https://ntrs.nasa.gov/search.jsp?R=19750002071 2020-03-23T03:30:33+00:00Z

NASA CR-13747

APPLIED SCIENCE ASSOCIATES, INC.
105 EAST CHATHAM STREET
POST OFFICE BOX 949
APEX, NORTH CAROLINA 27502
PHONE (919) 362-7256

Engineering Studies Related to the Skylab Program

by

G. S. Hayne

(NASA-CR-137473) ENGINEERING STUDIES RELATED TO THE SKYLAB PROGRAM Final Report (Applied Science Associates, Inc., Apex, N.C.) 105 p HC $5.25 CSCL 22C
N75-10143
Unclas
G3/15 53457

Final Report for Task F

September, 1974

Prepared under
NASA Contract No. NAS6-2307

for

National Aeronautics and Space Administration
Wallops Flight Center
Wallops Island, Virginia 23337

NOV 1974
RECEIVED
NASA STI FACILITY
INPUT BRANCH

TABLE OF CONTENTS

Page

SECTION I. INTRODUCTION

This final report on Task F of Contract NAS6-2307 must be regarded as a report of progress to date in the problem of analysis and interpretation of the waveform-related data from the S-193 radar altimeter experiment. The waveform analyses are not complete and a continuing series of different "cut and try" calculations must be done; the procedures which are presented in this report provide the basic building blocks for this continuing search.

In very brief summary of the results, two major problem areas exist: 1.) determining the sample-and-hold (S&H) dc offsets, and also the parameters for a theoretical radar mean return waveform from a set of averaged S&H data as input, and 2.) performing a time-realignment of individual sets of S&H results prior to averaging in order to correct for the altimeter tracker time-jitter. For the first problem area, a general-purpose chi-squared minimizing computer program, X2MIN, has been used to best-fit the non-time-realigned S&H averages, and the results are encouraging - a set of S&H offsets is found and some consistency is observed between these offsets seen in various "good" data passes. Time-realignment and averaging, the second problem area, is not in such good shape; we have thus far been unable to produce any improvement in our knowledge of the waveform by use of time-realignment before averaging, and therefore we have no confidence that any need exists for applying this cumbersome procedure to waveform data. These two areas, line-fitting and time-realignment, will be more thoroughly defined and discussed in the subsequent sections of this report.

For the waveform experiments the altimeter's tracker must be operating ("locked") when S&H data are acquired. This eliminates from any further consideration here the dual-pulse mode, Mode 3, and the Nadir Align Mode. Table 1 summarizes the (nominal) pulsewidths, IF bandwidths, S&H positions and spacing, and the number of frames of data for the remaining altimeter modes and submodes for which the altitude tracker is locked.

Virtually all our effort for this report has been spent on Mode 5, submode 2, the "brute force" short pulse (a nominal 20 ns pulse) experiment which was intended as a direct comparison submode for Mode 5's pulse compression submode, SM1. Since M5,SM1 was apparently operating incorrectly until

	Mode	Submode	Pointing Angle	IF Bandwidth	Pulse Width	S&H Spacing	Sub-submode	S&H Start	# of Data Frames(totals)
PULSE SHAPE EXPERIMENT	1	0 (narrow-band, on-nadir)	0.°	10 MHz	100 ns	25 ns	0	0 ns	15
							1	200 ns	15
							2	400 ns	15 (45)
		1 (wide-band, on-nadir)	0.°	100 MHz	100 ns	25 ns	0	0 ns	19
							1	200 ns	20
							2	400 ns	20 (59)
		2 (wide-band, off-nadir)	.431°	100 MHz	100 ns	25 ns	0	0 ns	19
							1	200 ns	20
							2	400 ns	20 (59)
CROSS-SECTION EXPERIMENT	2	0	0.°	10 MHz	100 ns	25 ns	0	0 ns	2
							1	200 ns	2
							2	400 ns	2 (6)
	[. . . tracker is disabled in submodes 2 through 5 . . .]								
		6	0.°	10 MHz	100 ns	25 ns	0	0 ns	2
							1	200 ns	2
							2	400 ns	2 (6)
PULSE-COMPRESSION EXPERIMENT	5	0 (long pulse)	0.°	10 MHz	100 ns	25 ns	0	0 ns	6
							1	200 ns	5
							2	400 ns	5 (16)
		1 (compressed pulse)	0.°	100 MHz	130 ns p.c.	10 ns	0	240 ns	64
							1	280 ns	15
							2	360 ns	5
							3	440 ns	15 (99)
		2 (short pulse)	0.°	100 MHz	20 ns	10 ns	0	240 ns	16
							1	280 ns	15
							2	360 ns	5
							3	440 ns	15 (51)

Table 1. Summary of Data Acquisition Submodes For Which Altitude Tracker Is Locked.

late in mission SL-3, only M5,SM2 remain as a source of short-pulse ocean-scattering information. The longer-pulse submodes (100 ns nominal) are important for determination of the ocean's radar backscattering cross-section, σ^o, whose effects are more pronounced in the trailing edge of the mean return waveform; apart from noting that the waveform fitting procedures described later in this report may also be used for refining the antenna pointing angle estimates which are necessary for σ^o calculations, we will not deal further here with σ^o. If any changes in rms ocean surface roughness are detectable in the S-193 altimeter data, these changes should manifest themselves in the leading edge of the short-pulse mean return waveform. Moreover, the tracker jitter should leave no measurable effect on the leading edge of the 100 ns mean return waveform while estimates made prior to Skylab launch indicated that the tracker jitter would have an appreciable effect (increasing the apparent risetime and also increasing the variance) on the leading edge of the 10 ns short pulse mean return waveform. (Note the distinction here between the nominal 10 ns design goal and the nominal 20 ns actually realized pulses; the tracker jitter corrections - the time-realignment procedures - were started with the 10 ns pulse in mind.)

Figure 1 sketches the overall waveform processing carried out by a series of labelled boxes; this is the diagram appropriate to an automatic waveform program which does not yet exist. Because of the number of different difficulties with the S-193 output data and of the only mixed successes in our work with the best of this data, such an overall program not only doesn't now exist but never will.

For instance, Box A of Figure 1 refers to editing of data, but because of the number of missing modes or submodes, of tracker loss-of-lock, and of other data drop-outs, it has not been possible to even begin to define an automatic data editing set of criteria. Instead the editing has been a manual operation; in fact only those submodes having no apparent loss-of-lock or data dropouts over an entire submode have been used in our work to date, deferring until later the questions of what to do about cases in which part of a submode should be edited out.

The waveform time-realignment procedure includes the functions of boxes E,F, and G of Figure 1 to produce the necessary tracker jitter time-correction,

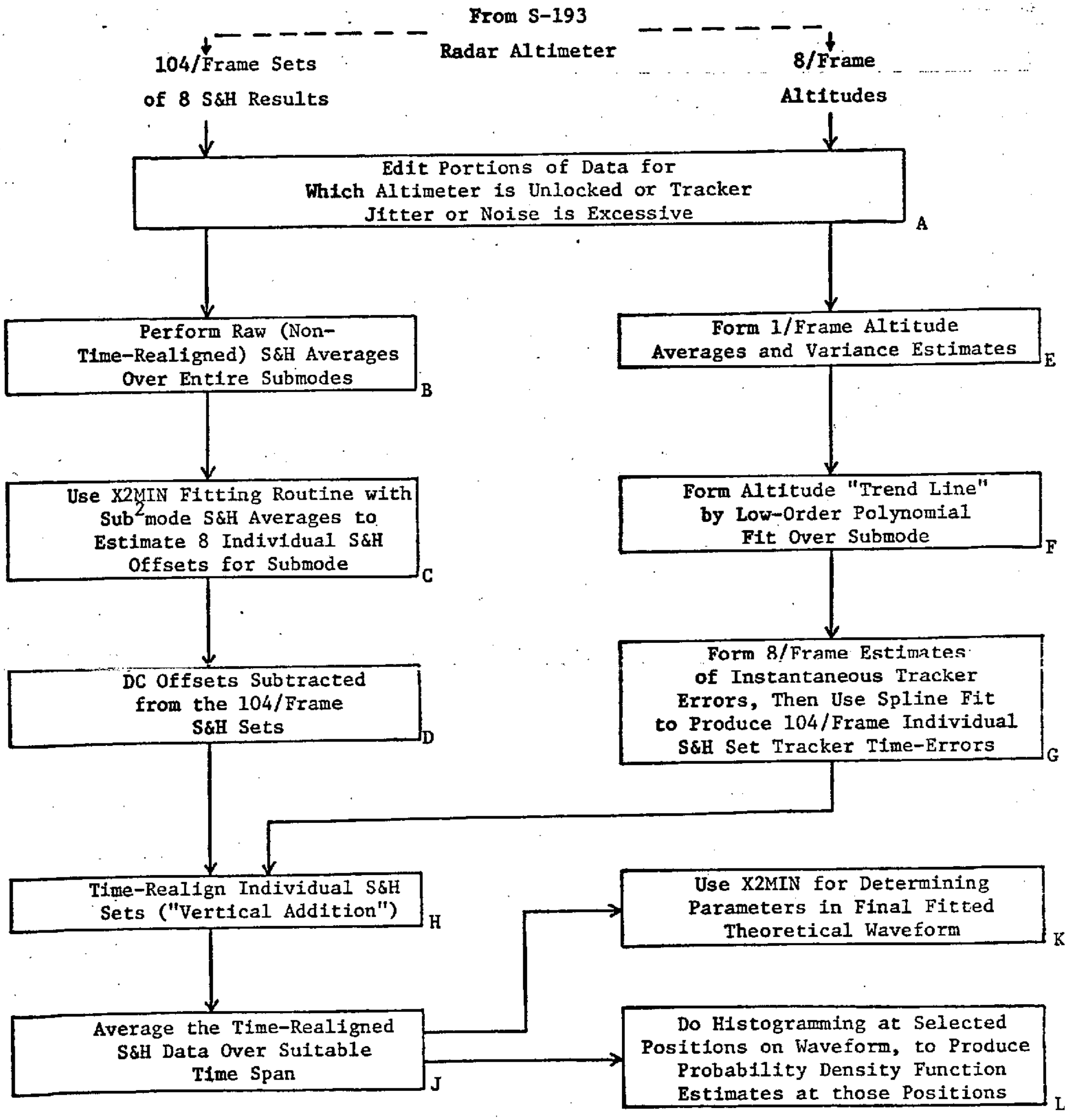


Figure 1. Idealized Overall Waveform Processing Procedure

then boxes H and J use these corrections to obtain the final time-realigned average waveform. At present, these two steps (that is, D&E&F and then G&J) are performed separately instead of within a single program. Box K is an entirely straightforward application of the line fitting program to the final time-realigned average waveform sample but has not been implemented because of difficulties to be discussed in the section on time-realignment. The programming for box L of Figure 1 has been completed and is sketched later in this report; at the time of this report's preparation the histogramming had not yet been done on actual altimeter waveform data.

The two following sections of this report describe first the line-fitting process, then the time realignment process. The computer programs developed or modified for these tasks are described briefly in the appropriate sections but the bulk of the computer program documentation is relegated to Appendix sections at the end of this report - source decks for all of these programs have been separately provided to Wallops Flight Center as these procedures have developed through the contract period. In some cases (notably the time-realignment main programs) further program modifications have been made after the programs have been installed at the Wallops computer; thus the programs now at Wallops vary somewhat from those in the Appendix sections of this report.

SECTION II. DETERMINATION OF S&H OFFSETS AND LINESHAPE PARAMETERS

The overall waveform processing diagram, Figure 1 of Section I, shows two different uses for the line fitting procedure to be described in this section; first, the submode S&H averages are used to find the individual S&H dc offsets, and later the final time-realigned averages are to be used to find the final set of descriptive parameters to characterize the waveform. The same general-purpose chi-squared-minimizing computer program, X2MIN, is used for both of these purposes. The program X2MIN is described in some detail in Appendix A, and Appendix B provides the function subroutine FX which is used with X2MIN to fit S-193 S&H data to the theoretical mean return waveform described later in this section.

First we will describe the S&H offset problem in general, and point out the lack of a good means to determine the needed S&H offsets from any of the calibration modes. Then after describing in general the method to be used, we will discuss the particular functional form which represents the theoretical mean return waveform. Following the pages on waveform fitting to the 4-parameter function plus S&H offsets, we present examples of waveforms and offsets determined in this way. Finally a means of examining the statistics at one or several points on the mean waveform is described.

Because the video output in the S-193 radar altimeter is ac-coupled to the S&H gates, the average of a set of readings from a S&H gate which is sampling a noise-only region of the mean waveform (i.e., a point prior to the leading edge of the waveform) should be zero. Because the S&H gates are less than ideal however, the average is non-zero in practice for a S&H gate with a noise-only input. Each S&H gate will differ from others, with the result that each S&H gate will have its own non-zero average output for noise-only input. Furthermore, the offset for each S&H gate may be different for changes in any of the following: IF bandwidth, receiver temperature, and S&H gate width. It is important to note that the dc-offset for each gate must be subtracted from all values obtained by that particular gate before any time-realignment and waveform averaging can be done. [As we will see in the section on time-realignment, any particular specified point, or "bin", on the mean return waveform receives contributions from not one but several different S&H gates as they are carried past the point of interest by the tracker jitter.]

In this work, we assume that the offset characterizing each of the 8 S&H gates is a simple additive effect and that once the offset is determined, all the S&H data may be corrected by simply subtracting the appropriate offset from each S&H data point. This necessary assumption could very well be incorrect; for example, the offsets might depend somewhat upon the level of the signal in, or might exhibit some sort of hysteresis-like effects, or could have other, more complicated behavior. Any case but that of the simple additive offset becomes virtually impossible to analyze from the limited amount of data available.

To determine the S&H offsets under the simple additive assumption, we should look at the output for a S&H array position in which all eight gates are sampling noise. Immediately we are in trouble for submode 1 and 2 of Mode 5, as there is no subsubmode satisfying this noise-only requirement.

The next possibility would be to use the calibration data step (CDS) data. Mode 5, submode 3 provide the CDS data for the short pulse mode, Mode 5, submode 2, and Figure 2 shows the results given by this CDS submode for SL-2 Pass 9. The solid line in the curve is a Gaussian which is best-fitted to the CDS data points. Here again, there is no subsubmode in the CDS data in which all eight S&H gates are in a non-signal region. Another difficulty with all the CDS data in our experience is that it is much less variable, much more reproducible pass-to-pass than is any of the actual data acquisition step (DAS) data in submode 2 of Mode 5. Figure 2 does, however, suggest an approach which might be used if we were to use CDS data to determine offsets; we could take the offsets as being the differences between the actual data points and a best-fitted theoretical curve. Figure 3 shows the CDS data of Figure 2 after subtracting the offsets determined in this manner, together with a new fitted Gaussian. The fit is somewhat better in Figure 3 than in Figure 2 .

This reasoning can be extended to DAS data as well. All our expectations are that the plateau region of the mean return waveform should be a smoothly varying function; consequently variations of S&H data about a smoothly varying fitted function might be attributable to S&H offset. The functional form to be fitted is based on earlier work at Applied Science Associates, and

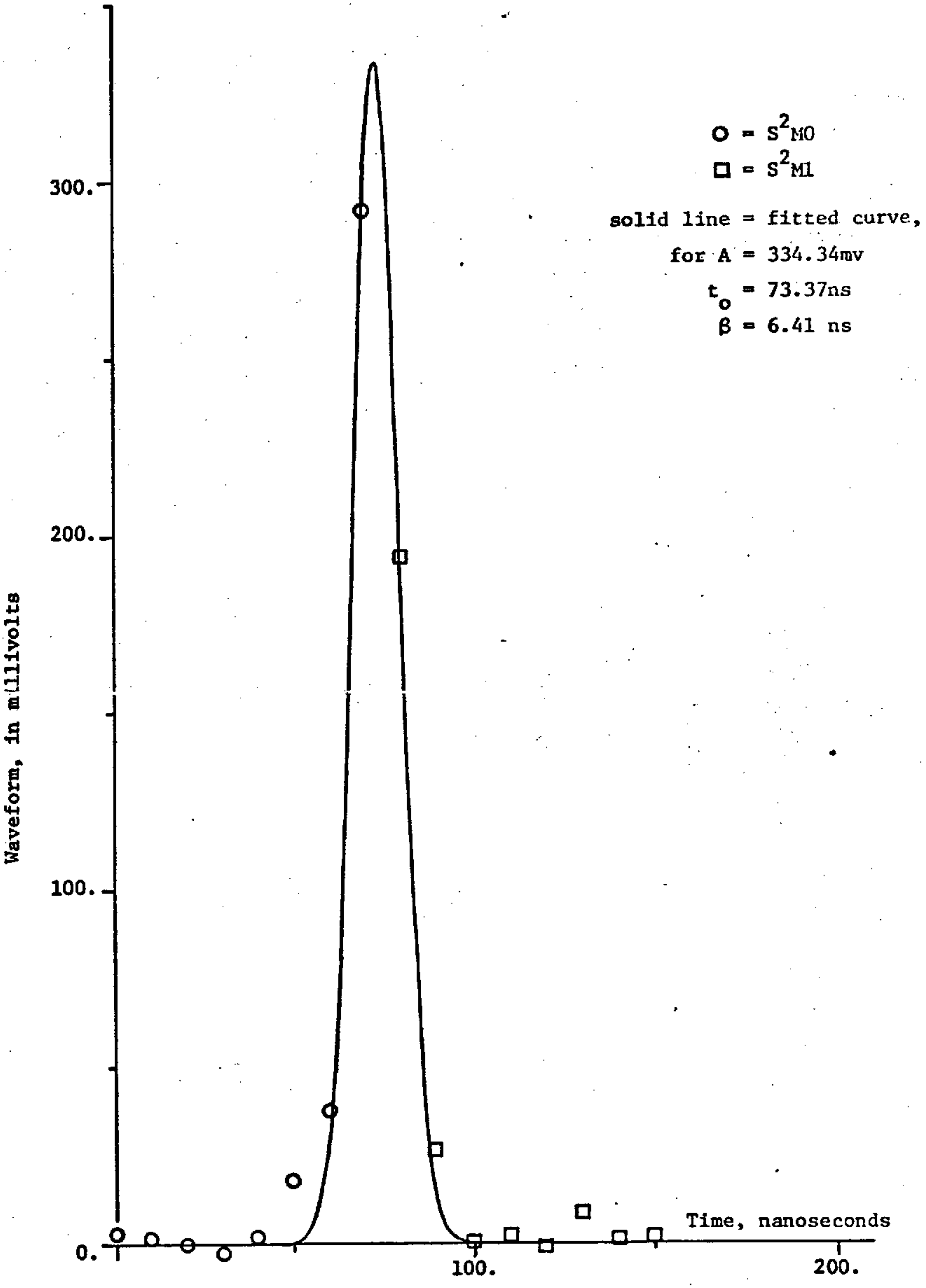


Figure 2. Results from SL-2, Pass 9, Mode 5, Submode 3 With Fitted Gaussian

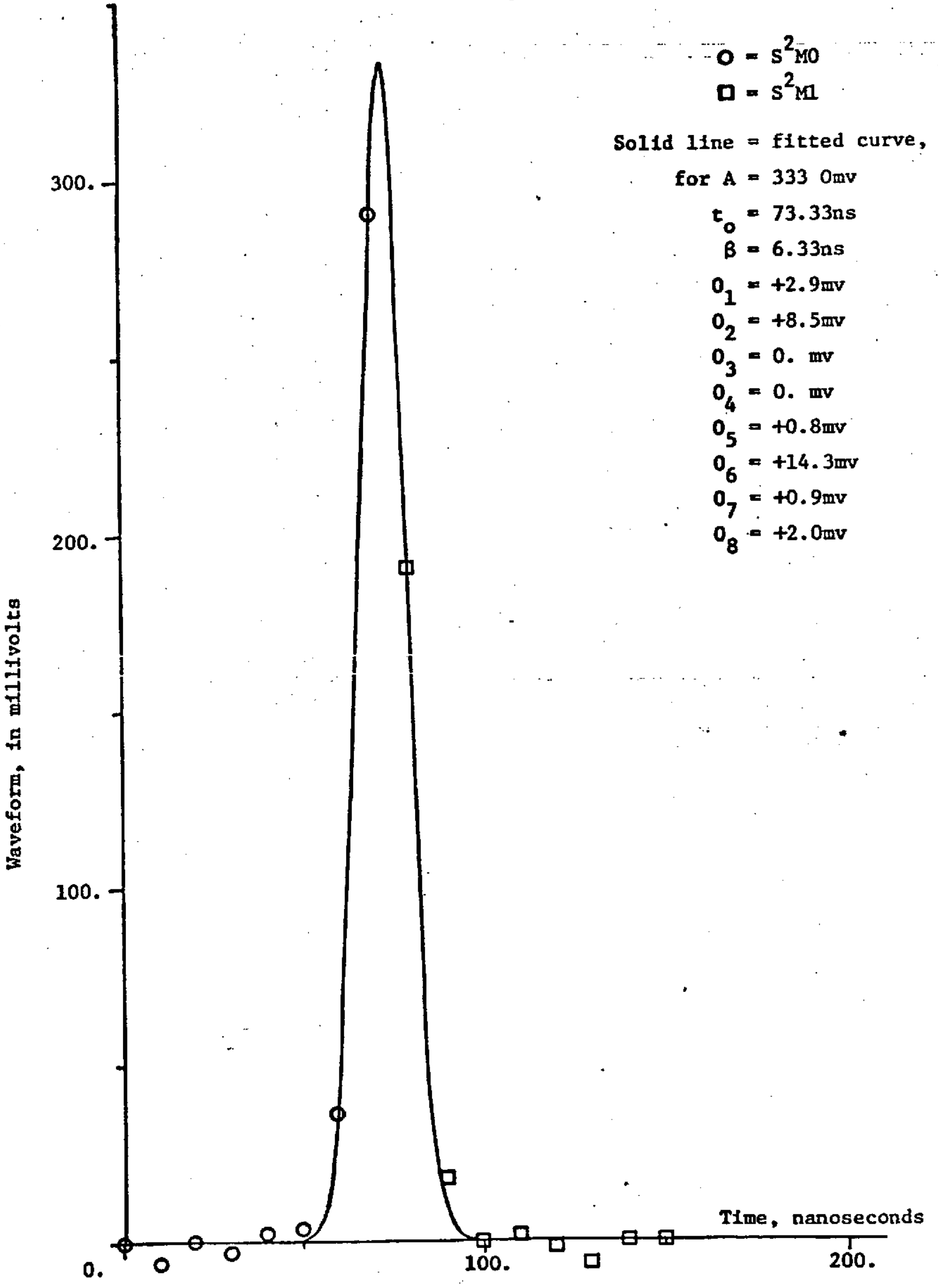


Figure 3. Results from SL-2, Pass 9, Mode 5, Submode 3
After Subtraction of Offsets, With Fitted Gaussian

will be described in the next paragraphs, but there remains the question of where to derive the offsets. Should we take simply subsubmode 3, the set of S&H positions starting latest in time on the M5, SM2 waveform? Or use some weighted combination of offsets derived from several subsubmodes? The answer to which we have evolved is that the eight S&H offsets should be treated as free parameters to be varied together with the four lineshape-related parameters (A, t_o, β, and ξ in the following paragraphs) for a 12-parameter fit to the 32 data points from the 4 subsubmodes of M5, SM2. This might seem to overlook the entire subject of time-realignment, since we are simply averaging all S&H data over the entire submode but since the primary effect of tracker jitter on the observed mean waveform should be to increase the leading edge risetime (β in the 12-parameter fit), this should not be a problem in offset determination.

The 32 waveform averages y_j and their corresponding times t_j are assigned an index j=1,32 according to the following rules: j=1,2,...,8 is assigned to the result from S&H gate 1,2,...,8 in sub-submode 0; j=9,10,...,16 is assigned to the sub-submode 1 result from S&H gate 1,2,...,8; and this sequential numbering is continued for sub-submodes 2 and 3. Thus arranged, data point j,(j=1,32), will have come from S&H gate k,(k=1,8), where the relationship between j and k is k=1 + j(mod 8). The theoretical waveform will be a function of four parameters (A,t_o,β,ξ) to be described in the following paragraph, and of the eight S&H offsets 0_k,(k=1,8).

The four-parameter waveshape function of time t, f(t), is derived from an earlier report [L. S. Miller and G. S. Brown, "Engineering Studies Related to the GEOS-C Radar Altimeter, Final Report for Task D, "NASA Contract NAS6-2307, Applied Science Associates, Inc., Apex, N. C., May 1974, pg. 70], and is given by

$$f(t) = AP(\frac{t - t_o}{\beta}) \exp [\frac{-\delta^2}{\gamma}(t - t_o)\cos|2\xi|] \, I_o(\frac{2\delta}{\gamma} \sqrt{t-t_o} \sin|2\xi|) \quad ,$$

for $t > t_o$, and by

$$f(t) = AP(\frac{t - t_o}{\beta}) \quad ,$$

for $t \leq t_o$. In these expressions P(z) is the probability function, which is related to the error function erf(z) by

$$P(z) = 1/2[1 + \mathrm{erf}(\frac{z}{\sqrt{2}})] \ ,$$

and $I_o(z)$ is the Bessel function of the second kind, order zero. The other symbols have the following meaning:

t = time, in ns
A = a general amplitude factor,
t_o = a reference time origin in nanoseconds
ξ = antenna angle off-nadir,
γ = antenna beamwidth-related constant
= .00055 $(ns)^{-1/2}$ for the S-193 radar altimeter(for SL-2 and SL-3; the beamwidth apparently changed during SL-4),
$\delta = 2\sqrt{c/h}$
$= 1.656\mathrm{x}10^{-3}(ns)^{-1/2}$ for S-193
β = a pulsewidth-risetime parameter

The parameter β is the product of two parameters σ_p and α of the earlier report, where

$$\alpha = \sqrt{(\frac{2\sigma_s}{c\sigma_p})^2 + 1}$$

$$\sigma_p = \sqrt{.181} \ \ PW$$

PW = the 3dB pulsewidth of the S-193 altimeter,
c = the speed of light
= .3 m/ns
and σ_s = rms ocean surface roughness.

Thus for a flat sea, β will be equal to σ_p = 7.66 ns (for a 18 ns pulsewidth as in submode 2 of Mode 5), and an increase in sea surface rms roughness will increase β.

The jth data point, for the data arranged by submode in the manner already described, is (y_j,t_j) where we let y_j be the averaged waveform sample. The theoretical function to be fitted will be

$$f_o(t_j) = f(t_j) + 0_k \ .$$

Here the subscript o denotes "observed", $f(t_j)$ is the four-parameter function of A, t_o, β, ξ given above, and 0_k is the offset of S&H date k with k and j related as before. We use a chi-squared minimizing computer routine X2MIN to best-fit f_o to the (y_j, x_j). This computer subroutine is described in Appendix A ; X2MIN requires a function subroutine FX which evaluates the function f_o at each of the input times t_j. FX also evaluates the derivatives of f_o with respect to each of the fitting parameters; these derivatives are evaluated at each of the 32 input times t_j.

We therefore need the derivatives of f_o with respect to A, t_o, β, ξ, and to 0_k, k = 1,8 at each point t_j. To handle the offsets, first we use the relationship between k and j to write

$$\frac{df_o}{d0_k}(t_j) = 1 \quad , \quad \text{if } k = 1 + j \pmod 8$$
$$= 0 \text{ otherwise}$$

Then since $f_o(t_j)$ and $f(t_j)$ are related by a simple additive constant (0_k),

$$\frac{df_o}{dp}(t_j) = \frac{df}{dp}(t_j)$$

where p is any one of the parameters A, t_o, β, ξ. Doing the differentiation, and summarizing here,

$$f_o(t_j) = AP\left(\frac{t_j - t_o}{\beta}\right) + 0_k \quad , \quad t_j \leq t_o$$

$$= AP\left(\frac{t_j - t_o}{\beta}\right) \exp\left[\frac{-\delta^2}{\gamma}(t_j - t_o)\cos|2\xi|\right] I_o\left(\frac{2\delta}{\gamma}\sqrt{t_j - t_o}\,\sin|2\xi|\right) + 0_k \quad , \quad t_j > t_o$$

$$\frac{df_o}{dA}(t_j) = P\left(\frac{t_j - t_o}{\beta}\right) \quad , \quad t_j \leq t_o$$

$$= P\left(\frac{t_j - t_o}{\beta}\right) \exp\left[\frac{-\delta^2}{\gamma}(t_j - t_o)\cos|2\xi|\right] I_o(\ldots) \quad , \quad t_j > t_o$$

$$\frac{df_o}{dt_o}(t_j) = \frac{-A}{\sqrt{2\pi}\,\beta} \exp\left[-\frac{(t_j - t_o)^2}{2\beta^2}\right] \quad , \quad t_j \leq t_o$$

$$= A \exp[\frac{-\delta^2}{\gamma} \cos|2\xi|(t_j - t_o)] \left\{ P(\frac{t_j - t_o}{\beta}) \frac{\delta}{\gamma} [\delta \cos|2\xi| I_o(\ldots)\right.$$

$$\left. - \frac{\sin|2\xi|}{\sqrt{t_j - t_o}} I_1(\ldots)] - \frac{I_o(\ldots)}{\sqrt{2\pi}\,\beta} \exp[\frac{-(t_j - t_o)^2}{2\beta^2}] \right\}, \quad t_j > t_o$$

$$\frac{df_o}{d\beta}(t_j) = \frac{-A(t_j - t_o)}{\sqrt{2\pi}\,\beta^2} \exp[\frac{-(t_j - t_o)^2}{2\beta^2}], \quad t_j \leq t_o$$

$$= \frac{-A(t_j - t_o)}{\sqrt{2\pi}\,\beta^2} \exp[-\frac{(t_j - t_o)^2}{2\beta^2}] \exp[\frac{-\delta^2}{\gamma}(t_j - t_o)\cos|2\xi|]\, I_o(\ldots)$$

and $\frac{df_o}{d\xi}(t_j) = 0 \quad , \quad t_j \leq t_o$

$$= AP(\frac{t_j - t_o}{\beta}) \frac{2\delta}{\gamma} \sqrt{t_j - t_o} \exp[\frac{-\delta^2}{\gamma}(t_j - t_o)\cos|2\xi|] \operatorname{sgn}(\xi) \left\{ 2\cos|2\xi| I_1(\ldots) \right.$$

$$\left. + \delta\sqrt{t_j - t_o} \sin|2\xi| I_o(\ldots) \right\} \quad , \quad t_j > t_o$$

In the above, sgn(ξ) is the sign function,

$$\operatorname{sgn}(\xi) = +1 \quad , \quad \xi \geq 0$$
$$\operatorname{sgn}(\xi) = -1 \qquad \xi < 0 \quad ,$$

and $I_o(\ldots)$ and $I_1(\ldots)$ have the same argument $(\frac{2\delta}{\gamma} \sqrt{t_j - t_o} \sin|2\xi|)$ in all the above expressions.

The function subroutine FX is written for the 12 parameters to be stored in an array A(I) with the order of parameter assignment as follows:

A(1) ← A, amplitude,
A(2) ← t_o, time origin,
A(3) ← β, rise time
A(4) ← ξ , antenna off-nadir angle,
A(5) ← 0_1 , offset of S&H #1
A(6) ← 0_2 offset of S&H #2
⋮ ⋮
A(12) ← 0_8 , offset of S&H #8

Appendix B provides a source listing for FX: a flow diagram is also provided to simplify program debugging or modification. Source listings are also provided for subroutines BESI to calculate $I_o(.)$ and $I_1(.)$, and for NDTR to calculate P(.). FX includes the option of subtraction of the individual gate offset from the input waveform data after finding those offsets. This option is enabled by INFLAG(4) = 1, and assumes the use of the REPEAT and the continuation flag on C (refer to Appendix A), and it is assumed that the following data deck organization will be employed.

Input Deck 1

1 Comment card
1 Size card - NP=32,NX=1,NA=4,NC=2,default CHILIM and MAXITR
1 Data Labels - 'TIME', 'WAVEFORM', 'SIGMA'
32 Data Point cards - (32 input data points)
4 Variable Param. cards(A)-initial neighborhood guesses for A(I),I=1,4
2 Const. param(C) - δ,γ

2nd Input Deck

1 Comment
1 Size card - NP=32,NX=1,NA=12,NC=2, default CHILIM and MAXITR
1 Data Label Card - REPE in columns 1-4
12 Variable Param (A) - continue Flags on A(1),...,A(4),zero as first guess for A(5),...,A(12)
2 Const. Param.(C) - contunue flags

3rd Input Deck

1 Comment
1 Size card-NP=32,NX=1,NA=4,NC=10, default CHILIM and MAXITR
1 Data Label - REPE in columns 1-4
4 Variable Param.(A) - continue flags
10 Const. Param.(C) - continue flags on all zeros

With this organization of the data deck for any one set of input averaged waveform points, there will be three successive sets of problem output and these will provide the following:

Output # 1

Prints out the results of the 4-parameter functional fit (A,t_o,β,ξ) under the assumption of zero offsets. Also prints out input data values of fitted function and the deviations for each input data point.

Output # 2	Prints out 12-parameter fit results, treating the 8 S&H offsets as parameters to be determined. Also prints out input data, fitted function, and deviations.
Output # 3	Prints out 4-parameter fit after the 8 offsets determined above have been subtracted from the 32 input data. The 8 offsets are printed out as C(3),C(4),...,C(10). Also prints out the offset-corrected input data, fitted function, and deviations. Use of the line-printer-plotting feature is also useful in this 3rd output to provide a quick-look check of the results.

The entire subject of best-fitting experimental data to an expected functional form is very complicated and often one uses least-squares or related procedures not because they are optimum but because they are readily available. Our own use of the function-fitting routine X2MIN(as just described) is in this spirit; if the first results of this approach were encouraging, we intended later work to verify its suitability.

The initial results of applying the linefitting offset estimation technique appeared very promising. Figure 4 presents the results for SL-2, Pass 9, Mode 5, Submode 2, and it is readily seen that the 12-parameter fit is better than the 4-parameter fit assuming zero offsets. Another example is provided by Figure 5 showing the results for SL-3, Pass 7/18, Mode 5, Submode 2. Another case analyzed was SL-2, Pass 6, M5, SM2, and results from these three passes just mentioned are entered in the first three lines of Table 2. This table also indicates the averages of the offsets determined from these three passes and we were gratified that the pattern of the offsets was very similar in these cases. These three cases happened to be among the earliest ones we used, and the approach seemed very promising.

However, there were difficulties soon encountered. Specifically, the case of SL-3, Pass 28/39 stopped our apparent progress on the offset-and-linefitting approaches. There was high interest in this particular pass as it appeared to be the first pass in the Skylab missions in which the pulse compression submode seemed to be operating and the obvious question was how the pulse compression, SM1, and the short pulse, SM2, submodes of Mode 5 compared.

