQRT(SRMX/SRMN)
     SQ=SQRT((G-1.)*ALS+1.)
     DSR=(SRMX-SRMN)/100.
```

```
      NPTS = 101
C           OUTPUT ARR,YY
      DO 2 I=1,NPTS
      J = NPTS - I + 1
      SR = SRMX - DSR * (I-1)
        AXX(J) = SR
        AR(J) = YLIN(123,SR,ARR,YY)
2     CONTINUE
C     OUTPUT AXX,AR
C     WRITE(103)(DAND(I,1),I=1,101),(DAND(I,2),I=1,101)
C     GOTO 65
C     CALL AREA(AT,SRMN)
C     CALL AREA1(AT1,SRMN)
C     OUTPUT AT,AT1
C     OUTPUT RD,ALP,ALS,G,SQ,F(1)
C     OUTPUT SRMN,SRMX,(SIG(I),I=1,MS+1),(SGL(I),I=1,M+1),XL,YL,ZL
      B = 2.5
        CALL OPTIM(B,NT,M,MS,NI,    WS,WV,YY,ARR)
      TSIA=N*WS
      OUTPUT WV,WS
C     OUTPUT TSIA
      IF(WV.EQ.0.) WV=1.E-30
      BS =NT*WS/WV
      BM =4.*SMP*FLDA/A1
479   FORMAT(6I12)
489   FORMAT (12F8.3)
      RETURN
65    END
C
C
      SUBROUTINE COEF2(C,I,J,K,DJ,M)
C
C     THIS SUBROUTINE CALCULATES THE DIFFERENCE BETWEEN TWO
C     AREA VALUES IN AN INTEGRATION STEP.
C
C     INPUTS :
C
C     I IS THE INDEX IN ARRAY SGL.
C
C     J IS THE INDEX IN ARRAY SIG.
C
C     K IS THE INTEGRATION STA/EP.
C
C     DJ IS THE VALUE OF SIGMA FOR A SINGLE INTEGRATION STEP.
C
C     M IS THE NUMBER OF SIGMAS - SIG.
C
C     OUTPUTS :
C
C     C IS THE AREA DIFFERENCE.
C
      COMMON/COE2/SIG(22),SGL(22),WI
```

```
      COMMON/ARE1/AR(101),DSR,SRMX,AXX(101)
      A2=0.
      SI=SIG(J)+(K-1)*DJ
      SR=SGL(I)/SI
      A1 = YLIN(101,SR,AXX,AR)
      IF(I.GT.M+1) GO TO 1
      IF(I.EQ.J) GO TO 1
      SR=SGL(I+1)/SI
      A2 = YLIN(101,SR,AXX,AR)
    1 C =A1-A2
      RETURN
      END
C
C
      SUBROUTINE LPICK(SIG,MS,   BP,DB,ALP,DALP)
C
C     THIS IS THE LOG NORMAL PICK SUBROUTINE; IT'S PURPOSE
C     IS TO GENERATE A GOOD STARTING POINT FOR THE LOG-NORMAL
C     SOLUTION.
C
C     INPUTS :
C
C     SIG IS THE CROSS SECTION ARRAY.
C
C     MS IS THE NUMBER OF SIGMAS IN THE ARRAY.
C
C     OUTPUTS :
C
C     ALP IS THE FITTING PARAMETER FOR THE LOG-NORMAL SOLUTION.
C
C     DALP IS THE FITTING PARAMETER INCREMENT FOR THE L-N SOL'N.
C
      DIMENSION SIG(22)
      DB=(SIG(MS+1)-SIG(1))/20.
       BP=(SIG(1)+SIG(MS+1))/2.
      A1=SQRT(LOG(SIG(MS+1))/2./SIG(1)+LOG(SIG(1))**2/2./SIG(1)
     1 /LOG(SIG(MS+1))-LOG(SIG(1))/SIG(1))
      DALP=A1/20.
       ALP=A1/2.
       RETURN
       END
C
C
      SUBROUTINE EPICK(SIG,BPI,BP,MS,DP)
C
C     THIS IS THE EXPONENTIAL PICK SUBROUTINE; IT'S PURPOSE
C     IS TO GENERATE A GOOD STARTING POINT FOR THE EXPONENTIAL
C     SOLUTION.
C
C     INPUTS :
C
C     SIG IS THE CROSS SECTION ARRAY.
```