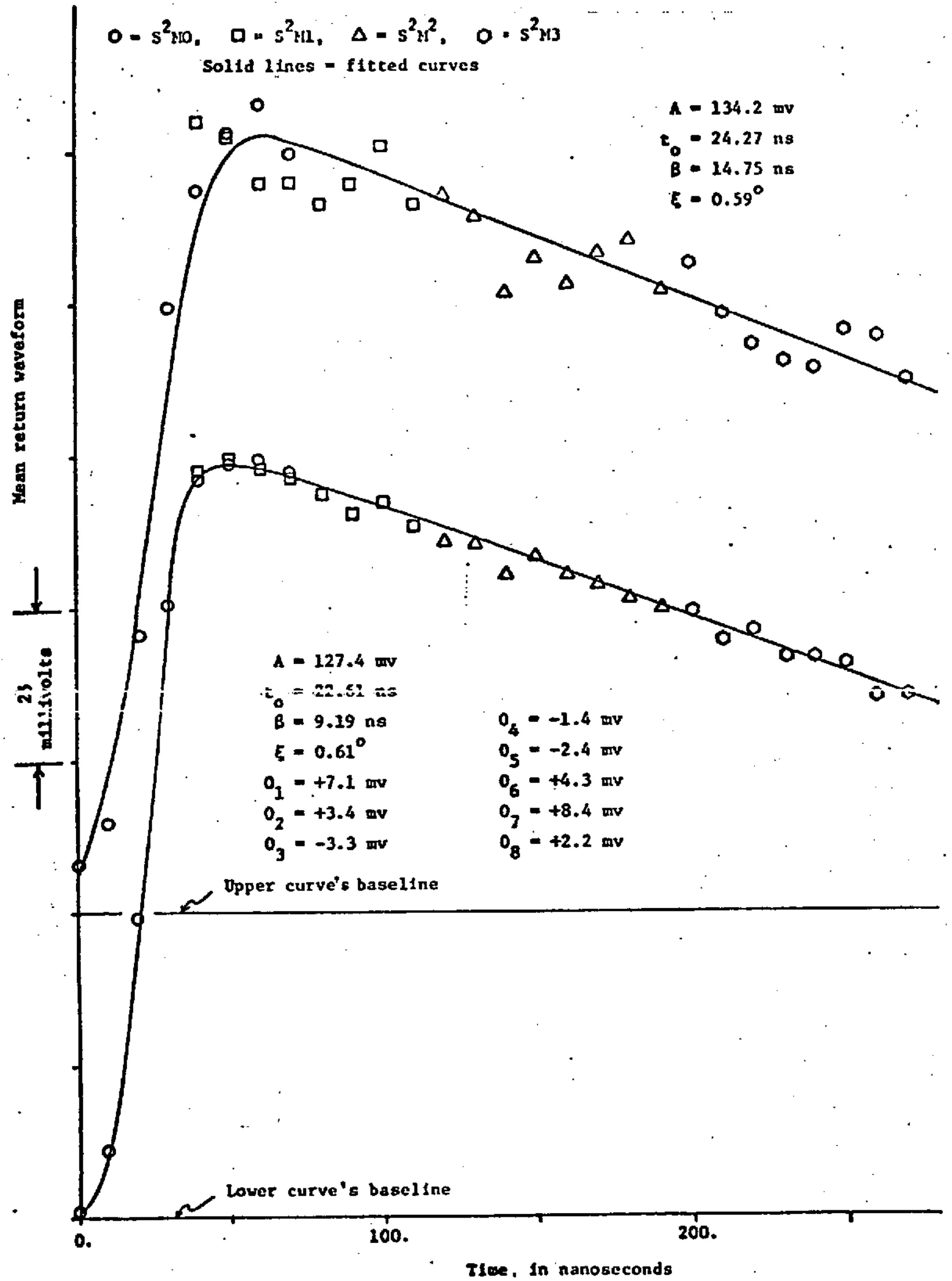


Figure 4. Linefitting Results For SL-2, Pass 9, Mode 5, Submode 2. Upper Curve is 4-Parameter Fit With Zero Offsets, Lower Curve is 12-Parameter Fit After Subtracting Offsets.

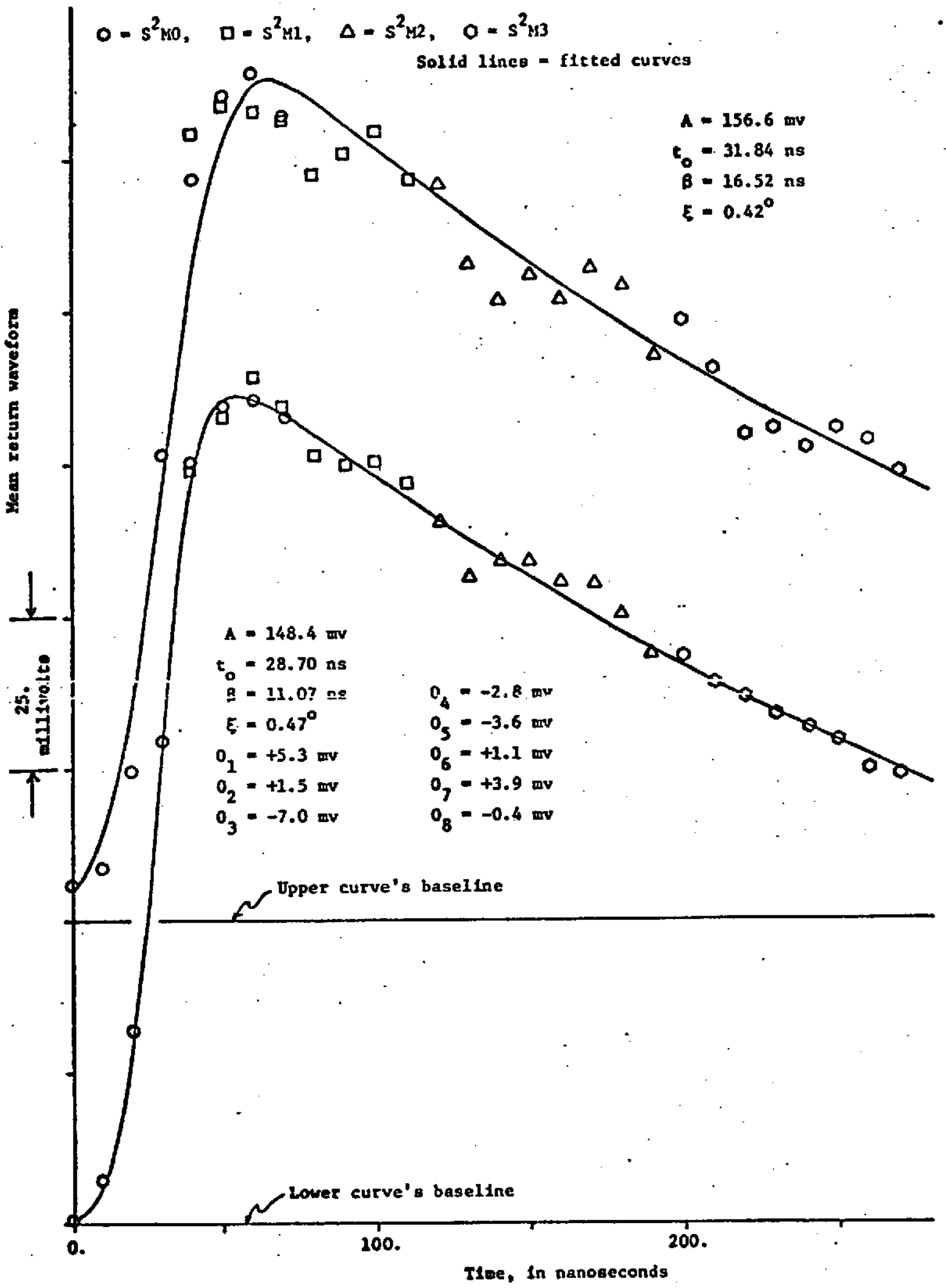


Figure 5. Linefitting Results For SL-3, Pass 7/18, Mode 5, Submode 2. Upper Curve Is 4-Parameter Fit With Zero Offsets, Lower Curve is 12-Parameter Fit After Subtracting Offsets.

(Pass)Mission	Mode(s)	SM	# Frames	O_1 δ_1	O_2 δ_2	O_3 δ_3	O_4 δ_4	O_5 δ_5	O_6 δ_6	O_7 δ_7	O_8 δ_8	Avg.
(SL-2) 6	5	2	45	+10.68 +8.12	+3.87 +1.31	-3.88 -6.44	-0.08 -2.64	-2.25 -4.81	+3.79 +1.23	+8.25 +5.69	+0.11 -2.45	+2.56
(SL-2) 9	5	2	43	+7.12 +4.84	+3.39 +1.11	-3.27 -5.55	-1.43 -3.71	-2.45 -4.73	+4.28 +2.00	+8.40 +6.12	+2.18 -0.10	+2.28
(SL-3)7/18	5	2	47	+5.30 +5.54	+1.51 +1.75	-6.96 -6.72	-2.84 -2.60	-3.58 -3.34	+1.15 +1.39	+3.89 +4.13	-0.40 -0.16	-0.24
(SL-3)28/39	5	1	95	-26.20 +2.48	-29.11 -0.43	-30.93 -2.24	-29.02 -0.34	-32.55 -3.86	-27.39 +1.30	-24.94 +3.74	-29.34 -0.66	-28.68
(SL-3)28/39	5	2	47	+37.18 +4.75	+30.89 -1.54	+28.78 -3.65	+32.05 -0.38	+28.38 -4.05	+34.37 +1.94	+36.36 +3.93	+31.44 -0.99	+32.43
(SL-4)79/24	5(2)	1	94	+48.54 -1.99	+50.93 +0.38	+34.74 -15.79	+51.05 +0.52	+50.48 -0.05	+59.78 +9.25	+64.92 +14.39	+43.80 -6.73	+50.53
Average of (SL-2) 6, (SL-2) 9, and (SL-3) 7/18			O_i δ_i	+7.70 +6.17	+2.92 +1.33	-4.70 -6.24	-1.45 -2.98	-2.76 -4.29	+3.07 +1.54	+6.85 +5.31	+0.63 -0.90	+1.53 0.

Table 2. Results of Linefitting. [Offsets O_i in Millivolts. δ_i Defined as $\delta_i = O_i - O_{avg}$]

(Pass)Mission	Mode(n)	SM	# Frames	$\Sigma\chi^2$	A	t_o	β	ξ
(SL-2) 6	5	2	45	15.9	98.9	12.40	8.54	0.761
(SL-2) 9	5	2	43	11.1	127.4	+22.61	9.19	0.609
(SL-3)7/18	5	2	47	29.26	148.4	+28.70	11.07	0.468
(SL-3)28/39	5	1	95	18.48	95.56	-11.61	15.07	0.871
(SL-3)28/39	5	2	47	24.96	43.70	+1.94	5.49	0.929
(SL-4)79/24	5(2)	1	94	25.34	65.0	+25.0	15.27	0.225

Table 2 (continued). Results of Linefitting. [A in Millivolts, t_o and β in Nanoseconds, ξ in Degrees]

Figure 6 presents the results of the linefitting with and without offset fitting and subtraction for SL-3, Pass 28/39, Mode 5, Submode 2, and the set of estimated offsets is also listed in Table 2. Notice that the average of these offsets is more than 10 times as large as in the first three cases discussed above. Notice also the considerable difference in risetimes for the fits with and without offset corrections for M5, SM2 of Pass 28/39.

Proceeding to the pulse compression and applying the same operations to the data of Submode 1 of Pass 28/39, Mode 5, the results of Figure 7 are obtained, with the offsets again listed in Table 2 . Here the difficulty is even more extreme; the fitting program prefers large <u>negative</u> offsets!

The problem here seems to be that there simply is not a sufficient number of data points in the earlier portions of the mean return waveform's leading edge, so that the fitting program cannot distinguish between the variables β and t_o (and also offsets). Figure 8 may be related to this point; it shows the various waveform sensitivities to changes in A, t_o, β, and ξ as calculated for the particular values of these as determined for SL-3, Pass 7/18, Mode 5, SM2 (see Figure 5). The function f_o itself will have the same shape as the derivative (df_o/dA) shown in Figure 8 , and it is apparent that the (df_o/dt_o) and $df_o/d\beta$) curves have somewhat the same behavior in the upper half of the "ramp" portion of f_o. In the lower half of the ramp these two derivatives are clearly distinguishable in their effect as they have opposite signs. We think that the major difference between the results for Mode 5, SM2, for SL-3, Pass 7/18 and Pass 28/39 is that the former pass had adequate S&H sampling in the earlier half of the ramp and the latter pass did not. We think that this may have been due to a greater pointing angle which led (because of the change in mean return waveshape as a result of increasing ξ) to a change in tracker bias point such that the tracker positioned the S&H gates too late in time, but we can't prove this conclusively from the data of Pass 28/39.

One other point that might be remarked on Figure 8 is that the sensitivity to ξ increases as one moves later in time. This is simply the familar result that the later plateau regions are more sensitive to pointing angle than is the ramp region where the pointing angle is less than the half-beamwidth of the antenna (when the pointing angle is greater than the half-beamwidth, the mean waveform is grossly distorted and no longer characterizable

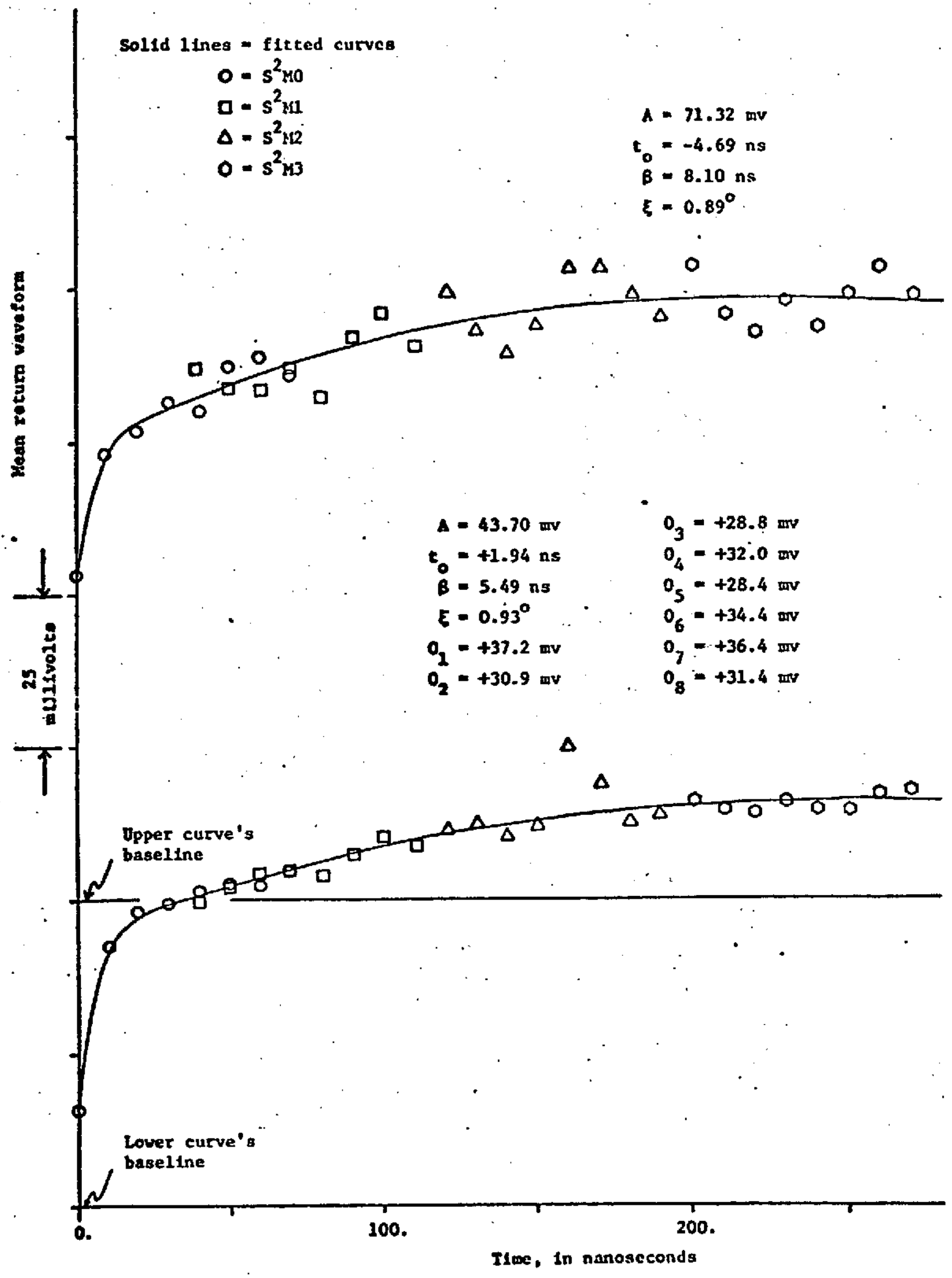


Figure 6. Linefitting Results For SL-3, Pass 28/39, Mode 5, Submode 2. Upper Curve Is 4-Parameter Fit With Zero Offsets, Lower Curve Is 12-Parameter Fit After Subtracting Offsets.

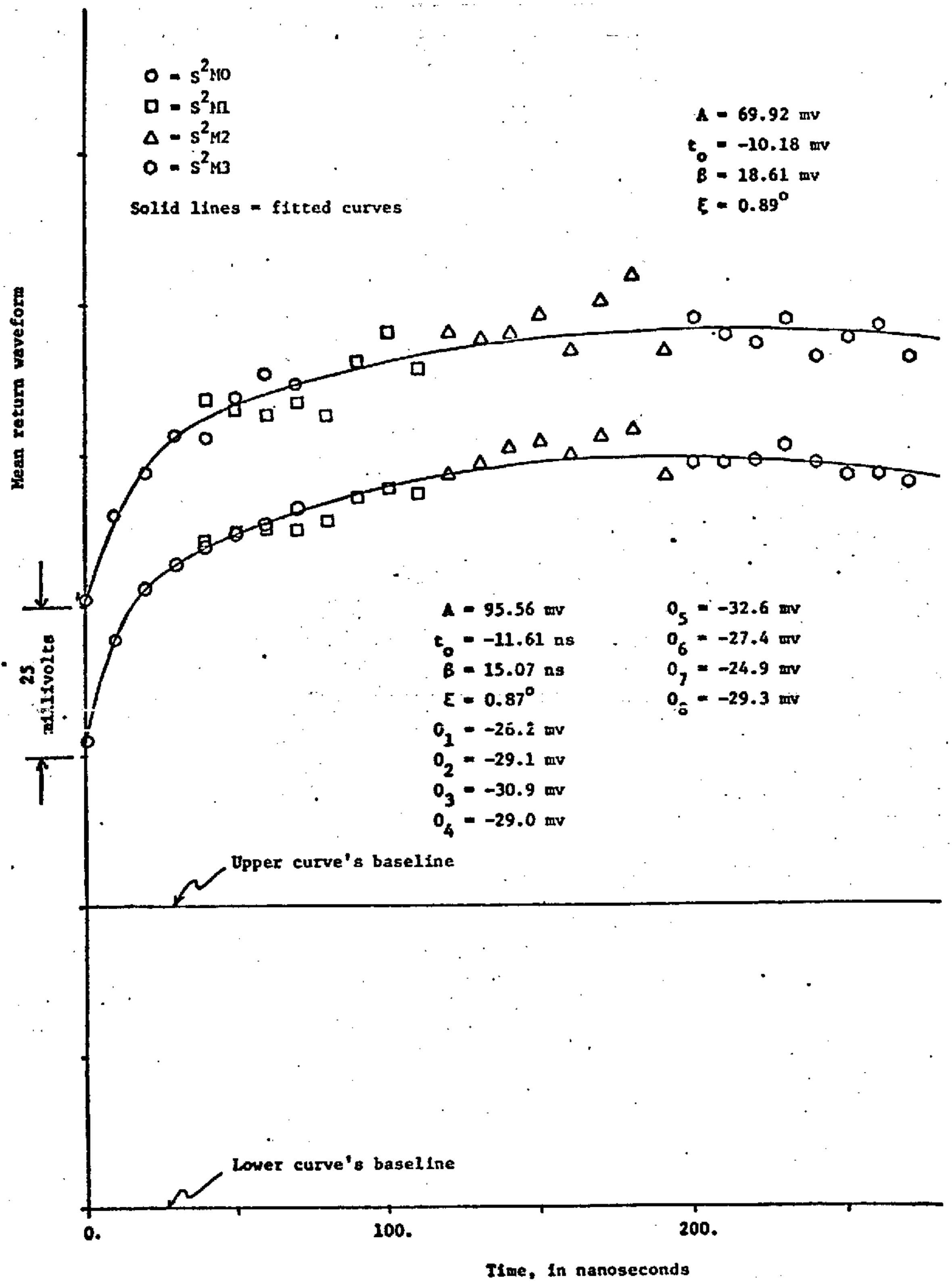


Figure 7. Linefitting Results For SL-3, Pass 28/39, Mode 5, Submode 1. Upper Curve Is 4-Parameter Fit With Zero Offsets, Lower Curve Is 12-Parameter Fit After Subtracting Offsets.

Positions of S&H Gates, M5, SM2.

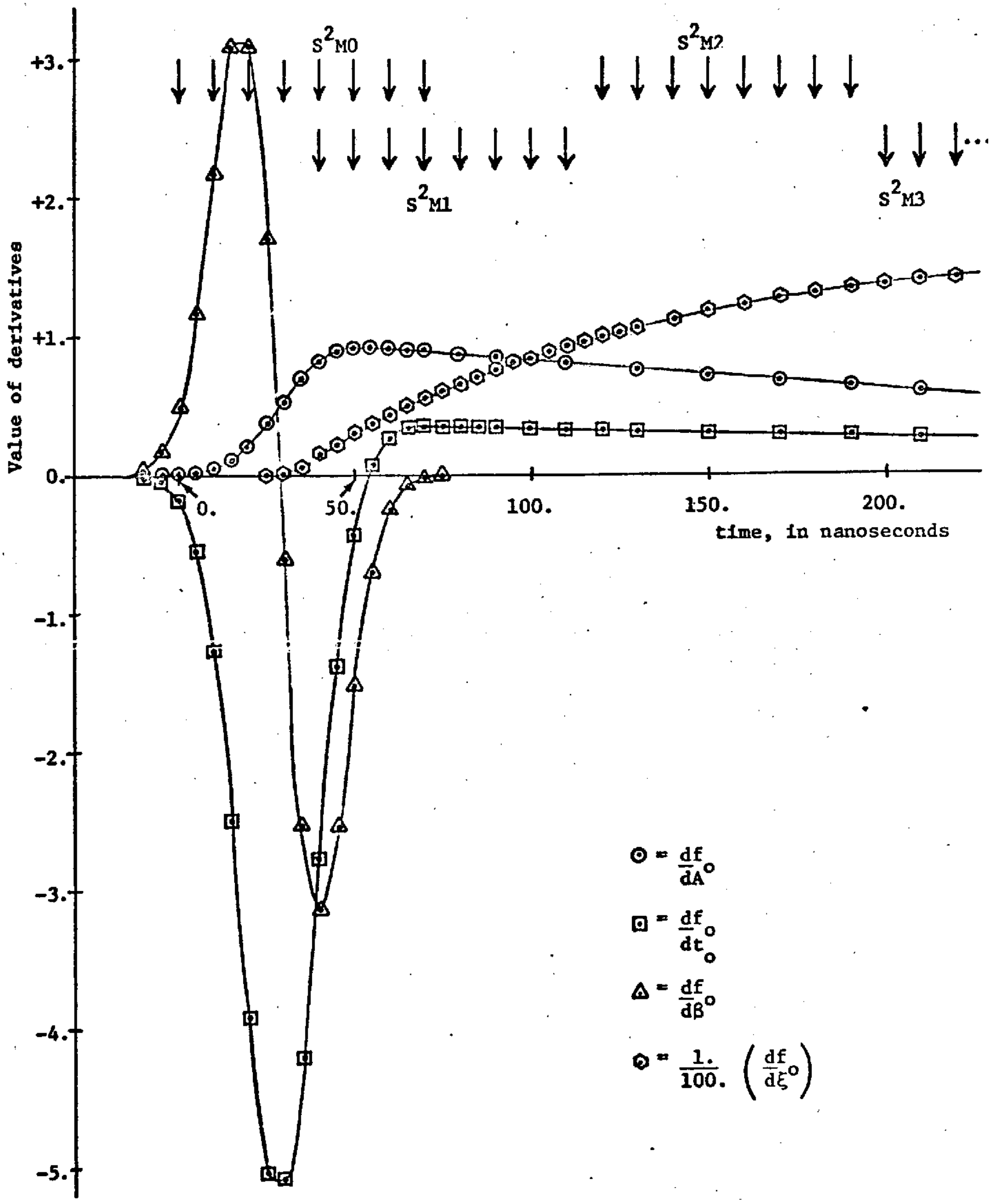


Figure 8 . Theoretical waveform derivatives calculated for A=148.4mv, t_o=28.7ns, β=11.07ns, and ξ=0.47 degrees, as found by 12-parameter fit to Pass 7/18 of SL-3, Mode 5, Submode 2. S&H gate positions in each subsubmode are indicated by arrows.

as having a ramp and a plateau). Thus for determining the pointing angle from the S&H waveform data (as needed in σ^o work for example), the 100ns pulsewidth submode would seem preferable to the 20ns submode.

Figure 9 shows again the S&H offset estimation via linefitting for SL-4, Pass 79/24, M5, SM1. Here a different problem is built into the again unacceptable results. It is now thought that the antenna characteristics changed between SL-3 and SL-4 and without adequate knowledge of the new antenna patterns, the fitting routine's constant γ is not adequately known. [Recall that γ is a beamwidth-related factor determined as specified by the earlier Applied Science Associates Report on Task D as cited in the discussion of f(t) several pages earlier.] In addition to the uncertainty in γ, the results of Figure 9, with offsets as listed in Table 2 , have very large positive offsets. These offsets show an apparently quite different distribution about the mean offset than all other entries in Table 2 [see specifically the δ_1 in that Table].

Finally, there is nothing in the preceeding discussion which restricts our efforts to Submodes 1 or 2 of Mode 5 except for the specific reference to 4 subsubmodes and to 32 data points, and the general linefitting procedure has also been applied to three different submode 0 results from different M5, SL-4 passes. These again are subject to the uncertainty in the beamwidth-related constant γ. The results from these passes are presented in Figures 10, 11, and 12, and here again the results are only partly consistent. There is very little more that can be said about the S&H offset problem at this time; a general approach has been described and the results are mixed. Some more variations on this should be carried out, and we again emphasize the view that the present report is only a statement of progress to date on an unfinished problem. The remaining pages in this section are addressed to a different waveform-related problem, that of the mean return waveform's statistical properties.

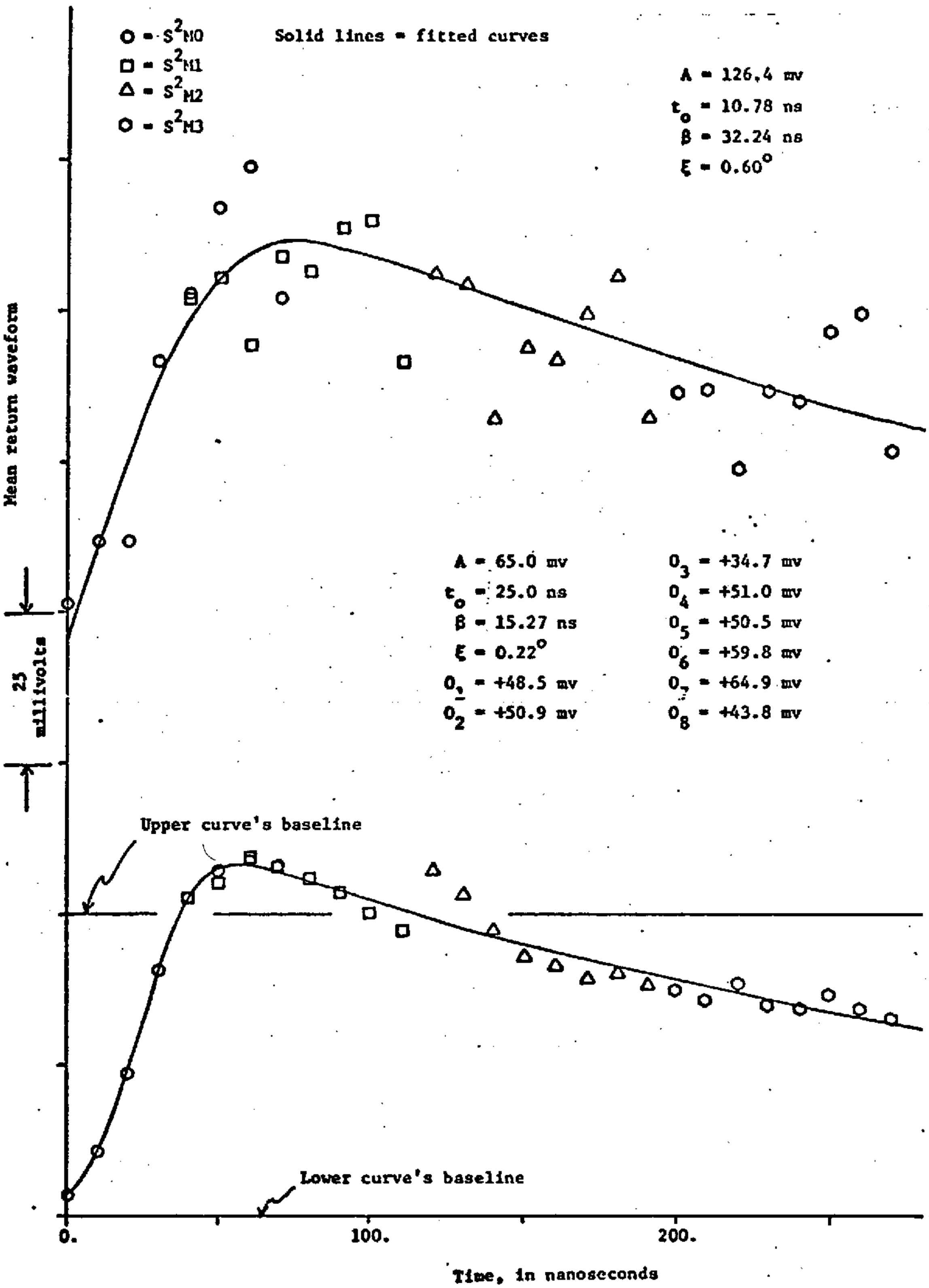


Figure 9. Linefitting Results For SL-4, Pass 79/24, Mode 5, Submode 1. Upper Curve Is 4-Parameter Fit With Zero Offsets, Lower Curve Is 12-Parameter Fit After Subtracting Offsets.

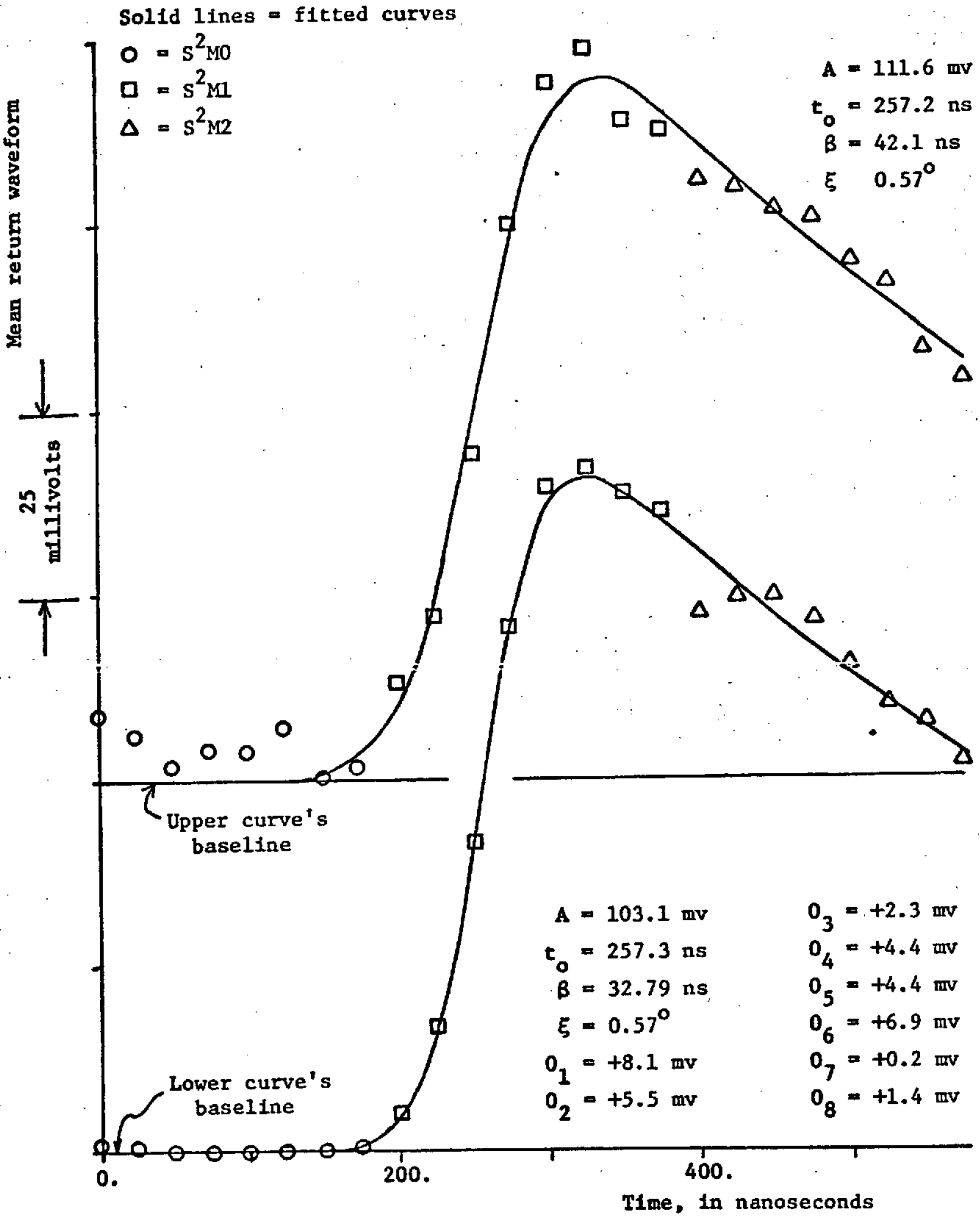


Figure 10. Linefitting Results for SL-4, Pass 78/23, Mode 5, Submode 0. Upper Curve Is 4-Parameter Fit With Zero Offsets, Lower Curve Is 12-Parameter Fit After Subtracting Offsets.

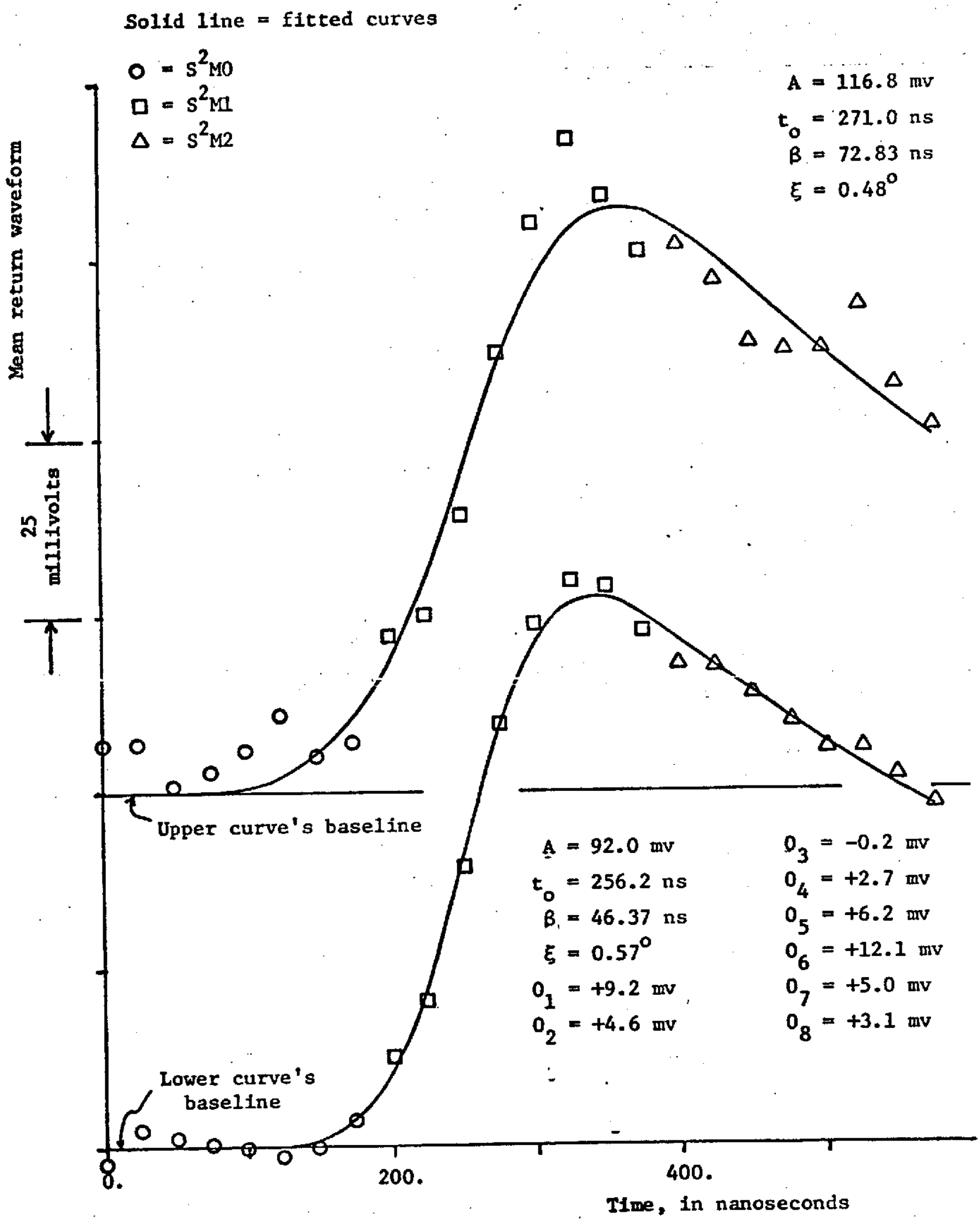


Figure 11. Linefitting Results For SL-4, Pass 83/29, Mode 5, Submode 0. Upper Curve Is 4-Parameter Fit With Zero Offsets, Lower Curve Is 12-Parameter Fit After Subtracting Offsets.

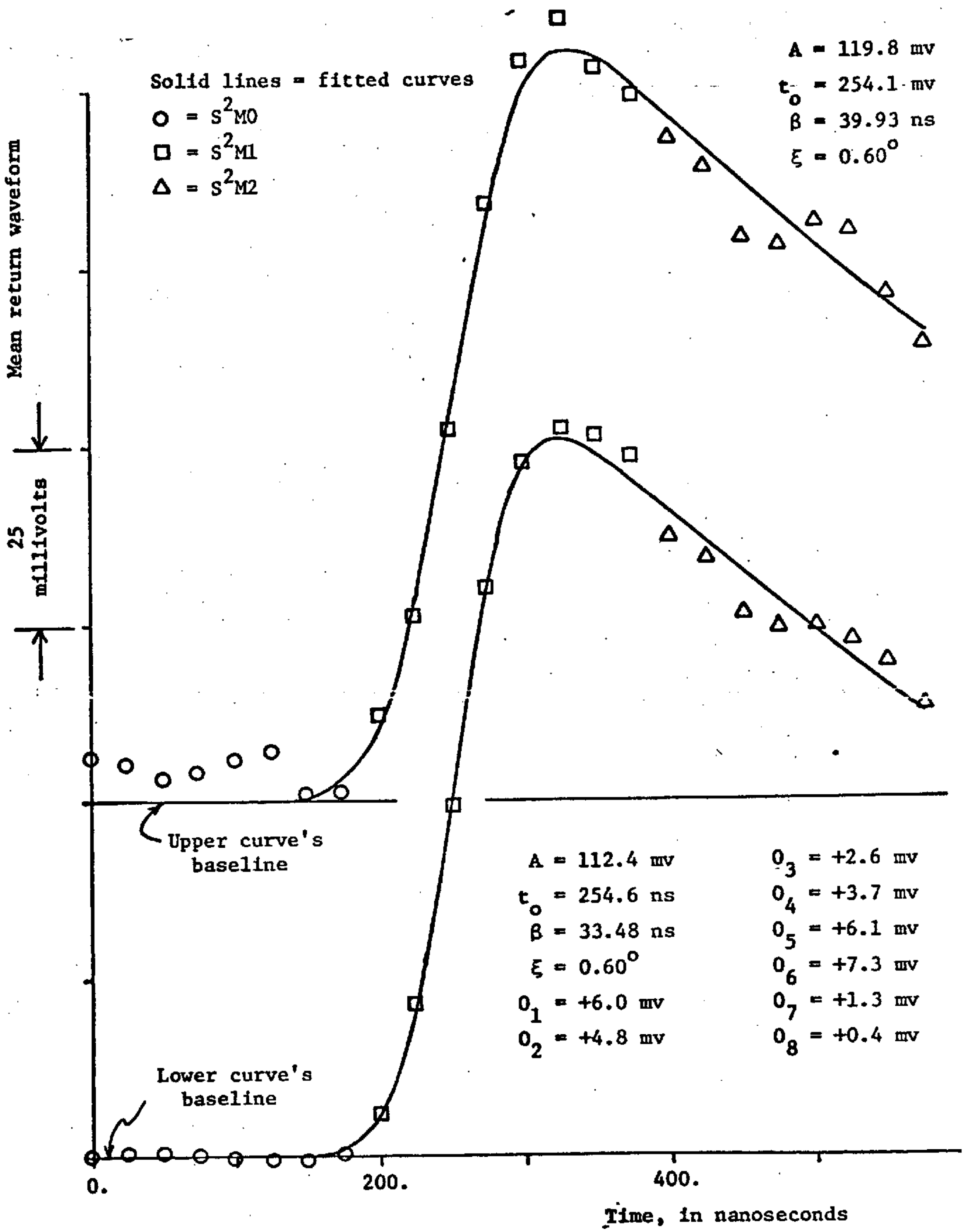


Figure 12. Linefitting Results For SL-4, Pass 86/32, Mode 5, Submode 0. Upper Curve Is 4-Parameter Fit With Zero Offsets, Lower Curve Is 12-Parameter Fit After Subtracting Offsets.

Most of the discussion thus far has been concerned with the concept of the mean return waveform, and the arithmetic averages of S&H gate reading have been used as estimates of this mean return waveform. It is useful to perform the sum of squares of gate readings at the same time standard in order to be able to estimate the standard deviation as well as the mean, and we practically always do provide standard deviation estimates in all our S&H-related programming.

But the mean and standard deviation are not sufficient to characterize an unknown distribution, and one needs additional information. In the following discussion we will ignore any tracker time-jitter effects by assuming that either such effects are negligible or have already been corrected for (by the time-realignment process for example) before the histogramming procedure is applied.

By selecting several points (at different times) on the mean waveform and producing a histogram at each point of the frequency of occurrence of each S&H voltage, we can get at each of these points a coarse estimate of the probability distribution function (pdf) of the S&H voltage. The "coarseness" of the pdf estimate is reduced as one increases the number of histogram bins but this cannot be done without limit because of the requirement of having an adequate number of samples in each bin. For this reason and because of practical limitations on how much computer space is to be tied up or how many output points have to be plotted, an upper limit of the order of 100 bins is reasonable for at least the first try at the problem of obtaining the waveform statistical properties via the histogram approach.

A general-purpose histogramming subroutine HIST has been written for this purpose.* An entry point HISTI is provided for initialization and another entry point HISTO prints out the results of histogramming. We will sketch here the use of HIST, HISTI, and HISTO for the individual inputs to

---

*At the time of this report's writing, this histogramming procedure had not yet been implemented. The subroutines HIST had been debugged and delivered to Wallops Flight Center and the first results were being obtained while this report was being prepared.

S&H gates 1,3,5 and 7 in each subsubmode of each submode of interest. For a first trial, the histogramming range should be from -0.050 volts to +0.450 volts, virtually the entire non-saturated range of the S-193 S&H gates.

We present immediately below the outline of the form a main program will have to carry this out, under the assumption that all S&H data in each given subsubmode will be used to achieve the maximum precision available for estimates of the statistical properties at S&H gates 1,3,5, and 7 on the mean return waveform. Appendix C provides a program description for subroutine HIST: it will be necessary to refer to that description of HIST to understand its use in the "main program" below.

"Main Program"

```
       other DIMENSION stuff
       DIMENSION JH1(100),JH2(100),JH3(100),JH4(100),SUMS1(2),SUMS2(2)
       DIMENSION SUMS3(2),SUMS4(2)
       DATA NB/100/,NSIGMA/1/,XBAR/.250/,SIGMA/.200/
       other initialization
 -- -> new submode
 |  ---- -> new subsubmode
 |  |          initialization within subsubmode
 |  |   CALL HISTI(XBAR,SIGMA,NSIG,NB,WB,XL,JH1,NHE1,SUMS1)
 |  |   CALL HISTI(XBAR,SIGMA,NSIG,NB,WB,XL,JH2,NHE2,SUMS2)
 |  |   CALL HISTI(XBAR,SIGMA,NSIG,NB,WB,XL,JH3,NHE3,SUMS3)
 |  |   CALL HISTI(XBAR,SIGMA,NSIG,NB,WB,XL,JH4,NHE4,SUMS4)
 |  |  --- ->loop to read S&H data, 104 sets per frame, for all S^2M frames
 |  |  | read S&H #1 into SH1
 |  |  | read S&H #3 into SH2
 |  |  | read S&H #5 into SH3
 |  |  | read S&H #7 into SH4
 |  |  | CALL HIST(SH1,WB,XL,JH1,NHE1,SUMS1)
 |  |  | CALL HIST(SH2,WB,XL,JH2,NHE2,SUMS2)
 |  |  | CALL HIST(SH3,WB,XL,JH3,NHE3,SUMS3)
 |  |  | CALL HIST(SH4,WB,XL,JH4,NHE4,SUMS )
 |  |  --- -end of loop for individual set of S&H values
 |  |   write heading for S&H #1
 |  |   CALL HISTO(NB,WB,JH ,NHE ,SUMS )
 |  |   write heading for S&H #2
 |  |   CALL HISTO(NB,WB,JH2,NHE2,SUMS2)
 |  |   write heading for S&H #3
 |  |   CALL HISTO(NB,WB,JH3,NHE3,SUMS3)
 |  |   write heading for S&H #4
 |  |   CALL HISTO(NB,WB,JH4,NHE4,SUMS4)
 |  ---- ---go to next S^2M
 |
 ---- --- --go to next SM

        finish out problem, etc.
```

SECTION III. WAVEFORM TIME-REALIGNMENT AND AVERAGING

The background for the results of this section is provided by an earlier Applied Science Associated report ["Final Report on Task A - Engineering Studies Related to the Skylab Program", prepared under Contract NAS6-2307, approx. July 1973]; that report is also the source for the weighting function used to estimate true altitude tracker position, given the average of the tracker's positions over an approximately 1/8 second time interval. We will first discuss briefly the need for time-realignment and the procedures to carry out a realignment followed by an averaging operation, then representative results will be presented. The computer program details involved in this are in Appendix D.

Because the 8 S&H gates are fed directly to the digital delay generator (DDG) in the S-193 radar altimeter's altitude tracking loop, finite altitude rates and/or tracker jitter will lead to each S&H set's sampling different time-positions on a return waveform. Simply averaging together all measurements by S&H gate #1, gate #2,...,8 to form average sample points #1, #2, ...,#8 without correcting for the tracker jitter should lead to a mean waveform distortion and an increase in the variance of the 8 sample points. A more detailed investigation of jitter effects on waveform and variance is provided by another Applied Science Associates Report ["Final Report on Task D - Engineering Studies Related to the GEOS-C Radar Altimeter" by L. S. Miller and G. S. Brown, prepared under Contract NAS6-2307, May, 1974]. Briefly, the mean waveform is "smeared", essentially by a convolution with the tracker jitter process's probability density function and there is an attendant increase in variance estimates. The waveshape effects should be negligible for the S-193 100 nanosecond pulsewidth modes but significant for the short pulse submodes of Mode 5, the 10 ns (nominal design pulsewidth) direct or pulse compression submodes.

The correction for the tracker jitter involves comparing instantaneous tracker position with "true" tracker position; this difference is a measure of how far the 8 S&H gate results must be repositioned earlier or later in time before adding up a number of separate sets of S&H data to determine an average waveform over a suitable averaging period. There are two unknown quantities to be estimated to find this instantaneous tracker error however, the "true tracker position" (or equivalently, the "true" altitude) and the instantaneous tracker position.

To determine the "true altitude" as a function of time, a low-order polynomial is least-squares fitted to the altitude data over an entire submode. This is in effect a type of smoothing operation. In practice, the maximum degree is 4 for the polynomial fitting [done in subroutine POLRG as explained in Appendix D].

The altitude outputs from the S-193 altimeter at the 8 per frame rate are <u>not</u> 8/frame measurements of the tracker's instantaneous position. Rather, each altitude output is an average of the tracker's position during the ~1/8 frame preceeding that output. Part of the Applied Science Associates Task A Report, July 1973, was devoted to deriving a weighting function to estimate tracker instantaneous positions, and that weighting function is used in this work.

The 8/frame instantaneous tracker position estimates are used to produce the needed 104/frame S&H time corrections, and a spline is used for this [see Appendix D]. The time-realigned sets of S&H readings are summed into a set of time bins [as described in the Applied Science Task A Report and summarized by the time sketch in subroutine VTADD in Appendix D of this report], and the results are written on tape on a frame-by-frame basis. A subsequent averaging operation reads the frame-by-frame tape, and prints and plots the results on a 10 frame basis [this last program was developed by Wallops personnel and is not described in this report].

Figure 13 summarizes the overall time-realignment and averaging as just described. Figure 14 shows a portion of actual altitude data as it comes out of box D of Figure 13 ; also indicated are the 8/frame instantaneous tracker error estimates out of box E of Figure 13. Then Figure 15 repeats these 8/frame instantaneous error estimates and shows the 104/frame individual tracker errors as produced by the spline - these 104/frame error estimates provide the time-corrections to be applied before summing the 104/frame sets of S&H data.

It is necessary that the S&H data already be corrected for dc offsets before carrying out the averaging after the time-realignment. This is because each time bin's contents includes contributions from several different S&H

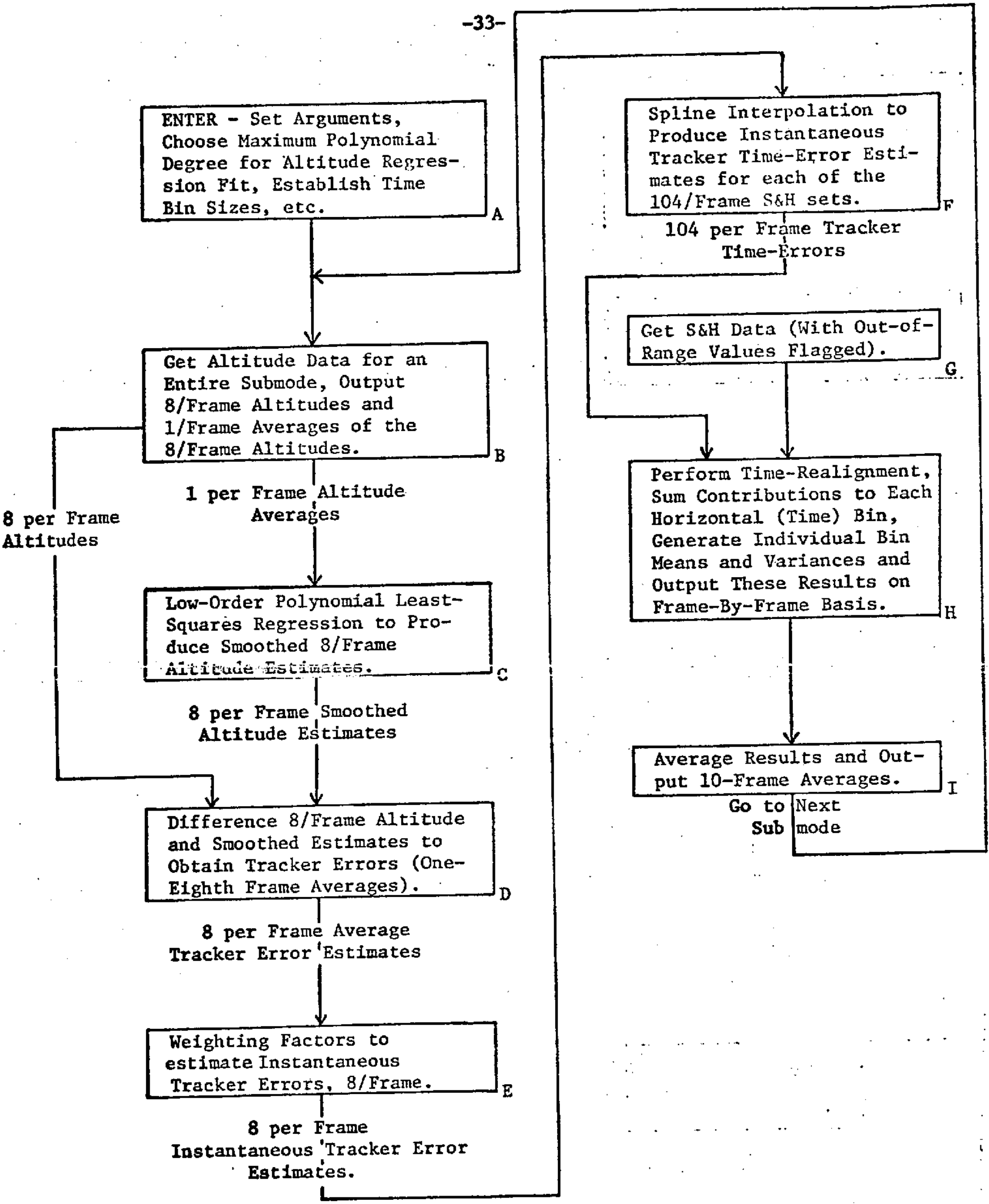


Figure 13. Time-Realignment and Averaging

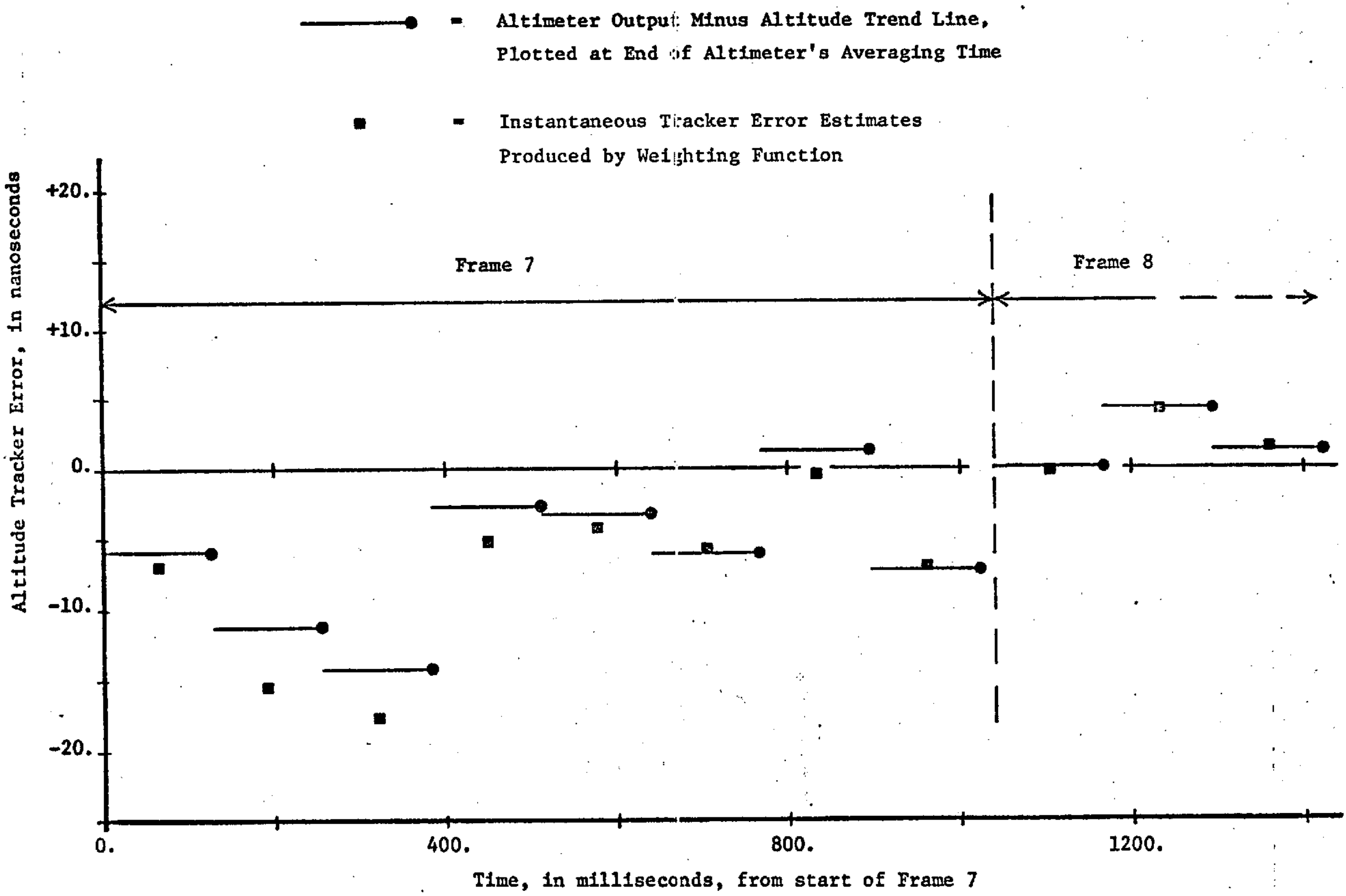


Figure 14. Tracker Time-Errors For Frame 7 And Part Of Frame 8, S-193 SL-2, Pass 9, Mode 5, Submode 1, Subsubmode 0.

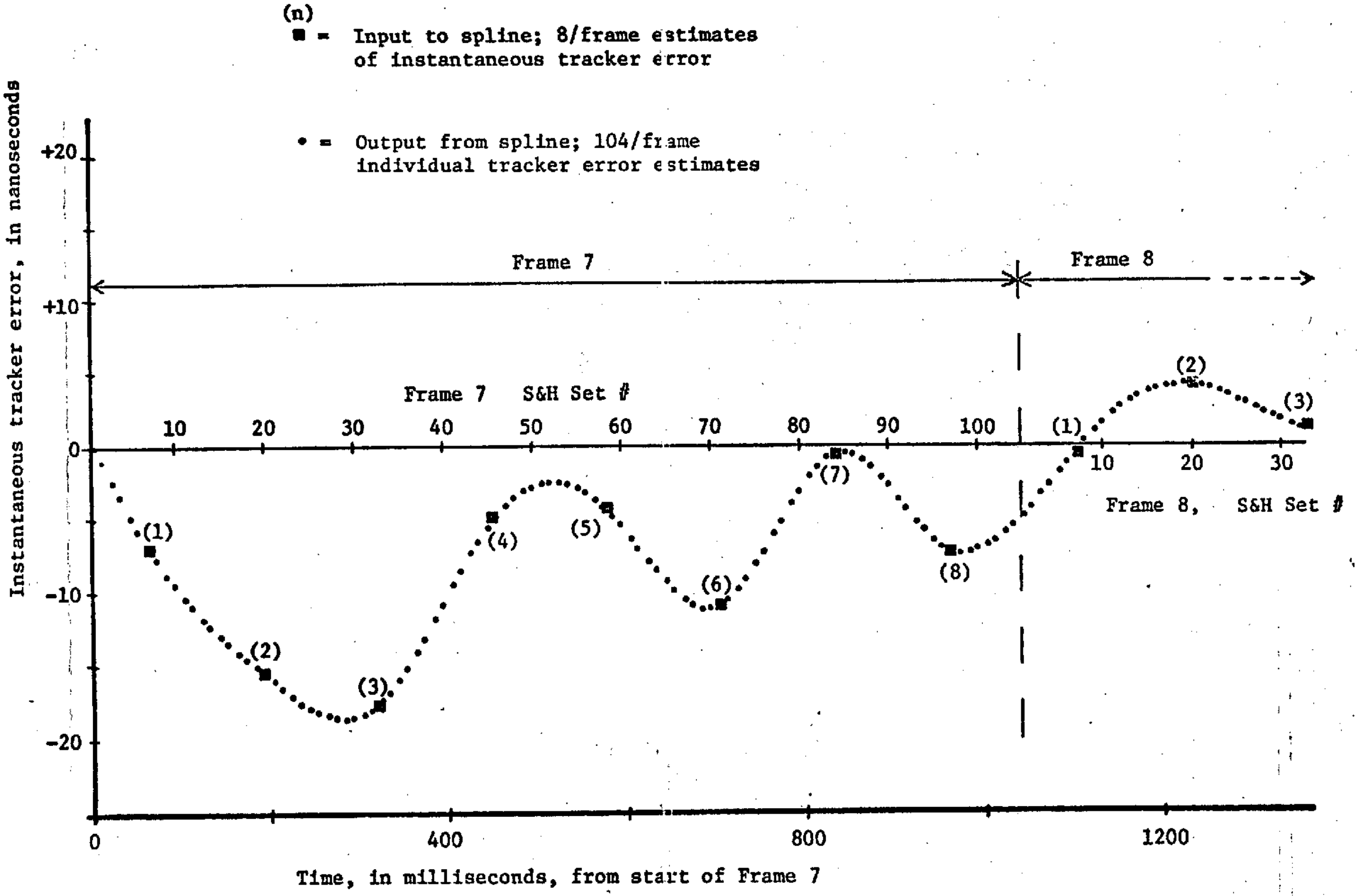


Figure 15. Tracker Instantaneous Error Estimates for Frame 7 and Part of Frame 8, S-193 SL-2, Pass 9, Mode 5, Submode 1, Subsubmode 0.

gates over the time period of a frame, and there is no way to calculate or correct for this combination of several different S&H offsets after the time-realignment is applied.

One additional problem (not shown on Figure 13) is that it is possible that some of the S&H readings in the primary data tape (from JSC in Houston) may fall outside the allowed region of the S&H calibration curves; this occurs because of the way in which the calibration curves were handled in the JSC processing. To avoid contamination of the waveforms by these invalid data, a test procedure has been built into the programs at Wallops to detect the occurrence of S&H readings outside the allowed range, and to flag these data so that they are not included in the waveform averages determined in the later program steps. Consequently there will not necessarily be 104 contributions per frame to the waveform averages, but 104 will be the maximum number possible per frame.

Decisions have yet to be made about how to handle date for which there are appreciable numbers of loss-of-lock indications in the altitude tracker's output. This is part of the large, general problem of data editing and, as already mentioned in the introduction to this report, no suitable set of data quality criteria has yet been found which would permit the implementation of an automatic data editing part of the overall waveform processing.

Figure 16 presents a preliminary indication of the time-realignment results from SL-2 Pass 9, Mode 5, submode 2; what has not been shown on the figure is that points shown are of unequal weights. The next step would be to form appropriate weighted means and then use these as input to a final pass through X2MIN to determine final mean return waveshape parameters. It is apparent however that no striking change in the leading edge risetime has appeared - and this is the parameter of the greatest interest in attempts to detect sea-state from S-193 waveform measurements. Another disturbing feature not displayed in the figure is that no changes were found in variances estimated from the time-realignment procedure as opposed to variances from simply summing up all of the measurements of a specified S&H gate - we had expected to see variance changes approaching 25%.

Figure 17 presents results for the time-realignment process applied to SL-3 Pass 28/39, Mode 5, Submode 1; no offset corrections have been applied

Data have been offset-corrected. Results shown are averages over 10 frames in each sub-submode (except sub-submode 2 for which only 5 frames of data are taken). Note that the individual points below for the time-realigned waveform are not of equal weight.

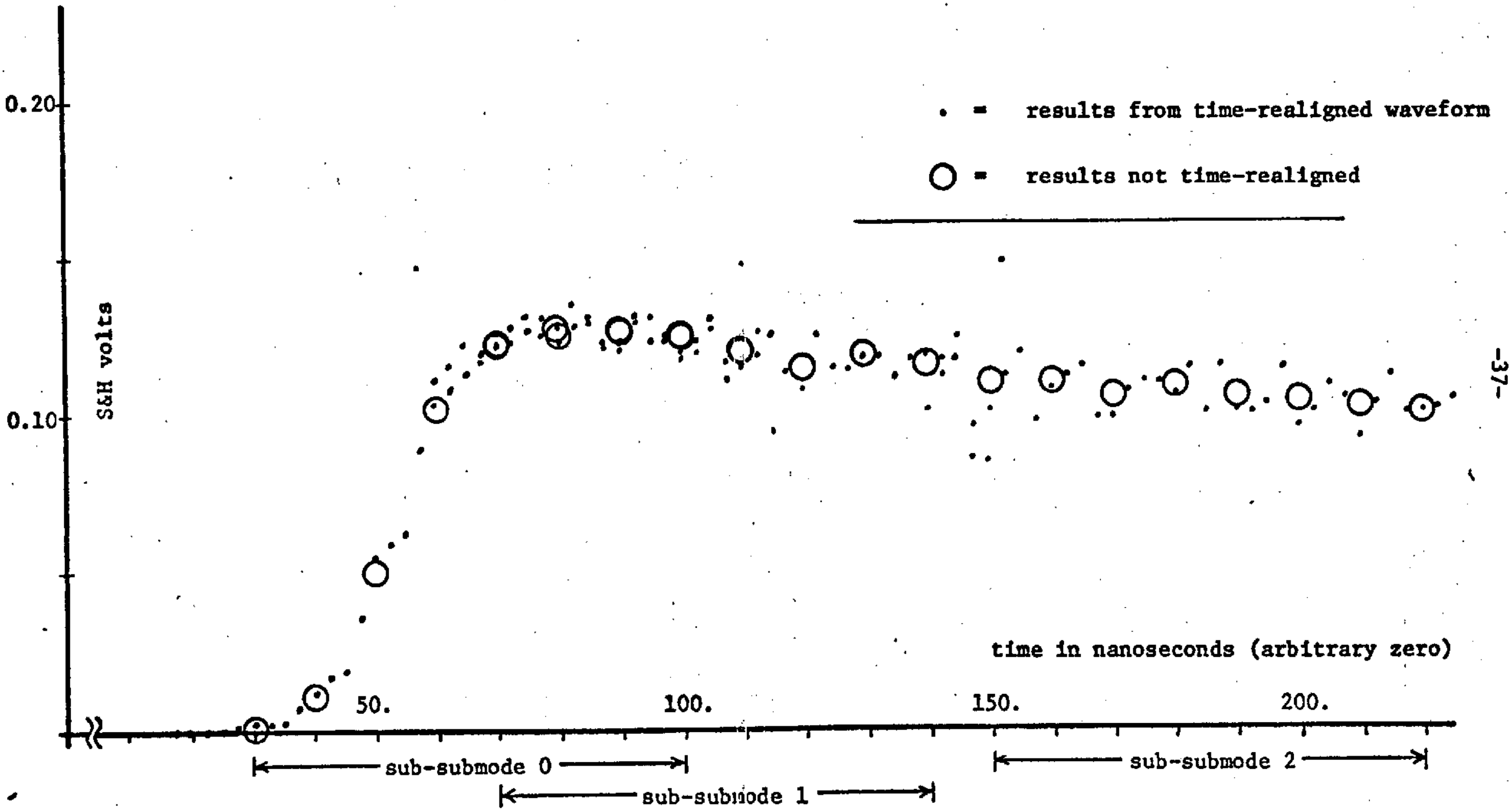


Figure 16. SL-2, Pass 9, Mode 5, Submode 2 Time Re-alignment Results

No offset corrections have been applied. Results shown are averages over 10 frames in each sub-submode (except sub-submode 2 for which only 5 frames of data are taken). Note that the individual points below for the time-realigned waveform are not of equal weight.