```
C
C      MS IS THE NUMBER OF SIGMAS IN THE ARRAY.
C
C      BP1 IS THE POWER LAW SOLUTION PARAMETER.
C
C      OUTPUT :
C
C      BP IS THE EXPONENTIAL POWER PICK.
C
C      DP IS THE EXPONENTIAL POWER INCREMENT.
C
       DIMENSION SIG(22)
       S1=BP1*LOG(SIG(1))/SIG(1)
       S2=BP1*LOG(SIG(MS+1))/SIG(MS+1)
       BP=S2
       DP= (S1-S2)/25.
       RETURN
       END
C
C
        SUBROUTINE PROW(BP,IP,SIP,M)
C
C      THIS SUBROUTINE COMPUTES NORMALIZED PROBABILITIES FOR
C      THE POWER LAW, EXPONENTIAL, AND LOG-NORMAL DISTRIBUTIONS
C      FOR THE INPUT VALUES OF BP AND SIP USING THE SIG ARRAY.
C
C      INPUTS :
C
C      BP IS THE FITTING PARAMETER.
C
C      IP DETERMINES WHICH SOLUTION TO USE.
C          IP = 1 => POWER LAW
C          IP = 2 => EXPONENTIAL
C          IP = 3 => LOG-NORMAL
C
C      SIP IS THE CROSS SECTION VALUE.
C
C      SO AND A3 ARE NORMALIZATION FACTORS REQUIRED ONLY
C                FOR THE LOG-NORMAL SOLUTION.
C
C      OUTPUTS :
C
C      W IS THE NORMALIZED PROBABILITY.
C
         COMMON/COE2/SIG(22),SGL(22),W
       COMMON/ALN/S0,A3
C      OUTPUT   IP,BP,SIP
        IF(SIP.LE.0.) OUTPUT SIP
        IF(SIP.LE.0) W=0
        IF(SIP.LE.0) RETURN
       GOTO (1,2,3)IP
     1 X=1.-BP
```

```
      A1=-(SIG(1)**X-SIG(M+1)**X)/X
      IF(BP.GT..999.AND.BP.LT.1.001)A1=LOG(SIG(M+1)/SIG(1))
      W=(SIP**(-BP))/A1
      RETURN
    2 A2=(EXP(-BP*SIG(1))-EXP(-BP*SIG(M+1)))/BP
      W=EXP(-BP*SIP)/A2
      RETURN
    3 W=(EXP(-(LOG(SIP)-LOG(BP))**2/2./S0**2))/A3
      RETURN
      END
C
C
      SUBROUTINE OPTIM(B,N,M,MS,NI   ,WSS,WVS,YY,ARR)
C
C     THIS SUBROUTINE OPTIMIZES THE BETA INVERSION AND SELECTS
C     THE CORRECT DISTRIBUTION OF CROSS SECTIONS TO PROVIDE
C     THE BEST FIT.
C
      DIMENSION PN(22)
      DIMENSION ARR(125),YY(125)
      COMMON/OPT/DS(22),NP(22),FSI,FN
      COMMON/COE2/SIG(22),SGL(22),WI
       COMMON/ALN/S0,A1
      COMMON/ARE1/AR(101),DSR,SRMX,AXX(101)
      OUTPUT IP,B
      NC=0
       IP=1
      IC=0
      BP=B
       IT=0
        BP=B
        DB=.1
   10 CONTINUE
      II=0
      RM=1.E+60
       IF(NC.GT.1)IC=1
      IF(IC.EQ.1)NC=0
       IP=2
      IF(IC.EQ.0)IP=1
       IF(IT.GE.2)IP=3
        IC=0
C
C         THIS CHECK NOW  DEFEATS THE EXP AND LOG-NORMAL SOLNS
C          TO USE THE EXP. AND LOG-NORM. SOLUTIONS AGAIN JUST
C         REMOVE THE STMT. BELOW (   IF (IP.EQ.2) RETURN   ).
C
C
        IF(IP.EQ.2) RETURN
C
       IF(IP.EQ.2) CALL EPICK(SIG,BP1,BP,MS,DB)
       IF(IT.EQ.2)CALL LPICK(SIG,MS,   AM,DB,S0,DALP)
    8   CONTINUE
```