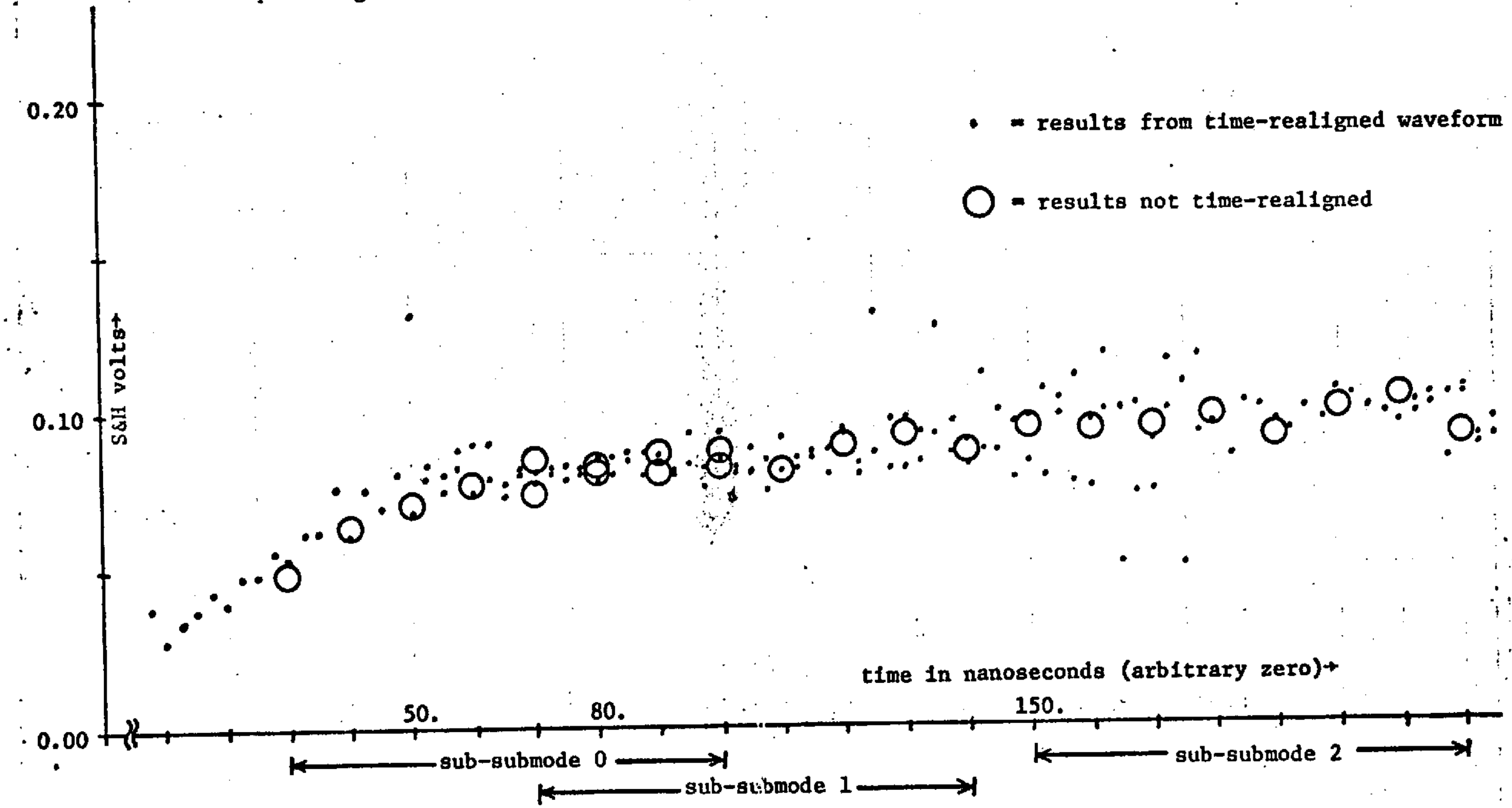


Figure 17. SL-3, Pass 28/39, Mode 5, Submode 1 (Pulse Compression) Time-Realignment Results.

to the S&H data before time realignment. This figure displays one of the hoped-for outcomes of the time-realignment as there seem to be data which are recoverable in the leading edge earlier than the first S&H gate. Since the S&H set started too late on the waveform, even in subsubmode 0, these data are very desireable. However, we again here have totally unknown S&H offsets. The only chance of recovering the desired information might be some sort of iterative approach using first the offset-determining procedure, then the time-realignment; by going back and forth, and carrying lineshape parameters from the time-realignment back to the offset determination, it might be possible to recover something. We have not yet tried this.

Returning to the SL-2 Pass 9, M5, SM2 time-realignment results, no significant changes were obtained for a variety of different guesses at S&H offsets. Eventually growing frustrated by not being able to make the situation better, we tried to make things worse just to determine that there were no overlooked sign errors. For instance, first we replaced the weighting function by a single unit weight(which is equivalent to simply bypassing the box E of Figure 13) - again, no significant changes were found in the results. Finally we reversed the sign of the time-error correction with the results shown in Figure 18. There it does appear that the risetime is shorter for the non-time-realigned results than for the deliberately erroreously corrected results.

These results all indicate that we are not now gaining enough information from the time-realignment to justify applying it in any automatic or routine fashion to the remainder of the S-193 waveform data. Some additional small-scale investigation of time-realignment should be continued on a case-by-case basis, and the work presented in this report providesthe basic building blocks for this. The change in the short-pulse mode's pulsewidth from nominal design width of 10 ns to an actual pulsewidth closer to 20 ns is probably the largest single factor contributing to our failure to realize benefits from time-realignment. It is worth noting that our S-193 experience further supports the conclusions of the earlier Applied Science Associates Report [our Task D of Contract NAS6-2307] which indicated that waveform time-realignment was not going to be necessary for GEOS-C if that system met its specifications.

Data have not been offset-corrected. Results shown are averages over 10 frames in each sub-submode (except sub-submode 2 for which only 5 frames of data are taken). Note that the individual points below for the time-realigned waveform are not of equal weight.

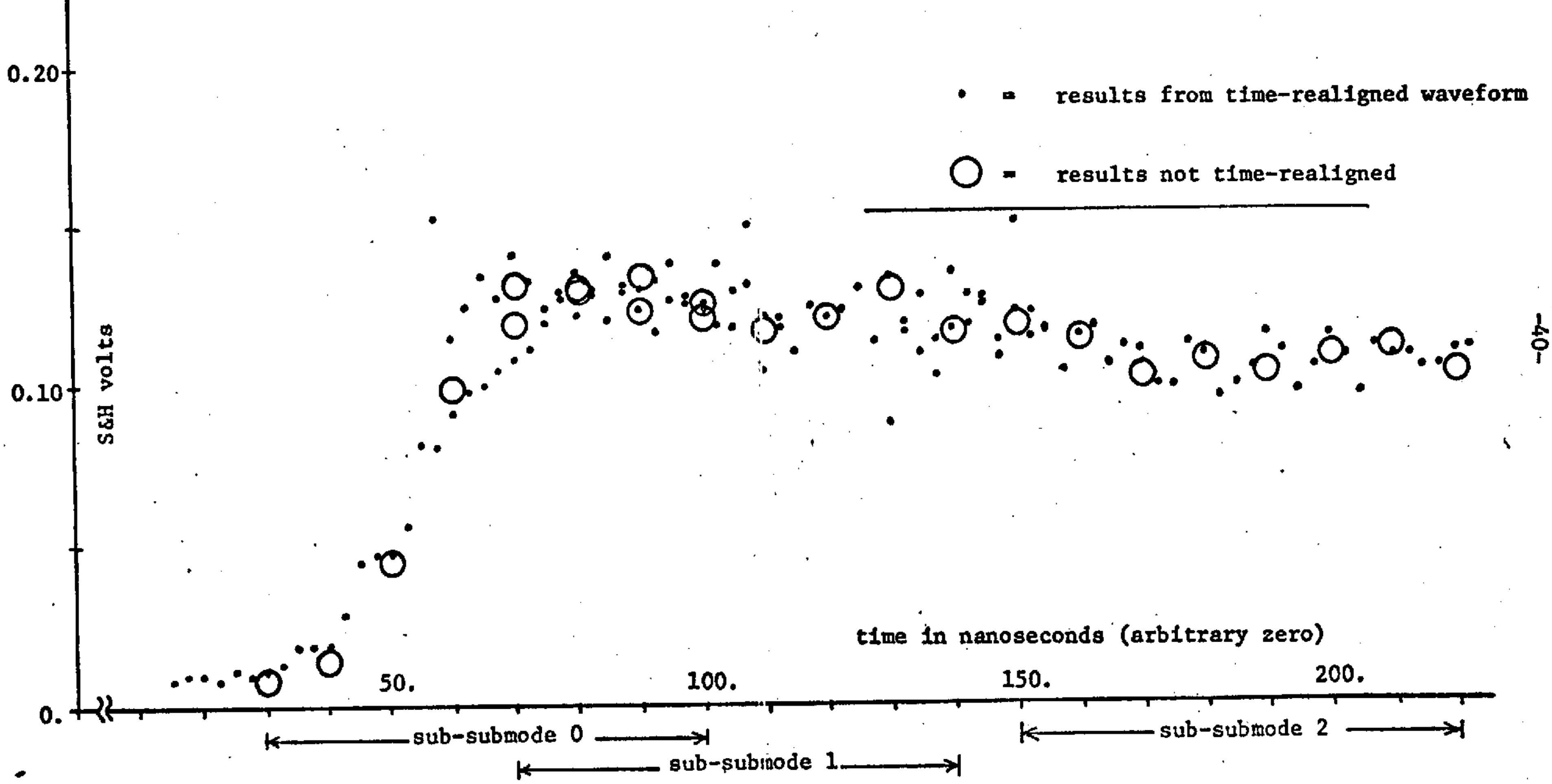


Figure 18. SL-2, Pass 9, Mode 5, Submode 2 Time-Realignment Results For Deliberate Error In The Sign Of The Time-Correction In The Realignment Procedure.

APPENDIX A. SUBROUTINE X2MIN FOR GENERAL FUNCTION FITTING TO DATA.

In our investigation of the S&H offset in waveform data from the S-193 radar altimeter, we used a general-purpose subroutine X2MIN for fitting a theoretical function to the experimental data. X2MIN is sufficiently general as to merit further description in this appendix. We did not invent X2MIN, and the version presented here differs only in very minor details from a program obtained from Professor Dwight W. Carpenter at the Duke University Physics Department (Dr. Carpenter is now at Limestone College, Gaffney, South Carolina). Dr. Carpenter's program was in turn an extensive revision of an earlier general-purpose fitting routine from the University of Illinois. The following description is entirely derived from Dr. Carpenter's program write-up at Duke University.

X2MIN is a FORTRAN subroutine to fit an arbitrary function to given data points by minimizing the weighted squared deviations of the points from the function. The function is of the form,

$$Y = FX(X_1, X_2, \ldots, X_{NX}, A_1, A_2, \ldots, A_{NA}, C_1, C_2, \ldots, C_{NC}).$$

The X's are the independent variables, the A's are the variable parameters to be adjusted by the program, and the C's are constant parameters. The data consists of a set of NP points of measured Y^i at given values of X_1^i, $X_2^i, \ldots, X_{NX}^i$ with associated error estimates σY^i; $i = 1$, NP.

$$\chi^2 = \sum_{i=1}^{NP} \frac{[Y^i - FX(X^i, A, C)]^2}{(\sigma Y^i)^2}$$

The program requires initial guesses for the variable parameters A. Successive corrections are made to the A's until a relative minimum in χ^2 is reached. Reasonably good initial guesses are often necessary. Additional constraints may be placed on the A's by providing an error estimate σA^i on the initial value of A^i. For each constrained A^i, the term

$$[(A^i_{initial} - A^i_{final})^2 / (\sigma A^i)^2]$$

is added to the χ^2.

The iteration continues until the fractional change in χ^2 in one step is less than a given limit (CHILIM), or until a given maximum number of interations (MAXITR) is reached.

The user must supply a function type FORTRAN subroutine, FUNCTION FX, which will evaluate FX (and, optionally, its derivatives with respect to the A's) for a given set of X's, A's, and C's. The data deck organization for any one problem is as follows:

Card Type	Information	No.	Short Cuts
Comment	Any	1	Repeat? →
Size	NP,NX,NA,NC,CHILIM,MAXITR,NPLOT	1	↓
Data Label	X,Y,σY names	1	↓ Repeat? →
Data Point	X,Y,σY values	NP	↓
Var. Param (A)	A name, initial value, step, constraint	NA	↓ ←
Const. Param (C)	C name, value	NC	↓

This may be followed by succeeding problems. The Short Cuts may be used on succeeding problems if portions of the input data are to be repeated. A single card which preceeds the first of the problem decks sets up a set of ten general flags INFLAG(10) which are available through the labelled COMMON area, COM1.

The calling procedure for X2MIN is as follows:

```
---Misc Job Control Cards -- (input on device 1, output on 3)---
                  COMMON/COM1/INFLAG(10)
                  READ(1,5)(INFLAG(I),I=1,10)
                5 FORMAT(10I5)
                  CALL X2MIN
                  STOP
                  END
```

Subroutine X2MIN calls the additional subroutines LSQMIN, MINV20, XPROB, and PLOT4 (with entry points PLOTWD, PLOT3, and PLOT3L) as well as function FX. Following details of the X2MIN input data and of FX in general, source listings are provided for X2MIN, LSQMIN, MINV20 and XPROB. In addition to facilitate use of X2MIN elsewhere and to ease debugging or modification, logical flow diagrams are provided for X2MIN and LSQMIN. Because the details are computer-dependent, no source is provided for PLOT4 (with

PLOTWD, PLOT3, and PLOT3L); however a description of PLOT4 is provided.

X2MIN Input Data

Six types of data cards are described in the order of their appearance in the data deck:

1. COMMENT CARDS

The comment cards will simply be read and printed out. At least one comment card must appear. Each card except the last must have .... (four periods) punched in columns 77-80; the last card must not have them.

If any comment card starts with the word REPEAT in columns 1-6, the program will expect the Variable Parameter (A) cards to follow the last comment card. The Size, Data Label, and Data Point information will be carried over from the preceding problem.

2. SIZE CARD

Seven numbers appear on one card in successive fields of 10:

NUMBER	FORMAT	DESCRIPTION	ALLOWED VALUES
NP	5X, I5	Number of data points	1-100
NX	5X, I5	Number of independent variables (X's)	1-10
NA	5X, I5	Number of variable parameters (A's)	1-20
NC	5X, I5	Number of constant parameters (C's)	0-20
CHILIM	5X, F5.0	$\Delta\chi^2/\chi^2$ convergence test. (If left blank, 0.001 is used)	Any positive value
MAXITR	5X, I5	Maximum number of iteractions taken. (If left blank, 30 is used)	Any positive integer
NPLOT	5X, I5	1 causes plots to be deleted	0 (blank)-1

3. DATA LABEL CARD

Alphabetic characters, 10 columns each, giving the names of the variables $X_1, X_2 \ldots X_{NX}$, Y, σY (NX + 2 names) in sequence. These will be used as column headings in the printout.

If the word REPEAT appears in the first 6 columns (instead of the labels), the Data Point Cards must be deleted. The Data Label and Data Point information will be carried over from the previous problem.

If the words STAT ERROR are inserted on the Data Label Card as the label for σY, the value taken for σY is $\sqrt{FX}$ (not $\sqrt{Y}$) for each point. A lower limit of 0.1 is arbitrarily imposed on FX for this calculation. The σY columns on the Data Point Cards are ignored in this option.

4. DATA POINT CARDS

One card for each data point as follows:

X_1, X_2, ...X_{NX}, Y, σY in format (8F10.0). If the σY are left blank (and the STAT ERROR is not specified), all points will be given equal weight and portions of the output dealing with errors will be deleted.

5. VARIABLE PARAMETER CARDS (A'S)

One card for each parameter as follows:

Parameter Name	(A10)	Alphabetic name of parameter
Continuation Flag	(A1)	If C, the initial value is taken as the final value in the preceding problem. If blank, the new initial value is taken from the card.
Initial Value	(F14.0)	Initial Value
Step Size	(F15.0)	Step (ΔA) used to find dY/dA, or blank if subroutine computes dY/dA directly.
Constraint	(F15.0)	Standard deviation (σA) assigned to initial value. If blank (or zero), no constraint is made.

6. CONSTANT PARAMETER CARDS (C's)

One card for each (if any) constant parameter as follows:

Parameter Name (A10)
Continuation Flag (A1) - as above in A's
Constant Value (F14.0)

EVALUATION OF DERIVATIVES

The program must evaluate the derivatives of FX with respect to the A parameters. This may be done in one of two ways for each parameter:

(1) A step size ΔA (non-zero) may be given on the A cards. The program then makes a step ΔA in A and takes ΔFX/ΔA as the derivative. The step size

must be chosen small enough that the derivative is essentially constant over the step, but large enough to make a significant change in FX. An appropriate step size is often critical. It may be checked by varying the step size to see that the results are independent of reasonable changes.

(2) A formula which evaluates dFX/dA for each A may be written into the function routine FX. In this case, the function must return

$$FX = dFX/dA(JA) \text{ whenever } JA > 0$$

and

$$FX = FX \text{ function whenever } JA = 0.$$

JA is a flag given the subroutine by the main program. The step size field must be left blank (or zero) for each parameter whose derivative is to be computed in this way.

The two methods may be intermixed if desired. The main program gives the flag JA > 0 for those A not given step sizes.

PRINTOUT

The printout contains the following information for each problem:

All input information.

The value of χ^2 and the A's at each iteration.

The final errors on the A's, $\sqrt{<(\Delta A)^2>}$

The degrees of freedom and χ^2 probability (high number for good fit, low for bad, in percent)

The value of the best fit function FX at every data point

The deviation of each data point Y^i from the best fit FX

The $(\text{deviation}/\sigma Y)^2$ for each point, i.e., the contribution to χ^2

The number of the worst point

A plot of the deviations in Y^i vs X_1^i

A plot of the Y^i vs X_1^i

A correlation matrix

The error matrix is $(EM)_{ij} = <\Delta A_i \cdot \Delta A_j>$ where ΔA_i is the error in the parameter A_i. The rms error on A_i is $\sqrt{<\Delta A_i \cdot \Delta A_i>}$. The correlation matrix is related to the error matrix by:

$$(CM)_{ij} = (EM)_{ij} / \sqrt{<\Delta A_i^2>} \sqrt{<\Delta A_j^2>} .$$

$(CM)_{ij}$ will range from 0.0 for uncorrelated parameters to ± 1.0 for completely correlated parameters.

ERROR MESSAGES

The message SINGULAR MATRIX will occur if a matrix is encountered which cannot be inverted as required. This usually indicates an error in the data or in FX.

FUNCTION FX

The function subroutine FX must be supplied. The FUNCTION, COMMON, and DIMENSION cards must be as shown. The routine must evaluate FX for point JP;

```
FUNCTION FX (JP,JA)
COMMON/COM1/INFLAG(10)
COMMON X,Y,A,NA,NX,NC,C,NP
DIMENSION X(100,10),Y(100),A(20),C(20)
---test on JP≤0---
FX = function evaluated for X(J,1),...X(JP,NX),A(1),...A(NA),
          C(1),...C(NC)
RETURN
END
```

In X(JP, JX), JP is the point nunber ($1 \le JP \le NP$) and JX is the independent variable number ($1 \le JX \le NX$). Immediately upon entering X2MIN for the first time, there is a statement "F = FX(-1,-1)"; after the first problem is finished, the program will execute the statement "F = FX (0,0)" immediately prior to reading each new problem. The values -1 and 0 for JP are provided for possible initialization in FX, and must be tested for. The use of COM1 is optional; only if some use is to be made of flags INFLAG(10) must COMI be provided in FX.

If the function is to evaluate derivatives explicitly (as described above as method 2) JA must be tested:

```
   IF (JA) 1, 1, 2
1  FX = FUNCTION
   RETURN
2  FX = derivative dFX/dA(JA)
   RETURN
   END
```

EXAMPLE

P is measured for various T and V for gasses with known R. One wants to find AA and BB in the following equation:

$$P = R*T/(V - BB) - AA/V^2$$

The FX subroutine might be:

```
  FUNCTION FX (JP,JA)
  COMMON X,Y,A,NA,NX,NC,C
  DIMENSION X(100,10),Y(100),A(20),C(20)
  IF (JP.LE.0) RETURN
  T=X(JP,1)
  V=X(JP,2)
  R=C(1)
  AA=A(1)
  BB=A(2)
  IF(JA) 1,1,2
1 FX=R*T/(V-BB)-AA/V**2
  RETURN
2 GO TO (3,4),JA
3 FX=-1.0/V**2
  RETURN
4 FX=R*T/(V-BB)**2
  RETURN
  END
```

A typical data deck for this problem might then appear as follows (notice the use of the REPEAT feature, and the continuation flag C on parameter BB in the repeated part):

```
Column 1
        10        20   25   30        40             55                     77-80
FIT AA AND BB FOR GAS
  NP=     5  NX=    2  NA=    2  NC=    1
      TEMP       VOL     PRESS   P ERROR
     300.0      27.6      18.4      0.10
     230.0      20.0      32.4      0.15
     240.0      20.0      15.2      0.10
        AA            .03
        BB            .01
         R           50.2
REPEAT WITH AA CONSTRAINED TO .03 +-.01                                     ....
AND USING STEP SIZES TO FIND DERIVATIVES (CONTINUE BB)
        AA            .03          .0001            .01
        BBC                         .001
         R           50.2
```

NOTES

The constraint in the A's is useful for two things:

(1) introducing outside knowledge about a parameter
(2) keeping a variable from running wild in the early stages of fitting. The constraint may be released in later stages.

The function FX has available to it NA and NC which may be useful in having the number of terms set at run time, or changed during a run.

The C's may be used as program switches by the function FX as well as actual numerical constants. Thus several different functions could be included in one FX. The flags INFLAG(10) are provided for additional switches, but the first three of these are used by PLOT4 as described later.

If the step sizes are used, one should see what the effect of the choice of size has on the results.

A high correlation coefficient between two parameters ($|C| \geq .9$) means variations in the two parameters have a similar effect on the function, making it hard to fit and causing large uncertainties in the values. One should try to find a parameterization which will give low correleations. Beware uf $C \geq .98$.

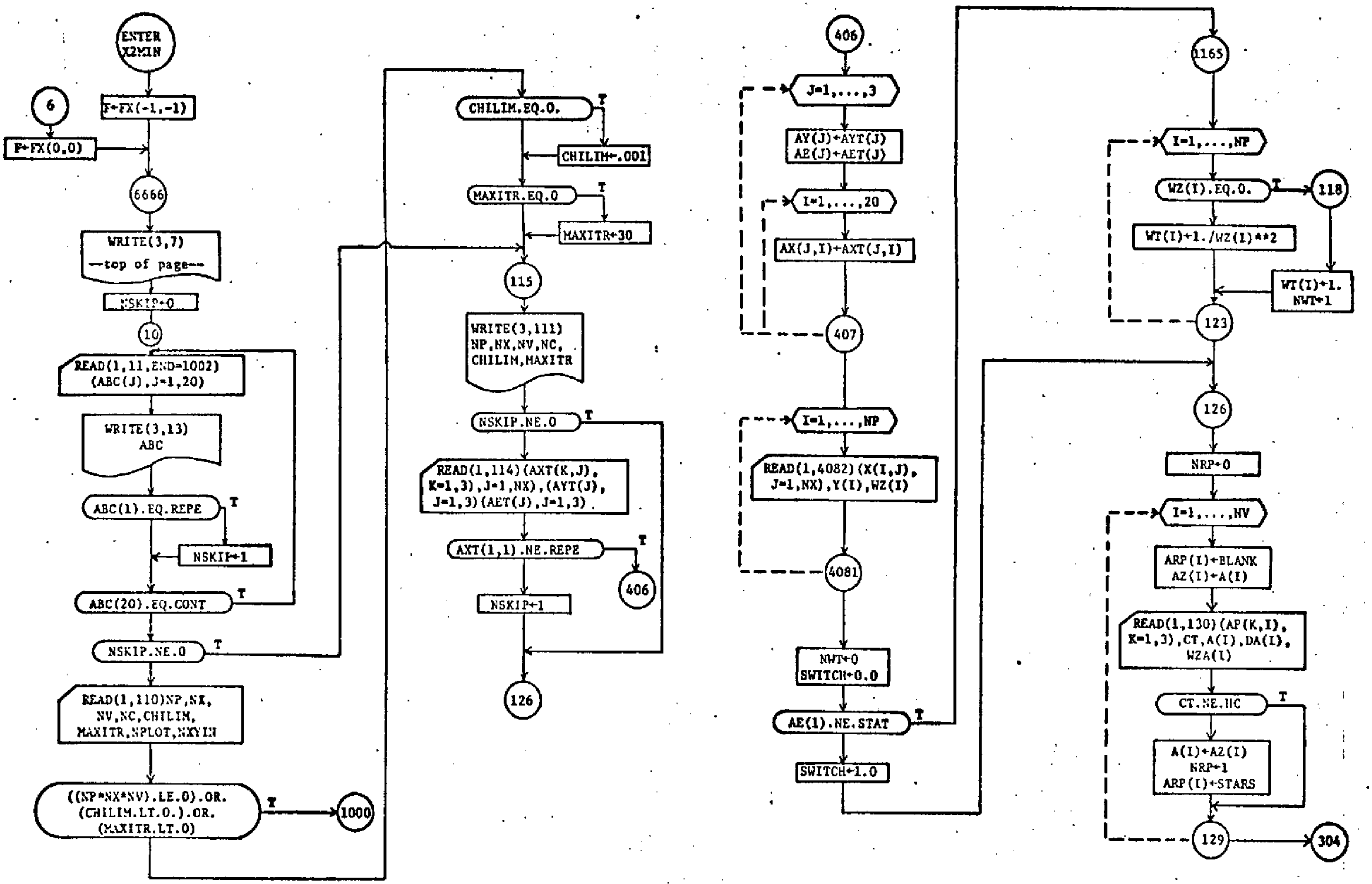
The following material will be presented in the order listed here: X2MIN source listing, X2MIN flow diagram, LSQMIN source, LSQMIN flow diagram, MINV20 source listing, XPROB source listing, and description (only) of PLOT4.

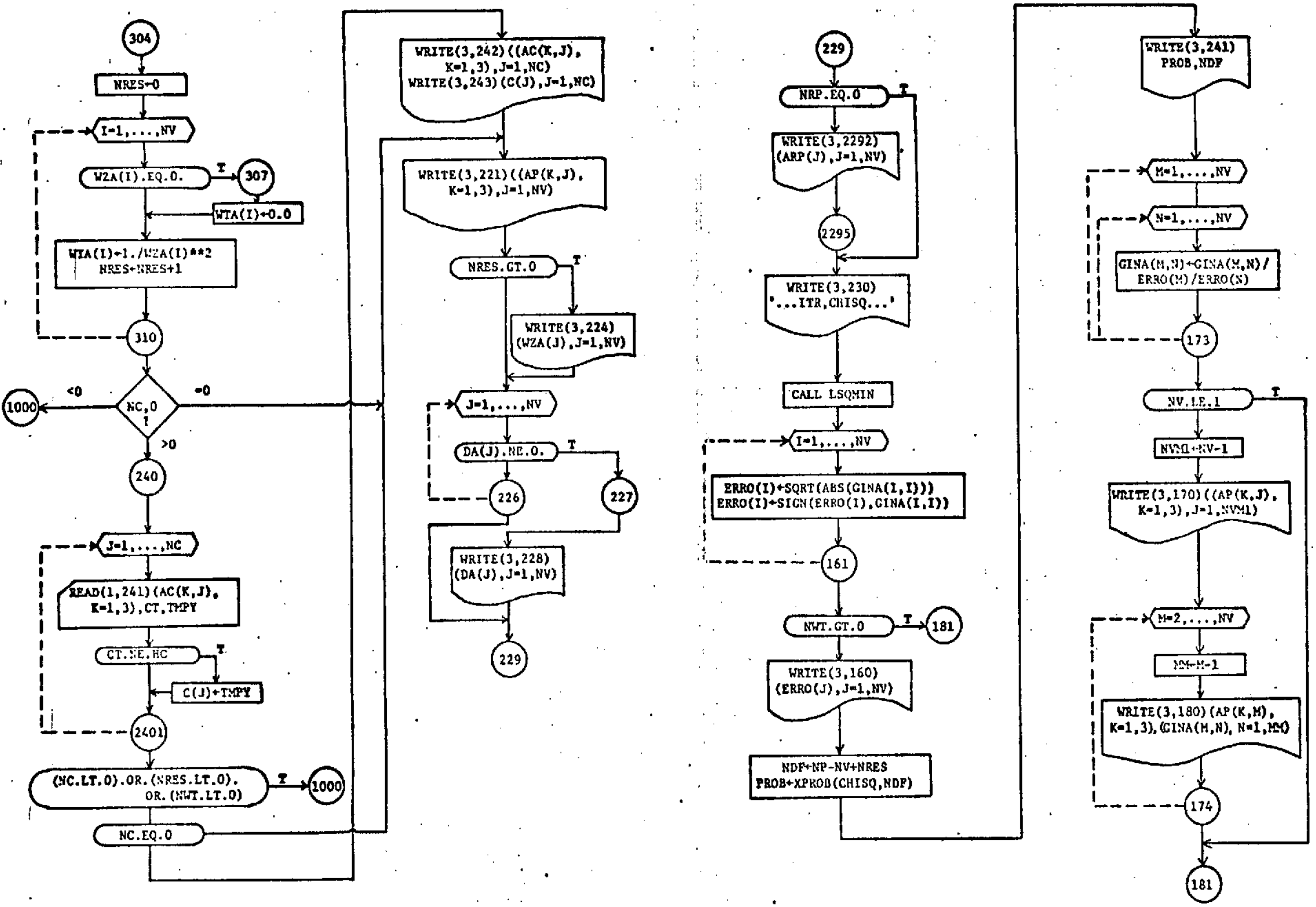
```
      SUBROUTINE X2MIN
      DIMENSION X(100,10),Y(100),DA(20),GINA(20,20),WT(100),A(20),AZ(20)
      DIMENSION WTA(20),TP(100),C(20),ERRO(20),AYT(3),AET(3),ARP(20)
      DIMENSION AX(3,20),AY(3),AE(3),AP(3,20),AC(3,20),ABC(20),XP(101)
      DIMENSION WZ(100),WZA(20), CSQ(100),AXT(3,20),YO(100),YP(101)
      COMMON X,Y,A,NV,NX,NC,C,NP,DA,GINA,WT,WTA,CHISQ,SWITCH,CHILIM,
     1   MAXITR,ITR,ARP,NRP,AX,AY,AE,AP,AC,WZA
      COMMON /COM1/INFLAG(10)
      REAL CONT/'....'/,STAT/'STAT'/,BLANK/'    '/,STARS/'****'/,
     1 DEV(3)/'DEVI','ATIO','N   '/,CHS(3)/'CHIS','Q   ','    '/,
     2 FUN(3)/'FUNC','TION','    '/,HC/'C   '/,REPE/'REPE'/
      F=FX(-1,-1)
      GO TO 6666
    6 F=FX(0,0)
C------------------------------------------------------- INITIALIZE
 6666 WRITE (3,7)
    7 FORMAT(1H1)
    9 NSKIP=0
C------------------------------------------------------- COMMENT CARDS
   10 READ (1,11,END=1000) (ABC(J),J=1,20)
   11 FORMAT (20A4)
      WRITE (3,13) ABC
   13 FORMAT(1H ,20A4)
   14 IF (ABC(1).EQ.REPE) NSKIP=1
   16 IF (ABC(20).EQ.CONT) GO TO 10
   13 IF (NSKIP.NE.0) GO TO 115
C------------------------------------------------------- SIZE CARD
  100 READ(1,110)NP,NX,NV,NC,CHILIM,MAXITR,NPLOT,NXFIN
  110 FORMAT(4(5X,I5),5X,F5.0,3(5X,I5))
      IF(((NP*NX*NV).LE.0).OR.(CHILIM.LT.0.).OR.(MAXITR.LT.0))GO TO 1000
      IF (CHILIM.EQ.0.) CHILIM=.001
      IF (MAXITR.EQ.0) MAXITR=30
  115 WRITE (3,111) NP,NX,NV,NC,CHILIM,MAXITR
  111 FORMAT(/' ',I4,' DATA PT''S,',I3,' INDEP.VAR''S,',I3,' VAR.PARAMET
     1ERS,',I3,' CONST.PARAMETERS'/'     DCHI/CHI LIMIT=',1PE10.2,
     2 ', ITERATION LIMIT=',I4)
      IF (NSKIP.NE.0) GO TO 126
C------------------------------------------------------- DATA LABEL CARD
  113 READ (1,114) ((AXT(K,J),K=1,3),J=1,NX),(AYT(J),J=1,3),(AET(J),J=1,3)
  114 FORMAT(8(2A4,A2))
      IF (AXT(1,1).NE.REPE) GO TO 406
      NSKIP=1
      GO TO 125
  406 DO 407 J=1,3
      AY(J)=AYT(J)
      AE(J)=AET(J)
      DO 407 I=1,20
  407 AX(J,I)=AXT(J,I)
      DO 4081 I=1,NP
C------------------------------------------------------- DATA POINT CARDS
 4081 READ (1,4082) (X(I,J),J=1,NX),Y(I),WZ(I)
 4082 FORMAT(8F10.2)
      NWT=0
      SWITCH =0.0
      IF (AY(1).NE.STAT) GO TO 1165
      SWITCH=1.
      GO TO 125
 1165 DO 123 I=1,NP
      IF (WZ(I).EQ.0.) GO TO 118
  117 WT(I)=1./WZ(I)**2
      GO TO 123
  118 WT(I)=1.
      NWT=1
  123 CONTINUE
  126 NRP=0
      DO 129 I=1,NV
      ARP(I)=BLANK
      AZ(I)=A(I)
C------------------------------------------------------- VARIABLE PARAM CARDS
      READ(1,130)(AP(K,I),K=1,3),CT,A(I),DA(I),WZA(I)
  130 FORMAT(2A4,A2,A1,F14.0,2F15.0)
      IF (CT.NE.HC) GO TO 129
      A(I)=AZ(I)
      NRP=1
      ARP(I)=STARS
  129 CONTINUE
  304 NRES=0
      DO 310 I=1,NV
      IF (WZA(I).EQ.0.) GO TO 307
      WTA(I)=1./WZA(I)**2
      NRES=NRES+1
      GO TO 310
  307 WTA(I)=0.0
  310 CONTINUE
  239 IF(NC)1000,220,240
  240 DO 2401 J=1,NC
C------------------------------------------------------- CONSTANT PARAM CARDS
      READ (1,241) (AC(K,J), K=1,3),CT,TMPY
      IF (CT.NE.HC) C(J)=TMPY
 2401 CONTINUE
  241 FORMAT(2A4,A2,A1,F14.0)
      IF ((NC.LT.0).OR.(NRES.LT.0).OR.(NWT.LT.0)) GO TO 1000
      IF (NC.EQ.0) GO TO 220
C------------------------------------------------------- WRITE CONSTANTS
      WRITE(3,242) ((AC(K,J),K=1,3),J=1,NC)
  242 FORMAT(10H0CONSTANTS,5X,8(1X,2A4,A2)/22X,8(1X,2A4,A2))
      WRITE(3,243) (C(J),J=1,NC)
  243 FORMAT(15X,1P9G13.4/22X,(8G13.4))
C------------------------------------------------------- WRITE VARIABLE NAMES
  220 WRITE(3,221) ((AP(K,J),K=1,3),J=1,NV)
  221 FORMAT(10H0PARAMETER,8X,8(1X,3A4),2A4,A2/26X,8(1X,3A4))
      IF (NRES.GT.0) WRITE (3,224) (WZA(J),J=1,NV)
  224 FORMAT(11H CONSTRAINT,4X,1P9G13.4/22X, (8G13.4))
      DO 226 J=1,NV
      IF (DA(J).NE.0.) GO TO 227
  226 CONTINUE
      GO TO 229
C------------------------------------------------------- WRITE STEP SIZES
  227 WRITE(3,228) (DA(J),J=1,NV)
  228 FORMAT (10H STEP SIZE, 5X, 1P9G13.4/22X, (8G13.4))
  229 IF (NRP.EQ.0) GO TO 2295
C------------------------------------------------------- WRITE CONTINUED PARAMS
      WRITE(3,2292) (ARP(J),J=1,NV)
 2292 FORMAT(10H CONTINUED,4X, 9(7X,A4,2X)/21X, 8(7X,A4,2X))
 2295 WRITE(3,230)
  230 FORMAT(13H  ITR   CHISQ)
C------------------------------------------------------- DO THE ITERATIONS
      CALL LSQMIN
      DO 161 I=1,NV
      ERRO(I) = SQRT(ABS(GINA(I,I)))
  161 ERRO(I) = SIGN(ERRO(I),GINA(I,I))
C------------------------------------------------------- PARAMETER ERRORS
      IF (NWT.GT.0) GO TO 181
      WRITE(3,160) (ERRO(J), J=1,NV)
  160 FORMAT (1H0,7X,5HERROR,2X, 1P9G13.4/22X, (8G13.4))
      NDF = NP - NV+ NRES
      PROB = XPROB(CHISQ, NDF)
C------------------------------------------------------- CHI-SQUARED PROBABILITY
      WRITE(3,341) PROB,NDF
  341 FORMAT (1H0, 19HCHISQ PROBABILITY  ,1PG10.3, 9H PER CENT,
     X  6X, I4,20H DEGREES OF FREEDOM.)
      DO 173 M=1,NV
```

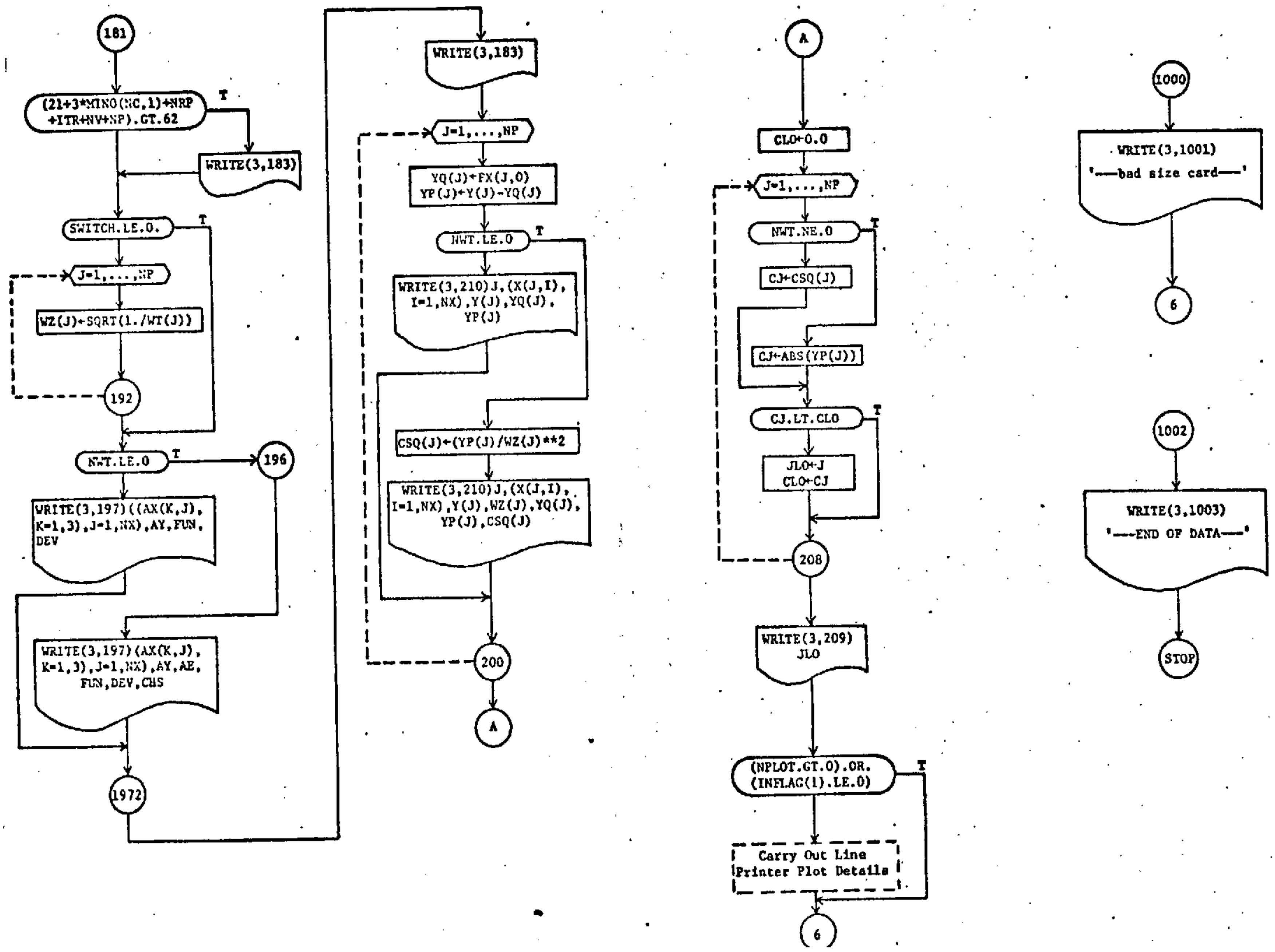
REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      DO 173 N=1,NV
  173 GINA(M,N) = GINA(M,N)/ERRO(M)/ERRC(N)
C--------------------------------------------- CORRELATIONS
      IF (NV.LE.1) GO TO 181
      NVM1=NV-1
      WRITE(3,170) ((AP(K,J),K=1,3),J=1,NVM1)
  170 FORMAT(/' CORRELATION MATRIX '/' ',17X,10(2A4,A3)/' ',21X,10(2A4,
     1 A3))
      DO 174 M=2,NV
      MM=M-1
  174 WRITE(3,180) (AP(K,M),K=1,3),(GINA(M,N),N=1,MM)
  180 FORMAT(' ',2A4,A2,5X,F6.3,9F11.3/' ',22X,10F11.3)
C--------------------------------------------- PAGE LENGTH
  181 IF ((21+3*MINO(NZ,1)+NPP+ITP+NV+NP).GT.62) WRITE (3,183)
  183 FORMAT(1H1)
      IF (SWITCH.LE.0.) GO TO 195
  191 DO 192 J=1,NP
  192 WZ(J) = SQRT(1.0/WT(J))
  195 IF (NWT.LE.0) GO TO 196
      WRITE(3,197)  ((AX(K,J),K=1,3),J=1,NX),AY,FUN,DEV
      GO TO 1972
  196 WRITE(3,197) ((AX(K,J),K=1,3),J=1,NX),AY,AE,FUN,DEV,CHS
  197 FORMAT(/' ',4X,9(1X,3A4)/' ',11X,9(1X,3A4))
 1972 WRITE(3,198)
  198 FORMAT (1H )
      DO 200 J=1,NP
C--------------------------------------------- FUNCTION AT DATA POINTS
      YO(J)=FX(J,0)
      YP(J)=Y(J)-YO(J)
      IF (NWT.LE.0) GO TO 199
      WRITE(3,210) J, (X(J,I),I=1,NX),Y(J),YO(J),YP(J)
      GO TO 200
  199 CSQ(J) = (YP(J)/WZ(J))**2
C--------------------------------------------- DATA POINT LIST
      WRITE(3,210)J,(X(J,I),I=1,NX),Y(J),WZ(J),YO(J),YP(J),CSQ(J)
  210 FORMAT(' ',I3,1PG12.5,1P8G13.5/(11X,9G13.5))
  200 CONTINUE
      CLO = 0.0
      DO 208 J=1,NP
      IF (NWT.NE.0) GO TO 203
      CJ=CSQ(J)
      GO TO 204
  203 CJ = ABS(YP(J))
  204 IF (CJ.LT.CLO) GO TO 208
      JLC=J
      CLO = CJ
  208 CONTINUE
C--------------------------------------------- WORST POINT
      WRITE(3,209)JLC
  209 FORMAT(1H0, 10X,  12H WORST POINT, I4)
      IF ((NPLOT.GT.0).OR.(INFLAG(1).LE.0)) GO TO 6
      IF (INFLAG(2).NE.0) CALL PLOTWD(INFLAG(2),INFLAG(3))
C--------------------------------------------- DATA AND FUNCTION PLOT
      IF(NX-1)1000,555,567
  555 XLO=1.0E75
      XHI=-1.0E75
      DO 556 K=1,NP
      XLO=AMIN1(X(K,1),XLO)
  556 XHI=AMAX1(X(K,1),XHI)
      IF (INFLAG(1).EQ.2) GO TO 5681
      CALL PLOT3('O',X,Y,NP)
      XST=(XHI-XLO)/100.0
      XZ=X(1,1)
      DO 558 K=1,101
      XR(K) =XLO+(K-1)*XST
      X(1,1)=XR(K)
  558 YR(K)=FX(1,0)
      X(1,1)=XZ
      CALL PLOT3L('*',XR,YR,101)
      IF (NWT.GT.0) GO TO 568
  563 DO 566 K=1,NP
      CALL PLOT3L('I',X(K,1),Y(K)-WZ(K),1)
      CALL PLOT3L('I',X(K,1),Y(K)+WZ(K),1)
      CALL PLOT3L('*',0.,0.,0).
      CALL PLOT3('-',X(K,1),Y(K)-WZ(K),1)
  566 CALL PLOT3('-',X(K,1),Y(K)+WZ(K),1)
      GO TO 568
  567 CALL PLOT3('*',X,YO,NP)
  568 CALL PLOT3('O',X,Y,NP)
      CALL PLOT4(10,AX,10,AY,80,ABC)
      IF (INFLAG(1).LE.1) GO TO 6
 5681 IF (INFLAG(2).NE.0) CALL PLOTWD(INFLAG(2),INFLAG(3))
C--------------------------------------------- DEVIATION PLOT
      CALL PLOT3('O',X,YP,NP)
      IF (NX.GT.1) GO TO 579
      CALL PLOT3L('-',XLO,0.0,1)
      CALL PLOT3L('-',XHI,0.0,1)
      CALL PLOT3L('*',0.,0.,0)
      IF (NWT.GT.0) GO TO 578
      DO 576 K=1,NP
      CALL PLOT3L('I',X(K,1),YP(K)+WZ(K),1)
      CALL PLOT3L('I',X(K,1),YP(K)-WZ(K),1)
      CALL PLOT3L('*',0.,0.,0)
      CALL PLOT3('-',X(K,1),YP(K)-WZ(K),1)
  576 CALL PLOT3('-',X(K,1),YP(K)+WZ(K),1)
  578 CALL PLOT3('O',X,YP,NP)
  579 CALL PLOT4(10,AX,9,'DEVIATION',80,ABC)
C--------------------------------------------- GO TO NEXT PROBLEM
      GO TO 6
C--------------------------------------------- BAD SIZE CARD
 1000 WRITE(3,1001)
 1001 FORMAT (14H0BAD SIZE CARD)
      STOP
C--------------------------------------------- END OF DECK
 1002 WRITE (3,1003)
 1003 FORMAT(//' -------- END OF DATA ---- GSH --------'/)
      STOP
      END
```

ORIGINAL PAGE IS OF POOR QUALITY

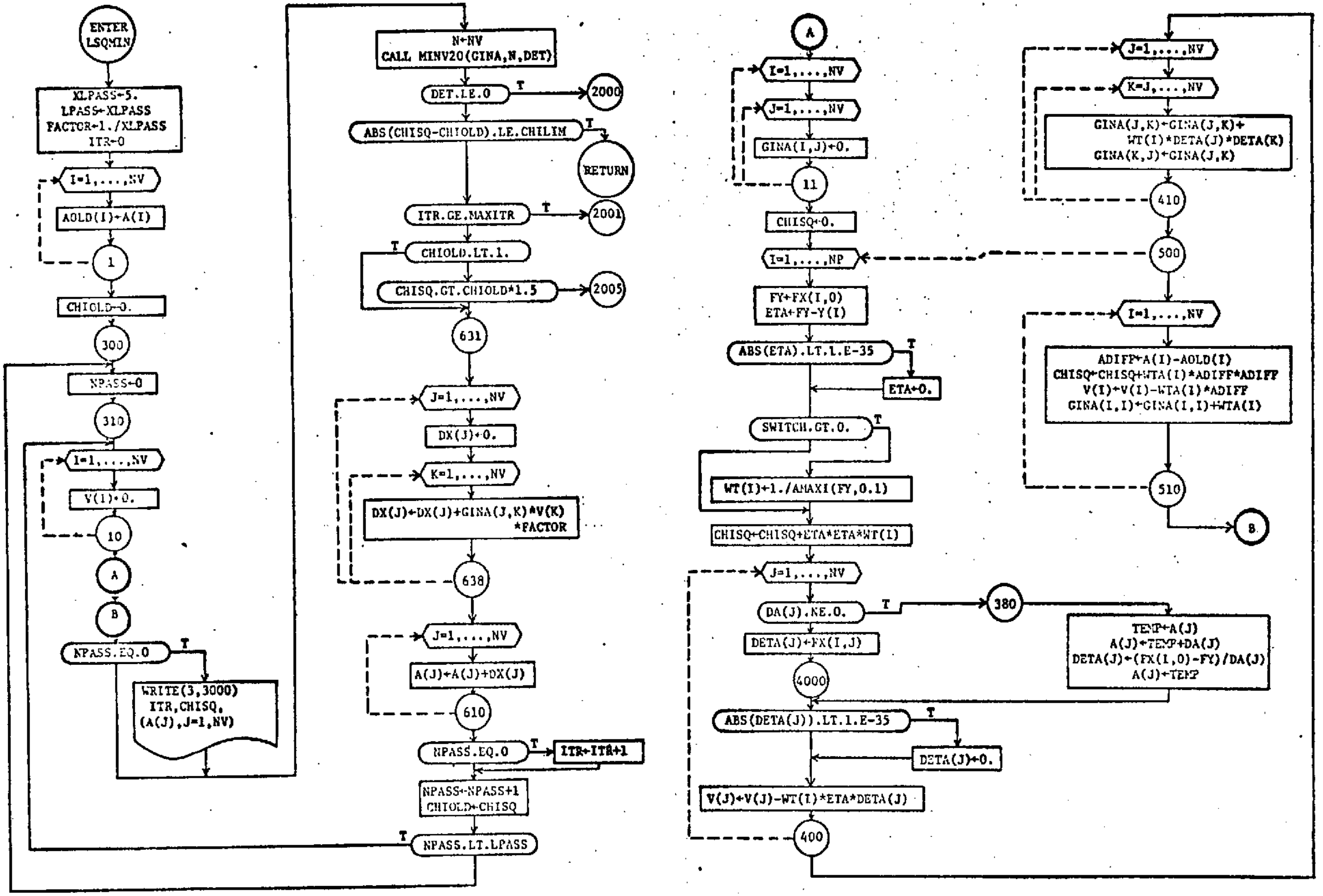

ENTER X2MIN
6
F←FX(-1,-1)
F←FX(0,0)
6666
WRITE(3,7)
—top of page—
NSKIP←0
10
READ(1,11,END=1002) (ABC(J),J=1,20)
WRITE(3,13) ABC
ABC(1).EQ.REPE
T
NSKIP←1
ABC(20).EQ.CONT
T
NSKIP.NE.0
T
READ(1,110)NP,NX, NV,NC,CHILIM, MAXITR,NPLOT,NXYIN
((NP*NX*NV).LE.0).OR. (CHILIM.LT.0.).OR. (MAXITR.LT.0)
T
1000
CHILIM.EQ.0.
T
CHILIM←.001
MAXITR.EQ.0
T
MAXITR←30
115
WRITE(3,111) NP,NX,NV,NC, CHILIM,MAXITR
NSKIP.NE.0
T
READ(1,114)(AXT(K,J), K=1,3),J=1,NX),(AYT(J), J=1,3)(AET(J),J=1,3)
AXT(1,1).NE.REPE
T
406
NSKIP←1
126
406
J=1,...,3
AY(J)←AYT(J) AE(J)←AET(J)
I=1,...,20
AX(J,I)←AXT(J,I)
407
I=1,...,NP
READ(1,4082)(X(I,J), J=1,NX),Y(I),WZ(I)
4081
NWT←0 SWITCH←0.0
AE(1).NE.STAT
T
SWITCH←1.0
1165
I=1,...,NP
WZ(I).EQ.0.
T
118
WT(I)←1./WZ(I)**2
WT(I)←1. NWT←1
123
126
NRP←0
I=1,...,NV
ARP(I)←BLANK AZ(I)←A(I)
READ(1,130)(AP(K,I), K=1,3),CT,A(I),DA(I), WZA(I)
CT.NE.HC
T
A(I)←AZ(I) NRP←1 ARP(I)←STARS
129
304

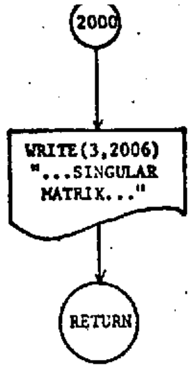

304
NRES←0
I=1,...,NV
WZA(I).EQ.0.
T
307
WTA(I)←0.0
WTA(I)←1./WZA(I)**2
NRES←NRES+1
310
<0
1000
NC,0
?
=0
>0
240
J=1,...,NC
READ(1,241)(AC(K,J),
K=1,3),CT,TMPY
CT.NE.HC
T
C(J)←TMPY
2401
(NC.LT.0).OR.(NRES.LT.0).
OR.(NWT.LT.0)
T
1000
NC.EQ.0
WRITE(3,242)((AC(K,J),
K=1,3),J=1,NC)
WRITE(3,243)(C(J),J=1,NC)
WRITE(3,221)((AP(K,J),
K=1,3),J=1,NV)
NRES.GT.0
T
WRITE(3,224)
(WZA(J),J=1,NV)
J=1,...,NV
DA(J).NE.0.
T
226
227
WRITE(3,228)
(DA(J),J=1,NV)
229
229
NRP.EQ.0
T
WRITE(3,2292)
(ARP(J),J=1,NV)
2295
WRITE(3,230)
'...ITR,CHISQ...'
CALL LSQMIN
I=1,...,NV
ERRO(I)←SQRT(ABS(GINA(I,I)))
ERRO(I)←SIGN(ERRO(I),GINA(I,I))
161
NWT.GT.0
T
181
WRITE(3,160)
(ERRO(J),J=1,NV)
NDF←NP-NV+NRES
PROB←XPROB(CHISQ,NDF)
WRITE(3,241)
PROB,NDF
M=1,...,NV
N=1,...,NV
GINA(M,N)←GINA(M,N)/
ERRO(M)/ERRO(N)
173
NV.LE.1
T
NVM1←NV-1
WRITE(3,170)((AP(K,J),
K=1,3),J=1,NVM1)
M=2,...,NV
MM←M-1
WRITE(3,180)(AP(K,M),
K=1,3),(GINA(M,N),N=1,MM)
174
181

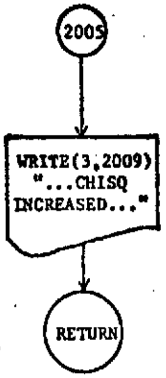

181
(21+3*MINO(NC,1)+NRP
+ITR+NV+NP).GT.62
T
WRITE(3,183)
SWITCH.LE.0.
T
J=1,...,NP
WZ(J)←SQRT(1./WT(J))
192
NWT.LE.0
T
196
WRITE(3,197)((AX(K,J),
K=1,3),J=1,NX),AY,FUN,
DEV
WRITE(3,197)(AX(K,J),
K=1,3),J=1,NX),AY,AE,
FUN,DEV,CHS
1972
WRITE(3,183)
J=1,...,NP
YQ(J)←FX(J,0)
YP(J)←Y(J)-YQ(J)
NWT.LE.0
T
WRITE(3,210)J,(X(J,I),
I=1,NX),Y(J),YQ(J),
YP(J)
CSQ(J)←(YP(J)/WZ(J))**2
WRITE(3,210)J,(X(J,I),
I=1,NX),Y(J),WZ(J),YQ(J),
YP(J),CSQ(J)
200
A
A
CLO←0.0
J=1,...,NP
NWT.NE.0
T
CJ←CSQ(J)
CJ←ABS(YP(J))
CJ.LT.CLO
T
JLO←J
CLO←CJ
208
WRITE(3,209)
JLO
(NPLOT.GT.0).OR.
(INFLAG(1).LE.0)
T
Carry Out Line
Printer Plot Details
6
1000
WRITE(3,1001)
'---bad size card---'
6
1002
WRITE(3,1003)
'---END OF DATA---'
STOP

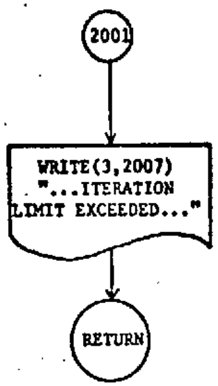
```
      SUBROUTINE LSQMIN
      DIMENSION WTA(20),GINA(20,20),WT(100),Y(100),A(20),DA(20),V(20),
     1DETA(20),DX(20)   ,X(100,10), AOLD(20),C(20)
      COMMON X,Y,A,NV,NX,NC,C,NP,DA,GINA,WT,WTA,CHISQ,SWITCH,CHILIM,
     1MAXITR,ITR
      XLPASS=5.
      LPASS=XLPASS
      FACTOR=1./XLPASS
      ITR=0
      DO 1 I=1,NV
1     AOLD (I) =A(I)
      CHIOLD=0.
300   NPASS = 0
310   CONTINUE
      DO 10 I=1,NV
10    V(I)=0.
      DO 11 I=1,NV
      DO 11 J=1,NV
11    GINA(I,J)=0.0
      CHISQ=0.
      DO 500 I=1,NP
      FY=FX(I,0)
      ETA = FY - Y(I)
      IF (ABS(ETA).LT.1.E-35) ETA=0.
      IF (SWITCH.GT.0.) WT(I)=1./AMAX1(FY,0.1)
      CHISQ=CHISQ+ETA*ETA*WT(I)
      DO 400 J=1,NV
      IF (DA(J).NE.0.) GO TO 380
      DETA(J)=FX(I,J)
      GO TO 4000
380   TEMP=A(J)
      A(J)=TEMP+DA(J)
      DETA(J)=(FX(I,0)-FY)/DA(J)
      A(J)=TEMP
4000  IF (ABS(DETA(J)).LT.1.E-35) DETA(J)=0.
400   V(J)=V(J)-WT(I)*ETA*DETA(J)
      DO 410 J=1,NV
      DO 410 K=J,NV
      GINA(J,K)=GINA(J,K)+WT(I)*DETA(J)*DETA(K)
410   GINA(K,J)=GINA(J,K)
500   CONTINUE
      DO 510 I=1,NV
      ADIFF=A(I)-AOLD(I)
      CHISQ=CHISQ+WTA(I)*ADIFF*ADIFF
      V(I)=V(I)-WTA(I)*ADIFF
      GINA(I,I)=GINA(I,I)+WTA(I)
510   CONTINUE
      IF(NPASS.EQ.0) WRITE(3,3000) ITR,CHISQ,(A(J),J=1,NV)
3000  FORMAT (1H ,I3,1PG12.5,9G13.5/ (G36.5, 7G13.5))
      N=NV
      CALL [illegible](GINA,N,DET)
      IF (DET.LE.0.) GO TO 2000
      IF ((ABS(CHISQ-CHIOLD)/CHISQ).LE.CHILIM) RETURN
      IF (ITR.GE.MAXITR) GO TO 2001
630   IF(CHIOLD.LT.1.E) GO TO 631
      IF(CHISQ.GT.CHIOLD*1.5) GO TO 2005
631   CONTINUE
      DO 638 J=1,NV
      DX(J)=0.0
      DO 638 K=1,NV
638   DX(J) = DX(J) + GINA(J,K)*V(K)*FACTOR
      DO 610 J=1,NV
610   A(J)=A(J)+DX(J)
      IF(NPASS.EQ.0) ITR = ITR + 1
      NPASS = NPASS + 1
      CHIOLD=CHISQ
      IF(NPASS.LT.LPASS) GO TO 310
      GO TO 300
700   RETURN
2000  WRITE(3,2006)
2006  FORMAT (15H0SINGULAR MATRIX)
      RETURN
2005  WRITE(3,2009)
2009  FORMAT('0CHISQ INCREASED')
      RETURN
2001  WRITE(3,2007)
2007  FORMAT (25H0ITERATION LIMIT EXCEEDED)
      RETURN
      END
```

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR


ENTER LSQMIN
XLPASS←5.
LPASS←XLPASS
FACTOR←1./XLPASS
ITR←0
I=1,...,NV
AOLD(I)←A(I)
1
CHIOLD←0.
300
NPASS←0
310
I=1,...,NV
V(I)←0.
10
A
B
NPASS.EQ.0
T
WRITE(3,3000) ITR,CHISQ, (A(J),J=1,NV)
N←NV
CALL MINV20(GINA,N,DET)
DET.LE.0
T
2000
ABS(CHISQ-CHIOLD).LE.CHILIM
T
RETURN
ITR.GE.MAXITR
T
2001
CHIOLD.LT.1.
T
CHISQ.GT.CHIOLD*1.5
2005
631
J=1,...,NV
DX(J)←0.
K=1,...,NV
DX(J)←DX(J)+GINA(J,K)*V(K) *FACTOR
638
J=1,...,NV
A(J)←A(J)+DX(J)
610
NPASS.EQ.0
T
ITR←ITR+1
NPASS←NPASS+1
CHIOLD←CHISQ
NPASS.LT.LPASS
T
A
I=1,...,NV
J=1,...,NV
GINA(I,J)←0.
11
CHISQ←0.
I=1,...,NP
FY←FX(I,0)
ETA←FY-Y(I)
ABS(ETA).LT.1.E-35
T
ETA←0.
SWITCH.GT.0.
T
WT(I)←1./AMAX1(FY,0.1)
CHISQ←CHISQ+ETA*ETA*WT(I)
J=1,...,NV
DA(J).NE.0.
T
380
TEMP←A(J)
A(J)←TEMP+DA(J)
DETA(J)←(FX(I,0)-FY)/DA(J)
A(J)←TEMP
DETA(J)←FX(I,J)
4000
ABS(DETA(J)).LT.1.E-35
T
DETA(J)←0.
V(J)←V(J)-WT(I)*ETA*DETA(J)
400
J=1,...,NV
K=J,...,NV
GINA(J,K)←GINA(J,K)+ WT(I)*DETA(J)*DETA(K)
GINA(K,J)←GINA(J,K)
410
500
I=1,...,NV
ADIFF←A(I)-AOLD(I)
CHISQ←CHISQ+WTA(I)*ADIFF*ADIFF
V(I)←V(I)-WTA(I)*ADIFF
GINA(I,I)←GINA(I,I)+WTA(I)
510
B







REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      FUNCTION XPROB (CHISQ,NDF)
C
C CALCULATE THE CHISQ PROBABILITY GIVEN CHISQ AND DEGREES OF FREEDOM NDF
C XPROB IS PROB THAT ONE SHOULD GET A FIT WORSE THAN ONE GIVEN. ANSWER
C IS EXPRESSED IN PER CENT. (HIGH CHISQ GIVES LOW XPROB.) FOR VERY LOW
C PROB, XPROB SET=0.0, FOR VERY HIGH SET=1.0. ACCURACY THREE PLACE IN
CGENERAL. METHOD GIVEN IN HANDBOOK OF MATHEMATICAL FUNCTIONS, NAT.
C BUREAU OF STANDARDS. EQUATION NUMBERS GIVEN BELOW.
C
      TPROBF(X)=1.0/(1.0+0.33267*X)
C                                   ERROR FN INTEGRAL, EQ. 26.2.16
      QPROBF(T)=Z*T*(.4361836-T*(.1201676-T*.9372980))
C                                   ERROR FUNCTION, EQ 26.2.1
      ZPROBF(X2)=C*EXP (-X2/2.0)
      C=1.0/SQRT(6.283185)
C
      DF=NDF
      X2=CHISQ
C
C                                   TEST FOR NONSENSE INPUT,RETURN XPROB=-100.
C                                   IF NONSENSE
C
      IF(X2) 91,11,11
 11   IF(NDF) 91,91,12
C
C                                   TEST SIZE OF VARIABLES FOR APPROPRIATE
C                                   SECTION
 12   IF(NDF-30) 21,51,51
 21   IF(X2-164.0) 22,81,81
 22   Z=ZPROBF(X2)
      IF( MOD (NDF,2)) 31,31,41
C
C                                   NDF EVEN,LESS THAN 30, EQ. 26.4.5
 31   S=1.0
      A=1.0
      L=(NDF-2)/2
      IF(L) 34,34,32
 32   DO 33 N=1,L
      FDO=2*N
      A=A*X2/FDO
 33   S=S+A
 34   XPROB=Z*S/C
      GO TO 101
C
C                                   NDF ODD, LESS THAN 30, EQ. 2626.4.4
 41   S=0.0
      X=SQRT (X2)
      A=1.0/X
      L=(NDF-1)/2
      IF(L) 44,44,42
 42   DO 43 N=1,L
      FDO=2*N-1
      A=A*X2/FDO
 43   S=S+A
 44   XPROB=2.0*(QPROBF(TPROBF(X))+Z*S)
      GO TO 101
C
C                                   NDF 30 OR GREATER, EQ. 26.4.14
 51   V=2.0/9.0/DF
      X=((X2/DF)**.333333333-1.0+V)/SQRT(V)
      IF(X-12.0) 52,81,81
 52   IF(X+12.0) 86,53,53
 53   Z=ZPROBF(X**2)
      XPROB=QPROBF(TPROBF(ABS (X)))
      IF(X) 54,101,101
 54   XPROB=1.0-XPROB
      GO TO 101
C
 81   XPROB=0.0
      GO TO 101
 86   XPROB=1.0
      GO TO 101
 91   XPROB=-1.0
C
 101  XPROB=XPROB*100.0
C
      RETURN
      END
```

```
      SUBROUTINE MINV20(A,N,DETERM)
C..THE INVERTED MATRIX ELEMENTS ARE STORED IN A(20,20)
      DIMENSION IPIVOT(20),A(20,20),INDEX(20,2),PIVOT(20)
C...INITIALIZATION
      DETERM=1.
      DO 20 J=1,N
   20 IPIVOT(J)=0
      DO 550 I=1,N
C...SEARCH FOR PIVOT ELEMENTS
      AMAX=0.
      DO 105 J=1,N
      IF (IPIVOT(J).EQ.1) GO TO 105
      DO 100 K=1,N
      IF (IPIVOT(K)-1) 80,100,740
   80 IF (ABS(AMAX).GE.ABS(A(J,K))) GO TO 100
      IROW=J
      ICOLUM=K
      AMAX=A(J,K)
  100 CONTINUE
  105 CONTINUE
      IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1
C...INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
      IF (IROW.EQ.ICOLUM) GO TO 260
      DETERM=-DETERM
      DO 200 L=1,N
      SWAP=A(IROW,L)
      A(IROW,L)=A(ICOLUM,L)
  200 A(ICOLUM,L)=SWAP
  260 INDEX(I,1)=IROW
      INDEX(I,2)=ICOLUM
      PIVOT(I)=AMAX
      DETERM=DETERM*PIVOT(I)
      IF (DETERM.EQ.0.) RETURN
C...DIVIDE PIVOT ROW BY PIVOT ELEMENT
      A(ICOLUM,ICOLUM)=1.
      DO 350 L=1,N
  350 A(ICOLUM,L)=A(ICOLUM,L)/PIVOT(I)
C...REDUCE NON-PIVOT ROWS
      DO 550 L1=1,N
      IF (L1.EQ.ICOLUM) GO TO 550
      SWAP=A(L1,ICOLUM)
      A(L1,ICOLUM)=0.
      DO 450 L=1,N
  450 A(L1,L)=A(L1,L)-A(ICOLUM,L)*SWAP
  550 CONTINUE
C...INTERCHANGE COLUMNS
      DO 710 I=1,N
      L=N+1-I
      IF (INDEX(L,1).EQ.INDEX(L,2)) GO TO 710
      IROW=INDEX(L,1)
      ICOLUM=INDEX(L,2)
      DO 705 K=1,N
      SWAP=A(K,IROW)
      A(K,IROW)=A(K,ICOLUM)
  705 A(K,ICOLUM)=SWAP
  710 CONTINUE
  740 RETURN
      END
```

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

PLOT4, the Line-Pointer-Plotting Subroutine Called by X2MIN

As now used, X2MIN calls a line-printer-plotting subroutine PLOT4 which has additional entry points PLOTWD, PLOT3 and PLOT3L. Since PLOT4 forms a plot image by storing one character in each byte (8 bits) of a 4-byte word, the subroutine is specific to the IBM 360/370 series computer we used; for this reason we do not provide here the source listing for PLOT4. However, a brief description follows for those users who would wish to incorporate line printer plots at their own computer. (The alternative to this would be to delete all statements in X2MIN which call PLOT4, etc.; that is, delete all statements between 209 and 579 in the X2MIN listing. Statement 209 must be left in, 579 must be removed, and the "go to 6" statement immediately following 579 will then appear immediately after statement 209 to provide the necessary transfer of control.)

The size of the plot is set up automatically at the first call of PLOT3, additional calls to PLOT3 or PLOT3L provide additional data to the plot, and the call to PLOT4 prints out the accumulated plot image and resets it. The automatic setup provides that all the points from the first PLOT3 call are included within the axis limits. The automatic setup sets 10 print spaces per grid line in the x-dimension and 5 print spaces per grid line in the y-dimension with the grid lines adjusted to give round or "nice" numbers and hence the points plotted won't in general extend to the edges of the plot space. The maximum grid widths allowed for NGX (the number of X-grid lines) and NGY (Y-grid) are both preset to 12, as appropriate to a line printer, but a call to PLOTWD(MAXGX,MAXGY) will change these limits to MAXGX (maximum NGX) and MAXGY (maximum NGY). Reducing MAXGX to 7 permits output plotting on the 80-character line of Teletype-like devices.

In X2MIN, the labelled common area COM1 provides ten flags, INFLAG(10), for general setup; the first three of these are used to set the type of output printer plots and the plot size. INFLAG (1) $\geq$ 3 produces printer plots of both of the fitted function with experimental points superimposed and of the deviations between experimental and fitted points, INFLAG(1) = 2 provides only the plot of the deviations, INFLAG(1) = 1 plots only the fitted function with experimental data, and INFLAG(1) $\leq$ 0 suppresses all plots. If INFLAG(2)

≠ 0, INFLAG(2) and INFLAG(3) carry new values for MAXGX and MAXGY in a PLOTWD(MAXGX,MAXGY) call; otherwise MAXGX and MAXGY are both 12.

PLOT3(C,X,Y,N) plots the character C at the N points X,Y(X and Y are each one-dimensional arrays); the plot character C replaces whatever was previously in the plot image.

PLOT3L(C,XX,YY,N) is called with either 0 or 1; if N=1, a line is drawn (using character C) to the single point XX,YY from the point input of the previous PLOT3L call, provided that N=1 in the previous call also. The first PLOT3L call with N=1 provides no output, and a N=0 call cuts the string to permit a new "first" N=1 call.

PLOT4(LX,XLAB,LY,YLAB,LT,TITLE) prints out the accumulated plot image and resets it. XLAB, YLAB, and TITLE are character strings for the x and y axes and the title, with LX,LY, and LT being the string lengths.