```
      II=II+1
      IF(II.GT.100) RETURN
      IF(BP.LT.0..AND.IP.EQ.1) RETURN
        IF(IP.LE.2)GOTO 102
        AM=AM+DB
        IF(AM.LT.1.) DB=0.
        IF(AM.LT.1.) AM=ABS(AM/10)
        IF(IT.EQ.3)S0=S0+DALP
        IF(IT.EQ.3)AM=BP1
C
C       NORMALIZATION OF LOG-NORMAL DISTRIBUTION
C
        A1=0.
        DO 101 J=1,MS
        DJ=DS(J)/NI
        DO 100 K=1,NI+1
       SIP=SIG(J)+(K-1)*DJ
        W=EXP(-(LOG(SIP)-LOG(AM))**2/2./S0**2)
100     A1=A1+W*DJ
        SI1=SIG(J)
        SI2=SIG(J+1)
        W1=EXP(-(LOG(SI1)-LOG(AM))**2/2./S0**2)
        W2=EXP(-(LOG(SI2)-LOG(AM))**2/2./S0**2)
101     A1=A1-.5*(W1+W2)*DJ
        BP=AM-DB
C      OUTPUT S0,A1
102     CONTINUE
C
C     RESET PARTICLE PROBABILITIES TO ZERO.
C
    6 DO 5 I=1,MS
    5 PN(I)=0.
      BP=BP+DB
        IF(IT.EQ.3)BP=BP1
      OUTPUT BP
      WS=0.
      WV=0.
C
C     THESE LOOPS PRODUCE THE AVERAGE CROSS SECTION AND FOCAL VOLUME
C     ( WS AND WV ).
C
       DO500 J=1,MS
       DJ=DS(J)/NI
      DO 501 K=1,NI +1
       SIP=SIG(J)+(K-1)*DJ
      SR=SGL(1)/SIP
      ASK=YLIN(101,SR,AXX,AR)
      CALL PPOW(BP,IP,SIP,MS)
       WS=WS+WI*SIP*DJ
501    WV=WV+WI*ASK*DJ
       SI1=SIG(J)
       SI2=SIG(J+1)
```

```
      SR=SGL(1)/SIG(J)
      AS1=YLIN(101,SR,AXX,AR)
      SR=SGL(1)/SI2
      AS2=YLIN(101,SR,AXX,AR)
      CALL PROW(BP,IP,SI1,MS)
      W1=WI
      CALL PROW(BP,IP,SI2,MS)
      W2=WI
C
C     SUBTRACT OFF END POINTS.
C
      WS=WS-.5*(W1*SI1+W2*SI2)*DJ
  500 WV=WV-.5*(W1*AS1+W2*AS2)*DJ
      WS=WS+1.E-12
      WSS=WS1
      WVS=WV1
      WS1=WS
      WV1=WV
C
C     N IS THE NUMBER OF SINGLE PARTICLE COUNTS THAT THIS
C       INVERSION IS DOING.
C
C     PREDICTED DENSITY = NO. PARTICLES / AVG. FOCAL VOL.
C
C     THESE LOOPS PRODUCE THE CROSS SECTION DISTRIBUTION
C     PROBABILITY ARRAY ( PN ).
C
      FN=N
      DNT=FN/WV
      BT=DNT*WS
      DO 1 I=1,M+1
      DO 1 J=1,MS
      DJ=DS(J)/NI
      DO 4 K=1,NI+1
      SIP=SIG(J)+(K-1)*DJ
      CALL PROW(BP,IP,SIP,MS)
      CALL COEF2(C,I,J,K,DJ,M)
    4 PN(I)=PN(I)+C*WI*DJ
      SIP=SIG(J)
      CALL PROW(BP,IP,SIP,MS)
      W1=WI
      SIP=SIG(J)+NI*DJ
      CALL PROW(BP,IP,SIP,MS)
      W2=WI
      CALL COEF2(C1,I,J,1,DJ,M)
      CALL COEF2(C2,I,J,NI +1,DJ,M)
    1 PN(I)=PN(I)-.5*(C1*W1+C2*W2)*DJ
      R=0.
      DO 2 I=1,MS
C
C     FN IS THE ACTUAL DATA
C
```