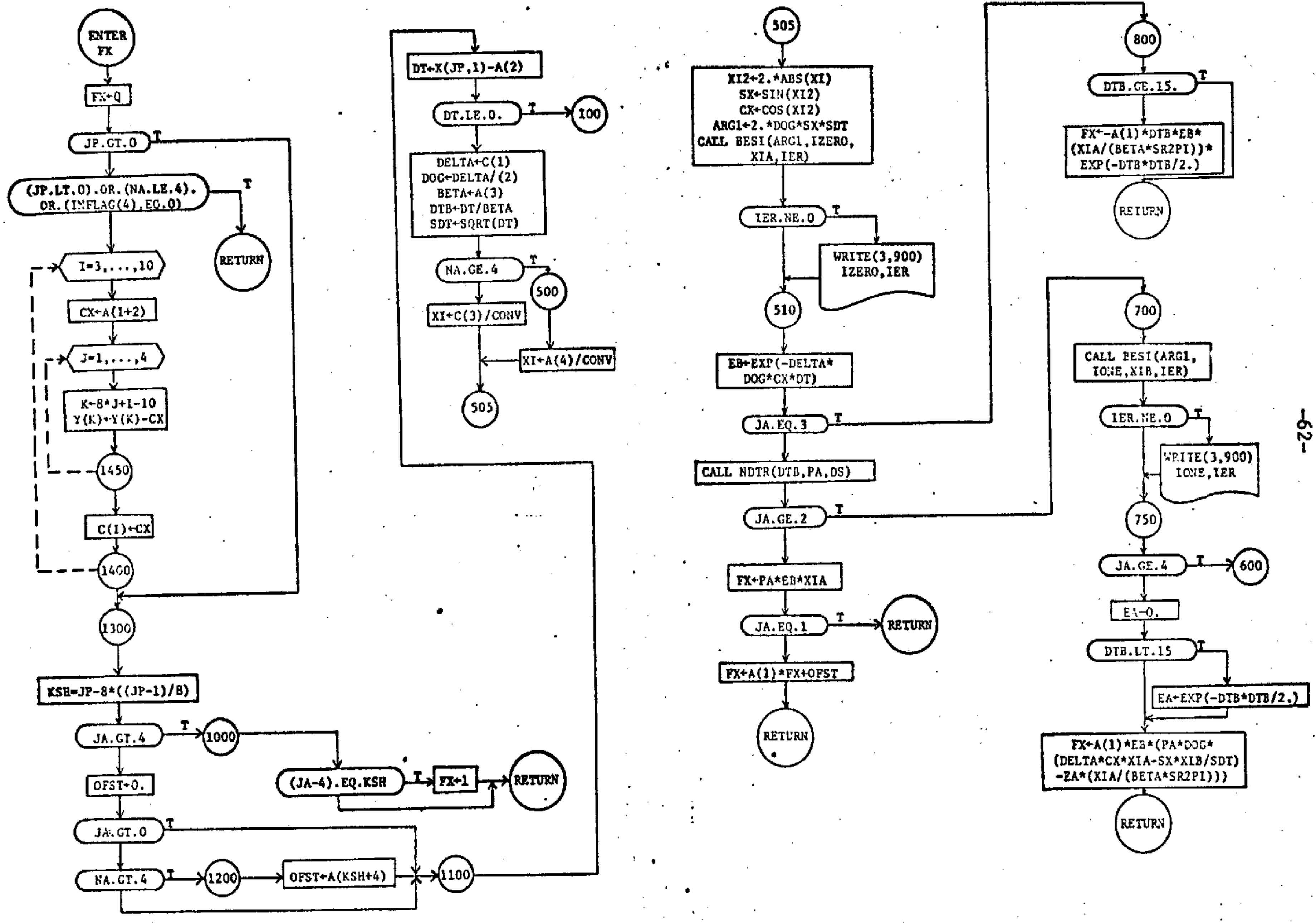
APPENDIX B. SUBROUTINE FX.

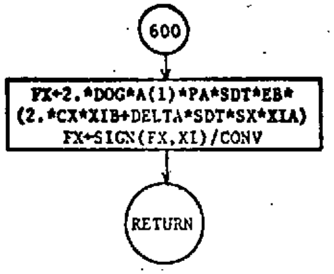
The following pages provide the source listing and the flow diagram for subroutine FX which supplies the lineshape function and its derivatives with respect to each of the 12 lineshape parameters; the mathematical description of this FX has already been given in Section II. FX is for use in, and follows the rules established by, the general line-fitting subroutine X2MIN as described in Appendix A.

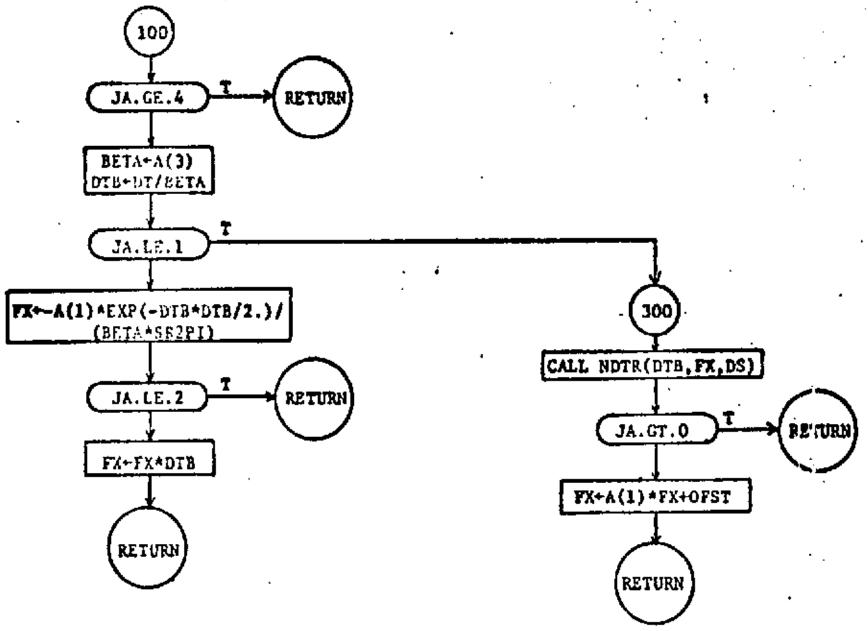
Because the Bessel Functions, I_o and I_1, and the probability function are needed in FX, listings are provided for the Bessel function subroutine BESI and the probability function subroutine NDTR. Both BESI and NDTR are taken directly from the IBM Scientific Subroutine Package and are described more fully there.

```
      FUNCTION FX(JP,JA)
      COMMON X,Y,A,NA,NX,NC,C
      COMMON/COM1/INFLAG(10)
      DIMENSION X(100,10),Y(100),A(20),C(20)
      DATA SP2PI/2.506628/,CONV/57.295780/,IZERO/0/,IONE/1/
      FX=0.
      IF (JP.GT.0) GO TO 1300
      IF ((JP.LT.0).OR.(NA.LE.4).OR.(INFLAG(4).EQ.0)) RETURN
      DO 1400 I=3,10
      CX=A(I+2)
      DO 1450 J=1,4
      K=8*J+I-10
 1450 Y(K)=Y(K)*CX
 1400 C(I)=CX
 1300 KSH=JP-8*((JP-1)/8)
      IF (JA.GT.4) GO TO 1000
      OFST=0.
      IF (JA.GT.0) GO TO 1100
      IF (NA.GT.4) GO TO 1200
      GO TO 1100
 1000 IF ((JA-4).EQ.KSH) FX=1.
      RETURN
C:: RETURN POINT FOR OFFSET DERIVATIVES...
 1200 OFST=A(KSH+4)
 1100 DT=X(JP,1)-A(2)
      IF (DT.LE.0.) GO TO 100
      DELTA=C(1)
      DOG=DELTA/C(2)
      BETA=A(3)
      DTB=DT/BETA
      SDT=SQRT(DT)
      IF (NA.GE.4) GO TO 500
      XI=C(3)/CONV
      GO TO 505
  500 XI=A(4)/CONV
  505 XI2=2.*ABS(XI)
      SX=SIN(XI2)
      CX=COS(XI2)
      ARG1=2.*DOG*SX*SDT
      CALL BESI(ARG1,IZERO,XIA,IER)
C---BESI=BESSEL FCN.,IBM SSP III; SEE PROGRAMMER'S MANUAL PAGE 365
      IF (IER.NE.0) WRITE (3,900) IZERO,IER
  900 FORMAT(' BESI',I3,' ,IER=',I3)
  510 EB=EXP(-DELTA*DOG*CX*DT)
      IF (JA.EQ.0) GO TO 800
      CALL NDTR(DTB,PA,DS)
C--- NDTR,FCN. FROM IBM SSP III; SEE PROGRAMMER'S MANUAL PAGE 78
      IF (JA.GE.2) GO TO 700
      FX=PA*EB*XIA
      IF (JA.EQ.1) RETURN
C:: JA=1,DT.GT.0.
      FX=A(1)*FX+OFST
      RETURN
C:: JA=0,DT.GT.0.
  400 IF (DTB.GE.15.) RETURN
      FX=-A(1)*DTB*EB*(XIA/(BETA*SP2PI))*EXP(-DTB*DTB/2.)
      RETURN
C:: JA=3,DT.GT.0.
  700 CALL BESI(ARG1,IONE,XIB,IER)
      IF (IER.NE.0) WRITE (3,900) IONE,IER
  750 IF (JA.GE.4) GO TO 600
      PA=0.
      IF (DTB.LT.15.) PA=EXP(-DTB*DTB/2.)
      FX=A(1)*EB*(PA*DOG*(DELTA*CX*XIA-SX*XIB/SDT)
     1 -PA*(XIA/(BETA*SP2PI)))
      RETURN
C:: JA=2,DT.GT.0.
  600 FX=2.*DOG*A(1)*PA*SDT*EB*(2.*CX*XIB+DELTA*SDT*SX*XIA)
      FX=SIGN(FX,XI)/CONV
      RETURN
C:: JA=4,DT.GT.0.
  100 IF (JA.GE.4) RETURN
      BETA=A(3)
      DTB=DT/BETA
      IF (JA.LE.1) GO TO 300
      FX=-A(1)*EXP(-DTB*DTB/2.)/(BETA*SP2PI)
      IF (JA.LE.2) RETURN
      FX=FX*DTB
      RETURN
  300 CALL NDTR(DTB,FX,DS)
      IF (JA.GT.0) RETURN
      FX=A(1)*FX+OFST
      RETURN
      END
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR


ENTER FX
FX←0
JP.GT.0
T
(JP.LT.0).OR.(NA.LE.4).OR.(INFLAG(4).EQ.0)
T
RETURN
I=3,...,10
CX←A(I+2)
J=1,...,4
K←8*J+I-10
Y(K)←Y(K)-CX
1450
C(I)←CX
1400
1300
KSH=JP-8*((JP-1)/8)
JA.GT.4
T
1000
(JA-4).EQ.KSH
T
FX←1
RETURN
OFST←0.
JA.GT.0
T
NA.GT.4
T
1200
OFST←A(KSH+4)
1100
DT←X(JP,1)-A(2)
DT.LE.0.
T
100
DELTA←C(1)
DOG←DELTA/(2)
BETA←A(3)
DTB←DT/BETA
SDT←SQRT(DT)
NA.GE.4
T
500
XI←C(3)/CONV
XI←A(4)/CONV
505
505
XI2←2.*ABS(XI)
SX←SIN(XI2)
CX←COS(XI2)
ARG1←2.*DOG*SX*SDT
CALL BESI(ARG1,IZERO,XIA,IER)
IER.NE.0
T
WRITE(3,900) IZERO,IER
510
EB←EXP(-DELTA*DOG*CX*DT)
JA.EQ.3
T
CALL NDTR(DTB,PA,DS)
JA.GE.2
T
FX←PA*EB*XIA
JA.EQ.1
T
RETURN
FX←A(1)*FX+OFST
RETURN
800
DTB.GE.15.
T
FX←-A(1)*DTB*EB*(XIA/(BETA*SR2PI))*EXP(-DTB*DTB/2.)
RETURN
700
CALL BESI(ARG1,IONE,XIB,IER)
IER.NE.0
T
WRITE(3,900) IONE,IER
750
JA.GE.4
T
600
EA←0.
DTB.LT.15
T
EA←EXP(-DTB*DTB/2.)
FX←A(1)*EB*(PA*DOG*(DELTA*CX*XIA-SX*XIB/SDT)-EA*(XIA/(BETA*SR2PI)))
RETURN


600
FX←2.*DOG*A(1)*PA*SDT*EB*
(2.*CX*XIB+DELTA*SDT*SX*XIA)
FX←SIGN(FX,XI)/CONV
RETURN


100
JA.GE.4
T
RETURN
BETA←A(3)
DTB←DT/BETA
JA.LE.1
T
FX←-A(1)*EXP(-DTB*DTB/2.)/
(BETA*SR2PI)
JA.LE.2
T
RETURN
FX←FX*DTB
RETURN
300
CALL NDTR(DTB,FX,DS)
JA.GT.0
T
RETURN
FX←A(1)*FX+OFST
RETURN

```
      SUBROUTINE BESI(X,N,BI,IER)
      DATA PI/3.141592653/
C----SEE PG.365,PROGRAMMER'S MANUAL,IBM SSP III....
      IER=0
      BI=1.
      IF (N) 150,15,10
   15 IF (X) 160,17,20
   17 RETURN
   10 IF (X.LT.0.) GO TO 160
   20 TOL=1.E-06
      IF (X.LE.12.) GO TO 40
   30 IF (X.GT.FLOAT(N)) GO TO 110
   40 XX=X/2.
   50 TERM=1.
      IF (N.LE.0) GO TO 70
   55 DO 60 I=1,N
      FI=I
      IF (ABS(TERM).GT.1.E-68) GO TO 60
   56 IER=3
      BI=0.
      RETURN
   60 TERM=TERM*XX/FI
   70 BI=TERM
      XX=XX*XX
      DO 90 K=1,1000
      IF (ABS(TERM).LE.ABS(BI*TOL)) RETURN
   80 FK=K*(N+K)
      TERM=TERM*(XX/FK)
   90 BI=BI+TERM
  100 RETURN
  110 FN=4*N*N
      IF (X.LT.170.) GO TO 115
  111 IER=4
      RETURN
  115 XX=1./(8.*X)
      TERM=1.
      BI=1.
      DO 130 K=1,30
      IF (ABS(TERM).LE.ABS(TOL*BI)) GO TO 140
  120 FK=(2*K-1)**2
      TERM=TERM*XX*(FK-FN)/FLOAT(K)
  130 BI=BI+TERM
      GO TO 40
  140 BI=BI*EXP(X)/SQRT(2.*PI*X)
      RETURN
  150 IER=1
      RETURN
  160 IER=2
      RETURN
      END
```

```
      SUBROUTINE NDTR(X,P,D)
C---SEE PG.78,PROGRAMMER'S MANUAL,IBM SSP III....
      AX=ABS(X)
      IF (AX.LT.10.) GO TO 50
      P=1.
      GO TO 100
   50 T=1./(1.+.2316419*AX)
      D=.3989423*EXP(-X*X/2.)
      P=1.-D*T*((((1.330274*T-1.821256)*T+1.781478)*T
     1  -.3565638)*T+.3193815)
  100 IF (X.LT.0.) P=1.-P
      RETURN
      END
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

APPENDIX C. SUBROUTINE HIST

The following several pages provide the description of the subroutine HIST (with entry points HISTI and HISTO) which is intended for use in obtaining the probability density function at one or several points on the mean return waveform. A brief description of how to use HIST for this has already been provided at the end of Section II of this report.

Subroutine Name and Argument List

```
Subroutine HISTI(XBAR,SIGMA,NSIG,NB,WB,XL,JH,NHE,SUMS)
Entry HIST(XI,WB,XL,JH,NHE,SUMS)
Entry HISTO(NB,WB,WL,JH,NHE,SUMS)
```

Description

Subroutine HISTI, with entry points HIST, HISTO, is to allow the histogramming of a number of observations of the variable XI. A call to HISTI initializes the subroutine and calculates the width and location of individual histogram bins. Calls to HIST generate the histogram desired, and a final call to HISTO causes print-out of the results; the call to HISTO does not disturb the histogramming storage locations so that one may display an intermediate result by HISTO and then continue to build up the histogram by more calls to HIST. All variables used are listed in calling lists so that it is possible to have several histograms being built up at the same time.

For present subroutine dimensions, a maximum of 100 histogram storage locations is allowed; these dimensions are easily increased if desired.

Subroutine Arguments (and Dimensions) in Calling List

XBAR SIGMA NSIG NB	These are input arguments to establish the range which the histogramming will cover uniformly with a total of NB histogram bins. XBAR is the value at which the histogramming routine is centered, and (NSIG*SIGMA) is the width of the total range to be histogrammed. Input data lying outside the range XBAR±(NSIG*SIGMA) will be entered either in bin #1 or bin #NB, depending on whether the input is below or above the range indicated.
WB XL	These values are calculated in HISTI, given the inputs XBAR, SIGMA, NSIG, and NB. WB is the width of an individual bin, and XL is used in index computation in HIST.
JH(100)	Integer array JH is the histogramming "count" storage.

NHE = Counter recording total number of histogram counts in the locations of JH(100).

SUMS(2) = Two double-precision variables used for estimating an overall mean and standard deviation for the histogrammed variable XI; SUMS(1) is a running subtotal of individual input values XI, and SUMS(2) stores the subtotal of the squares of XI.

XI = Individual input value of the variable to be histogrammed.

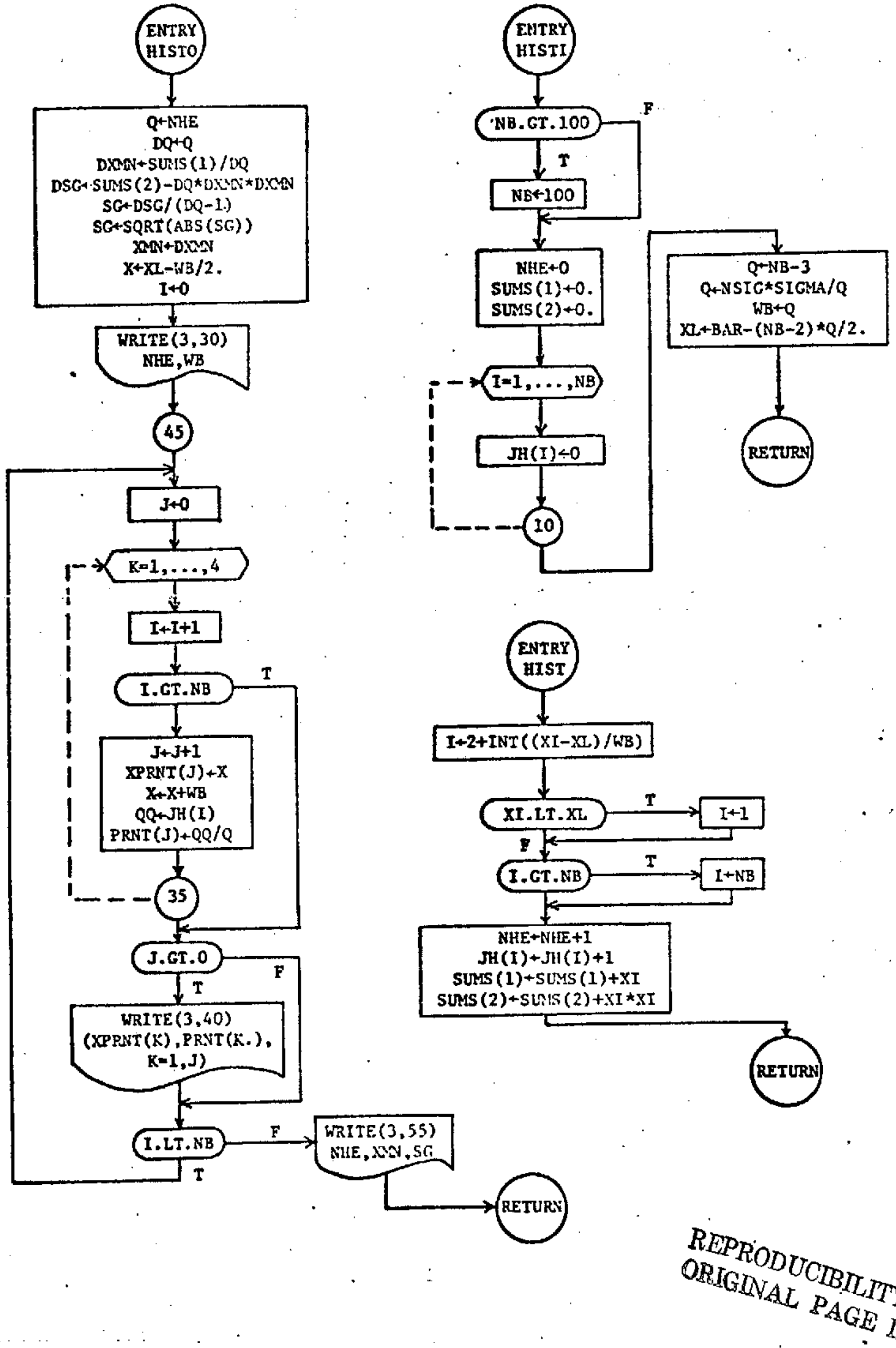
Input/Output Performed by Subroutine

No input except through calling list. HISTO writes (on unit 3) the probability density function estimates obtained from the histogram, together with estimates of the mean and standard deviation. The pd.f. estimates may be converted to total histogram-bin counts, if desired, by multiplying all pd.f. estimates by NHE, the total number of histogram entries.

Program Listing for HISTI, HIST, and HISTO

```
      SUBROUTINE HISTI(XBAR,SIGMA,NSIG,NB, WB,XL, JH,NHE,SUMS)
      DIMENSION JH(100),XPRNT(10),PPRNT(10),SUMS(2)
      DOUBLE PRECISION SUMS,DXMN,DSG,DQ
      IF (NB.GT.100) NB=100
      NHE=0
      SUMS(1)=0.D0
      SUMS(2)=0.D0
      DO 10 I=1,NB
   10 JH(I)=0
      Q=NB-3
      Q=NSIG*SIGMA/Q
      WB=Q
      XL=XBAR-(NB-2)*Q/2.
      RETURN
      ENTRY HIST(XI, WB,XL, JH,NHE,SUMS)
      I=2+INT((XI-XL)/WB)
      IF (XI.LT.XL) I=1
      IF (I.GT.NB) I=NB
   20 NHE=NHE+1
      JH(I)=JH(I)+1
      SUMS(1)=SUMS(1)+XI
      SUMS(2)=SUMS(2)+XI*XI
      RETURN
      ENTRY HISTO(NB, WB,XL, JH,NHE,SUMS)
      Q=NHE
      DQ=Q
      DXMN=SUMS(1)/DQ
      DSG=SUMS(2)-DQ*DXMN*DXMN
      SG=DSG/(DQ-1.D0)
      SG=SQRT(ABS(SG))
      XMN=DXMN
      X=XL-WB/2.
      I=0
      WRITE (3,30) NHE,WB
   30 FORMAT(/' FOLLOWING IS HISTOGRAM-DERIVED PDF FOR',I5,
     1' HISTOGRAM ENTRIES,'/' IN FORM (CENTER,PDF).  BIN WIDTH=',E13.6)
   45 J=0
      DO 35 K=1,4
      I=I+1
      IF (I.GT.NB) GO TO 50
      J=J+1
      XPRNT(J)=X
      X=X+WB
      QQ=JH(I)
   35 PPRNT(J)=QQ/Q
   50 IF (J.GT.0) WRITE (3,40) (XPRNT(K),PPRNT(K), K=1,J)
   40 FORMAT(' ',4('(',F8.3,',',F7.5,') '))
      IF (I.LT.NB) GO TO 45
      WRITE (3,55) NHE,XMN,SG
   55 FORMAT(' FOR ABOVE',I5,' ENTRIES,MEAN=',E13.6,', AND STD DEV=',
     1 E13.6/)
      RETURN
      END
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR



REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

APPENDIX D. PROGRAMS FOR TIME-REALIGNMENT AND AVERAGING.

The various program descriptions in Appendix D carry out details of the time-realignment process as described in Section III of the report. Our main program to do this is (arbitrarily) divided into two major subroutines, PART1 and PART2. PART1 determines the instantaneous tracker error estimates at the 104 times per frame for the S&H data, and PART2 is intended to make use of those time-error estimates.

For now, PART2 only prints out the 104 per frame time corrections. The phrase "vertical addition" has been used in the past to denote the summing into time bins of the properly time-realigned sets of S&H data, and a subroutine VTADD (with entry points VTADI and VTADO) is provided to accomplish this. If the waveform computations were all to be done here, we would modify PART2 to incorporate VTADD, but this becomes intimately involved with specific details of where the S&H data are stored and how to access those data. Instead, the work has been continued at Wallops Flight Center where a single main program now incorporates the two programs PART1 and PART2 plus the setup and calling details to use VTADD. This present report documents PART1 and PART2 as of approximately February 1974; since then, the development and modification work has been done at Wallops. The present Wallops programs differ only slightly however from the material presented in this Appendix.

The organization of the rest of Appendix D is as follows: program description, list, and flow chart for PART2; then description, list, and flow for POLRG, the polynomial regression routine used by PART1; source listings only for GDATA, MINV, MULTR, and ORDER which are needed by POLRG; a description, listing, and flow chart for SPLINE, the third-degree spline interpolation used by PART1; description, listing, and flow chart for WEIGHT and XCNVLV; and finally the general description, listing, and flow chart for VTADD, together with a note on timing details. Notice that while PART1 requires a subroutine FDATA to obtain the altitude data upon which PART1 will operate, we supply no details of FDATA. This is because FDATA again is specifically dependant upon how the altitude data is stored or how it is available to the program. It is important to notice that the altitudes returned by FDATA are to be in nanoseconds (i.e., the two-way ranging time).

Subroutine Name and Argument List

Subroutine PART1(PR)

Description

PART1 is the first of a pair of subroutines (PART2, called from PART1, is the other) performing the individual sample-and-hold (S&H) tracker error calculation as of February 1974. The 104 per frame tracker errors, in nanoseconds, are printed out; the "vertical averaging" of the individual sets of 8 S&H results has not been implemented locally (this has been accomplished however at NASA/Wallops, and involves relatively few changes and additions within PART2).

PART1 reads general problem parameters and then acquires altitude data, an entire submode at a time, from FDATA. A least-squares, polynomial regression analysis determines a set of coefficients COE which characterize the fitted altitude over the submode. These coefficients COE and the 8 per frame altitude averages are used to derive average altitude residuals (8 per frame); a weighting function is then used to estimate 8 per frame instantaneous tracker errors which are used to set up a spline function. PART2 then will use the spline in determining the 104 per frame tracker errors. The weighting function is based on the work described in the "Task A Final Report", (Contract NAS6-2307, Applied Science Associates, approx. July 1973).

Subroutine Arguments (and Dimensions) in Calling List

PR(20) = One line of problem identification to be printed out by a 20A4 Format.

Input/Output Performed by Subroutine

A number of problem parameters are read, and a general problem heading is written each time through the subroutine. The principal input to PART1 comes through the call to FDATA which provides the altitude data input for the entire problem. The quantities read in directly in PART1 are listed below:

ML = Lower limit to the degree of fit to be employed. POLRG will perform a regression analysis for all degrees between ML and M, starting at ML and incrementing up to M (unless there is no improvement in the sum of errors squared for a degree less than M in which case the analysis stops at that point).

M = Maximum degree for the least-squares polynomial regression fit to be performed by POLRG, called by PART1. (Reading M=0 will retain the already-used values for M,ML,NPLOT,NTABXY, and NTEST.)

NPLOT = Integer describing how the results of POLRG are to be printed out, according to the rules established by POLRG:

NPLOT = 5 - Print a table of data input, fitted value, and residuals for each separate degree of the regression,

NPLOT = 4 - Same as NPLOT=5 but in addition call a PLOT subroutine at each separate degree,

NPLOT = 3 - Print the table of values and residuals only for the final degree fitted,

NPLOT = 2 - Same as NPLOT=3 but in addition call PLOT at the final degree,

NPLOT = 1 - No table of residuals, no plot.

NTABXY = Integer determining whether to print out table of weighted vs. unweighted tracker error estimates, 8 per frame, in PART1A:

NTABXY = 0 - No printout
NTABXY ≠ 0 - Print table

NTEST = Integer setting how much of the entire problem is done:

NTEST = 2 - Go from start through POLRG only
NTEST = 1 - Go through the weighting process (which

estimates instantaneous tracker, given the 8 per frame average tracker errors)

NTEST = 0 - Do entire problem including determining and printing out the 104 per frame tracker errors (in nanoseconds).

Other Important Subroutine Variables (and Dimensions)

X(2200) N	X contains the one-per-frame averages of the 8 per frame altitudes and also contains the times corresponding to these frame-averages. The rules for location of these input data to POLRG are set by the requirements of GDATA called by POLRG: FDATA puts the data for an entire submode into the required form. N is the number of such one-frame altitude averages. Once the coefficients COE have been determined X becomes unnecessary, and it is used as temporary storage for the option NTABXY≠0.
T(1600) Y(1600) NY	FDATA also returns the 8 per frame altitudes in array Y with the corresponding times in array T. NY is the total number of such points, and will be equal to the number of frames times 8.
COE(11)	This array contains the coefficients determined by the least-squares polynomial regression; hence, all the smoothed altitude history for the entire submode is contained in COE upon return from POLRG.

Other Subroutines Called

FDATA(M,N,X,T,Y,NY) - Obtains the altitude data, an entire submode at a time. M is necessary in the argument list in order that FDATA stores the one-per-frame altitude averages in X.

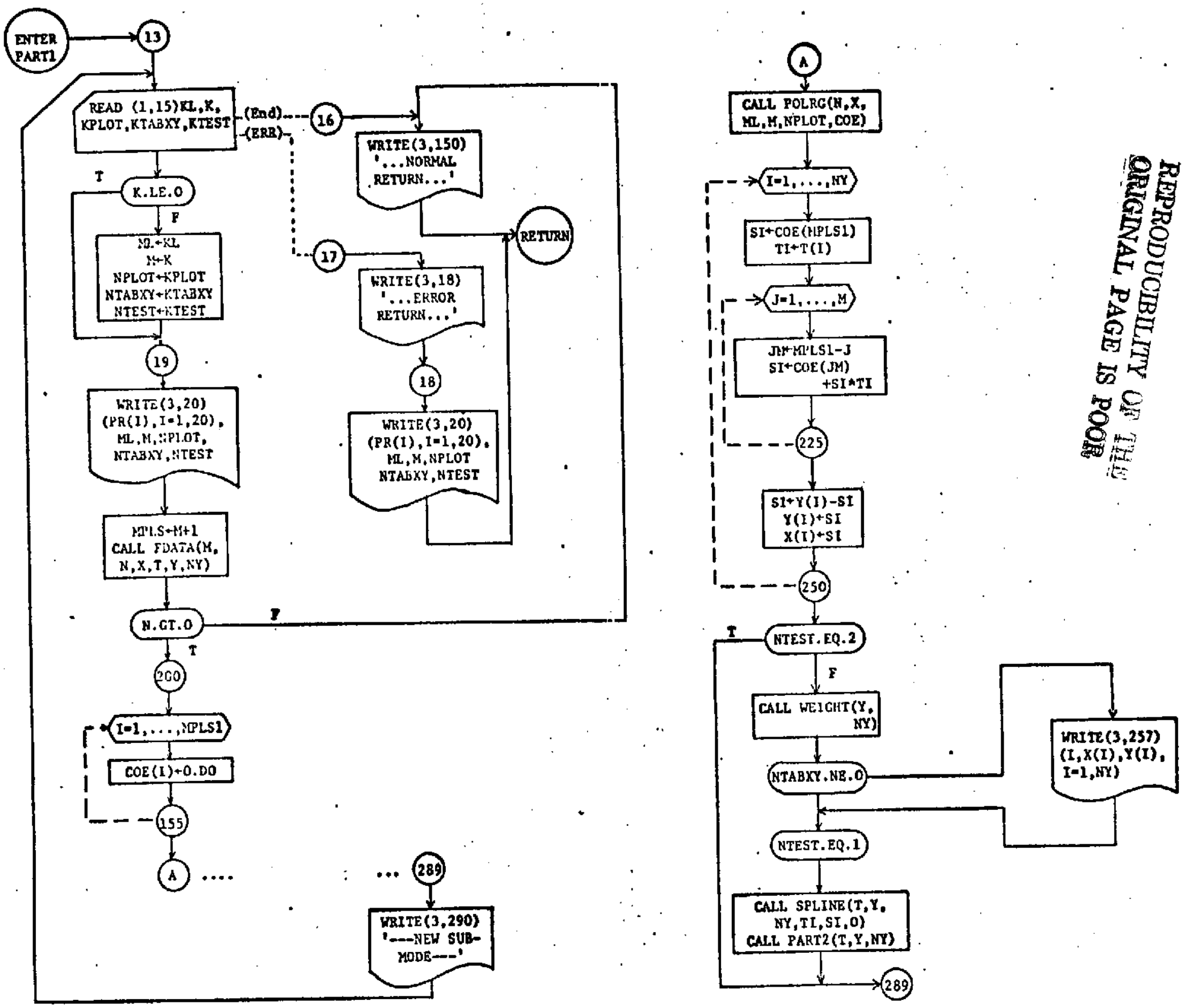
POLRG(N,X,ML,M,NPLOT,COE) - Performs the least-squares polynomial regression analysis on the frame-average altitudes and times in X, to return the polynomial coefficients COE.

WEIGHT(Y,NY) - Performs the weighting function necessary to convert average tracker errors to estimate instantaneous tracker errors. The averages in Y are converted to instantaneous estimates in Y upon the return from WEIGHT. (Since the Y contents are changed in value by calling WEIGHT, the Y values are copied over into X before calling WEIGHT, in case a "before-and-after" printout is desired as signalled by NCADXY).

SPLINE(T,Y,NY,TI,SI,0) - Calling SPLINE with the last argument zero uses T,Y,and NY as the input to set up spline-fit coefficients within subroutine SPLINE; subsequent calls with the last argument equal 1 will return in SI the spline result corresponding to an input time TI. TI and SI are unused arguments in the spline set-up step.

```
      SUBROUTINE PART1(PP)
      DIMENSION X(1100),T(200),Y(200),COE(11),PP(20)
      DOUBLE PRECISION X,T,Y,COE,TI,SI,SGLIM
      DATA ML/3/,M/4/,NPLOT/0/,NTABXY/0/,NTEST/0/,SGLIM/20.D0/
   13 READ (1,15,END=16,ERR=17) KL,K,KPLOT,KTABXY,KTEST
   15 FORMAT(20I3)
      IF (K.LE.0) GO TO 19
      ML=KL
      M=K
      NPLOT=KPLOT
      NTABXY=KTABXY
      NTEST=KTEST
   19 WRITE (3,20) (PP(I), I=1,20),ML,M,NPLOT,NTABXY,NTEST
   20 FORMAT(//' ',20A4/' PROBLEM PARAMETERS: ML=',I4,',M=',I4,
     1 ',NPLOT=',I4,',NTABXY=',I4,', AND NTEST=',I4//)
      MPLS1=M+1
  100 CALL FDATA(M,N,X,T,Y,NY,SGLIM)
      IF (N.GT.0) GO TO 200
   16 WRITE (3,150)
  150 FORMAT(//' ... NORMAL RETURN TO MAIN PROG...'//)
      RETURN
   17 WRITE (3,18)
   18 FORMAT(//' ...ERROR RETURN TO MAIN PROG...'//)
      WRITE (3,20) (PP(I), I=1,20),ML,M,NPLOT,NTABXY,NTEST
      RETURN
  200 DO 155 I=1,MPLS1
  155 COE(I)=0.D0
      CALL POLRG(N,X,ML,M,NPLOT,COE)
      DO 250 I=1,NY
      SI=COE(MPLS1)
      TI=T(I)
      DO 225 J=1,M
      JM=MPLS1-J
  225 SI=COE(JM)+SI*TI
      SI=Y(I)-SI
      Y(I)=SI
  250 X(I)=SI
      IF (NTEST.EQ.2) GO TO 289
      CALL WEIGHT(Y,NY)
      IF (NTABXY.NE.0) WRITE (3,257) (I,X(I),Y(I), I=1,NY)
  257 FORMAT(//' FOLLOWING ARE I:X(I):Y(I) - - -'/
     1 (' ',2(I5,':',D13.6,':',D13.6,2X)))
      IF (NTEST.EQ.1) GO TO 289
      WRITE (3,288)
  288 FORMAT(//' (SET UP SPLINE...)')
      CALL SPLINE(T,Y,NY,TI,SI,0)
      CALL PART2(T,Y,NY)
  289 WRITE (3,290)
  290 FORMAT(//' GO TO FDATA FOR NEW SUBMODE'/)
      GO TO 13
      END
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR



REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

Subroutine Name and Argument List

Subroutine PART2(T,Y,NY)

Description

PART2 continues from PART1 the individual S&H tracker error calculation (104 per frame) as of Feb. 1974. PART2 uses the spline coefficients set up in PART1; for now, the 104 per frame tracker errors are just printed out, but eventually the vertical averaging will have to be performed over a frame of data. This will necessitate modifying PART2. (Such modifications are being carried out at NASA/Wallops.)