```
C     FP IS THE PREDICTED VALUES
C
C     R IS THE DIFFERENCE BETWEEN THE PREDICTED AND
C     ACTUAL DATA SQUARED.
C
      FN=NP(I)
      FP=PN(I)*DNT
C     OUTPUT R,FN,FP,I
       IF(I.LE.M) R=(FN-FP)**2+R
    2  CONTINUE
      NC=NC+1
C      OUTPUT NC,DNT
      IF(NC.NE.2)GOTO 11
      IF(R.LT.RM)GOTO 11
C       OUTPUT R,RM
C
C     SAVE OLD VALUE
C
      RM=R
      DB=-DB
       IF(IT.EQ.3) DALP=-DALP
C
C     JUST IN CASE SOLUTION IS WORSE ON 2ND PASS.
C
      GOTO 8
   11 CONTINUE
C
C     IF NEW SOLUTION BETER THAN PREVIOUS ONE
C     THEN CONTINUE TO ITERATE.
C
      IF(R.LT.RM) RM=R;GO TO 8
   19 FORMAT( 5F10.0)
   29 FORMAT(F10.3,4E12.3)
   39 FORMAT(5F12.4)
   49 FORMAT(5F10.0)
      IT=IT+1
C
C     WSS IS THE AVERAGE SIGMA BAR.
C
C     WVS IS THE AVERAGE FOCAL VOLUME.
C
C     1. POWER LAW SOL'NS.
C
      IF(IT.EQ.1)RP=RM;WSP=WSS;WVP=WVS
C
C     2. EXPONENTIAL SOL'NS.
C
      IF(IT.EQ.2)RE=RM;WSE=WSS;WVE=WVS
C
C     3. LOG-NORMAL SOL'NS.
C
```

```
      IF(IT.GT.2)RLN=RM;WSLN=WSS;WVLN=WVS
C
      BP1=BP-DB;OUTPUT BP1
      IF(BP.LT.0.) BP=0.
    7 BP=BP-DB
      IF(IT.LT.4) GOTO 10
      OUTPUT RP,RE,RLN
       OUTPUT WSP,WSE, WSLN,WVP,WVE,WVLN
      WSS=WSP
      WVS=WVP
C
C     IF THE POWER LAW SOL'N IS BEST
C     THEN RETURN FROM HERE.
C
      IF(RP.LT.RE.AND.RP.LT.RLN)RETURN
C
      WSS=WSE
      WVS=WVE
C
C     IF EXPONENTIAL SOL'N IS BEST
C     THEN RETURN FROM HERE.
C
      IF(RE.LT.RLN)RETURN
C
       WSS=WSLN
       WVS=WVLN
C
C     IF LOG-NORMAL SOL'N IS BEST
C     THEN RETURN FROM HERE.
C
      RETURN
C
      END
C
C
      FUNCTION YLIN(N,XX,X,Y)
C
C     THIS FUNCTION SUBPROGRAM PERFORMS A LINEAR INTERPOLATION.
C
C     N - NUMBER OF DATA POINTS
C     XX - X VALUE FOR WHICH Y VALUE MUST BE INTERPOLATED
C     X - DEPENDENT VARIABLE ARRAY
C     Y - INDEPENDENT VARIABLE ARRAY FOR WHICH
C         INTERPOLATIONS ARE PERFORMED
      DIMENSION X(1),Y(1)
      J=1
      DO 10 I=J,N
      IF (XX .LT. X(I)) GO TO 11
   10 CONTINUE
      I=N
   11 IF (I .EQ. 1) I=2
      J=I-1
```

```
      P=(XX-X(J))/(X(I)-X(J))
      IF (P .LT. 0. .OR. P .GT. 1.) P = 0.0
      YLIN=Y(J) +P*(Y(I)-Y(J))
      RETURN
      END
```