Subroutine Arguments (and Dimensions) in the Calling List

Y(1600) = The 8 per frame instantaneous tracker error estimates, are used to set up SPLINE. Y is used in the computations within SPLINE.

T(1600) = The times corresponding to the Y data.

NY = Number of elements in Y,T; NY is 8 times the number of data frames.

Input/Output Performed by Subroutine

The 104 S&H tracker error corrections per frame are printed out as are the 8 per frame values of Y, the tracker error input data to SPLINE. The time at the start of each frame is also printed out.

Other Subroutines Called

SPLINE(T,Y,NY,TI,SI,1) - With the last argument in the list non-zero, SPLINE returns in SI the spline-determined tracker error corresponding to input time TI.

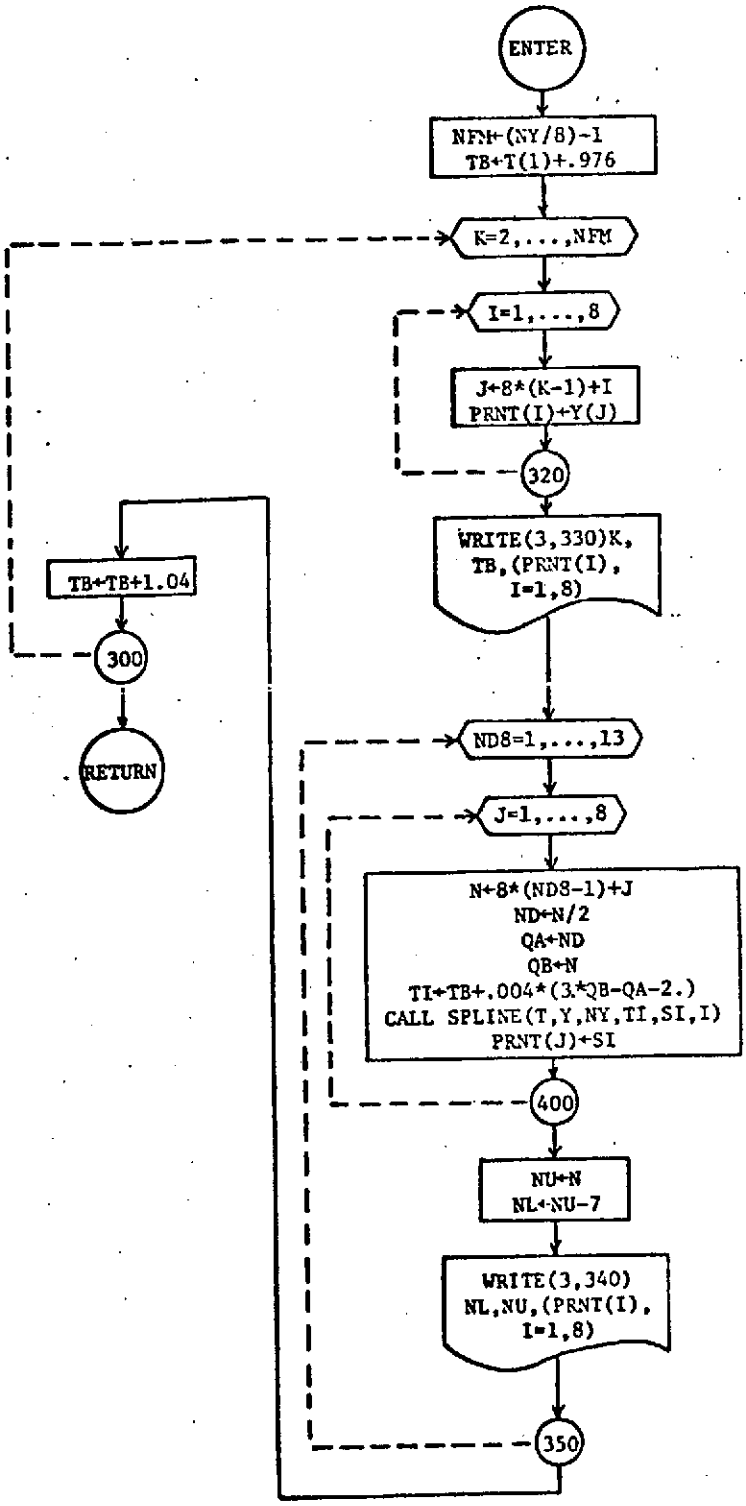
Other Important Subroutine Variables (and Dimensions)

PRNT(8) = An array used to store 8 successive values of the tracker error in order to print one line. There will be 13 such lines for each frame, since (13 lines/frame) times (8 values/line) equals 104 values/frame.

N = Integer from 1 to 104 specifying which of the 104 S&H times in a given frame is to be found. Notice that the calculation of TI in PART2 makes specific use of the truncation which occurs in integer division in FORTRAN.

```
      SUBROUTINE PART2(T,Y,NY)
      DIMENSION T(800),Y(800),PRNT(20)
      DOUBLE PRECISION T,Y,TI,SI,PRNT,TB,QA,QB
      NPM=(NY/8)-1
      TB=T(1)+0.976D0
      DO 300 K=2,NPM
      DO 320 I=1,8
      J=8*(K-1)+I
  320 PRNT(I)=Y(J)
      WRITE (3,330) K,TB,(PRNT(I), I=1,8)
  330 FORMAT(/' FRAME #',I3,',START-FRAME T=',F8.3,';LEFT CCL=',
     1 'SGH #, 8 WEIGHTED (ALT-FIT):'/8(1X,F7.2))
      DO 350 NDB=1,13
      DO 400 J=1,8
      N=(NDB-1)*8+J
      ND=N/2
      QA=ND
      QB=N
      TI=TB+.004D0*(QB*3.D0-QA-2.D0)
      CALL SPLINE(T,Y,NY,TI,SI,1)
  400 PRNT(J)=SI
      NU=N
      NL=NU-7
      WRITE (3,340) NL,NU,(PRNT(I), I=1,8)
  340 FORMAT(' ',I3,',',I4,8(1X,F7.2))
  350 CONTINUE
  300 TB=TB+1.04D0
      RETURN
      END
```

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR


ENTER
NFM←(NY/8)-1
TB←T(1)+.976
K=2,...,NFM
I=1,...,8
J←8*(K-1)+I
PRNT(I)←Y(J)
320
WRITE(3,330)K,
TB,(PRNT(I),
I=1,8)
ND8=1,...,13
J=1,...,8
N←8*(ND8-1)+J
ND←N/2
QA←ND
QB←N
TI←TB+.004*(3.*QB-QA-2.)
CALL SPLINE(T,Y,NY,TI,SI,I)
PRNT(J)←SI
400
NU←N
NL←NU-7
WRITE(3,340)
NL,NU,(PRNT(I),
I=1,8)
350
TB←TB+1.04
300
RETURN

Subroutine Name and Argument List

POLRG(N,X,ML,M,NPLOT,COE)

Description

POLRG is a double-precision and modified version of the sample program of the same name in the IBM Scientific Subroutine Package, Version III, (IBM Form H20-0205-3). This is a least-squares polynomial regression subroutine using other IBM Scientific Subroutine Package subroutines to perform the analysis. Our POLRG is called from subroutine PART1 of the Skylab tracker error program, but may be used for other general polynomial regression applications.

Data is carried into POLRG by the array X, and a regression analysis is performed for progressively increasing degree of polynomial (starting from degree ML) up to a maximum degree M. Printout or plotting is controlled by NPLOT. If there is no reduction in the residual sum of squares between two successive degrees of the polynomials, the subroutine terminates the problem before completing the analysis for the highest degree polynomial specified. When the problem is either completed or terminated, the polynomial coefficients are returned in array COE.

Subroutine Arguments (and Dimensions) in Calling List

X(1100),N - The input data is supplied in the double precision array X. The number of input variable pairs is N. The individual values of the independent variable are stored in the first N locations of X, or in the first column if one views X as a matrix of dimension N by M+1. The individual values of the dependent variable are stored in the last column of this matrix. The dimension of X must be equal to or greater than N*(M+1).

ML - ML is the degree of polynomial at which the problem starts. ML must be equal to or less than M; the problem starts at ML and increments up to M(unless no improvement in the residual sum of squares is reached, terminating the analysis before M is reached).

M - The maximum degree specified for the polynomial.

NPLOT - Integer describing how the results of POLRG are to be printed out, according to the rules established by POLRG:

NPLOT = 5 - Print a table of data input, fitted value, and residuals for each separate degree of the regression,

NPLOT = 4 - Same as NPLOT=5 but in addition call a PLOT subroutine at each separate degree,

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

NPLOT = 3 - Print the table of values and residuals only for the final degree fitted,

NPLOT = 2 - Same as NPLOT=3 but in addition call PLOT at the final degree,

NPLOT = 1 - No table of residuals, no plot.

COE(11) - The double-precision polynomial coefficients determined by POLRG. If we let XI be an individual value of the independent variable, and YI be the corresponding value on the fitted curve, then

YI = COE(1) + COE(2)*XI + COE(3)*XI**2
+ . . . + COE(11)*XI**10.

The dimension of COE must be at least M+1.

Input/Output Performed by Subroutine

The output from POLRG is controlled by NPLOT as described above.

Other Important Subroutine Variable

DI(100),D(66), B(10),E(10),SB(10), T(10),XBAR(11),STD(11), SUMSQ(11),ISAVE(11), ANS(10) - These are various work spaces. DI,D,B, E,SB,T,XBAR,STD,SUMSQ, and ANS are all double-precision variables. The dimension of DI must be at least M*M. The dimension of D must be at least (M+2)*(M+1)/2. The dimensions of B,E,SB, and T must be at least M, and the dimensions of SBAR,STD, SUMSQ, and ISAVE must be at least M+1. Our POLRG is set up for a maximum M=10 and converts any higher value to 10, and for M=10 the maximum N allowable (for present dimension) is 100.

P(600) - P carries information for the plot of observed data and/or printout of estimates. The dimension of P should be at least 3*N if a plot is desired.

Other Subroutines Called

The following subroutines are all from the IBM Scientific Subroutine Package, and are unchanged except that they have been converted to double precision. The descriptions are from the Scientific Subroutine Package documentation and are repeated here for convenience of reference.

GDATA(N,M,X,XBAR,STD,D,SUMSQ) - This subroutine generates independent variables up to the Mth power and calculates means, standard deviations, sums of cross-products of deviations from means, and product moment correlation coefficients.

ORDER(MM,D,MM,I,ISAVE,DI,E) - Constructs from a larger matrix of correlation coefficients a subset matrix of intercorrelations among independent variables and a vector of intercorrelations of independent variables with dependent variable.

MINV(DI,I,DET,B,I) - Uses standard Gauss-Jordan method to invert a matrix and calculate the determinant.

MULTR(N,I,XBAR,STD,SUMSQ,DI,E,ISAVE,B,SB,T,ANS) - Performs a multiple regression analysis for a dependent variable and a set of independent variables.

PLOT(LA,P,N,3,0,1) - A special plot subroutine can be provided for the input data and the fitted resulting curve. The IBM documentation provides a PLOT subroutine but in our modified POLRG it is important to notice that P is a double-precision array. Most of our waveform work has been done using a dummy PLOT routine.

Remarks

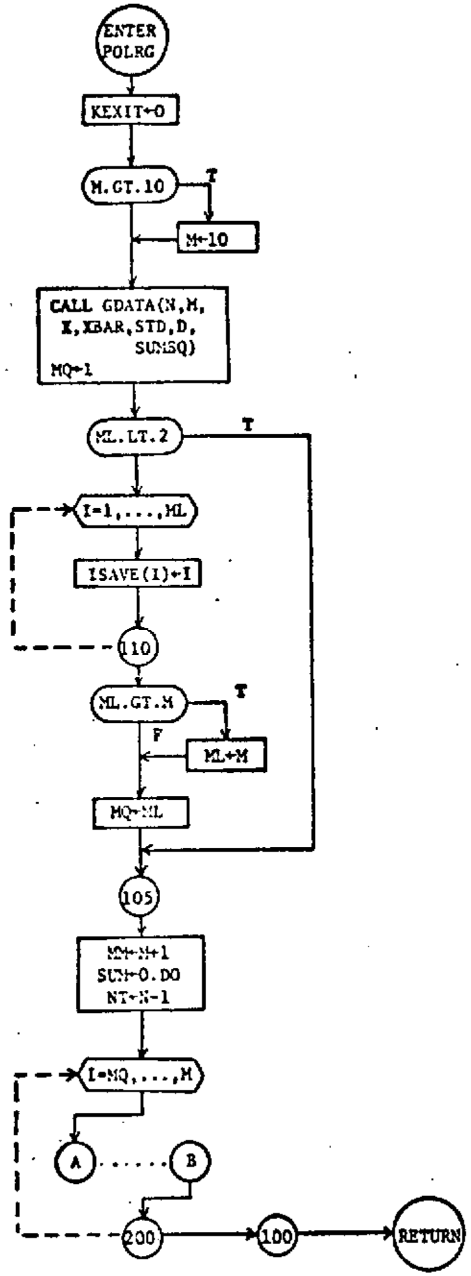
Following first the source listing and then the flow diagram for POLRG, we provide the source listings for GDATA,ORDER,MINV, and MULTR; for any further details of these routines, reference to the IBM documentation is necessary.

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

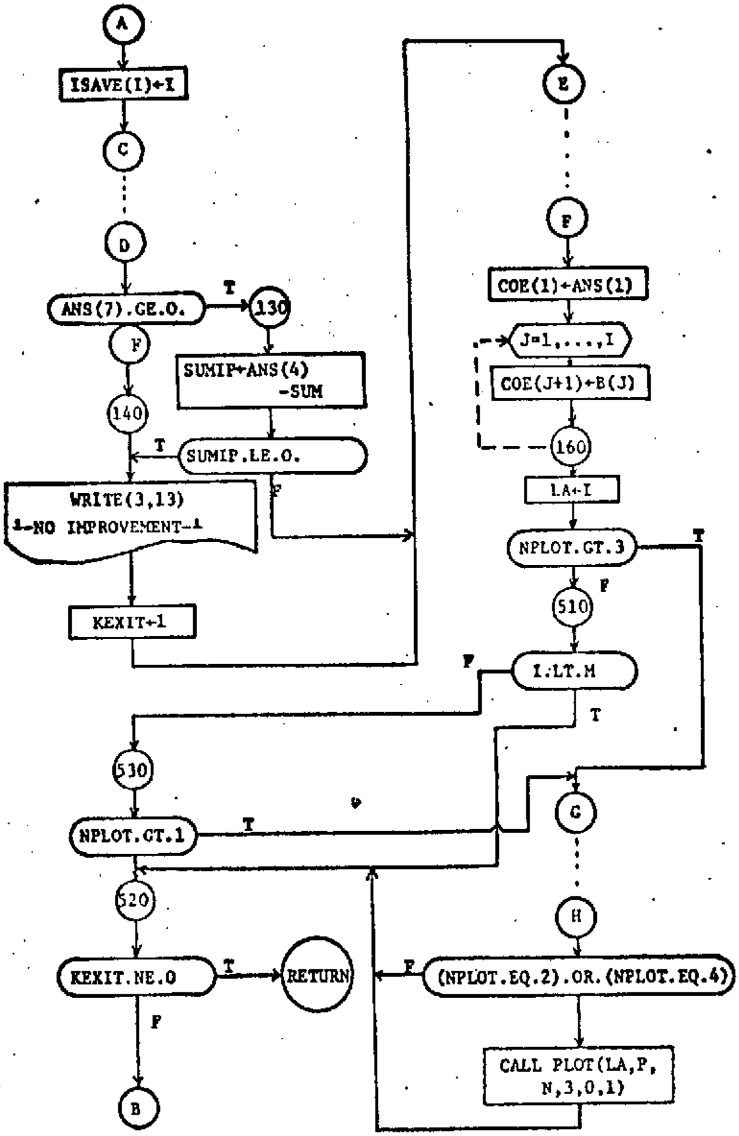
```
      SUBROUTINE POLRG(N,X,ML,M,NPLOT,COE)
      DIMENSION X(1),DI(100),D(66),B(10),E(10),SB(10),T(10)
      DIMENSION XBAR(11),STD(11),COE(11),SUMSQ(11),ISAVE(11)
      DIMENSION ANS(10),P(600)
      DOUBLE PRECISION X,XBAR,STD,D,SUMSQ,DI,E,B,SB,T,ANS,DET,COE
      DOUBLE PRECISION P,RESID
    5 FORMAT('1POLYNOMIAL REGRESSION OF DEGREE',I3)
    6 FORMAT('0   INTERCEPT',F20.12)
    7 FORMAT('0   REGRESSION COEFFICIENTS'/(6D20.12))
    8 FORMAT('0'/24X,'ANALYSIS OF VARIANCE FOR',I4,
     1 '  DEGREE POLYNOMIAL'/)
    9 FORMAT('0',5X,'SOURCE OF VARIATION',7X,'DEGREE OF',7X,'SUM OF',
     1 9X,'MEAN',10X,'F',9X,'IMPROVEMENT IN TERMS'/33X,'FREEDOM',
     2 8X,'SQUARES',7X,'SQUARE',7X,'VALUE',8X,'OF SUM OF SQUARES')
   10 FORMAT('0  DUE TO REGRESSION',12X,I6,F17.5,F14.5,F13.5,F20.5)
   11 FORMAT('  DEVIATION ABOUT REGRESSION   ',I6,F17.5,F14.5)
   12 FORMAT(8X,'TOTAL',19X,I6,F17.5///)
   13 FORMAT('0  NO IMPROVEMENT')
   14 FORMAT('0'//27X,'TABLE OF RESIDUALS'//' OBSERVATION NO.',5X,
     1 'X VALUE',7X,'Y VALUE',7X,'Y ESTIMATE',7X,'RESIDUAL'/)
   15 FORMAT(' ',3X,I6,F18.5,F14.5,F17.5,F15.5)
      KEXIT=0
      IF (M.GT.10) M=10
      CALL GDATA(N,M,X,XBAR,STD,D,SUMSQ)
      MO=1
      IF (ML.LT.2) GO TO 105
      DO 110 I=1,ML
  110 ISAVE(I)=I
      IF (ML.GT.M) ML=M
      MO=ML
  105 MM=M+1
      SUM=0.D0
      NT=N-1
      DO 200 I=MO,M
      ISAVE(I)=I
      CALL ORDER(MM,D,MM,I,ISAVE,DI,E)
      CALL MINV(DI,I,DET,B,T)
      CALL MULTR(N,I,XBAR,STD,SUMSQ,DI,E,ISAVE,B,SB,T,ANS)
      WRITE (3,5) I
      IF (ANS(7)) 140,130,130
  130 SUMIP=ANS(4)-SUM
      IF (SUMIP) 140,140,150
  140 WRITE (3,13)
      KEXIT=1
  150 WRITE (3,6) ANS(1)
      WRITE (3,7) (B(J), J=1,I)
      WRITE (3,8) I
      WRITE (3,9)
      SUM=ANS(4)
      WRITE (3,10) I,ANS(4),ANS(6),ANS(10),SUMIP
      NI=ANS(8)
      WRITE (3,11) NI,ANS(7),ANS(9)
      WRITE (3,12) NT,SUMSQ(MM)
      COE(1)=ANS(1)
      DO 160 J=1,I
  160 COE(J+1)=B(J)
      LA=I
      IF (NPLOT-2) 510,510,220
  510 IF (I-M) 520,530,530
  530 IF (NPLOT.LE.1) GO TO 520
  220 NP3=N+N
      DO 230 K=1,N
      NP3=NP3+1
      P(NP3)=COE(1)
      L=K
      DO 230 J=1,LA
      P(NP3)=P(NP3)+X(L)*COE(J+1)
  230 L=L+N
      N2=N
      L=N*M
      DO 240 K=1,N
      P(K)=X(K)
      N2=N2+1
      L=L+1
  240 P(N2)=X(L)
      WRITE (3,14)
      NP2=N
      NP3=N+N
      DO 250 K=1,N
      NP2=NP2+1
      NP3=NP3+1
      RESID=P(NP2)-P(NP3)
  250 WRITE (3,15) K,P(K),P(NP2),P(NP3),RESID
      IF ( (NPLOT.EQ.2).OR.(NPLOT.EQ.4) ) CALL PLOTDP(LA,P,N,3,0,1)
  520 IF (KEXIT) 100,200,100
  200 CONTINUE
  100 RETURN
      END
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

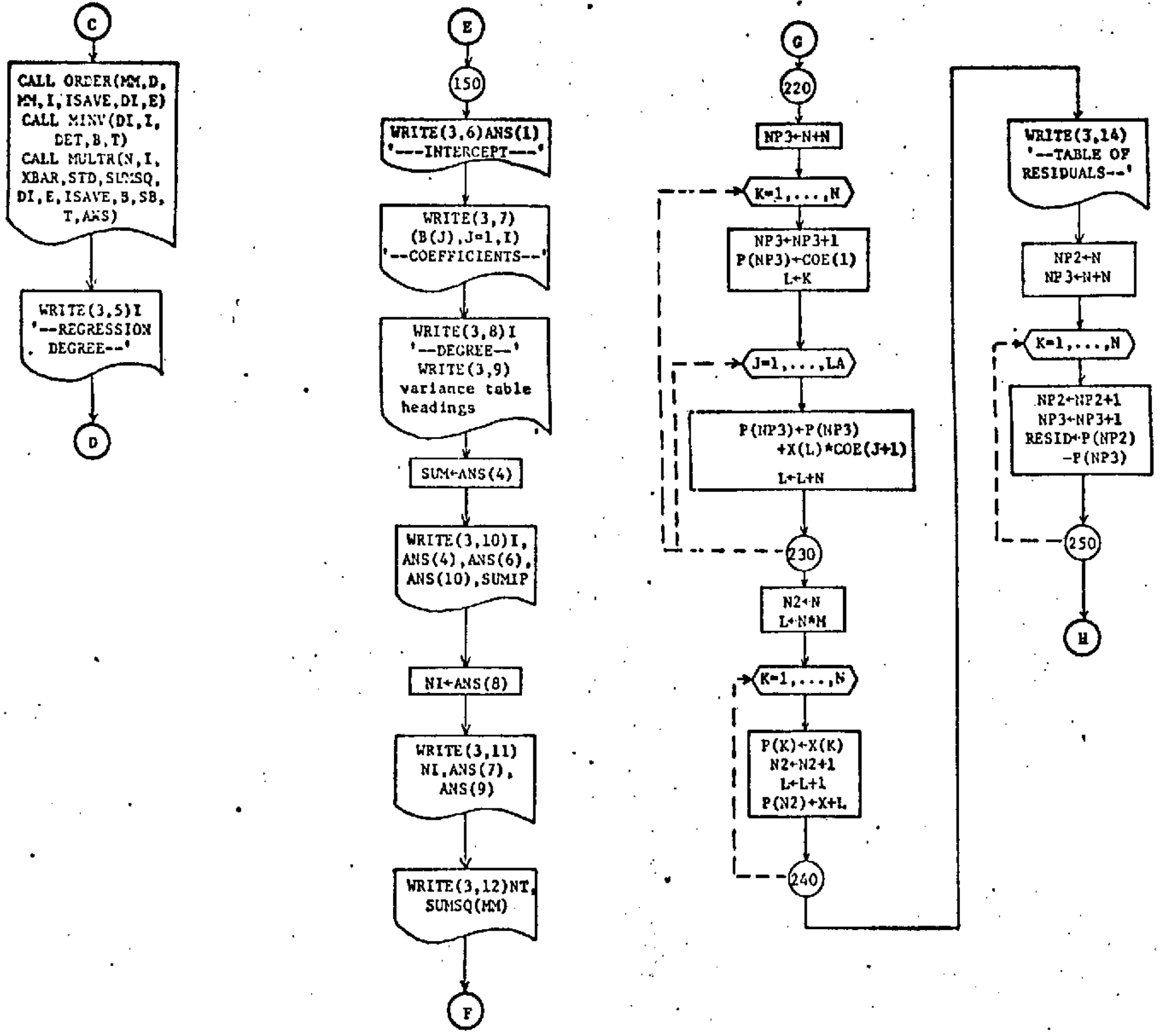
POLRG Flow Chart



REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR



POLRG Flow Chart (continued)



```
      SUBROUTINE GDATA(N,M,X,XBAR,STD,D,SUMSQ)
      DIMENSION X(1),XBAR(1),STD(1),D(1),SUMSQ(1)
      DOUBLE PRECISION X,XBAR,STD,D,SUMSQ,T1,T2
      IF (M-1) 105,105,90
   90 L1=0
      DO 100 I=2,M
      L1=L1+N
      DO 100 J=1,N
      L=L1+J
      K=L-N
  100 X(L)=X(K)*X(J)
  105 MM=M+1
      DF=N
      L=0
      DO 115 I=1,MM
      XBAR(I)=0.D0
      DO 110 J=1,N
      L=L+1
  110 XBAR(I)=XBAR(I)+X(L)
  115 XBAR(I)=XBAR(I)/DF
      DO 130 I=1,MM
  130 STD(I)=0.D0
      L=((MM+1)*MM)/2
      DO 150 I=1,L
  150 D(I)=0.D0
      DO 170 K=1,N
      L=0
      DO 170 J=1,MM
      L2=N*(J-1)+K
      T2=X(L2)-XBAR(J)
      STD(J)=STD(J)+T2
      DO 170 I=1,J
      L1=N*(I-1)+K
      T1=X(L1)-XBAR(I)
      L=L+1
  170 D(L)=D(L)+T1*T2
      L=0
      DO 175 J=1,MM
      DO 175 I=1,J
      L=L+1
  175 D(L)=D(L)-STD(I)*STD(J)/DF
      L=0
      DO 180 I=1,MM
      L=L+I
      SUMSQ(I)=D(L)
  180 STD(I)=DSQRT(DABS(D(L)))
      L=0
      DO 190 J=1,MM
      DO 190 I=1,J
      L=L+1
  190 D(L)=D(L)/(STD(I)*STD(J))
      DF=SQRT(DF-1.)
      DO 200 I=1,MM
  200 STD(I)=STD(I)/DF
      RETURN
      END
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      SUBROUTINE MINV(A,N,D,L,M)
      DIMENSION A(1),L(1),M(1)
      DOUBLE PRECISION A,D,BIGA,HOLD
      D=1.0D0
      NK=-N
      DO 80 K=1,N
      NK=NK+N
      L(K)=K
      M(K)=K
      KK=NK+K
      BIGA=A(KK)
      DO 20 J=K,N
      IZ=N*(J-1)
      DO 20 I=K,N
      IJ=IZ+I
   10 IF (DABS(BIGA)-DABS(A(IJ))) 15,20,20
   15 BIGA=A(IJ)
      L(K)=I
      M(K)=J
   20 CONTINUE
      J=L(K)
      IF (J-K) 35,35,25
   25 KI=K-N
      DO 30 I=1,N
      KI=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
   30 A(JI)=HOLD
   35 I=M(K)
      IF (I-K) 45,45,38
   38 JP=N*(I-1)
      DO 40 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)
   40 A(JI)=HOLD
   45 IF (BIGA) 48,46,48
   46 D=0.D0
      RETURN
   48 DO 55 I=1,N
      IF (I-K) 50,55,50
   50 IK=NK+I
      A(IK)=A(IK)/(-BIGA)
   55 CONTINUE
      DO 65 I=1,N
      IK=NK+I
      HOLD=A(IK)
      IJ=I-N
      DO 65 J=1,N
      IJ=IJ+N
      IF (I-K) 60,65,60
   60 IF (J-K) 62,65,62
   62 KJ=IJ-I+K
      A(IJ)=HOLD*A(KJ)+A(IJ)
   65 CONTINUE
      KJ=K-N
      DO 75 J=1,N
      KJ=KJ+N
      IF (J-K) 70,75,70
   70 A(KJ)=A(KJ)/BIGA
   75 CONTINUE
      D=D*BIGA
      A(KK)=1.0D0/BIGA
   80 CONTINUE
      K=N
  100 K=K-1
      IF (K) 150,150,105
  105 I=L(K)
      IF (I-K) 120,120,108
  108 JQ=N*(K-1)
      JP=N*(I-1)
      DO 110 J=1,N
      JK=JQ+J
      HOLD=A(JK)
      JI=JP+J
      A(JK)=-A(JI)
  110 A(JI)=HOLD
  120 J=M(K)
      IF (J-K) 100,100,125
  125 KI=K-N
      DO 130 I=1,N
      KI=KI+N
      HOLD=A(KI)
      JI=KI-K+J
      A(KI)=-A(JI)
  130 A(JI)=HOLD
      GO TO 100
  150 RETURN
      END
```

```
      SUBROUTINE MULTR(N,K,XBAR,STD,D,RX,RY,ISAVE,B,SB,T,ANS)
      DIMENSION XBAR(1),STD(1),D(1),RX(1),RY(1),ISAVE(1),B(1),SB(1),
     1 T(1),ANS(1)
      DOUBLE PRECISION XBAR,STD,D,RX,RY,B,SB,T,ANS,RM,BO,SSAR,
     1 SSDR,SY,FN,FK,SSARM,SSDRM,F
      MM=K+1
      DO 100 J=1,K
  100 B(J)=0.D0
      DO 110 J=1,K
      L1=K*(J-1)
      DO 110 I=1,K
      L=L1+I
  110 B(J)=B(J)+RY(I)*RX(L)
      RM=0.D0
      BO=0.D0
      L1=ISAVE(MM)
      DO 120 I=1,K
      RM=RM+B(I)*RY(I)
      L=ISAVE(I)
      B(I)=B(I)*(STD(L1)/STD(L))
  120 BO=BO+B(I)*XBAR(L)
      BO=XBAR(L1)-BO
      SSAR=RM*D(L1)
  122 RM=DSQRT(DABS(RM))
      SSDR=D(L1)-SSAR
      FN=N-K-1
      SY=SSDR/FN
      DO 130 J=1,K
      L1=K*(J-1)+J
      L=ISAVE(J)
  125 SB(J)=DSQRT(DABS((RX(L1)/D(L))*SY))
  130 T(J)=B(J)/SB(J)
  135 SY=DSQRT(DABS(SY))
      FK=K
      SSARM=SSAR/FK
      SSDRM=SSDR/FN
      F=SSARM/SSDRM
      ANS(1)=BO
      ANS(2)=RM
      ANS(3)=SY
      ANS(4)=SSAR
      ANS(5)=FK
      ANS(6)=SSARM
      ANS(7)=SSDR
      ANS(8)=FN
      ANS(9)=SSDRM
      ANS(10)=F
      RETURN
      END
```

```
      SUBROUTINE ORDER(M,R,NDEP,K,ISAVE,RX,RY)
      DIMENSION R(1),ISAVE(1),RX(1),RY(1)
      DOUBLE PRECISION R,RX,RY
      MM=0
      DO 130 J=1,K
      L2=ISAVE(J)
      IF (NDEP-L2) 122,123,123
  122 L=NDEP+(L2*L2-L2)/2
      GO TO 125
  123 L=L2+(NDEP*NDEP-NDEP)/2
  125 RY(J)=R(L)
      DO 130 I=1,K
      L1=ISAVE(I)
      IF (L1-L2) 127,128,128
  127 L=L1+(L2*L2-L2)/2
      GO TO 129
  128 L=L2+(L1*L1-L1)/2
  129 MM=MM+1
  130 RX(MM)=R(L)
      ISAVE(K+1)=NDEP
      RETURN
      END
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

Subroutine Name and Argument List

Subroutine SPLINE (X,Y,N,TJ,SSJ,MODE)

General Background

Given a set of n points $(x,y,),(x_2,y_2),--,(x_n,y_n)$ arranged in ascending order in the independent variable x, a third degree spline is a means of interpolation to find the value of y corresponding to a specified x_s for $x_1 \leq x_s \leq x_n$. The program accomplishes this by connecting each pair of adjacent points with a section of a third degree polynomial. There will be a different polynomial for each interval and these polynomials are "matched up" at the (x,y) points by requiring that the first and second derivatives be continuous at those points. The spline function is in some senses the "smoothest" interpolating function. As in all interpolating procedures, the interpolating function goes through each input point, but the spline exhibits considerably less oscillatory behavior between input points than does Lagrangian interpolation, for example.

The spline function of this subroutine is derived directly from the chapter, "Spline Functions, Interpolation, and Numerical Quadrature," by T.N.E. Greville, in *Mathematical Methods for Digital Computers, Vol. II*, A. Ralston and H. S. Wilf, editors, (New York: J. Wiley & Sons, 1967), pp 156-168. Our spline differs from Greville's in trading running time for storage space; since we were concerned about the overall size of our time-realignment programs, we recompute each time into the routine several quantities which Greville's program stores in arrays. The index search procedure is modified also; since our use of SPLINE will usually be for a series of monotonically increasing independent variables, we let the index search start at the last index found by the program.

SPLINE operates in two modes: an initial settingup with the input points x,y; and the return of a value SSJ for an input value TJ of the independent variable (corresponding to X). We use an indicator, MODE, in the argument list of SPLINE to distinguish between these modes.

Subroutine Argument (and Dimension) in Calling List

X(800),Y(800) = The N input points (X,Y) for setting up SPLINE
N when called with MODE = 0. The successive values

of X must be distinct and must be in ascending order.

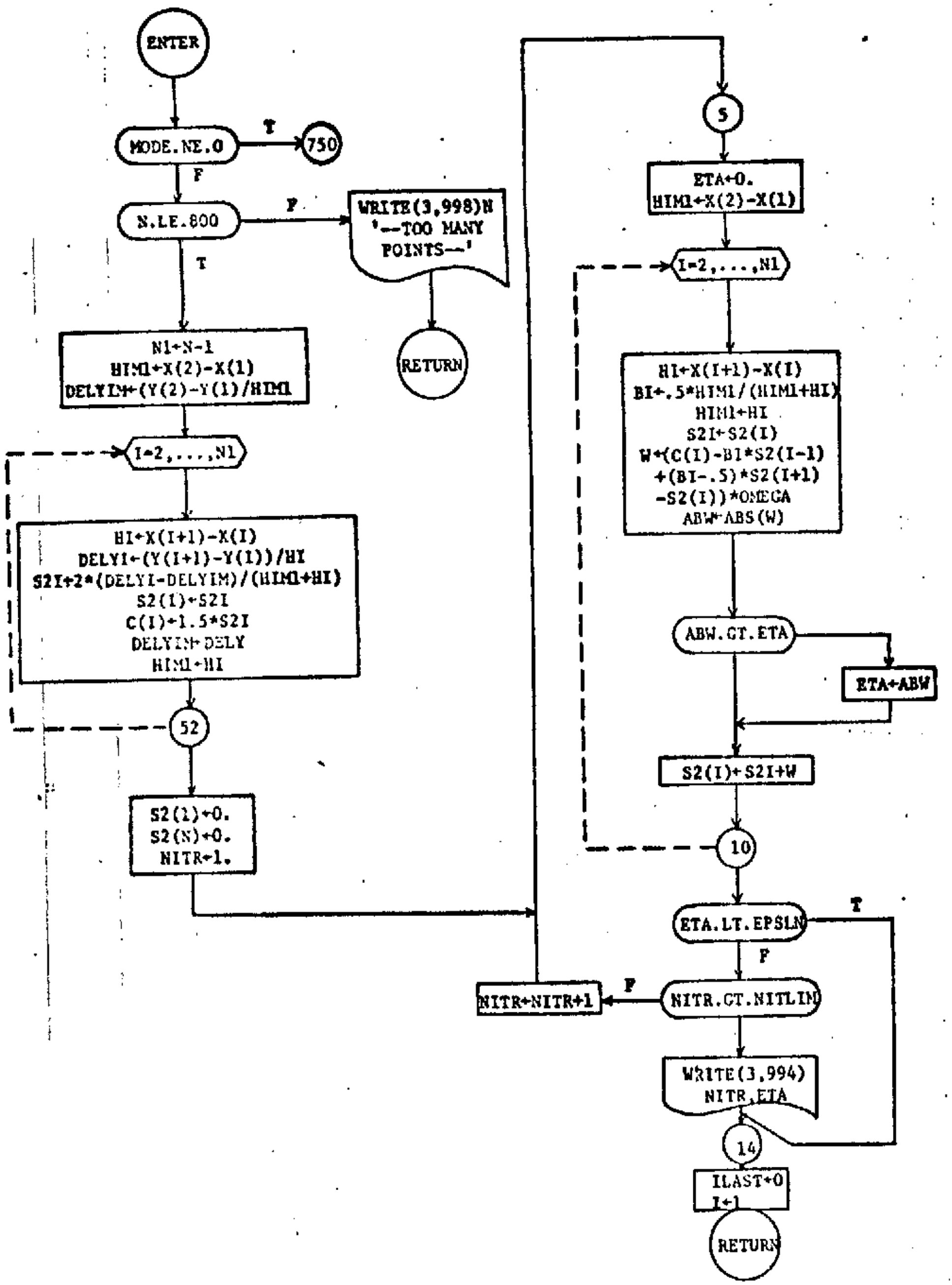
TJ,SSJ = Output of value SSJ is returned for input independent variable value TJ, when MODE = 1.

Input/Output Performed by Subroutine

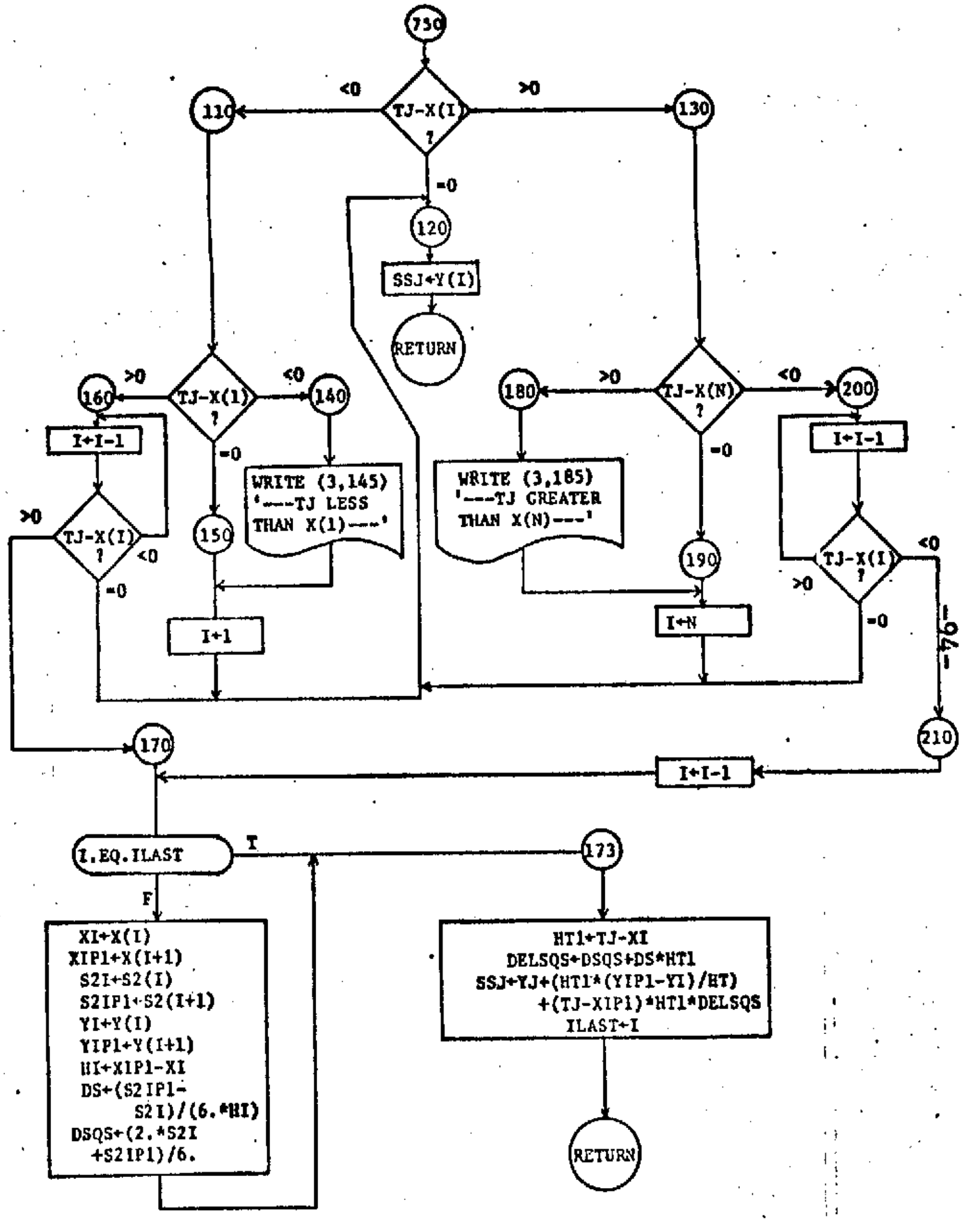
None if TJ (in MODE=1) satisfies $X(1) \leq TJ \leq X(N)$. If not, an error message is written indicating that TJ was outside the allowed range.

```
      SUBROUTINE SPLINE(X,Y,N,TJ,SSJ,MODE)
      DIMENSION X(800),Y(800),S2(800),C(800)
      DOUBLE PRECISION X,TJ,XI,XIP1,Y,YI,YIP1,SSJ,S2,S2I,S2IP1,BI,W,
     1 OMEGA,DELYI,DELYIM,HI,HIM1,C,DS,DSOS,HT1,DELSQS
      DATA EPSLN/.0001/,NITLIM/10/,OMEGA/1.0717967680D0/
      IF (MODE.NE.0) GO TO 750
      IF (N.LE.800) GO TO 2
      WRITE (3,998) N
  998 FORMAT(/' ERROR..HAD',I5,' INPUT POINTS,SETSPL HANDLES ONLY 800')
      RETURN
    2 N1=N-1
    3 HIM1=X(2)-X(1)
      DELYIM=(Y(2)-Y(1))/HIM1
    4 DO 52 I=2,N1
      HI=X(I+1)-X(I)
      DELYI=(Y(I+1)-Y(I))/HI
      S2I=2.D0*(DELYI-DELYIM)/(HIM1+HI)
      S2(I)=S2I
      C(I)=1.5D0*S2I
      DELYIM=DELYI
   52 HIM1=HI
      S2(1)=0.D0
      S2(N)=0.D0
      NITR=1
    5 ETA=0.
      HIM1=X(2)-X(1)
    6 DO 10 I=2,N1
      HI=X(I+1)-X(I)
      BI=.5D0*HIM1/(HIM1+HI)
      HIM1=HI
      S2I=S2(I)
    7 W=(C(I)-BI*S2(I-1)+(BI-.5D0)*S2(I+1)-S2(I))*OMEGA
    8 ABW=DABS(W)
      IF (ABW.GT.ETA) ETA=ABW
   10 S2(I)=S2I+W
   13 IF (ETA.LE.EPSLN) GO TO 14
      IF (NITR.GT.NITLIM) GO TO 995
      NITR=NITR+1
      GO TO 5
  995 WRITE (3,994) NITR,ETA
  994 FORMAT(/' NUMBER OF ITERATIONS=',I3,' AND EPSILON=',E12.5//)
   14 I=1
      ILAST=0
      RETURN
  750 IF (TJ-X(I)) 110,120,130
  110 IF (TJ-X(1)) 140,150,160
  140 WRITE (3,145) TJ
  145 FORMAT(' ARGUMENT TJ=',D12.5,', LESS THAN X(1)')
  150 I=1
  120 SSJ=Y(I)
      RETURN
  160 I=I-1
      IF (TJ-X(I)) 160,120,170
  130 IF (TJ-X(N)) 200,190,180
  180 WRITE (3,185) TJ
  185 FORMAT(' ARGUMENT TJ=',D12.5,', GREATER THAN X(N)')
  190 I=N
      GO TO 120
  200 I=I+1
      IF (TJ-X(I)) 210,120,200
  210 I=I-1
  170 IF (I.EQ.ILAST) GO TO 173
      XI=X(I)
      XIP1=X(I+1)
      S2I=S2(I)
      S2IP1=S2(I+1)
      YI=Y(I)
      YIP1=Y(I+1)
      HI=XIP1-XI
      DS=(S2IP1-S2I)/(6.D0*HI)
      DSOS=(2.D0*S2I+S2IP1)/6.D0
  173 HT1=TJ-XI
      DELSQS=DSOS+DS*HT1
      SSJ=YI+(HT1*(YIP1-YI)/HI)+(TJ-XIP1)*HT1*DELSQS
      ILAST=I
      RETURN
```

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR



ENTER
MODE.NE.0
T
750
F
N.LE.800
F
WRITE(3,998)N '--TOO MANY POINTS--'
RETURN
T
N1←N-1
HIM1←X(2)-X(1)
DELYIM←(Y(2)-Y(1))/HIM1
I=2,...,N1
HI←X(I+1)-X(I)
DELYI←(Y(I+1)-Y(I))/HI
S2I←2*(DELYI-DELYIM)/(HIM1+HI)
S2(I)←S2I
C(I)←1.5*S2I
DELYIM←DELY
HIM1←HI
52
S2(1)←0.
S2(N)←0.
NITR←1.
5
ETA←0.
HIM1←X(2)-X(1)
I=2,...,N1
HI←X(I+1)-X(I)
BI←.5*HIM1/(HIM1+HI)
HIM1←HI
S2I←S2(I)
W←(C(I)-BI*S2(I-1)
+(BI-.5)*S2(I+1)
-S2(I))*OMEGA
ABW←ABS(W)
ABW.GT.ETA
ETA←ABW
S2(I)←S2I+W
10
ETA.LT.EPSLN
T
F
NITR←NITR+1
F
NITR.GT.NITLIM
WRITE(3,994) NITR,ETA
14
ILAST←0
I←1
RETURN



750
TJ-X(I) ?
<0
>0
110
130
=0
120
SSJ←Y(I)
RETURN
>0
160
TJ-X(1) ?
<0
140
=0
I←I-1
>0
TJ-X(I) ?
<0
=0
150
WRITE (3,145) '---TJ LESS THAN X(1)---'
I←1
180
>0
TJ-X(N) ?
<0
200
=0
WRITE (3,185) '---TJ GREATER THAN X(N)---'
190
I←N
I←I-1
TJ-X(I) ?
<0
>0
=0
210
170
I←I-1
I.EQ.ILAST
T
173
F
XI←X(I)
XIP1←X(I+1)
S2I←S2(I)
S2IP1←S2(I+1)
YI←Y(I)
YIP1←Y(I+1)
HI←XIP1-XI
DS←(S2IP1-S2I)/(6.*HI)
DSQS←(2.*S2I+S2IP1)/6.
HT1←TJ-XI
DELSQS←DSQS+DS*HT1
SSJ←YJ+(HT1*(YIP1-YI)/HT)
+(TJ-XIP1)*HT1*DELSQS
ILAST←I
RETURN

Subroutine Namd and Argument List

Subroutine WEIGHT (Y,NY)

General Description

WEIGHT is called to apply a weighting function to a set of NY input data point. Y. Upon return from WEIGHT, the original Y will be replaced by the set of values from applying the NW individual weights to the input Y. The subroutine XCNVLV with an entry point XCFILL is used for the convolutional weighting. Since generally NW will be considerably smaller than NY, the total storage needed is only NW+NY rather than the 2*NY which would be needed for a straightforward digital convolution.

The number of weights, NW, and the weight values, W, are set by the DATA statement in WEIGHT. Note that with NW = 17, if W(9) is set to 1.0 and the other weights to 0., the original Y is returned from W.

Subroutine Arguments (and Dimensions) in Calling List

Y(800),NY = Input points (number = NY) entering WEIGHT; on return, Y is replaced by the weighted result.

Other Important Subroutine Variables

WEIGHT,NW = WEIGHT (NW) contains the weight as set by DATA statement.

TEMP = A working temporary array. The dimensions of W and TEMP need to be at least as large as NW.

Input/Output Performed by Subroutine

None

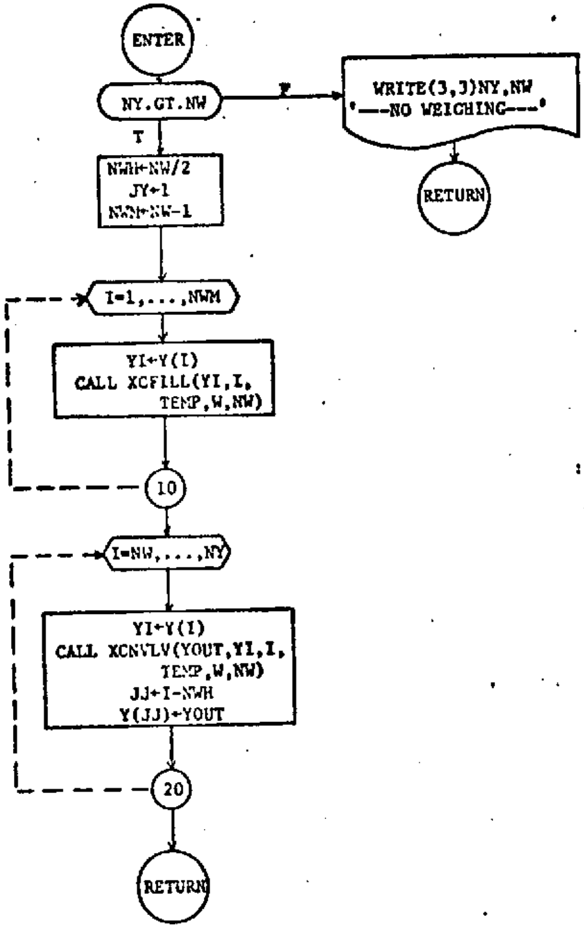
Other Subroutines Called

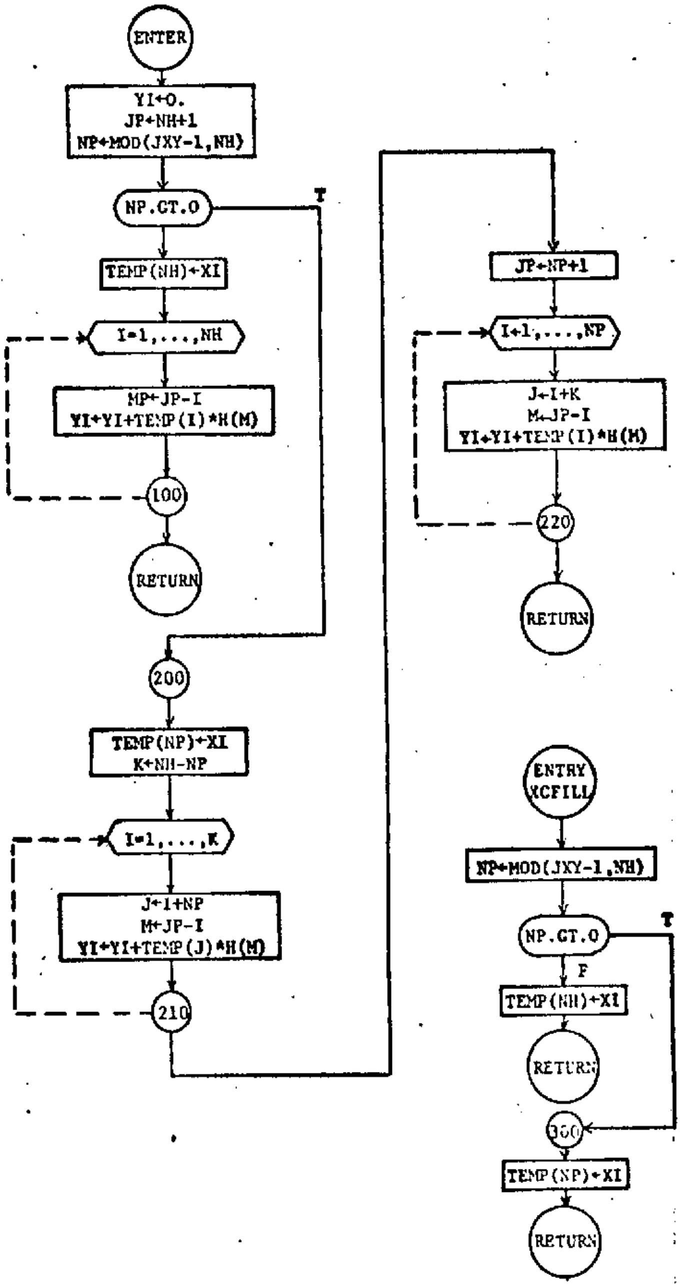
XCNVLV (with entry XCFILL) - Carries out digital convolution when called repeatedly with successive points. XCNVLV listing only is provided here.

```
      SUBROUTINE WEIGHT(Y,NY)
      DIMENSION Y(890),W(17),TEMP(17)
      DOUBLE PRECISION Y,YI,W,TEMP,YOUT
      DATA NW/17/,W/-.001D0,-.0055D0,-.0145D0,-.03D0,-.0485D0,-.057D0,
     1 -.0185D0,.14D0,1.07D0,.14D0,-.0185D0,-.057D0,-.0485D0,
     2 -.03D0,-.0145D0,-.0055D0,-.001D0/
      IF (NY.GE.NW) GO TO 5
      WRITE (3,3) NY,NW
    3 FORMAT(//' NO WEIGHTING DONE, BECAUSE NY=',I3,',BUT NW=',I3//)
      RETURN
    5 NWH=NW/2
      JY=1
      NWM=NW-1
      DO 10 I=1,NWM
      YI=Y(I)
   10 CALL XCFILL(YI,I,TEMP,W,NW)
      DO 20 I=NW,NY
      YI=Y(I)
      CALL XCNVLV(YOUT,YI,I,TEMP,W,NW)
      JJ=I-NWH
   20 Y(JJ)=YOUT
      RETURN
      END

      SUBROUTINE XCNVLV(YI,XI,JXY,TEMP,H,NH)
      DIMENSION TEMP(NH),H(NH)
      DOUBLE PRECISION TEMP,H,YI,XI
      YI=0.D0
      JP=NH+1
      NP=MOD(JXY-1,NH)
      IF (NP.GT.0) GO TO 200
      TEMP(NH)=XI
      DO 100 I=1,NH
      M=JP-I
  100 YI=YI+TEMP(I)*H(M)
      RETURN
  200 TEMP(NP)=XI
      K=NH-NP
      DO 210 I=1,K
      J=I+NP
      M=JP-I
  210 YI=YI+TEMP(J)*H(M)
      JP=NP+1
      DO 220 I=1,NP
      J=I+K
      M=JP-I
  220 YI=YI+TEMP(I)*H(M)
      RETURN
      ENTRY XCFILL(XI,JXY,TEMP,H,NH)
      NP=MOD(JXY-1,NH)
      IF (NP.GT.0) GO TO 300
      TEMP(NH)=XI
      RETURN
  300 TEMP(NP)=XI
      RETURN
      END
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR


ENTER
NY.GT.NW
F
T
WRITE(3,J)NY,NW
'---NO WEIGHING---'
RETURN
NWH←NW/2
JY←1
NWM←NW-1
I=1,...,NWM
YI←Y(I)
CALL XCFILL(YI,I,
TEMP,W,NW)
10
I=NW,...,NY
YI←Y(I)
CALL XCNVLV(YOUT,YI,I,
TEMP,W,NW)
JJ←I-NWH
Y(JJ)←YOUT
20
RETURN


ENTER
YI←0.
JP←NH+1
NP←MOD(JXY-1,NH)
NP.GT.0
T
TEMP(NH)←XI
I=1,...,NH
MP←JP-I
YI←YI+TEMP(I)*H(M)
100
RETURN
200
TEMP(NP)←XI
K←NH-NP
I=1,...,K
J←I+NP
M←JP-I
YI←YI+TEMP(J)*H(M)
210
JP←NP+1
I=1,...,NP
J←I+K
M←JP-I
YI←YI+TEMP(I)*H(M)
220
RETURN
ENTRY
XCFILL
NP←MOD(JXY-1,NH)
NP.GT.0
T
F
TEMP(NH)←XI
RETURN
300
TEMP(NP)←XI
RETURN

Subroutine Name and Argument List

Subroutine VTADI(MB,EPS,SANDH,SUM1,SUM2,NSUM)

Entry VTADD(EPS,SANDH)

Entry VTADO(SUM1,SUM2,NSUM)

Description

This subroutine accomplishes the "vertical averaging" of a number of sets of sample-and-hold results once each individual sample-and-hold time tracker error correction EPS has been found. VTADI initializes the storage locations and sets up several interval variables, VTADD is used to enter each successive sample-and-hold set SANDH in bins appropriate to the individual tracker error EPS, and VTADO gets ready for outputting the means, standard deviations, and total number of entries at each storage bin.

Subroutine Arguments (and Dimensions) in Calling List

MB = Integer multiplier determining bin width, number of bins. The individual bin width is GS/MB (where GS is the S&H gate separation, 10 nanoseconds in the case of SKYLAB S-193). 1≤MB≤10

EPS = Individual tracker error in nanoseconds at time corresponding to the input SANDH.

SANDH(8) = Input set of 8 sample-and-hold gate readings.

SUM1(151) = The set of double-precision bins into which the different inputs are summed; the call to VTADI zeroes all these and the call to VTADO converts all the subtotals to individual bin means.

SUM2(151) = A set of double-precision bins into which the squares of SANDH are added. VTADI zeroes all SUM2, and the call to VTADO converts the sum of squares into individual bin standard deviation estimates.

NSUM(151) = An integer array accompanying SUM1 and SUM2 which records the number of entries to each individual bin. The call to VTADI zeroes all NSUM.

Other Important Subroutine Variables

GS = Sample-and-hold gate separation in nanoseconds (GS=10.0 for S-193 Mode 5, submodes 1 and 2, and GS=25.0 for the other submodes of Modes 1 and 5).

NBB = An integer locating the base bin, calculated from WB in VTADI. All 8 S&H locations are calculated relative to NBB via JB in VTADD.

Input/Output Performed by Subroutine

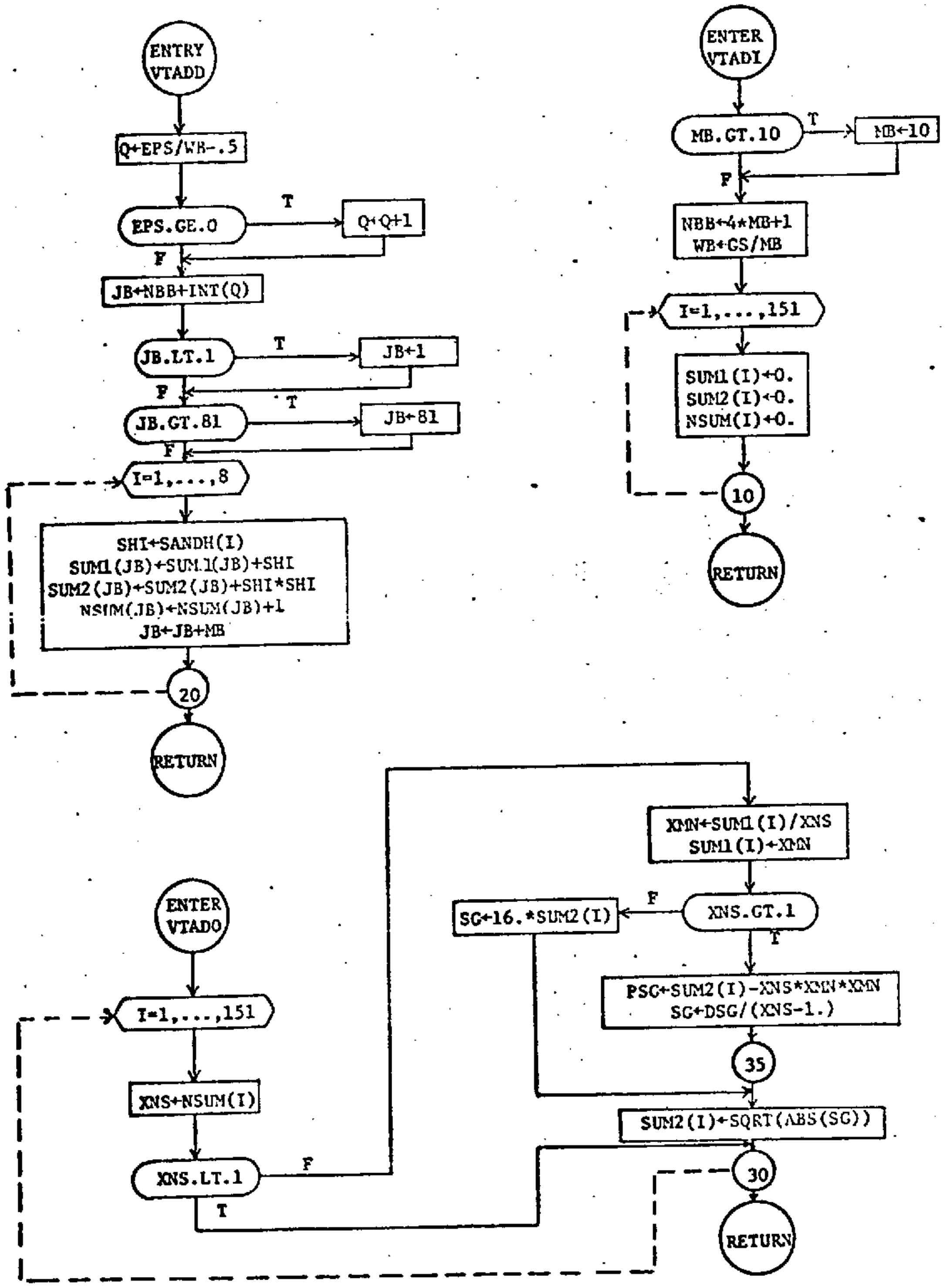
None except through calling list.

Other Subroutines Called

None.

```
      SUBROUTINE VTADI(MB,EPS,SANDH,SUM1,SUM2,NSUM)
      DIMENSION SANDH(8),SUM1(151),SUM2(151),NSUM(151)
      DOUBLE PRECISION SUM1,SUM2,SHI,DSG,XMN
      DATA GS/10./
      IF (MB.GT.10) MB=10
      NBB=4*MB+1
      WB=GS/MB
      DO 10 I=1,151
      SUM1(I)=0.D0
      SUM2(I)=0.D0
   10 NSUM(I)=0
      RETURN
      ENTRY VTADD(EPS,SANDH)
      Q=EPS/WB-.5
      IF (EPS.GE.0.) Q=Q+1.
      JB=NBB+INT(Q)
      IF (JB.LT.1) JB=1
      IF (JB.GT.81) JB=81
      DO 20 I=1,8
      SHI=SANDH(I)
      SUM1(JB)=SUM1(JB)+SHI
      SUM2(JB)=SUM2(JB)+SHI*SHI
      NSUM(JB)=NSUM(JB)+1
   20 JB=JB+MB
      RETURN
      ENTRY VTADO(SUM1,SUM2,NSUM)
      DO 30 I=1,151
      XNS=NSUM(I)
      IF (XNS.LT.1.) GO TO 30
      XMN=SUM1(I)/XNS
      SUM1(I)=XMN
      IF (XNS.GT.1.) GO TO 40
      SG=16.*SUM2(I)
      GO TO 35
   40 DSG=SUM2(I)-XNS*XMN*XMN
      SG=DSG/(XNS-1.)
   35 SUM2(I)=SQRT(ABS(SG))
   30 CONTINUE
      RETURN
      END
```

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR


ENTRY VTADD
Q←EPS/WB-.5
EPS.GE.0
T
Q←Q+1
F
JB←NBB+INT(Q)
JB.LT.1
T
JB←1
F
JB.GT.81
T
JB←81
F
I=1,...,8
SHI←SANDH(I)
SUM1(JB)←SUM1(JB)+SHI
SUM2(JB)←SUM2(JB)+SHI*SHI
NSUM(JB)←NSUM(JB)+1
JB←JB+MB
20
RETURN
ENTER VTADI
MB.GT.10
T
MB←10
F
NBB←4*MB+1
WB←GS/MB
I=1,...,151
SUM1(I)←0.
SUM2(I)←0.
NSUM(I)←0.
10
RETURN
ENTER VTADO
I=1,...,151
XNS←NSUM(I)
XNS.LT.1
F
T
XMN←SUM1(I)/XNS
SUM1(I)←XMN
XNS.GT.1
F
SG←16.*SUM2(I)
T
PSG←SUM2(I)-XNS*XMN*XMN
SG←DSG/(XNS-1.)
35
SUM2(I)←SQRT(ABS(SG))
30
RETURN

Time and Position Details for Bins in VTADD

The accompanying sketch shows the bin numbering and positioning details. Notice that this is specific to eight sample-and-hold gates as in the Skylab S-193 experiment (although the changes necessary to go to a different number of S&H gates would be simple ones), and that an additional four gate positions are located on either side of the original eight to allow for tracker jitter of up to ± 4 GS.

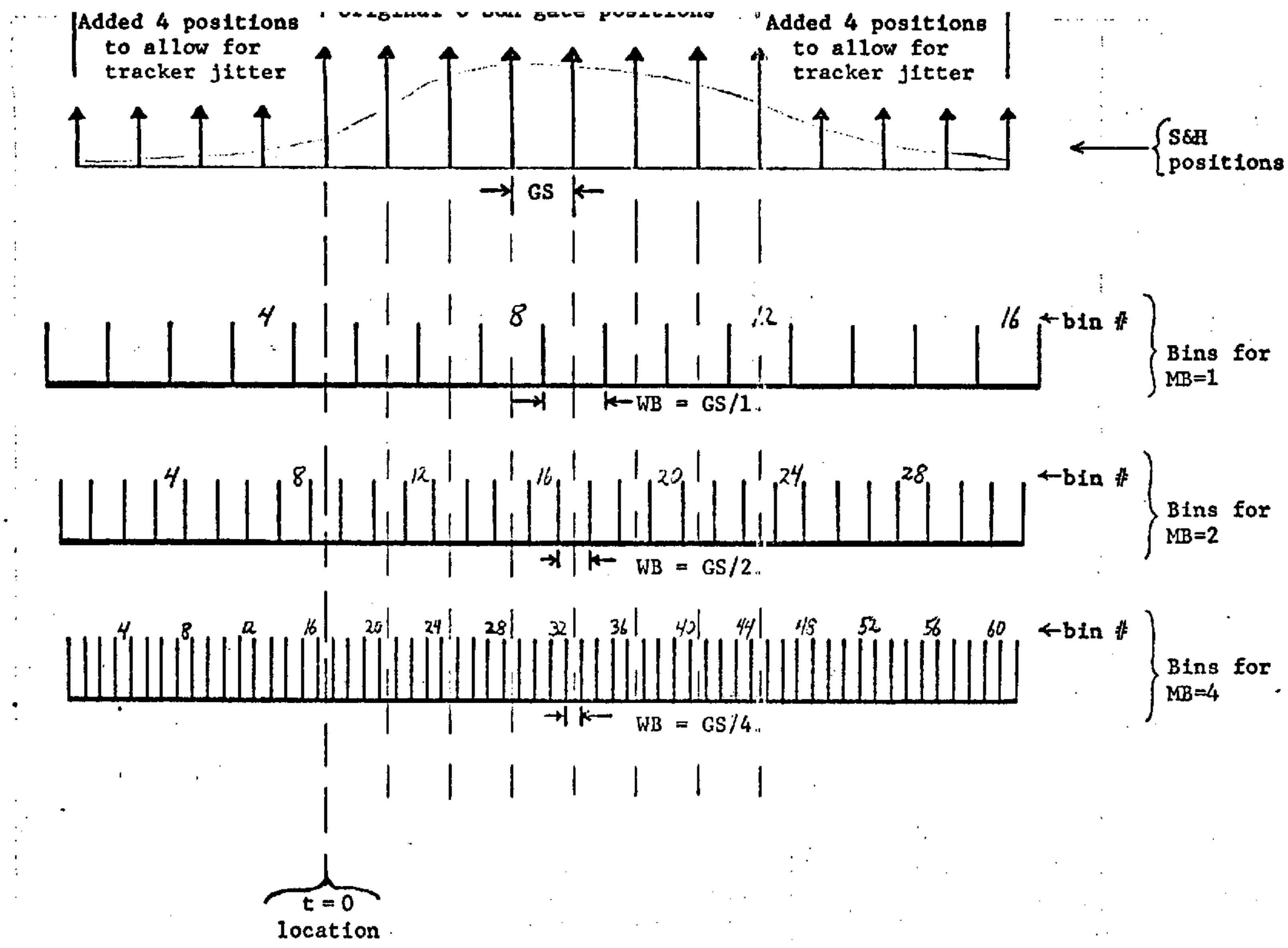
The total number of "vertical averaging" bins is (15*MB)+1, and VTADD allows for MB$\leq$10, so that the maximum number of bins, at MB=10, is 151.

If one uses JSO as an index to denote the original S&H gate numbers (JSO=1,...,8), and JMB as the index to denote the bin number corresponding to JSO for a specified MB, the relation between JSO and JMB is

JMB=(JSO+3)*MB+1 .

Also, relative to a local time origin t=0 located at the position of the first of the original gate positions, the time at the center of any specified bin JMB is given by

time=(JMB-4*MB-1)*GS/MB .



Sketch of Time and Position Relations among VTADD Bins

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR