NASA Technical Memorandum 73293

NOSS Altimeter Detailed Algorithm Specifications

(NASA-TM-73293) NOSS ALTIMETER DETAILED ALGORITHM SPECIFICATIONS (NASA) 199 p HC A09/MF A01 CSCL 17I
N82-24605
Unclas
G3/43 09955

D. W. Hancock, III

and

J. D. McMillan

March 1982

National Aeronautics and
Space Administration

Goddard Space Flight Center
Wallops Flight Center
Wallops Island, Virginia 23337

NASA Technical Memorandum 73293

NOSS Altimeter Detailed Algorithm Specifications

D. W. Hancock, III

NASA Goddard Space Flight Center
Wallops Flight Center
Wallops Island, Virginia 23337

and

J. D. McMillan

EG&G Washington Analytical Services Center
P. O. Box 476
Pocomoke City, Maryland 21851

National Aeronautics and
Space Administration

Goddard Space Flight Center
Wallops Flight Center
Wallops Island, Virginia 23337

NOSS ALTIMETER DETAILED ALGORITHM SPECIFICATIONS

by

David W. Hancock, III
NASA Wallops Flight Center
Wallops Island, VA 23337

and

James D. McMillan
EG&G Washington Analytical Services Center
P.O. Box 476
Pocomoke City, MD 21851

August 1981

TABLE OF CONTENTS

ACRONYMS AND SYMBOLS

AAFE	Advanced Applications Flight Experiments
AGC	Automatic gain control
ALT	Altimeter
Cal	Calibration
CPU	Central Processor Unit
CW	Continuous wave
CZCS-2	NOSS Coastal-Zone Color Scanner
EM	Electromagnetic
EU	Engineering Unit
FNOC	Fleet Numerical Oceanography Center
GDR	Seasat Geophysical Data Record
GEOS	Geodynamic Experimental Ocean Satellite (GEOS-3)
GSFC	Goddard Space Flight Center
JPL	Jet Propulsion Laboratory
LAMMR	Large Antenna Multichannel Microwave Radiometer
MSL	Mean sea level
NOAA	National Oceanic and Atmospheric Administration
NOSS	National Oceanic Satellite System
PPF	NOSS Primary Processing Facility
PRF	Pulse Repetition Frequency
SACU	Synchronizer Acquisition Calibration Unit
S/C	Spacecraft
SCATT	Scatterometer
SDR	Seasat Sensor Data Record
Seasat	Seasat Spacecraft
SF	Seasat Sensor File
SSH	Sea-Surface Height
SWH	Significant Waveheight
TBD	To Be Determined
TM	Telemetry Mode
TWT	Traveling-Wave Tube
WFC	Wallops Flight Center
σ°	Backscatter Cross-Section Per Unit Scattering Area
[ ]	Maximum Integer Function

INTRODUCTION

This document contains a detailed description of the NOSS altimeter algorithms and data sets. The algorithm/data set numbering scheme is

$$X.Y(S) - N.\{n\}$$

X - Sensor
    A = Altimeter
    C = CZCS-2
    L = LAMMR
    S = Scatterometer
Y - Type
    A = Algorithm
    D = Data Set
S - Source (in this report, if the source is omitted from the reference number, the altimeter is the implied source)
    A = Altimeter
    C = CZCS-2
    L = LAMMR
    S = Scatterometer
    M = Mission contractor
N Level
    1 = Level 1
    2 = Level 2

n Algorithm/Data Set Number (as many levels of this number as needed may be used; e.g., $X.Y(S) - N.n_1.n_2.n_3$).

Alternative algorithms or data are indicated by a letter after the number (e.g., X.Y.(S) - 1.2A; X.Y(S) - 1.2B).

Note that many coefficients and table entries presented in this document are taken from the documentation of the Seasat software and are therefore subject to change as the NOSS instrument package is developed.

MANAGEMENT SUMMARY

The purpose of the NOSS Altimeter Detailed Algorithm Specifications is to document the details of the algorithms and data sets presented in the NOSS Algorithm Freeze Report, Volume 1 in a form suitable for 1) development of the benchmark software by the Data Processing and Analysis Section at NASA Wallops Flight Center and 2) delivery to the spacecraft contractor as a guide for coding the operational software.

This document was to be the result of a two-year algorithm development effort to completely define the NOSS operational software well in advance of launch. The NOSS program was cancelled after a six-month start on this report. Therefore the algorithms reported in detail are ones which are established altimeter processing. The algorithms which required some additional development before documenting for production have only been scoped. This was necessary since the required analysis effort was not funded. The following processing description is taken from the NOSS Altimeter Algorithm Specification (Ref. 1).

The level 1 processor converts the data to engineering units and applies first order corrections to the data for known instrument variations. Also level 1 quality control monitoring is done in the calibration module and the health status module. Both of these modules produce reports that require engineering evaluation. No calibration tables are automatically modified because users require a constant data set with updates only when significant changes have occurred. Required external inputs to level 1 are the spacecraft ephemeris data, the LAMMR update file, and world surface map data. Some of the table entries will be determined after post launch calibration experiments. The calibration trend file is an important internal file which allows for the analysis of altimeter characteristics over long periods. The resulting level 1 output file contains all altimeter data at full rate with all instrument applied corrections and status flags on instrument health. By removing the corrections and knowing the conversion factors the original level 0 data can be recovered if needed.

The level 2 processor provides quality geophysical measurements derived from altimeter parameters to oceanographic users. In addition, it will provide ice sheet measurements to the ice user community. Its output data rate and content will be compressed to (nominally) once per second. The altimeter parameters significant waveheight, wind speed and surface height, are basically direct calculations. Additional products will be ocean waveheight distribution skewness, dominant wavelength, significant slope, rain rate, ocean backscatter, and sea-ice boundary. Ice-sheet height, surface slope, and roughness will be calculated over ice sheets. In order to provide quality products some of the corrections require iterative processing and several external files are required as input to determine the best corrections. The parameters required from these files are the LAMMR level 2 pathlength correction, FNOC atmospheric pressure, LAMMR T_B, ionospheric electron density, solar and lunar ephemerides, geoid, and tide. The resulting level 2 output file contains only geophysical data and associated corrections. By maintaining the corrections on file an individual user may apply variations from his own research.

OVERVIEW

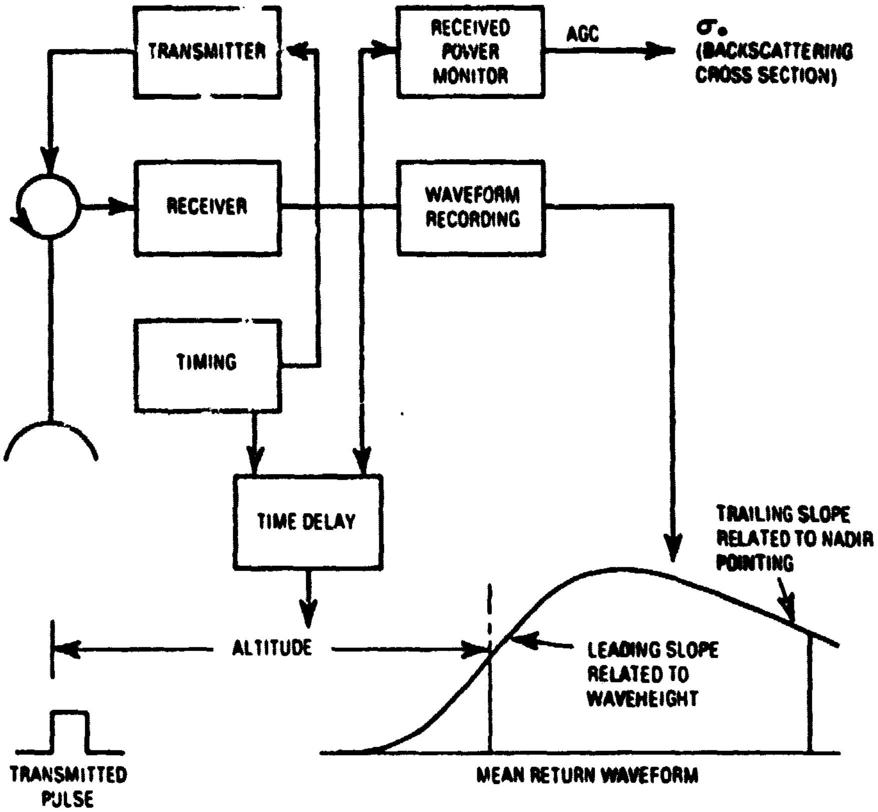
The background and objectives for the NOSS altimeter are defined in Reference 1. It is repeated here that functionally, the altimeter measures the spacecraft height above mean sea level (MSL), and the significant waveheight (SWH) and backscatter coefficient ($\sigma°$) of the ocean surface beneath the spacecraft (Ref. 2). The altimeter is a 13.56 GHz monostatic radar system that tracks in range only using a 1 m parabolic antenna pointed at the satellite nadir. Its high resolution, coupled with a high transmitted pulse rate of 926 Hz, permits the realization of 10 cm altitude precision.

The basic idea behind satellite altimetry is to utilize the highly stable platform provided by a spacecraft as a moving reference system from which vertical measurements to the ocean surface are made (Ref. 3). Referring to Figure 1, altimeter systems provide three measurements:

(1) Altitude - The elapsed time between the time of transmission of an RF pulse of energy and its reception back at the altimeter, after having been scattered from the ocean surface below, is essentially a measurement of the height of the satellite above mean sea level. When merged with accurate orbital information, the results can be related to changes in mean sea level due to such spatially varying quantities as gravity anomalies and such time varying quantities as tides, winds, and currents.

(2) Return Pulse Shape (Waveform) - The slope and duration of the leading edge of the return pulse can be related to the significant waveheight of the ocean surface below. In addition, through a deconvolution process, the surface height distribution can be recovered, including its skewness. It has been shown (Ref. 4) that skewness can then be related to such additional oceanographic parameters as dominant wavelength, swell/sea ratio, etc. Finally, the slope and duration of the trailing edge of the return pulse can be related to the attitude of the satellite (angle of the measurement axis with respect to the subsatellite point).

(3) Return Pulse Amplitude - The amplitude of the return pulse, which is determined from the AGC used to normalize the incoming waveform, can be related to the backscatter coefficient ($\sigma°$) of the surface below, which in turn, can be related to wind speed over the ocean as well as certain ice related parameters.



	APPLICATIONS
σ°	ICE EDGES PERCENT ICE COVER WIND SPEED
WAVEFORMS	LAND SURFACE ROUGHNESS WAVE HEIGHTS (SWH) SWELL FREQUENCY SPECTRA INFORMATION SATELLITE POINTING
ALTITUDE	STORM SURGES AND TIDES OCEAN GEOID OCEAN CURRENTS LAND AND ICE CONTOUR MAPS CROP INFORMATION

Figure 1. Block Diagram Depicting Satellite Altimetry Concept and Potential Applications.

Key Assumptions

In the development of the altimeter processing algorithms, the following key assumptions have been made:

1. Because of the similarity between the proposed NOSS altimeter instrument and the Seasat altimeter instrument and because many of the altimeter algorithms required by the NOSS processing software were developed for Seasat and verified in an operational (although not real-time) environment, many of the Seasat altimeter algorithms have been adopted for use by the NOSS altimeter processing software.
2. The similarity between the NOSS altimeter and the Seasat altimeter permits the adoption of Seasat calibration and processing tables for use as a starting point in developing those tables for the NOSS altimeter. All tables and constants, whether determined from Seasat documentation, from documentation of other altimeter instruments (i.e., GEOS-3), or some other source, must be flexible until the final NOSS values are determined. A few of these table entries will not be defined until well after the launch of the spacecraft.
3. In order to facilitate the generation of the benchmark software by the Data Processing and Analysis Section of the Wallops Flight Center and because of the general acceptance of the FORTRAN language in the scientific community, all altimeter algorithms have been expressed in FORTRAN-like instructions. However, these instructions should not necessarily be assumed to be in optimum programming form.
4. All data required from the input files (i.e., ephemeris files, FNOC file, LAMMR level 2 file, etc.) must be available and current as the altimetry processing software requires it. Failure to supply any of the input files must not result in the abnormal termination of the altimeter processing software, but may produce degraded output products that will be flagged as such.
5. All input data files required by the altimeter processing software, as well as all output reports generated by the altimeter processing software, will be maintained at the PPF on data-storage devices for a minimum of one week on a daily rotating basis. The trend file and solar/lunar ephemeris file will be maintained for one year on a monthly rotating basis.
6. Because the spacecraft is designed to have two altimeters that, although physically similar, may have different electronic characteristics, all software must have the capability for processing two independent altimeters. This will require two sets of input tables, two sets of output reports, and documentation stating which of the two altimeters was used in taking the data.

7. The prefix "instrument," as used in this report, relates to the altimeter housekeeping parameters that are supplied directly from the spacecraft. For example, the "instrument voltage" is the altimeter voltage as detected by the spacecraft monitor.
8. All processing of altimeter data will be handled by using data stored record by record. This will mean that a level 0 input data record will be read into core and modified as the appropriate altimeter processing algorithms are accessed. After all modules have been processed, the data record will be transferred from core to the output file.
9. The following algorithms have only been scoped:

A.A-1.1.8	EU Rain Subcom
A.A-1.1.9	EU Waveform
A.A-1.3.3	Trend File Processing
A.A-1.4	Adaptive Resolution
A.A-1.5.1	Spacecraft Ephemeris Interpolation
A.A-1.5.2	Subsatellite Point Calculation
A.A-2.1	Contamination Processing
A.A-2.3.2	Waveform Altitude Correction
A.A-2.3.3	Waveform SWH Correction
A.A-2.3.4	Waveform SWH Bias
A.A-2.4	Atmospheric Corrections
A.A-2.6.3.1	Solar/Lunar Ephemeris Interpolation
A.A-2.6.4	EM Bias
A.A-2.7	Ice Sheet Height
A.A-2.8	Sea Ice
A.A-2.9	Quality Control

10. Subsystem interfaces are handled by the Mission Contractor so that the required data are colocated in time and space for meeting the requirements of the altimeter algorithms.

SUBSYSTEM DESCRIPTION

General

This section summarizes the logic of the altimeter level 1 and level 2 subsystem software. Included are the altimeter subsystem flow charts, a narrative description of each of the major modules, and a description of the subsystem interfaces with other PPF software. Tables I and II provide an index to the level 1 and level 2 modules and data sets.

Figures 2 and 3 show the NOSS altimeter software subsystem for level 1 and level 2, respectively. Figures 4 and 5 are more detailed for level 1 and level 2, respectively, showing the submodules development status. The shaded submodules need more development and are not given in detail in this report. A brief description of each of the software modules follows in the Narrative Description, and Data Description. The Module Descriptions contain detailed descriptions of each of the modules, submodules, and data sets.

Narrative Description

The following paragraphs provide a general description of each of the major modules of the NOSS altimeter processing software, indexed by the algorithm reference numbers (see Tables II and III).

Level 1 Components

1. 1.1--Engineering Unit Conversion - This module will convert the counts in the telemetry data stream to engineering (functional) units. The input to this module will be the raw (level 0) data in counts, and the output will be altimeter parameters and housekeeping data, expressed in engineering units, to be used by the other modules of the altimeter processing software.

   The method of conversion to engineering units will vary with different parameters. Some conversions will simply require a table look-up or a temperature correction, whereas others will require the evaluation of a polynomial. It should be noted, however, that not all parameters in the telemetry stream require conversion to engineering units because some parameters are merely flags whose bits are used to determine a status or mode of operation.

2. 1.2--Level 1 Altitude Module - This module will calculate certain sensor-related corrections to the altitude and automatic gain control (AGC). These corrections are performed in the following submodules: (1) 1.2.1--Altitude time-tag corrections; (2) 1.2.2--Altitude calibration zone bias; (3) 1.2.3--Altitude center-of-gravity offset; and (4) 1.2.4--Calibration mode bias. The altitude time-tag corrections are functions of track mode and altitude. The calibration zone bias will correct the altitude measurements to a common datum using information derived from data that are taken directly over laser tracking stations in the calibration area. The center-of-gravity offset will account for the location of the altimeter antenna with respect to the spacecraft center of gravity and the expenditure of the onboard fuel, and the calibration mode bias compensates for differences determined by comparing calibration mode data with preflight tables.

TABLE I. LEVEL 1 MODULES AND DATA SETS

Identifier	Title
Modules	
A.A-0.0	Altimeter Main Driver
A.A-1.0	Level 1 Driver
A.A-1.1	Engineering Units Conversion
A.A-1.1.1	EU Date and Time
A.A-1.1.1.1	EU Bit Extraction
A.A-1.1.2	EU Altitude, SWH, and AGC
A.A-1.1.2.1	EU Polynomial Fit
A.A-1.1.3	EU Gate Amplitude
A.A-1.1.4	EU Status Extraction
A.A-1.1.5	EU Engineering Subcom #1
A.A-1.1.6	EU Engineering Subcom #2
A.A-1.1.7	EU Engineering Subcom #3
A.A-1.1.8	EU Rain Subcom
A.A-1.1.9	EU Waveform, CW, or Dump
A.A-1.2	Level 1 Altitude Correction
A.A-1.2.1	Time Tag Correction
A.A-1.2.2	Cal Zone Bias
A.A-1.2.3	Center of Gravity Offset
A.A-1.2.4	Cal Mode Bias
A.A-1.3	Cal Mode Driver
A.A-1.3.1	Cal 1
A.A-1.3.1.1	Cal Mode Statistical Accumulation
A.A-1.3.2	Cal 2
A.A-1.3.3	Trend Processing
A.A-1.4	Adaptive Resolution
A.A-1.5	Location Processing
A.A-1.5.1	Spacecraft Ephemeris Interpolation
A.A-1.5.2	Subsatellite Point Calculation
A.A-1.6	Health/Status Monitor Driver
A.A-1.6.1	HS 1-Day Wrap-Up
A.A-1.6.1.1	HS Statistical Accumulation
A.A-1.6.2	HS n-Minute Wrap-Up
A.A-1.6.3	HS Status
A.A-1.6.4	HS Non-Subcom
A.A-1.6.5	HS Engineering Subcom #1
A.A-1.6.6	HS Engineering Subcom #2
A.A-1.6.7	HS Engineering Subcom #3
Data Sets	
A.D-1.71	Trend File
A.D-1.72	EU Coefficient File
A.D(M)-1.81	Level 0 Date File
A.D(M)-1.82	Spacecraft Ephemeris File
A.D(M)-1.83	World Surface Map File
A.D(L)-1.84	LAMMR Update File
A.D(M)-1.85	Executive Parameter File
A.D-1.91	Calibration Report
A.D-1.92	Trend Report
A.D-1.93	Altimeter Diagnostics
A.D-1.94	Level 1 Output File

TABLE II. LEVEL 2 MODULES AND DATA SETS

Identifier	Title
Modules	
A.A-2.0	Level 2 Driver
A.A-2.1	Contamination
A.A-2.2	Data Compression
A.A-2.3	Waveform Module
A.A-2.3.1	Waveform Processor Driver
A.A-2.3.1.1	Derivative
A.A-2.3.1.1.1	Convolution
A.A-2.3.1.1.1.1	Surface Elevation Distribution
A.A-2.3.1.1.1.2	Flat Sea Response
A.A-2.3.1.1.1.3	Fast Fourier Transform
A.A-2.3.1.1.1.4	Fast Fourier Synthesizing
A.A-2.3.1.2	Matrix Inversion
A.A-2.3.2	Altitude Correction
A.A-2.3.3	SWH Correction
A.A-2.3.4	SWH Bias
A.A-2.3.5	Waveform Products
A.A-2.4	Atmospheric Corrections
A.A-2.4.1	Rain Gate
A.A-2.4.2	Barotropic Effect
A.A-2.4.3	Ionospheric Correction
A.A-2.4.4	Dry Tropospheric Refraction
A.A-2.4.5	Wet Tropospheric Refraction
A.A-2.4.6	Radar Backscatter Coefficient
A.A-2.5	Wind and Radar Backscatter Driver
A.A-2.5.1	Radar Backscatter Coefficient
A.A-2.5.2	Wind Speed
A.A-2.6	Level 2 Altitude Correction
A.A-2.6.1	Geoid Height
A.A-2.6.1.1	Bilinear Interpolation
A.A-2.6.2	Tide Height
A.A-2.6.3	Solid Earth Tide Height
A.A-2.6.3.1	Solar/Lunar Ephemeris Interpolation
A.A-2.6.4	EM Bias
A.A-2.6.5	Sea Surface Height
A.A-2.7	Ice Sheet Height
A.A-2.8	Sea Ice
A.A-2.9	Quality Control
Data Sets	
A.D-2.71	Geoid File
A.D-2.72	Tide File
A.D(M)-2.81	FNOC File
A.D(M)-2.82	Ionospheric Data File
A.D(M)-2.83	LAMMR Level 2 File
A.D(M)-2.84	LAMMR T_B File
A.D(M)-2.85	Solar/Lunar Ephemeris File
A.D-2.91	Level 2 Output File
A.D-2.92	Wind σ° Output File

ORIGINAL PAGE IS
OF POOR QUALITY

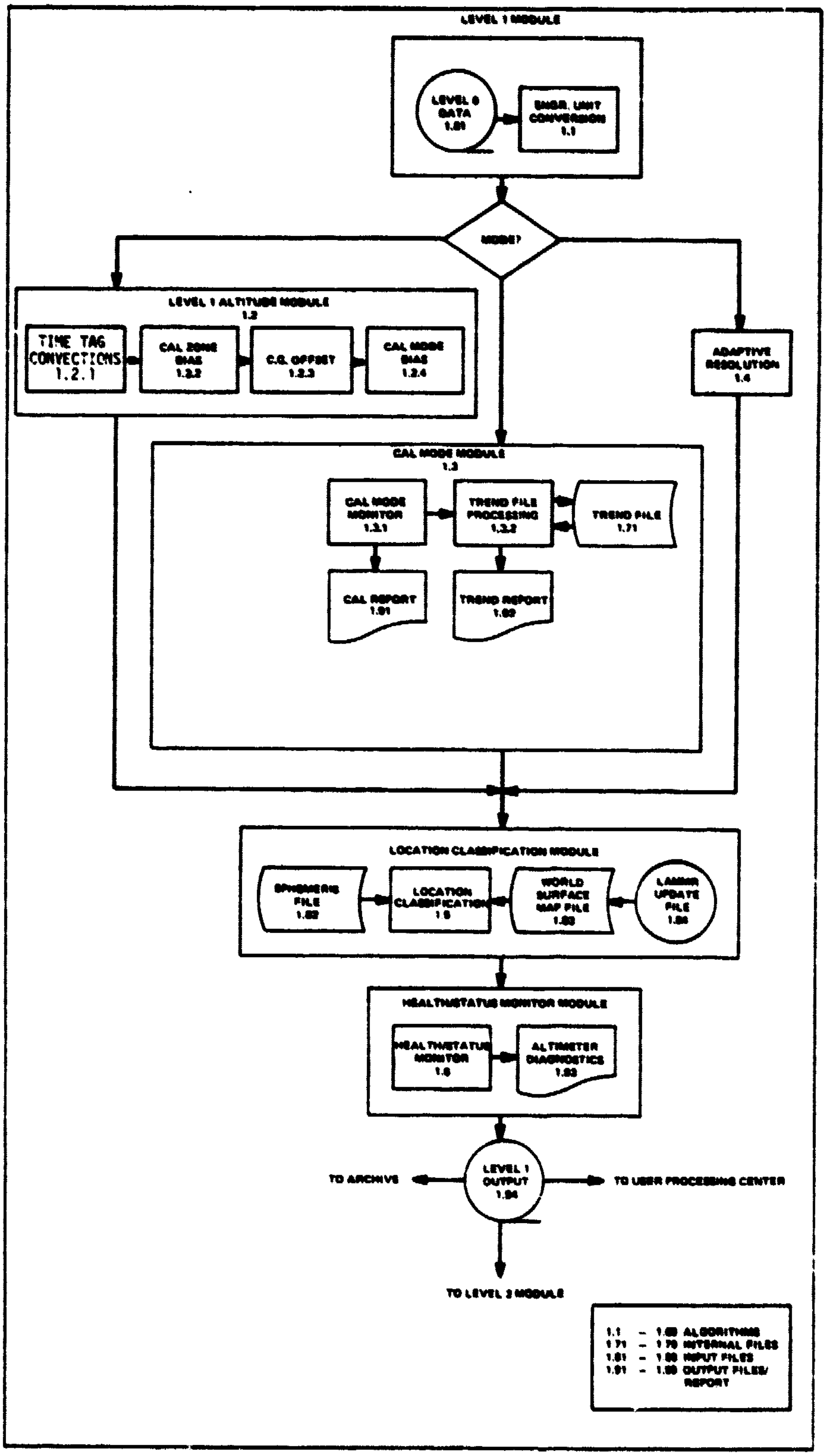


Figure 2. Altimeter Subsystem Level 1 Flow Chart.

ORIGINAL PAGE IS
OF POOR QUALITY

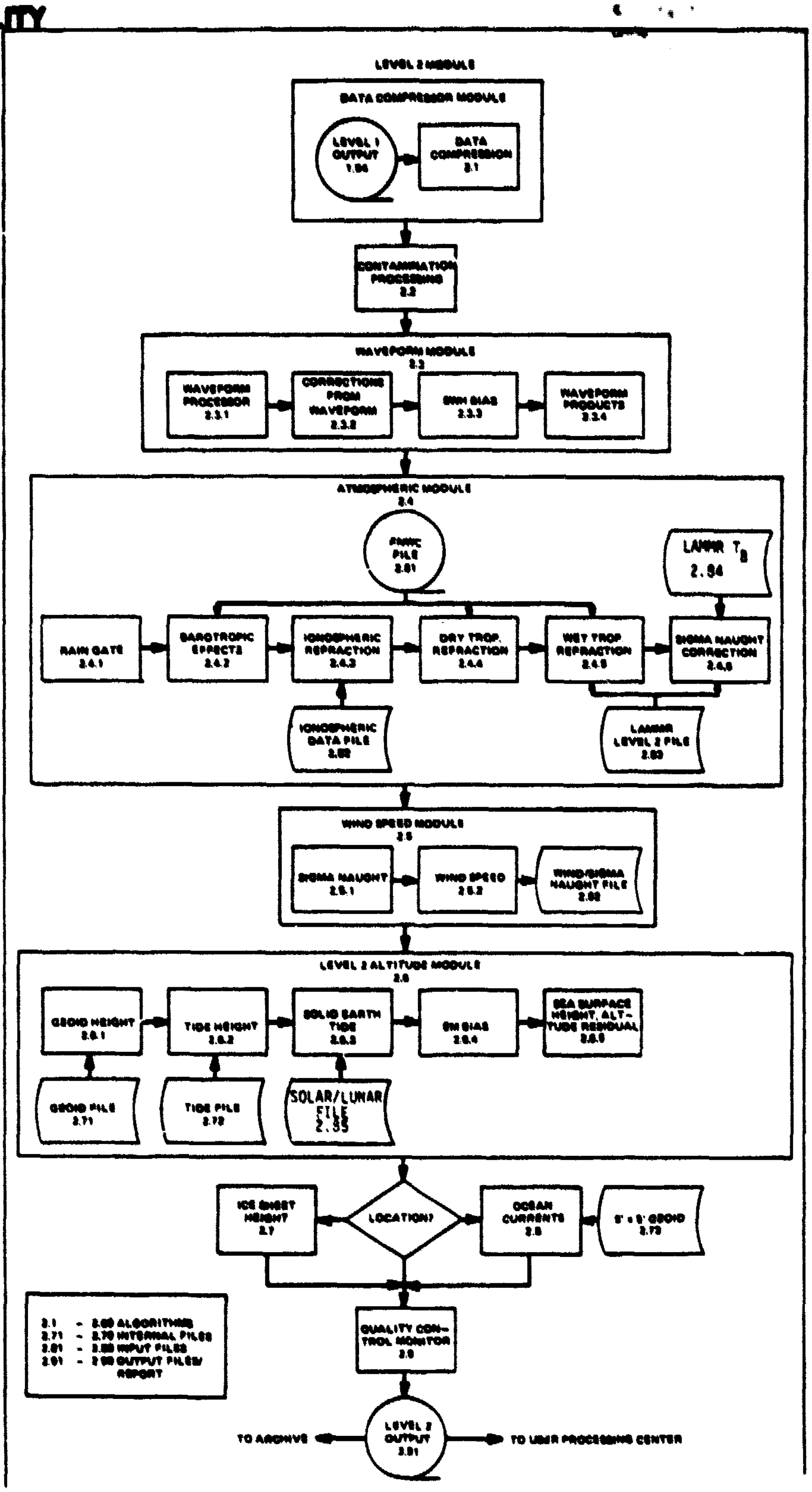


Figure 3. Altimeter Subsystem Level 2 Flow Chart.

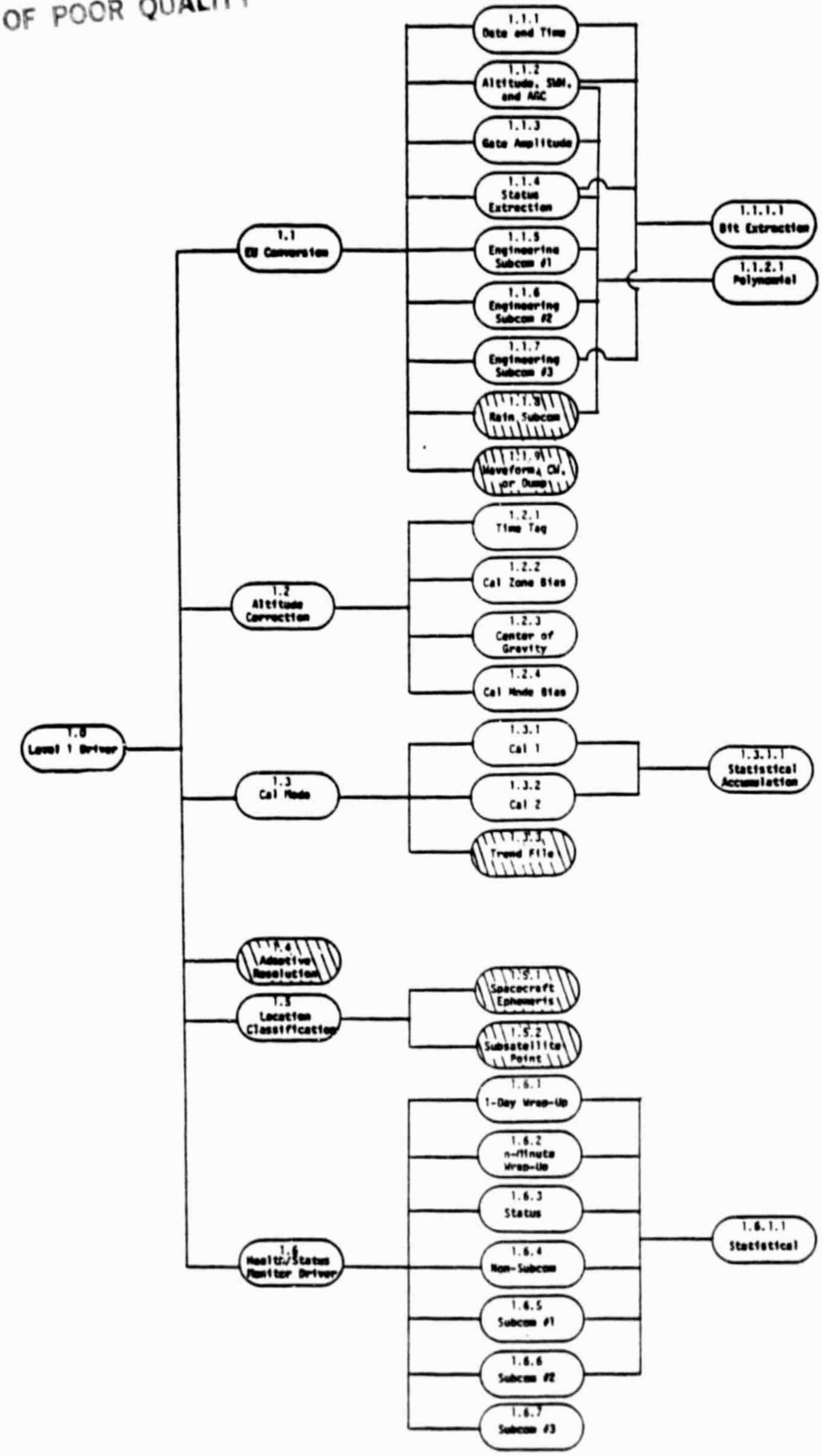


Figure 4. Altimeter Level 1 Status Chart.

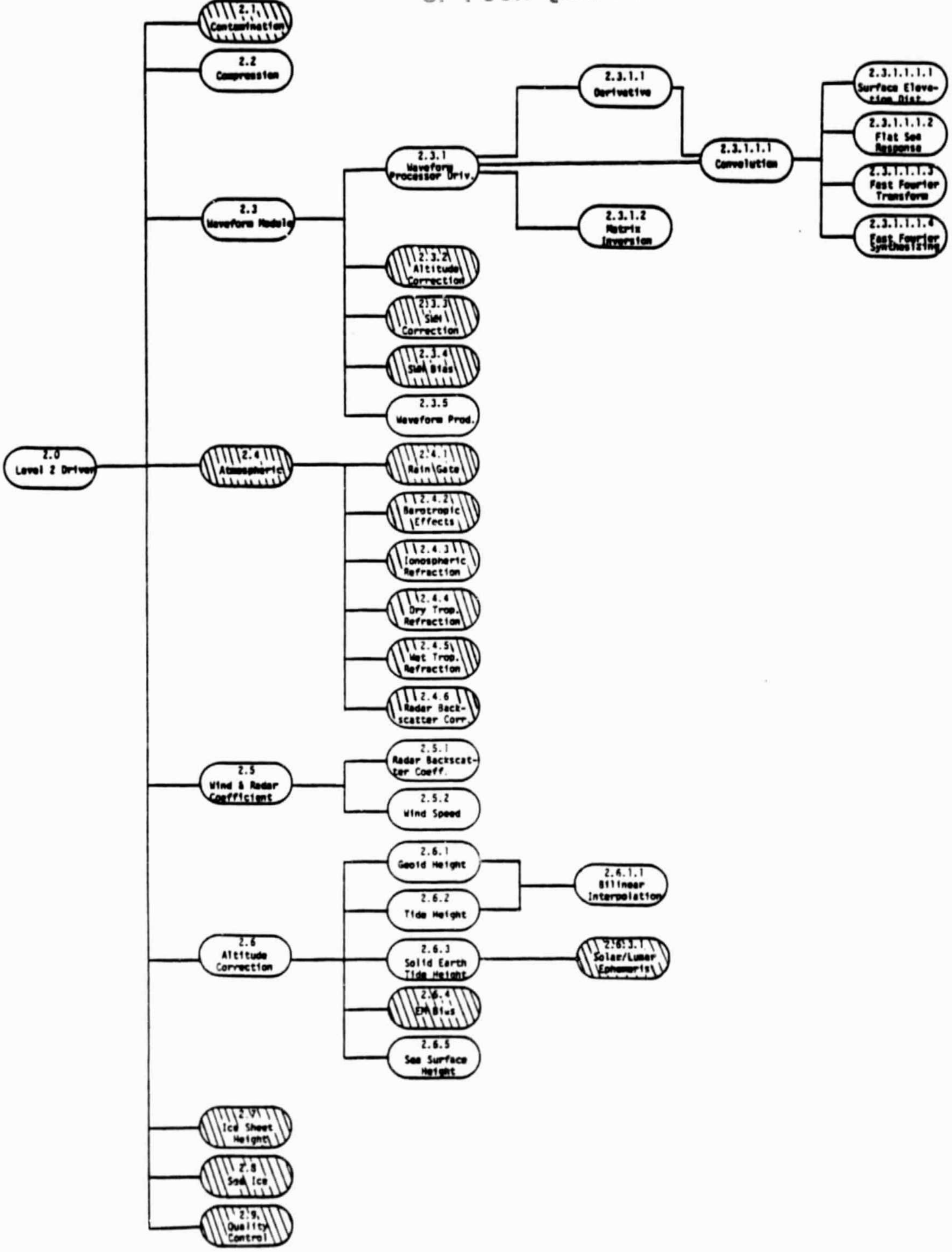


Figure 5. Altimeter Level 2 Status Chart.

Note that the time-tag corrections must be performed immediately after the engineering unit conversion so that all ephemeris data will be correctly interpolated in the location classification module.

3. 1.3--Calibration (Cal) Mode Module - The altimeter processing software will monitor all calibration mode data in this module, which contains three submodules: (1) 1.3.1--Cal 1 processor; (2) 1.3.2--Cal 2 processor; and (3) 1.3.3--Trend File processor. The altimeter has internal calibration modes to detect changes in altitude, AGC, and other parameters attributable to aging, temperature, voltage fluctuation, etc. This mode will be employed for 60 seconds about once per day. The cal mode module will process the calibration mode data and generate calibration reports of comparisons with preflight nominal calibrations. These reports will require Mission Contractor interpretation of flagged changes to be significant enough to justify updates to the cal mode bias submodule tables.

   The trend file processor is designed to identify long-range (one month to one year) trends in the calibration mode data. To this end, a file of trends will be automatically maintained by the altimeter processing software, and significant changes in altimeter parameters will be calculated and flagged. A human decision of the appropriate action to be taken to correct the flagged parameters will be required. It may take two to three weeks to verify that the characteristic is valid and, if valid, to implement a correction. Any faster changes will be detected by health status monitoring.

4. 1.4--Adaptive Resolution - Adaptive resolution is a mode in which the altimeter has detected surface slope changes and automatically switched to a wider pulse width and different track constants to maintain lock. This will occur primarily over ice sheets and land. This module will correct time tags, AGC, and altitude in the adaptive resolution mode for offsets due to the selected pulse width and tracker characteristics. These corrections will make the level 1 output products consistent with the normal track mode. In addition, CW mode data will be processed in this module. Although CW is normally used for acquisition only, the system can be commanded to this mode for continued operation.

5. 1.5--Location Classification Module - This module will classify the subsatellite point as either land, water, or ice based on a world surface map. The LAMMR subsystem will update the ice fields of the world surface map. In addition, this module will merge and interpolate the satellite ephemeris data in order to calculate the latitude and longitude of the subsatellite point and the spacecraft height above the reference ellipsoid.

6. 1.6--Health/Status Monitor - This module will automatically monitor critical instrument parameters and set off system alarms when a potentially damaging or dangerous condition is observed. It will set quality flags for altitude, tem-

perature, voltage, and current to be output to the health/status monitor report and to the level 1 output data file. The analysis of this output will require Mission Contractor interpretation and interaction. The protection of the altimeter will be the responsibility of the control center. This monitor will be a backup of control-center decisions and will assist in early alert of trends. The output should be reviewed daily.

Level 2 Components

1. 2.1--Contamination Processing - This module will classify the nature of the subsatellite point as either land, water, or ice. This classification, which is derived from analysis of the altimeter data with a resolution of approximately 1 km, should not be confused with the classification performed in the location classification module, which is derived from a world surface map that has a resolution of approximately 10 km. The software will estimate the time and location of land/water and ice/water interfaces. In addition, a flag will be set to indicate if the data are appropriate for processing by the waveform processor.
2. 2.2--Data Compression - The altimeter level 1 output data rate is 20 frames per second. The data compression software will smooth the data to a selectable rate (nominally once per second) and will edit nonproduction data modes such as calibration mode, trigger kill, and standby. The rate is variable and selected by the Project. The software will also calculate standard deviations for most parameters for subsequent quality analysis.
3. 2.3--Waveform Module - This module will process waveform data in order to calculate significant waveheight (SWH), attitude, and skewness, as well as altitude corrections and certain ocean-wave parameters. The calculation of the SWH, attitude, and skewness parameters requires an iterative calculation of a best-fit solution to the waveform data that is described in Ref. 4. That calculation is the convolution of the antenna pattern, surface distribution, and radar pulse. If the solution fails to converge, then the SWH calculated onboard will be used with backup table bias corrections to SWH and altitude.
4. 2.4--Atmospheric Module - This module will calculate the atmospheric corrections to the spacecraft altitude and the radar backscatter coefficient. It will also process data from the rain gate. The altitude corrections from this module consist of the combined effects of ionospheric refraction, wet and dry tropospheric refraction, and atmospheric pressure. The prime wet tropospheric refraction correction will employ data from a file built by LAMMR algorithms. FNOC data will be used when LAMMR data are not available.

   The radar backscatter coefficient correction accounts for the effects of the atmosphere on return power. This correction will be based on data from the

LAMMR T_B file and will use SCATT-supplied algorithms. The correction will not be applied until the validity of the LAMMR data has been established (after launch). The rain-gate processing detects the presence of rain at the subsatellite point and is used for σ° correction.

5. 2.5--Wind-Speed Module - This module will calculate the radar backscatter coefficient and the ground wind speed. The calculated radar backscatter coefficient (σ°) is a function of AGC, altitude, and attitude, and the altimeter estimated wind speed is a function of σ°. Note that the atmospheric correction to σ° for rain will be applied before the wind speed is calculated.
6. 2.6--Level 2 Altitude Module - This module will correct the altitude measurements for electromagnetic (EM) bias, using derived coefficients that account for the difference between the radar-observed sea-surface height distribution and the geometrical sea-surface height. This module will also calculate the sea surface height and the altimeter residual. In order to calculate these two parameters, it is necessary to evaluate the geoid height, the ocean tide height, and the solid Earth tide height.
7. 2.7--Ice-Sheet Height - This module will be employed only over areas of ice interest. The altimeter data will be corrected for waveform shape changes that cause track-point shifts. This correction will be done by a software retracking process designed for ice-sheet processing.
8. 2.8--Sea Ice Products - This module computes the sea ice-related products mean-squared slope and percent smooth area. In addition, special retracking will be performed to reduce the noise on the altitude measurement over sea ice.
9. 2.9--Quality-Control Monitor - As a final step in the level 2 processing, the altimeter software will analyze the contents of the level 2 output file in order to classify the quality of the data. The data will be flagged as being of questionable quality when: (a) prescribed standard deviation tolerances are exceeded, (b) the number of rejected points in the various smoothing algorithms exceeds acceptable limits, or (c) operational threshold limits are exceeded.

Data Interface Descriptions (Summary)

This section describes the subsystem interfaces between the altimeter processing software and other NOSS PPF software. These interfaces are in the form of input files and output files and reports, which are described in more detail in (Ref. 1).

The subsystem interfaces are as follows:

(a) A.D-1.71--Trend File - To maintain a history file of the calibration mode data in order to assist in the identification of long-range trends in that data. The file will be read from and written to by the trend file processor, A.A-1.3.2.

(b) A.D-1.72--EU Coefficient File - To provide engineering unit conversion coefficients for both altimeters. It is used to load the operating altimeter's coefficients into standard arrays for the data processing. The mission contractor is to supply this file.

(c) A.D(M)-1.81--Level 0 Data File - This file is supplied to the altimeter software by the Mission Contractor and contains all of the raw altimeter data in counts. Appendix A gives the expected altimeter telemetry contents.

(d) A.D(M)-1.82--Spacecraft Ephemeris File - This file is supplied by the Mission Contractor and contains the spacecraft ephemeris information needed to accurately identify the position of the spacecraft.

(e) A.D(M)-1.83--World Surface Map File - This file is supplied by the Mission Contractor and contains the locations of the land/water and ice/water boundaries. The file is updated by the LAMMR update file, A.D(L)- 1.84.

(f) A.D(L)-1.84--LAMMR Update File - This file supplied by the LAMMR processing software is used to update the ice-field locations. The Mission Contractor will handle the interface between A.D(L)-1.84 and A.D.(M)-1.83.

(g) A.D(M)-1.85--Executive Parameter File - This file is used by the user to select various program options before starting job execution.

(h) A.D-1.91--Calibration Report - This report is generated by the altimeter processing software to summarize the calibration mode data for human interpretation to determine if any requirements exist to update parameter calibration tables.

(i) A.D-1.92--Trend Report - This report is generated to identify long-range trends in the altimeter data for human interpretation to determine if any requirements exist to update parameter calibration tables.

(j) A.D-1.93--Altimeter Diagnostic Report - This report is generated to identify altimeter parameters that have exceeded tolerances and require human interpretation to decide if the operational status of an altimeter must be modified.

(k) A.D-1.94--Level 1 Output File - This file is supplied by the altimeter processing software to the PPF for archiving, level 2 processing, and users. It contains all level 1 altimeter data. The Mission Contractor is responsible for the formatting of this file. Table III lists the contents of the altimeter level 1 output record. Also the data in Table A-2 are output once per second.

(l) A.D-2.71--Geoid File - It provides global geoid-height estimates to be written on the level 2 output file, A.D-2.91, which are used in the calculation of the altimeter residual, A.A-2.6.5. The file is read by the geoid height module, A.A-2.6.1. Numerous geoid models are currently available with more expected by the mid 1980's. Most of the geoid models that are currently accepted as the most accurate have been generated by, and are available from, GSFC.

TABLE III. ALTIMETER LEVEL 1 OUTPUT RECORD CONTENT*

Name	Length (bytes)
Time	8
Orbit number	2
Latitude	4
Longitude	4
Spacecraft altitude	8
Solar latitude	4
Instrument attitude	2
World classification	1
Zone flag	2
Altimeter number/mode	1
Health flags	6
Altitude	8
Altitude rate	4
Altitude error	4
SWH	2
AGC	2
Delta time	2
Delta altitude for CG	2
Delta altitude for CZ	2
Delta altitude for CM	2
Delta AGC for CM	2
Delta altitude for AR	2
Delta AGC for AR	2
TM words 12 to 93	164
Solar position vector	24
Lunar position vector	24
Spare	28
Total	316

* Plus special record once per second (see Table A.2).

(m) A.D-2.72--Tide Height - It provides global ocean-tide height estimates to be written on the level 2 output file A.D-2.91, which are used in calculating the altimeter residual, A.A-2.6.5. The file is read by the tide height module, A.A-2.6.2. This comes from Government-furnished information developed by E. W. Schwiderski, Naval Surface Weapons Center, Dahlgren, Virginia.

(n) A.D(M)-2.81--FNOC File - This file is supplied by the Mission Contractor and contains meteorological data used in calculating altitude corrections.

(o) A.D(M)-2.82--Ionospheric Data File - This file is supplied to the altimeter software by the Mission Contractor and contains sunspot and solar flux data for calculating ionospheric refraction. Ref. 5 contains more detailed information.

(p) A.D.(L)-2.83--LAMMR Level 2 File - This file is supplied by the LAMMR processing software and contains data for calculating the wet tropospheric refraction altitude correction and the sigma-naught atmospheric correction.

(q) A.D(L)-2.84--LAMMR T_B File - This file is supplied by the LAMMR processing software and contains information relating to the sigma-naught atmospheric correction. Ref. 6 contains a more detailed description.

(r) A.D(M)-2.85--Solar/Lunar Ephemeris File - This file is supplied by the Mission Contractor and contains position vector of the sun and the moon as a function of time.

(s) A.D-2.91--Level 2 Output File - This file is supplied by altimeter processing software and contains all level 2 altimeter data in corrected geophysical form. The Mission Contractor is responsible for the formatting of this file. Table IV lists the contents of an output record.

(t) A.D-2.92--Wind/Sigma-Naught File - This file is supplied by the altimeter processing software to the SCATT and CZCS-2 subsystems and contains wind speed, sigma-naught, SWH, and rain-rate estimates. The Mission Contractor is responsible for the interface between the subsystems.

Common Description

Fortran-like code has been used to describe the algorithms. This lead to the definition of labeled commons for communication between the various submodules. The commons allow most variables to be readily available. These commons are described in detail by Tables V through X.

TABLE IV. ALTIMETER LEVEL 2 OUTPUT RECORD CONTENTS

Name	Length (bytes)
Time	8
Orbit number	4
Latitude	4
Longitude	4
Spacecraft altitude	8
Solar latitude	4
Instrument attitude	2
World classification	1
Zone flag	2
Altimeter number/mode	1
Quality flags	6
Contamination flag	2
Number of frames compressed	2
Altitude	8
Altitude standard deviation and total correction	4
Delta altitude corrections (16)	32
Atmospheric measurements	8
SWH	2
SWH standard deviations	4
Delta SWH corrections	4
Ocean backscatter σ°	2
Delta σ° correction/method	4
Wind speed	2
Rain rate/quality	4
AGC	2
AGC standard deviation	4
Rain gates 1 and 2	4
Waveform-derived amplitude	2
Waveform-derived baseline	2
Waveform-derived attitude	2
Skewness	2
Sea-surface height	2
Altitude residual	2
EM bias	2
Geoid height	2
Tide height	2
Solid-Earth tide	2
Ice boundary/quality	1
Mean square slope (sea ice)	2
Percent smoothness (sea ice)	2
Ice delta correction	2
Ice delta AGC correction	2
Ice-sheet slope	2
Ice-sheet roughness	2
Spares	35
Total	200

TABLE V. COMMON EXECUT EXECUTION CONTROL (IEXEC)

1. input data type
    = 0, process both level 1 & 2 (default)
    = 1, process level 1 only
    = 2, process level 2 only
2. year -1900 (i.e., 85)
3. rev number
4. number minutes for averaging period in Health Status Monitor (A.A-1.6)
5. compression period in hundredths of a second

TABLE VI. COMMON STATUS STATUS PARAMETERS (ISTAT)

1. command out to SACU
2. HV on
3. HV ready
4. TWT fault
5. parity
6. memory dump
7. rain processing enable
8. CAL I, II
9. mode command
10. status #1 bits 2 and 1
11. channel select
12. ATU mode
13. gate width
14. ACQ/TRK
15. chirp ACQ step
16. reacquire flag
17. $\Delta H > T_{\Delta H}$
18. chirp/cw
19. high voltage ON/OFF
20. TWTA fault reset
21. trigger kill
22. calibrate mode 1
23. calibrate mode 2
24. TWT heater ON/OFF
25. altimeter designator
26. program version
27. tracker type
28. resolution step
29. TWT fault override
30. LVPS current
31. AT number
32. acquisition constant index
33. track constant index
34. AGC threshold index
35. α, β, and AGC acquisition
36. α, β, and AGC track index
37. ΔH gate width index
38. L_6-E_6 track index
39. height error index
40. L_6-E_6 acquisition index
41. waveheight curve offset
42. acquisition height offset
43. subtrack direction flag
44. zone flag
45. waveform processor convergence flag
46.
47.
48.
49.
50.

TABLE VII. COMMON EUCOEF ENGINEERING UNIT CONVERSION

EUC

1. altitude
2. altitude rate
3. altitude error
4. SWH
5. AGC word
6. noise gate amplitude
7. plateau gate amplitude
8. attitude gate amplitude
9. transmit power
10. TWT beam current
11. TWT cathode voltage
12. TWT HVPS temperature
13. TWT collector temperature
14. receiver temperature
15. noise gate amplitude
16. plateau gate amplitude
17. attitude gate amplitude
18. transmit power
19. UCFM temperature
20. DDL temperature
21. DDL ASSY temperature
22. HSWS temperature
23. DFB temperature #1
24. AT #1 temperature
25. AT #2 temperature
26. ICU temperature
27. SACU temperature
28. LVPS temperature
29. LVPS 38V current
30. +28V S/C bus isolated
31. +28V
32. +15V
33. -15V
34. +7V
35. -9V
36. +5V
37. -5.2V
38. +1.00V REF
39. 0.657V REF
40. SACU PLO LOCK
41. MTU temperature
42. DFB temperature #2

TABLE VIII(a). COMMON HSCOM HEALTH STATUS VARIABLES

Array	Explanation
Z(I)	data to be averaged
ZL(I)	lower edit limits
ZU(I)	upper edit limits
ZE(I)	expected mean values
S1D(I,J)	summations of unedited points (1-day averaging)
S2D(I,J)	summations of unedited points squared (1-day averaging)
N1D(I,J)	number of unedited points (1-day averaging)
N2D(I,J)	number of edited points (1-day averaging)
ZMIND(I,J)	minimum unedited points (1-day averaging)
ZMAXD(I,J)	maximum unedited points (1-day averaging)
ZMD(I,J)	means (1-day averaging)
ZVD(I,J)	variances (1-day averaging)
ZSD(I,J)	standard deviations (1-day averaging)
S1M(I,J)	summations of unedited points (n-minute averaging)
S2M(I,J)	summations of unedited points squared (n-minute averaging)
N1M(I,J)	number of unedited points (n-minute averaging)
N2M(I,J)	number of edited points (n-minute averaging)
ZMINM(I,J)	minimum unedited points (n-minute averaging)
ZMAXM(I,J)	maximum unedited points (n-minute averaging)
ZMM(I,J)	means (n-minute averaging)
ZVM(I,J)	variances (n-minute averaging)
ZSM(I,J)	standard deviations (n-minute averaging)
NEWDAY	= 0 for same day = 1 for new day
ZAL(I,J)	lower alarm limits
ZAU(I,J)	upper alarm limits
NCNT(K)	frame counters

for explanations of subscripts I, J, and K see Table IX(b)

TABLE VIII(b). EXPLANATIONS OF SUBSCRIPTS I, J, AND K IN TABLE VIII(a)

I	Explanation
1	spacecraft attitude
2	altitude rate
3	altitude error
4	SWH
5	AGC word
6	AGC gate amplitude
7	noise gate amplitude
8	transmit power (chirp mode)
9	transmit power (CW mode)
10	TWT beam current
11	TWT cathode voltage
12	TWT HVPS temperature
13	TWT collector temperature
14	receiver temperature
15	noise gate amplitude
16	plateau gate amplitude
17	attitude gate amplitude
18	transmit power (chirp mode)
19	transmit power (CW mode)
20	UCFM temperature
21	DDL temperature
22	DDL ASSY temperature
23	HSWS temperature
24	DFB temperature no. 1
25	AT no. 1 temperature
26	AT no. 2 temperature
27	ICU temperature
28	SACU temperature
29	LVPS temperature
30	LVPS 38V current
31	+28V S/C bus isolated
32	+28V
33	+15V
34	-15V
35	+7V
36	-9V
37	+5V
38	-5.2V
39	+1.00V REF
40	0.657V REF
41	SACU PLO LOCK
42	MTU temperature
43	DFB temperature no. 2
44	TWT HVPS temperature
45	TWT collector temperature
46	receiver temperature
47	noise gate amplitude
48	plateau gate amplitude
49	attitude gate amplitude
50	transmit power (chirp mode)
51	transmit power (CW mode)
52	UCFM temperature
53	DDL temperature
54	DDL ASSY temperature
55	HSWS temperature
56	DFB temperature no. 1
57	AT no. 1 temperature
58	AT no. 2 temperature
59	ICU temperature
60	SACU temperature
61	LVPS temperature
62	LVPS 38V current
63	+28V S/C bus isolated
64	+28V
65	+15V
66	-15V
67	+7V
68	-9V
69	+5V
70	-5.2V
71	+1.00V REF
72	0.657V REF
73	SACU PLO LOCK
74	MTU temperature
75	DFB temperature no. 2

TABLE VIII(b) (continued)

J	Explanation
1	altimeter no. 1
2	altimeter no. 2

K	Explanation
1	HV on
2	HV off
3	altimeter on
4	standby
5	calibrate
6	trigger kill (last command sent)
7	track 1
8	track 2
9	track 3
10	track 4
11	TWT fault reset (last command sent)
12	test mode 1 (CW)
13	test mode 2
14	test mode 3
15	test mode 4
16	adapt. resolution
17	TBD
18	TBD
19	TBD
20	rain processing
21	ACQ/TRK
22	chirp mode
23	CW mode
24	trigger kill (SACU mode command)
25	TWTA fault reset (SACU mode command)

TABLE IX. COMMON SENSOR LEVEL 1 PARAMETERS (WLEV1)

1. day of year
2. seconds past midnight
3. microseconds
4. altitude rate
5. altitude
6.
7. altitude error
8. SWH
9. AGC word
10. AGC gate amplitude
11. early gate amplitude
12. late gate amplitude
13. middle gate amplitude
14. gate normalization factor
15. noise gate amplitude
16. plateau gate amplitude
17. attitude gate amplitude
18. transmit power
19. TWT beam current
20. TWT cathode voltage
21. TWT HVPS temperature
22. TWT collector temperature
23. receiver temperature
24. noise gate amplitude
25. plateau gate amplitude
26. attitude gate amplitude
27. transmit power
28. UCFM temperature
29. DDL temperature
30. DDL ASSY temperature
31. HSWS temperature
32. DFB temperature #1
33. AT #1 temperature
34. AT #2 temperature
35. ICU temperature
36. SACU temperature
37. LVPS temperature
38. LVPS 38V current
39. +28V S/C bus isolated
40. +28V
41. +15V
42. -15V
43. +7V
44. -9V
45. +5V
46. -5.2V
47. +1.00V REF
48. 0.657V REF
49. SACU PLO LOCK
50. MTU temperature
51. DFB temperature #2
52. spare
53. spare

. .

. .

. .

59. spare
60. relay status
61. bits 1 through 8 spare
62. parameter select 1
63. parameter select 2
64. parameter select 3
65. parameter select 4
66. spare
67. spare

. .

. .

. .

114. altitude acceleration
115. modified julian date (uncorrected)
116.
117. modified julian date (corrected)
118.
119. time correction

120. year - 1900
121. month
122. day
123. hour
124. minute
125. second
126. altitude cal zone bias
127. cal zone bias standard deviation
128. altitude C.G. offset
129. C.G. offset standard deviation
130. altitude cal mode bias
131. cal mode bias standard deviation
132. AGC cal mode bias
133. level 1 corrected altitude
134.
135. corrected AGC
136. spacecraft attitude
137.
height above the reference ellipsoid
138.
139. status #1
140. status #2
141. status #3
142. status #4
143. status #5
144. AGC cal mode standard deviation
145. Cal/Atten status
146 altitude adapt. res. correction
147. altitude adapt. res. correction
standard deviation
148. AGC adapt. res. correction
149. AGC adapt. res. correction
standard deviation
150. spare
151. geodetic latitude
152. longitude
153. spare
154. spare
. .
. .
. .
159 spare
160. waveform sample #1
161. waveform sample #2
. .
. .
. .
222. waveform sample #63
223. rain subcom parameter #1
. .
. .
. .
242 rain subcom parameter #20

TABLE X. COMMON GEOPHY LEVEL 2 PARAMETERS (WLEV2)

1. orbit number
2. altimeter number
3. contamination flag
4. compressed interval
5. mean compressed time
6. " " "
7. compressed latitude
8. compressed longitude
9. mean compressed ellipsoid height
10. " " " "
11. compressed solar latitude
12. mean compressed instrument attitude
13. number of attitude measurements used in compression interval
14. compressed zone flag
15. compressed world classification
16. std dev compressed altitude
17. mean compressed altitude
18. " " "
19. " " " rate
20. " " " error
21. " " SWH
22. " " AGC
23. " " delta time
24. " " delta altitude for Center Grav
25. " " delta altitude for Cal Zone
26. " " delta altitude for Cal Mode
27. " " delta AGC for Cal Mode
28. " " delta altitude for Adap Res
29. " " delta AGC for Adap Res
30. " " AGC gate
31. " " early gate
32. " " late gate
33. " " middle gate
34. spare
35. mean compressed noise gate
36. " " plateau gate
37. " " attitude gate

38. mean MTU temperature
39. mean DFB temperature
40. mean TWT collector temperature
41. spare
. .
. .
44. spare
45. std dev compressed altitude rate
46. " " " " error
47. " " " SWH
48. " " " AGC
49. " " compressed AGC gate
50. " " compressed early gate
51. " " compressed late gate
52. " " compressed middle gate
53. spare
54. std dev compressed noise gate
55. " " compressed plateau gate
56. " " compressed attitude gate
57. spare
58. compressed rain sub-comm 1 (mean)
59. " " " 2 "
60. " " " 3 "
61. " " " 4 "
62. " " " 5 "
63. " " " 6 "
64. " " " 7 "
65. " " " 8 "
66. " " " 9 "
67. " " " 10 "
68. " " " 11 "
69. " " " 12 "
70. " " " 13 "
71. " " " 14 "
72. " " " 15 "
73. " " " 16 "
74. " " " 17 "
75. " " " 18 "
76. " " " 19 "

77. compressed rain sub-comm 20 (mean)
78. " " " 1 (std dev)
79. " " " 2 " "
80. " " " 3 " "
81. " " " 4 " "
82. " " " 5 " "
83. " " " 6 " "
84. " " " 7 " "
85. " " " 8 " "
86. " " " 9 " "
87. " " " 10 " "
88. " " " 11 " "
89. " " " 12 " "
90. " " " 13 " "
91. " " " 14 " "
92. " " " 15 " "
93. " " " 16 " "
94. " " " 17 " "
95. " " " 18 " "
96. " " " 19 " "
97. " " " 20 " "
98. spare
99. mean compressed transmit power
100. std dev compressed transmit power
101. compressed waveform sample no. 1 (mean)
102. " " " 2 "
103. " " " 3 "
104. " " " 4 "
105. " " " 5 "
106. " " " 6 "
107. " " " 7 "
108. " " " 8 "
109. " " " 9 "
110. " " " 10 "
111. " " " 11 "
112. " " " 12 "
113. " " " 13 "
114. " " " 14 "
115. " " " 15 "

116. compressed waveform sample no. 16 (mean)
117. " " " 17 "
118. " " " 18 "
119. " " " 19 "
120. " " " 20 "
121. " " " 21 "
122. " " " 22 "
123. " " " 23 "
124. " " " 24 "
125. " " " 25 "
126. " " " 26 "
127. " " " 27 "
128. " " " 28 "
129. " " " 29 "
130. " " " 30 "
131. " " " 31 "
132. " " " 32 "
133. " " " 33 "
134. " " " 34 "
135. " " " 35 "
136. " " " 36 "
137. " " " 37 "
138. " " " 38 "
139. " " " 39 "
140. " " " 40 "
141. " " " 41 "
142. " " " 42 "
143. " " " 43 "
144. " " " 44 "
145. " " " 45 "
146. " " " 46 "
147. " " " 47 "
148. " " " 48 "
149. " " " 49 "
150. " " " 50 "
151. " " " 51 "
152. " " " 52 "
153. " " " 53 "
154. " " " 54 "

155. compressed waveform sample no. 55 (mean)
156. " " " 56 "
157. " " " 57 "
158. " " " 58 "
159. " " " 59 "
160. " " " 60 "
161. " " " 61 "
162. " " " 62 "
163. " " " 63 "
164. " " " 1 (std dev)
165. " " " 2 " "
166. " " " 3 " "
167. " " " 4 " "
168. " " " 5 " "
169. " " " 6 " "
170. " " " 7 " "
171. " " " 8 " "
172. " " " 9 " "
173. " " " 10 " "
174. " " " 11 " "
175. " " " 12 " "
176. " " " 13 " "
177. " " " 14 " "
178. " " " 15 " "
179. " " " 16 " "
180. " " " 17 " "
181. " " " 18 " "
182. " " " 19 " "
183. " " " 20 " "
184. " " " 21 " "
185. " " " 22 " "
186. " " " 23 " "
187. " " " 24 " "
188. " " " 25 " "
189. " " " 26 " "
190. " " " 27 " "
191. " " " 28 " "
192. " " " 29 " "
193. " " " 30 " "

194. compressed waveform sample no. 31 (std dev)
195. " " " 32 " "
196. " " " 33 " "
197. " " " 34 " "
198. " " " 35 " "
199. " " " 36 " "
200. " " " 37 " "
201. " " " 38 " "
202. " " " 39 " "
203. " " " 40 " "
204. " " " 41 " "
205. " " " 42 " "
206. " " " 43 " "
207. " " " 44 " "
208. " " " 45 " "
209. " " " 46 " "
210. " " " 47 " "
211. " " " 48 " "
212. " " " 49 " "
213. " " " 50 " "
214. " " " 51 " "
215. " " " 52 " "
216. " " " 53 " "
217. " " " 54 " "
218. " " " 55 " "
219. " " " 56 " "
220. " " " 57 " "
221. " " " 58 " "
222. " " " 59 " "
223. " " " 60 " "
224. " " " 61 " "
225. " " " 62 " "
226. " " " 63 " "
227. waveform processor rss of fit
228. spare
229. waveform processor convergence flag
230. waveform processor refined SWH estimate
231. waveform processor refined attitude estimate
232. waveform processor skewness estimate

233. waveform processor waveform amplitude estimate
234. waveform processor altitude correction
235. waveform processor baseline estimate
236. $\ddot{h}$
237. SWH correction for SWH and attitude
238. altitude correction for SWH, attitude and H
239. spare
240. std dev of SWH correction for SWH and attitude
241. SWH Cal zone bias
242. mean square slope
243. Fresnel power reflection coefficient
244. percent smooth area
245. significant slope
246. dominant wavelength
247. dominant frequency
248. dominant phase speed
249. dominant wave number
250. spare
251. rain rate estimate
252. rain rate quantity flag
253. rain rate quality flag
254. smoothed rain rate estimate
255. std dev smoothed rain rate estimate
256. atmospheric pressure at sea surface
257. atmospheric water vapor at sea surface
258. atmospheric temperature at sea surface
259. FNOC data present flag
260. altitude correction for barotropic effects
261. std dev of altitude correction for bar. effects
262. geomagnetic latitude
263. Ec minimum
264. Ec maximum
265. Beta
266. solar flux
267. ionospheric refraction altitude correction
268. std dev of ionospheric refraction altitude correction
269. dry tropospheric refraction altitude correction
270. std dev dry tropospheric refraction alt. correction
271. wet tropospheric refraction altitude correction

272. std dev wet tropospheric refraction alt correction
273. LAMMR quality data flag
274. sigma naught atmospheric correction
275. spare
276. radar backscatter coefficient
277. wind speed at 10 meters
278. wind speed at 19½ meters
279. level 2 corrected altitude
280. " " " "
281. geoid height
282. tide height
283. solid earth tide height
284. altitude correction from EM bias
285. std dev alt correction from EM bias
286. EM sea state bias quality flag
287. sea surface height
288. altitude residual
289. spare
290. ice sheet height correction
291. mean surface roughness
292. sea surface height retrack quality flag
293. spare
294. quality flag for altitude std dev
295. " " " altitude rate
296. " " " radar backscatter
297. " " " AGC std dev
298. " " " attitude
299. " " " MTU temp
300. " " " DFB tem
301. " " " noise gate
302. " " " AGC gate
303. " " " TWT collector temp
304. " " " SWH std dev
305. " " " EM sea state bias
306. " " " data validity
307. spare
308. "
309. "
310. "

311.
312.
313. lunar inertial position vector
314.
315.
316.
317.
318.
319. solar inertial position vector
320.
321.
322.
323. spare
324. "
325. "

MODULE DESCRIPTIONS

This section contains the detailed descriptions of the altimeter processing algorithms.

The requirements and logic for each completed module of the altimeter processing algorithms are provided in this section. For each of these modules, the title and function, inputs and outputs, tables, and processing are presented. For the modules not completed functional requirements are presented.

ALTIMETER MAIN DRIVER MODULE A.A-0.0

TITLE: PROGRAM ALTMTR
FUNCTION: Main driver module for the altimeter software
REFERENCE: Not applicable
CONTROL: None
SUPPORT: A.A-1.0 ALT1DR level 1 driver module
A.A-2.0 ALT2DR level 2 driver module
ACCESS: Not applicable
INPUTS: A.D-1.85 executive parameter file
A.D-1.72 EU conversion coefficient file

A.D-1.81 level 0 data file
A.D-1.94 level 1 data file

OUTPUTS: A.D-1.94 level 1 data file
A.D-2.91 level 2 data file

TABLES: Table 0.0a A.D-1.94 and A.D-2.91 Header Record (see Appendix B)
Table 0.0b A.D-1.94 Header Record #2 (see Appendix B)

ALGORITHM:

```
1.  Read the execution-time data files (A.D-1.85 and A.D-1.72) into arrays
    IEXEC and EUC
2.  Write header record(s) (see Tables 0.0a and 0.0b) on
          A.D-1.94 if IEXEC(1) equals 0 or 1
          A.D-2.91 if IEXEC(1) equals 0 or 2
3.  IF (IEXEC(1) .NE. 2) GO TO 4
    a.  Read header record from A.D-1.94
    b.  GO TO 8
4.  Read one minor frame from the level 0 data file (A.D-1.81); if end of
    data encountered GO TO 12
5.  Process the level 0 data
          CALL ALT1DR
6.  Write the minor frame on the level 1 data file (A.D-1.94)
7.  Determine if level 2 data is to be processed
          IF (IEXEC(1) .EQ. 1) GO TO 4
          GO TO 9
8.  Read one minor frame from the level 1 data file (A.D-1.94); if end of
    data encountered GO TO 12
9.  Process the level 1 data
          CALL ALT2DR (IFLAG)
10. If IFLAG = 0, write the minor frame on the level 2 data file (A.D-2.91)
11. Go back to process the next minor frame
          IF (IEXEC(1) .EQ. 0) GO TO 4
          GO TO 8
12. Normal termination
          STOP
```

COMMENTS: 1. The following TYPE statements must be included in this module

```
      DIMENSION EU(8,100,2)
      COMMON /EXECUT/ IEXEC(100)
      COMMON /TLMTRY/ ITLM(100)
      COMMON /STATUS/ ISTAT(100)
      COMMON /SENSOR/ WLEV1(300)
      COMMON /GEOPHY/ WLEV2(325)
```

LEVEL 1 DRIVER MODULE A.A-1.0

TITLE: SUBROUTINE ALT1DR

FUNCTION: Driver module for the altimeter level 1 software

REFERENCE: Not applicable

CONTROL: A.A-0.0 ALTMTR altimeter main driver module

SUPPORT:
A.A-1.1 EUCONV EU conversion module
A.A-1.2 AL1COR level 1 altitude correction module
A.A-1.3 CALMOD cal mode module
A.A-1.4 ADPRES adaptive resolution module
A.A-1.5 LOCATE location classification module
A.A-1.6 HSMNTR health/status monitor module

ACCESS: CALL ALT1DR

INPUTS:
IEXEC = array containing executive parameters
EUC = array containing EU conversion coefficients
ITLM = array containing telemetry data

OUTPUTS:
ISTAT = array containing status words
WLEV1 = array containing level 1 output products

TABLES: None

ALGORITHM:

```
1.  Convert counts to engineering units
          CALL EUCONV
2.  Determine the altimeter mode
          IF (ISTAT(9).EQ.1 .OR. NCAL.EQ.1) GO TO 3
          IF (ISTAT(9).GE.3 .AND. ISTAT(9).LE.6) GO TO 4
          IF (ISTAT(9).EQ.8 .OR. ISTAT(9).EQ.12) GO TO 5
          GO TO 6
3.  Cal mode data (NCAL = 1 for cal mode; NFINAL = 1 for final call to
    cal mode module)
          NCAL = 1
          NFINAL = 0
          IF (ISTAT(9) .NE. 1) NFINAL = 1
          CALL CALMOD (NFINAL)
          IF (ISTAT(9) .EQ. 1) GO TO 6
          IF (ISTAT(9) .EQ. 8 .OR. ISTAT(9).EQ. 12) GO TO 5
          IF (ISTAT(9) .LT. 3 .OR. ISTAT(9).GT.6) GO TO 6
4.  Track mode data (NCAL = 0 for non-cal mode)
          NCAL = 0
          CALL AL1COR
          GO TO 6
```

5. Adaptive resolution mode data

   `CALL ADPRES`

6. Classify the subsatellite point

   `CALL LOCATE`

7. Monitor critical instrument parameters

   `CALL HEALTH`

8. End of algorithm

   `RETURN`

COMMENTS:

1. The following TYPE statements must be included in the code of this module:

```
COMMON /EXECUT/ IEXEC(100)
COMMON /EUCOEF/ EUC(8,100)
COMMON /TLMTRY/ ITLM(100)
COMMON /STATUS/ ISTAT(100)
COMMON /SENSOR/ WLEV1(300)
DATA NCAL/0/
```

2. Spacecraft engineering data handling has not yet been defined

EU CONVERSION MODULE A.A-1.1

TITLE: SUBROUTINE EUCONV

FUNCTION: Conversion of telemetry stream from counts to engineering units

REFERENCE: <u>SEASAT-1 Radar Altimeter Phase I Engineering Assessment Report</u>, WFC-135-80-001, December 1980.

CONTROL: A.A-1.0 ALT1DR level 1 driver module

SUPPORT:

A.A-1.1.1	EUTIME	EU date and time conversion submodule
A.A-1.1.2	EUALT	EU altitude, SWH, & AGC conversion submodule
A.A-1.1.3	EUGATE	EU gate amplitude conversion submodule
A.A-1.1.4	EUSTAT	EU status extraction submodule
A.A-1.1.5	EU1COM	EU engineering subcom #1 conversion submodule
A.A-1.1.6	EU2COM	EU engineering subcom #2 conversion submodule
A.A-1.1.7	EU3COM	EU engineering subcom #3 conversion submodule
A.A-1.1.8	EURAIN	EU rain subcom conversion submodule
A.A-1.1.9	EUWFRM	EU waveform, CW, or dump submodule

ACCESS: CALL EUCONV

INPUTS:

A.D-1.72 = EU conversion coefficient file

ISTAT(25) = altimeter designator (1 or 2)

OUTPUTS: None

TABLES: None

ALGORITHM:

```
1.  Extract the date and time
          CALL EUTIME
2.  Break out the status words
          CALL EUSTAT
3.  Load the appropriate EU coefficients into array EUC from file A.D-1.72
    if IALT ≠ ISTAT(25); then set
          IALT  =  ISTAT(25)
4.  Convert the altitude, altitude rate, altitude error, SWH, and AGC
          CALL EUALT
5.  Convert the gate amplitudes
          CALL EUGATE
6.  Convert the engineering subcom #1
          CALL EU1COM
7.  Convert the engineering subcom #2
          CALL EU2COM
8.  Convert the engineering subcom #3
          CALL EU3COM
9.  Convert the rain subcom
          CALL EURAIN
10.  Process waveform, CW, or dump data
          CALL EUWFRM
11. End of algorithm
          RETURN
```

COMMENTS: 1. The following TYPE statements must be included in the code of this module

```
COMMON /STATUS/ ISTAT(100)
COMMON /EUCOEF/ EUC(8,100)
DATA /IALT/ 0
```

EU DATE AND TIME CONVERSION SUBMODULE A.A-1.1.1

TITLE: SUBROUTINE EUTIME

FUNCTION: Extraction of date and time from telemetry stream

REFERENCE: "SEASAT-A Sensor Data Record Tape Specification Interface Control Document," JPL 622-57, September 1978

CONTROL: A.A-1.1 EUCONV EU conversion submodule

SUPPORT: A.A-1.1.1.1 IEUBIT EU bit extraction submodule

ACCESS: CALL EUTIME

INPUTS:
ITLM(1-4) = time (GMT)
IEXEC(2) = year - 1900

OUTPUTS:
WLEV1(1) = day of year
WLEV1(2) = seconds past midnight
WLEV1(3) = microseconds
TMJDO = modified julian date in (d,$1.0x10^{-8}$)

TABLES: None

ALGORITHM:

```
1.  Split the timing words between seconds and microseconds
          I1  =  IEUBIT(ITLM(3),10,5)
          I2  =  IEUBIT(ITLM(3),4,1)
2.  Concatenate ITLM(1), ITLM(2), and I1 into a 26-bit right justified
    string called J1 with ITLM(1) being the MSB and I1 being the LSB
3.  Concatenate I2 and ITLM(4) into a 14-bit right justified string called
    J2 with I2 being the MSB and ITLM(4) being the LSB
4.  Compute the day of year and seconds and microseconds past midnight
          WLEV1(1)  =  J1/86400 + 1
          WLEV1(2)  =  FLOAT(J1) - 86400.0*WLEV1(1)
          WLEV1(3)  =  J2
5.  Compute the modified julian date
          INDEX     =  IEXEC(2) - 74
          TREF      =  IREF(INDEX)
          TMJDO     =  (DBLE(WLEV1(3))*1.0D-0.6+DBLE(WLEV1(2))) /
                       86400.0D+00
          1 + DBLE(WLEVL(1) + TREF
6.  End of algorithm
          RETURN
```

COMMENTS:

1. The following TYPE statements must be included in the code of this submodule

```
          DOUBLE PRECISION TMJDO,TREF
          DIMENSION IREF(25)
          COMMON /EXECUT/ IEXEC(100)
          COMMON /TLMTRY/ ITLM(100)
          COMMON /SENSOR/ WLEV1(300)
          EQUIVALENCE (TMJDO,WLEV1(115))
          DATA IREF/   0, 365, 731,1096,1461,1826,2192,
         1          2557,2922,3287,3653,4018,4383,4748,
         2          5114,5479,5844,6209,6575,6940,7305,
         3          7670,8036,8401,8766/
```

EU BIT EXTRACTION SUBMODULE A.A-1.1.1.1

TITLE: INTEGER FUNCTION IEUBIT

FUNCTION: To extract bits I1 through I2 from IWORD

REFERENCE: Not applicable

CONTROL:
A.A-1.1.1 EUTIME EU date and time conversion submodule
A.A-1.1.2 EUALT EU altitude, SWH, & AGC conversion submodule
A.A-1.1.4 EUSTAT EU status extraction submodule
A.A-1.1.7 EU3COM EU engineering subcom #3 conversion submodule

SUPPORT: None

ACCESS: J = IEUBIT(IWORD,I1,I2)

INPUTS:
IWORD = the telemetry word (10 bits, right justified)
I1 = the first bit to be extracted
I2 = the last bit to be extracted

OUTPUTS: IEUBIT = the extracted bits (right justified)

TABLES: None

ALGORITHM:

```
1. Extract the desired bits
        IEUBIT = FLD(36-I1,I1-I2+1,IWORD)
2. End of algorithm
        RETURN
```

COMMENTS:

1. Step #1 is machine-dependent (above is for HW625). Argument #1 is the starting bit, argument #2 is the number of bits, argument #3 is the array address. The HW625 numbers the bits from 0 to 35 with 0 being the MSB.
2. The 10 bits in the telemetry word (IWORD) are right justified. The numbering sequence used for I1 and I2 is as follows: the LSB is bit #1 and the MSB is bit #10.
3. This submodule is included here only for ease of specification of the algorithms. It may be replaced or eliminated to suit the particular hardware and software configuration chosen for the mission processing.

EU ALTITUDE, SWH, & AGC CONVERSION SUBMODULE A.A-1.1.2

TITLE: SUBROUTINE EUALT

FUNCTION: EU conversion of altitude, altitude rate, altitude error, SWH, and AGC

REFERENCE: <u>SEASAT-1 Radar Altimeter Phase I Engineering Assessment Report</u>, WFC-135-80-001, December 1980

CONTROL: A.A-1.1 EUCONV EU conversion module

SUPPORT: A.A-1.1.1.1 IEUBIT EU bit extraction submodule
A.A-1.1.2.1 EUPOLY EU polynomial fit submodule

ACCESS: CALL EUALT

INPUTS:
ITLM(5-7) = altitude counts
ITLM(8) = altitude rate counts
ITLM(9) = altitude error counts
ITLM(10) = SWH counts
ITLM(11) = AGC word counts
EUC = EU conversion coefficients
AGCW = AGC attenuator (see Table 1.1.2)

OUTPUTS:
ALT = level 0 altitude in (m,0.001)
WLEV1(4) = altitude rate in (m/s,0.01)
WLEV1(7) = altitude error in (m,0.01)
WLEV1(8) = SWH in (m,0.01)
WLEV1(9) = AGC word in (dB,0.01)

TABLES: Table 1.1.2 AGC Word Lookup Table (see Appendix B)

ALGORITHM:

```
1.  Convert the altitude from counts to meters
    a.  Concatenate the 10 bits contained in each of ITLM(5), ITLM(6),
        and ITLM(7) into a right justified 30-bit string called I1 with
        ITLM(5) being the MSB and ITLM(7) being the LSB
    b.  D    =  DBLE(FLOAT(I1))
        ALT  =  D*EUC(5,1) + EUC(6,1)
2.  Convert the altitude rate from counts to meters per second
           I1        =  IEUBIT(ITLM(8),10,10)
           IC        =  ITLM(8) - 1024*I1
           WLEV1(4)  =  EUPOLY(EUC(1,2),IC,0.0)
3.  Convert the altitude error from counts to meters
           I1        =  IEUBIT(ITLM(9),10,10)
           I2        =  IEUBIT(ITLM(9),10,5)
           I3        =  IEUBIT(ITLM(9), 4, 1)
           MAG       =  I2-64*I1
           C         =  2.0**(-I3) * MAG
           WLEV1(7)  =  EUC(6,3) + C*(EUC(5,3) + C*(EUC(4,3)
         1              + C*(EUC(3,3) + C*(EUC(2,3)
         2              + C*(EUC(1,3)))))) + EUC(8,3)
4.  Convert the SWH from counts to meters
           WLEV1(8)  =  EUPOLY(EUC(1,4),ITLM(10),0.0)
```

5. Convert the AGC word from counts to dB

```
I1        =  IEUBIT(ITLM(11),10,5)
I2        =  IEUBIT(ITLM(11),4,1)
WLEV1(9)  =  AGCW(I1+1) + EUPOLY(EUC(1,5), I2, WLEV1(23)-TMPRCV)
```

AGCW is an array of AGC values and I1 is the attenuator setting (see Table 1.1.2)

6. End of algorithm

```
RETURN
```

COMMENTS: 1. The following TYPE statements must be included in the code of this submodule

```
DOUBLE PRECISION D,ALT
COMMON /EUCOEF/ EUC(8,100)
COMMON /TLMTRY/ ITLM(100)
COMMON /SENSOR/ WLEV1(300)
COMMON /TMPREF/ TMPRCV,TMPMTU
COMMON /T112/   AGCW(64)
EQUIVALENCE (ALT,WLEV1(5))
```

EU POLYNOMIAL CONVERSION SUBMODULE A.A-1.1.2.1

TITLE: FUNCTION EUPOLY

FUNCTION: EU polynomial conversion from counts to engineering units

REFERENCE: Not applicable

CONTROL:

A.A-1.1.2	EUALT	EU altitude, SWH, & AGC conversion submodule
A.A-1.1.3	EUGATE	EU gate amplitude conversion submodule
A.A-1.1.4	EUSTAT	EU status extraction submodule
A.A-1.1.5	EU1COM	EU engineering subcom #1 conversion submodule
A.A-1.1.6	EU2COM	EU engineering subcom #2 conversion submodule
A.A-1.1.8	EURAIN	EU rain subcom conversion submodule

SUPPORT: None

ACCESS: E = EUPOLY(C,L,T)

INPUTS:
C = array of EU conversion coefficients
L = telemetry counts
T = temperature in (°C,0.01)

OUTPUTS: EUPOLY = engineering units

TABLES: None

ALGORITHM: 1. Convert counts to engineering units

```
D      = FLOAT(L)
```

```
        EUPOLY = C(6) + D*(C(5) + D*(C(4) + D*(C(3) + D*(C(2)
                           + D*(C(1)))))) + C(7)*T + C(8)
2.  End of algorithm
        RETURN
```

COMMENTS:

1. The polynomial evaluated is

$$EUPOLY = C(1)*L^{5} + C(2)*L^{4} + C(3)*L^{3} + C(4)*L^{2} + C(5)*L + C(6) + C(7)*T + C(8)$$

2. The following TYPE statement must be included in the code of this submodule

```
DIMENSION C(8)
```

3. The C array is a column vector of the doubly subscripted array EUC

EU GATE AMPLITUDE CONVERSION SUBMODULE A.A-1.1.3

TITLE: SUBROUTINE EUGATE

FUNCTION: EU conversion of

1. AGC gate amplitude
2. early gate amplitude
3. late gate amplitude
4. middle gate amplitude
5. gate normalization factor
6. noise gate amplitude
7. plateau gate amplitude
8. attitude gate amplitude

REFERENCE: *SEASAT-1 Radar Altimeter Phase I Engineering Assessment Report*, WFC-135-80-001, December 1980

CONTROL: A.A-1.1 EUCONV EU conversion module

SUPPORT: A.A-1.1.2.1 EUPOLY EU polynomial fit submodule

ACCESS: CALL EUGATE

INPUTS:

ITLM(12) = AGC gate amplitude counts

ITLM(13) = early gate amplitude counts

ITLM(14) = late gate amplitude counts

ITLM(15) = middle gate amplitude counts

ITLM(16) = gate normalization factor counts

ITLM(17) = noise gate amplitude counts

ITLM(18) = plateau gate amplitude counts

ITLM(19) = attitude gate amplitude counts

ITLM(20) = transmit power counts

EUC = EU conversion coefficients
TMPRCV = receiver reference temperature in (°C,0.1)
TMPMTU = MTU reference temperature in (°C,0.1)

OUTPUTS:
WLEV1(10) = AGC gate amplitude in (v,0.1)
WLEV1(11) = early gate amplitude in (v,0.1)
WLEV1(12) = late gate amplitude in (v,0.1)
WLEV1(13) = middle gate amplitude in (v,0.1)
WLEV1(14) = gate normalization factor
WLEV1(15) = noise gate amplitude in (v,0.1)
WLEV1(16) = plateau gate amplitude in (v,0.1)
WLEV1(17) = attitude gate amplitude in (v,0.1)
WLEV1(18) = transmit power in (kw,0.1)

TABLES: None

ALGORITHM:

```
1.  Extract the AGC, early, late, and middle gate amplitudes
          WLEVL(10)  =  ITLM(12)
          WLEV1(11)  =  ITLM(13)
          WLEV1(12)  =  ITLM(14)
          WLEV1(13)  =  ITLM(15)
2.  Extract the gate normalized factor
          WLEV1(14)  =  ITLM(16)
3.  Convert the noise, plateau, and attitude gate amplitudes from counts
    to volts
          WLEV1(15)  =  EUPOLY(EUC(1,6),ITLM(17),WLEV1(23)-TMPRCV)
          WLEV1(16)  =  EUPOLY(EUC(1,7),ITLM(18),WLEV1(23)-TMPRCV)
          WLEV1(17)  =  EUPOLY(EUC(1,8),ITLM(19),WLEV1(23)-TMPRCV)
4.  Convert the transmit power from counts to kilowatts
          D          =  1.14334068*(WLEV1(50)-TMPMTU) + ITLM(20)
          WLEV1(18)  =  EUC(6,9) + D*(EUC(5,9) + D*(EUC(4,9)
         1  + D*(EUC(3,9) + D*(EUC(2,9) + D*(EUC(1,9)))))) + EUC(8,9)
5.  End of algorithm
          RETURN
```

COMMENTS:

1. The following TYPE statements must be included in the code of this submodule

```
          COMMON /EUCOEF/ EUC(8,100)
          COMMON /TLMTRY/ ITLM(100)
          COMMON /SENSOR/ WLEV1(300)
          COMMON /TMPREF/ TMPRCV,TMPMTU
```

EU STATUS EXTRACTION SUBMODULE A.A-1.1.4

TITLE: SUBROUTINE EUSTAT

FUNCTION: EU conversion of

1. cal atten/SACU status
2. status #1
3. status #2
4. status #3
5. status #4
6. status #5

REFERENCE: <u>SEASAT-1 Radar Altimeter Phase I Engineering Assessment Report</u>, WFC-135-80-001, December 1980

CONTROL: A.A-1.1 EUCONV EU conversion module

SUPPORT:
A.A-1.1.1.1 IEUBIT EU bit extraction submodule
A.A-1.1.2.1 EUPOLY EU bit polynomial fit submodule

ACCESS: CALL EUSTAT

INPUTS:
ITLM(21) = cal atten/SACU status
ITLM(22) = status #1
ITLM(23) = status #2
ITLM(24) = status #3
ITLM(25) = status #4
ITLM(26) = status #5

OUTPUTS:
ISTAT(1-28) = broken out status words
WLEV1(139) = status #1
WLEV1(140) = status #2
WLEV1(141) = status #3
WLEV1(142) = status #4
WLEV1(143) = status #5

TABLES: Table A.1 TM Format (see Appendix A)

ALGORITHM:

```
1.  Decode the cal atten/SACU status
          ISTAT(1)    = IEUBIT(ITLM(21),8,5)
          ISTAT(2)    = IEUBIT(ITLM(21),3,3)
          ISTAT(3)    = IEUBIT(ITLM(21),2,2)
          ISTAT(4)    = IEUBIT(ITLM(21),1,1)
2.  Decode status #1
          WLEV1(139) = ITLM(22)
          ISTAT(5)    = IEUBIT(ITLM(22),10,10)
          ISTAT(6)    = IEUBIT(ITLM(22),9,9)
          ISTAT(7)    = IEUBIT(ITLM(22),8,8)
```

```
               ISTAT(8)    = IEUBIT(ITLM(22),7,7)
               ISTAT(9)    = IEUBIT(ITLM(22),6,3)
               ISTAT(10)   = IEUBIT(ITLM(22),2,1)
          3.  Decode status #2
               WLEV1(140) = ITLM(23)
               ISTAT(11)   = IEUBIT(ITLM(23),10,5)
               ISTAT(12)   = IEUBIT(ITLM(23),4,1)
          4.  Decode status #3
               WLEV1(141) = ITLM(24)
               ISTAT(13)   = IEUBIT(ITLM(24),10,8)
               ISTAT(14)   = IEUBIT(ITLM(24),7,7)
               ISTAT(15)   = IEUBIT(ITLM(24),6,5)
               ISTAT(16)   = IEUBIT(ITLM(24),4,4)
               ISTAT(17)   = IEUBIT(ITLM(24),3,3)
          5.  Decode status #4
               WLEV1(142) = ITLM(25)
               ISTAT(18)   = IEUBIT(ITLM(25),8,8)
               ISTAT(19)   = IEUBIT(ITLM(25),7,7)
               ISTAT(20)   = IEUBIT(ITLM(25),6,6)
               ISTAT(21)   = IEUBIT(ITLM(25),5,5)
               ISTAT(22)   = IEUBIT(ITLM(25),4,4)
               ISTAT(23)   = IEUBIT(ITLM(25),3,3)
               ISTAT(24)   = IEUBIT(ITLM(25),2,2)
          6.  Decode status #5
               WLEV1(143) = ITLM(26)
               ISTAT(25)   = IEUBIT(ITLM(26),10,10)
               ISTAT(26)   = IEUBIT(ITLM(26),9,6)
               ISTAT(27)   = IEUBIT(ITLM(26),5,4)
               ISTAT(28)   = IEUBIT(ITLM(26),3,1)
          7.  End of algorithm
               RETURN
```

COMMENTS: 1. The following TYPE statements must be included in the code of this submodule

```
               COMMON /TLMTRY/ ITLM(100)
               COMMON /STATUS/ ISTAT(100)
               COMMON /SENSOR/ WLEV1(300)
```

EU ENGINEERING SUBCOM #1 CONVERSION SUBMODULE A.A-1.1.5

TITLE: SUBROUTINE EU1COM

FUNCTION: EU conversion of engineering subcom #1

REFERENCE: <u>SEASAT-1 Radar Altimeter Phase I Engineering Assessment Report</u>, WFC-135-80-001, December 1980

CONTROL: A.A-1.1 EUCONV EU conversion module

SUPPORT: A.A-1.1.2.1 EUPOLY EU polynomial conversion submodule

ACCESS: CALL EU1COM

INPUTS:
ITLM(27) = EU subcom #1 counts
EUC = EU conversion coefficients
TMPRCV = receiver reference temperature in (°C,0.1)
TMPMTU = MTU reference temperature in (°C,0.1)
ISTAT(11) = subcom counter (1 to 20)

OUTPUTS: WLEV1(19-37) = subcom #1 in engineering units

TABLES: Table A.1 TM Format (see Appendix A)

ALGORITHM:

```
1.  Branch to the appropriate subcom word
        N = ISTAT(11)
        GO TO (2,3,4,...,20,21), N
2.  Convert the TWT beam current from counts to amps
        WLEV1(19) = EUPOLY(EUC(1,10),ITLM(27),0.0)
        GO TO 22
3.  Convert the TWT cathode voltage from counts to kilowatts
        WLEV1(20) = EUPOLY(EUC(1,11),ITLM(27),0.0)
        GO TO 22
4.  Convert the TWT HVPS temperature from counts to °C
        WLEV1(21) = EUPOLY(EUC(1,12)ITLM(27),0.0)
        GO TO 22
5.  Convert the TWT collector temperature from counts to °C
        WLEV1(22) = EUPOLY(EUC(1,13),ITLM(27),0.0)
        GO TO 22
6.  GO TO 22
7.  Convert the receiver temperature from counts to °C
        WLEV1(23) = EUPOLY(EUC(1,14),ITLM(27),0.0)
        GO TO 22
8.  Convert the noise gate amplitude from counts to volts
        WLEV1(24) = EUPOLY(EUC(1,15),ITLM(27),WLEV1(23)-TMPRCV)
        GO TO 22
```

```
9.  Convert the plateau gate amplitude from counts to volts
          WLEV1(25) = EUPOLY(EUC(1,16),ITLM(27),WLEV1(23)-TMPRCV)
          GO TO 22
10. Convert the attitude gate amplitude from counts to volts
          WLEV1(26) = EUPOLY(EUC(1,17),ITLM(27),WLEV1(23)-TMPRCV)
          GO TO 22
11. Convert the transmit power from counts to kilowatts
          D         = 0.3658690176*(WLEV1(50)-TMPMTU) + ITLM(27)
          WLEV1(27) = EUC(6,18) + D*(EUC(5,18) + D*(EUC(4,18)
         1  + D*(EUC(3,18) + D*(EUC(2,18) + D*(EUC(1,18)))))) + EUC(8,18)
          GO TO 22
12. Convert the UCFM temperature from counts to °C
          WLEV1(28) = EUPOLY(EUC(1,19),ITLM(27),0.0)
          GO TO 22
13. Convert the DDL temperature from counts to °C
          WLEV1(29) = EUPOLY(EUC(1,20),ITLM(27),0.0)
          GO TO 22
14. Convert the DDL ASSY temperature from counts to °C
          WLEV1(30) = EUPOLY(EUC(1,21),ITLM(27),0.0)
          GO TO 22
15. Convert the HSWS temperature from counts to °C
          WLEV1(31) = EUPOLY(EUC(1,22),ITLM(27),0.0)
          GO TO 22
16. Convert the DFB temperature #1 from counts to °C
          WLEV1(32) = EUPOLY(EUC(1,23),ITLM(27),0.0)
          GO TO 22
17. Convert the AT#1 temperature from counts to °C
          WLEV1(33) = EUPOLY(EUC(1,24),ITLM(27),0.0)
          GO TO 22
18. Convert the AT#2 temperature from counts to °C
          WLEV1(34) = EUPOLY(EUC(1,25),ITLM(27),0.0)
          GO TO 22
19. Convert the ICU temperature from counts to °C
          WLEV1(35) = EUPOLY(EUC(1,26),ITLM(27),0.0)
          GO TO 22
20. Convert the SACU temperature from counts to °C
          WLEV1(36) = EUPOLY(EUC(1,27),ITLM(27),0.0)
          GO TO 22
```

```
21. Convert the LVPS temperature from counts to °C
          WLEV1(37) = EUPOLY(EUC(1,28),ITLM(27),0.0)
22. End of algorithm
          RETURN
```

COMMENTS: 1. The following TYPE statements must be included in the code of this submodule

```
          COMMON /EUCOEF/ EUC(8,100)
          COMMON /TLMTRY/ ITLM(100)
          COMMON /STATUS/ ISTAT(100)
          COMMON /SENSOR/ WLEV1(300)
          COMMON /TMPREF/ TMPRCV,TMPMTU
```

EU ENGINEERING SUBCOM #2 CONVERSION SUBMODULE A.A-1.1.6

TITLE: SUBROUTINE EU2COM

FUNCTION: EU conversion of engineering subcom #2

REFERENCE: <u>SEASAT-1 Radar Altimeter Phase I Engineering Assessment Report,</u> WFC-135-80-001, December 1980

CONTROL: A.A-1.1 EUCONV EU conversion module

SUPPORT: A.A-1.1.2.1 EUPOLY EU polynomial conversion submodule

ACCESS: CALL EU2COM

INPUTS:
ITLM(28) = EU subcom #2 counts
EUC = EU conversion coefficients
ISTAT(11) = subcom counter (1 to 20)

OUTPUTS: WLEV1(38-51) = subcom #2 in engineering units

TABLES: Table A.1 TM Format (see Appendix A)

ALGORITHM:

```
1. Branch to the appropriate subcom word
          N = ISTAT(11)
          GO TO (2,3,4,...,20,21), N
2. Convert the LVPS 38V current from counts to amps
          WLEV1(38) = EUPOLY(EUC(1,29),ITLM(28),0.0)
          GO TO 22
3. Convert the +28V S/C bus isolated from counts to volts
          WLEV1(39) = EUPOLY(EUC(1,30),ITLM(28),0.0)
          GO TO 22
4. Convert the +28V from counts to volts
          WLEV1(40) = EUPOLY(EUC(1,31),ITLM(28),0.0)
          GO TO 22
```

```
5.  Convert the +15V from counts to volts
          WLEV1(41) = EUPOLY(EUC(1,32),ITLM(28),0.0)
          GO TO 22
6.  Convert the -15V from counts to volts
          WLEV1(42) = EUPOLY(EUC(1,33),ITLM(28),0.0)
          GO TO 22
7.  Convert the +7V from counts to volts
          WLEV1(43) = EUPOLY(EUC(1,34),ITLM(28),0.0)
          GO TO 22
8.  Convert the -9V from counts to volts
          WLEV1(44) = EUPOLY(EUC(1,35),ITLM(28),0.0)
          GO TO 22
9.  Convert the +5V from counts to volts
          WLEV1(45) = EUPOLY(EUC(1,36),ITLM(28),0.0)
          GO TO 22
10. Convert the -5.2V from counts to volts
          WLEV1(46) = EUPOLY(EUC(1,37),ITLM(28),0.0)
          GO TO 22
11. Convert the +1.00V REF from counts to volts
          WLEV1(47) = EUPOLY(EUC(1,38),ITLM(28),0.0)
          GO TO 22
12. Convert the 0.657V REF from counts to volts
          WLEV1(48) = EUPOLY(EUC(1,39),ITLM(28),0.0)
          GO TO 22
13. Convert the SACU PLO LOCK from counts to volts
          WLEV1(49) = EUPOLY(EUC(1,40),ITLM(28),0.0)
          GO TO 22
14. Convert the MTU temperature from counts to °C
          WLEV1(50) = EUPOLY(EUC(1,41),ITLM(28),0.0)
          GO TO 22
15. GO TO 22
16. Convert the DFB temperature #2 from counts to °C
          WLEV1(51) = EUPOLY(EUC(1,42),ITLM(28),0.0)
          GO TO 22
17. GO TO 22
18. GO TO 22
19. GO TO 22
20. GO TO 22
21. CONTINUE
```

```
22. End of algorithm
        RETURN
```

COMMENTS: 1. The following TYPE statements must be included in the code of this submodule

```
COMMON /EUCOEF/ EUC(8,100)
COMMON /TLMTRY/ ITLM(100)
COMMON /STATUS/ ISTAT(100)
COMMON /SENSOR/ WLEV1(300)
```

EU ENGINEERING SUBCOM #3 CONVERSION SUBMODULE A.A-1.1.7

TITLE: SUBROUTINE EU3COM

FUNCTION: EU conversion of engineering subcom #3

REFERENCE: <u>SEASAT-1 Radar Altimeter Phase I Engineering Assessment Report</u>, WFC-135-80-001, December 1980

CONTROL: A.A-1.1 EUCONV EU conversion module

SUPPORT: A.A-1.1.1.1 IEUBIT EU bit extraction submodule

ACCESS: CALL EU3COM

INPUTS: ITLM(29) = EU subcom #3 counts
ISTAT(11) = subcom counter (1 to 20)

OUTPUTS: ISTAT(29-42) = subcom #3 statuses

TABLES: Table A.1 TM Format (see Appendix A)

ALGORITHM:

```
1. Branch to the appropriate subcom word
        N = ISTAT(11)
        IF (N .GT. 6) GO TO 8
        GO TO (2,3,4,5,6,7), N
2. Decode word #1
        ISTAT(29) = IEUBIT(ITLM(29),3,3)
        ISTAT(30) = IEUBIT(ITLM(29),2,2)
        ISTAT(31) = IEUBIT(ITLM(29),1,1)
        GO TO 8
3. GO TO 8
4. Decode word #3
        ISTAT(32) = IEUBIT(ITLM(29),10,9)
        ISTAT(33) = IEUBIT(ITLM(29),8,7)
        ISTAT(34) = IEUBIT(ITLM(29),6,3)
        GO TO 8
```

```
5.  Decode word #4
        ISTAT(35) = IEUBIT(ITLM(29),10,7)
        ISTAT(36) = IEUBIT(ITLM(29),6,3)
        GO TO 8
6.  Decode word #5
        ISTAT(37) = IEUBIT(ITLM(29),10,9)
        ISTAT(38) = IEUBIT(ITLM(29),8,7)
        ISTAT(39) = IEUBIT(ITLM(29),6,5)
        ISTAT(40) = IEUBIT(ITLM(29),4,3)
        GO TO 8
7.  Decode word #6
        ISTAT(41) = IEUBIT(ITLM(29),10,7)
        ISTAT(42) = IEUBIT(ITLM(29),6,3)
8.  End of algorithm
        RETURN
```

COMMENTS: 1. The following TYPE statements must be included in the code of this submodule

```
COMMON /TLMTRY/ ITLM(100)
COMMON /STATUS/ ISTAT(100)
```

EU RAIN SUBCOM CONVERSION SUBMODULE A.A-1.1.8

TITLE: SUBROUTINE EURAIN

FUNCTION: EU conversion of the rain detection subcom

REFERENCE: Goldhirsh, J. and E. Walsh, "Precipitation Measurements from Space Using A Modified Seasat Type Radar Altimeter," JHU/APL S1R81U-022, May 1981.

CONTROL: A.A-1.1 EUCONV EU conversion module

SUPPORT: A.A-1.1.2.1 EUPOLY EU polynomial fit submodule

ACCESS: CALL EURAIN

INPUTS:
EUC = telemetry data
ITLM = telemetry data
ISTAT = status words
WLEV1 = level 1 output products

OUTPUTS: WLEV1 = level 1 output products

TABLES: None

ALGORITHM:

```
1.  End of algorithm
        RETURN
```

COMMENTS:

1. The following TYPE statements must be included in the code of this submodule

```
COMMON /EUCOEF/ EUC(8,100)
COMMON /TLMTRY/ ITLM(100)
COMMON /STATUS/ ISTAT(100)
COMMON /SENSOR/ WLEV1(300)
```

2. Algorithm will be defined at a later date
3. If subcom counter skips, set output parameter to -9999

EU WAVEFORM, CW, OR DUMP SUBMODULE A.A-1.1.9

TITLE: SUBROUTINE EUWFRM

FUNCTION: EU processing of waveform, CW, or dump data

REFERENCE: None

CONTROL: A.A-1.1 EUCONV EU conversion module

SUPPORT: None

ACCESS: CALL EUWFRM

INPUTS:
EUC = EU conversion coefficients
ITLM = telemetry data
ISTAT = status words

OUTPUTS: WLEV1 = level 1 output products

TABLES: None

ALGORITHM:

```
1.  Check for CW code
        IF (ISTAT(18) .EQ. 0) GO TO 5
2.  Check for memory dump
        IF (ISTAT(6) .EQ. 1) GO TO 6
3.  Extract the waveform samples
        DO 10  I = 31,93
        J        = I + 129
        10 WLEV1(J) = ITLM(I)
        GO TO 7
5.  Process CW mode
        GO TO 7
6.  Process memory dump
        GO TO 7
7.  End of algorithm
        RETURN
```

COMMENTS: 1. The following TYPE statements must be included in the code of this submodule

```
COMMON /EUCOEF/ EUC(8,100)
COMMON /TLMTRY/ ITLM(100)
COMMON /STATUS/ ISTAT(100)
COMMON /SENSOR/ WLEV1(300)
```

2. Algorithms for CW mode and memory dump will be defined at a later date

LEVEL 1 ALTITUDE CORRECTION MODULE A.A-1.2

TITLE: SUBROUTINE ALICOR

FUNCTION: To act as the driver module for the calculation of the sensor-related corrections to the altitude, the AGC, and the time, for all non-calibration and non-adaptive resolution mode data

REFERENCE: Not applicable

CONTROL: A.A-1.0 ALT1DR level 1 driver module

SUPPORT:

A.A-1.2.1	TIMCOR	time tag correction submodule
A.A-1.2.2	CZBIAS	cal zone bias submodule
A.A-1.2.3	OFFSET	C.G. offset submodule
A.A-1.2.4	CMBIAS	cal mode bias submodule

ACCESS: CAL ALICOR

INPUTS:

ALTO = uncorrected altitude in (m,0.001)
TMJDO = uncorrected modified julian date in (d,1.0x10^{-8})
ISTAT(25) = altimeter designation flag (1 or 2)
WLEV1(9) = uncorrected AGC in (dB,0.01)
WLEV1(119) = time tag correction in (s,0.0001)
WLEV1(126) = altitude cal zone bias in (m,0.001)
WLEV1(128) = altitude C.G. offset in (m,0.001)
WLEV1(130) = altitude cal mode bias in (m,0.001)
WLEV1(132) = AGC cal mode bias in (dB,0.01)

OUTPUTS:

ALT1 = corrected altitude in (m,0.001)
TMJD1 = corrected modified julian date in (d,1.0x10^{-8})
WLEV1(120) = year - 1900
WLEV1(121) = month
WLEV1(122) = day
WLEV1(123) = hour
WLEV1(124) = minute

WLEV1(125) = second
WLEV1(135) = corrected AGC in (dB,0.01)

TABLES: None

ALGORITHM:

```
1. Determine the net time tag correction
        CALL TIMCOR
2. Correct the modified julian date
        DELT        = DBLE(WLEV1(119)) / 86400.0D+00
        TMJD1       = TMJD0 + DELT
3. Determine the hour
        MJD         = TMJD1
        DHOUR       = (TMJD1-MJD) * 24.0D+00
        IHOUR       = DHOUR
        WLEV1(123) = IHOUR
4. Determine the minute
        DMIN        = (DHOUR-IHOUR) * 60.0D+00
        IMIN        = DMIN
        WLEV1(124) = IMIN
5. Determine the second
        WLEV1(125) = (DMIN-IMIN) * 60.0D+00
6. If same day, and same altimeter as last time, skip to Step #10
        IF (IHOUR .LT. 12) MJD = MJD + 1
        IF (ISTAT(25) .NE. LALT) GO TO 7
        IF (MJD .EQ. LMJD) GO TO 10
7. Determine the year
        ICHECK = 0
        DO 100 I = 2,25
        J = I - 1
        IF (MJD .LT. IREF(I)) GO TO 150
    100 CONTINUE
    150 IYEAR = J + 74
        WLEV1(120) = IYEAR
8. Determine the month
        K = 1
        IF (MOD(IYEAR,4) .EQ. 0) K = 2
        IDOY = MJD - IREF(J)
        DO 200 I = 2,12
        J = I - 1
        IF (IDOY .LT. JREF(I,K)) GO TO 250
    200 CONTINUE
```

```
   250 IMON = J
       WLEV1(121) = IMON
9.  Determine the day
       IDAY       = IDOY - JREF (IMON,K)
       WLEV1(122) = IDAY
10. Determine if the altitude and AGC correction must be computed
       IF (ICHECK .EQ. 1) GO TO 15
11. Determine the cal zone bias
       CAL CZBIAS
12. Determine the C.G. offset
       CALL OFFSET
13. Determine the cal mode bias
       CALL CMBIAS
14. Check the validity of the altitude and AGC corrections
       DH1  = WLEV1(126)
       DH2  = WLEV1(128)
       DH3  = WLEV1(130)
       DAGC = WLEV1(132)
       IF (DH1 .LT. -9000.0) DH1 = 0.0
       IF (DH2 .LT. -9000.0) DH2 = 0.0
       IF (DH3 .LT. -9000.0) DH3 = 0.0
       IF (DAGC .LT. -9000.0) DAGC = 0.0
       DALT = DBLE(DH1+DH2+DH3)
       IF (DALT.GT.-9000.0D+00 .AND. DAGC.GT.-9000.0) ICHECK = 1
       LALT = ISTAT(25)
       LMJD = MJD
15. Apply the altitude and AGC corrections
       ALT1       = ALT0 + DALT
       WLEV1(135) = WLEV1(9) + DAGC
16. End of algorithm
       RETURN
```

COMMENTS: 1. The following TYPE statements must be included in the code of this module:

```
       DOUBLE PRECISION TMJD0,TMJD1,DELT,ALT0,ALT1,DALT
       DIMENSION IREF(25),JREF(12,2)
       COMMON /STATUS/ ISTAT(100)
       COMMON /SENSOR/ WLEV1(300)
       EQUIVALENCE (ALT0,WLEV1(  5)) , (TMJD0,WLEV1(115))
       EQUIVALENCE (ALT1,WLEV1(133)) , (TMJD1,WLEV1(117))
```

```
      DATA LMJD/-9999/ , LALT/-9999/
      DATA IREF/   0, 365, 731,1096,1461,1826,2192,
     1          2557,2922,3287,3653,4018,4383,4748,
     2          5114,5479,5844,6209,6575,6940,7305,
     3          7670,8036,8401,8766/
      DATA JREF/  0, 31, 59, 90,120,151,181,212,
     1          243,273,304,334,  0, 31, 60, 91,
     2          121,152,182,213,244,274,305,335/
```

2. The year, month, and day, and the altitude and AGC corrections (Step #7 through Step #14) are only computed once per day
3. The altitude and AGC corrections are scheduled to be computed only once per day. However that calculation might be required more frequently because of mission operation design.

TIME TAG CORRECTION SUBMODULE A.A-1.2.1

TITLE: SUBROUTINE TIMCOR

FUNCTION: To compute the time tag correction, which accounts for the difference between the telemetry stream time tag (the time at which the reflected pulse reaches the telemeter) and the time at which the altimeter pulse is actually reflected from the earth's surface. The correction consists of a constant component, which is mode dependent, and a variable component, which is altitude dependent.

REFERENCE: "Seasat Algorithm Development Facility Altimeter Sensor Algorithm Specifications," JPL 622-207 Rev. A, March 1980

CONTROL: A.A-1.2 AL1COR level 1 altitude correction module

SUPPORT: None

ACCESS: CALL TIMCOR

INPUTS:
ISTAT(9) = operate status
ISTAT(25) = altimeter designator (1 or 2)
ALTO = uncorrected altitude in (m,0.001)
DTC = array of time corrections from Table 1.2.1
ALTLO = altitude lower limit in (m,1.0)
ALTHI = altitude higher limit in (m,1.0)

OUTPUTS: WLEV1(119) = time tag correction in (s,0.0001)

TABLES: Table 1.2.1 Constant Time Tag Correction (see Appendix B)

ALGORITHM: 1. Check input values

```
      IF (ISTAT(9).LT.1 .OR. ISTAT(9).GT.4) GO TO 5
```

```
      IF (ISTAT(25).LT.1 .OR. ISTAT(25).GT.2) GO TO 5
      IF (ALTO.LT.ALTLO .OR. ALTO.GT.ALTHI) GO TO 5
2.  Determine the constant time tag correction
      MODE        = ISTAT(9)
      IALT        = ISTAT(25)
      DT1         = DTC(MODE,IALT)
3.  Determine the variable time tag correction
      DT2         = ALTO / (C*2.0D+00)
4.  Determine the net time tag correction
      WLEV1(119) = DT1 + DT2
      GO TO 7
5.  Input out of range - print warning message and all input variables
6.  WLEV1(119) = -9999.0
7.  End of algorithm
      RETURN
```

COMMENTS: 1. The following TYPE statements must be included in the code of this submodule

```
      DOUBLE PRECISION ALTO,C
      COMMON /STATUS/ ISTAT(100)
      COMMON /SENSOR/ WLEV1(300)
      COMMON /ALTLIM/ ALTLO,ALTHI
      COMMON /T121/   DTC(4,2)
      EQUIVALENCE (ALTO,WLEV1(5))
      DATA C/2.99792458D+08/
```

CAL ZONE BIAS SUBMODULE A.A-1.2.2

TITLE: SUBROUTINE CZBIAS

FUNCTION: To set the altitude cal zone bias, which maximizes the absolute accuracy of the altitude measurement, and its associated standard deviation. The correction is based upon the analysis of altimeter data segments taken directly over laser tracking stations.

REFERENCE: None

CONTROL: A.A-1.2 AL1COR level 1 altitude correction module

SUPPORT: None

ACCESS: Call CZBIAS

INPUTS:
ISTAT(25) = altimeter designation flag (1 or 2)
TMJD1 = modified julian date in (d,1.0×10^{-8})

DCZB = array of time from Table 1.2.1
CZB = array of cal zone biases from Table 1.2.2
SCZB = array of standard deviations from Table 1.2.2

OUTPUTS:
WLEV1(126) = altitude cal zone bias in (m,0.001)
WLEV1(127) = cal zone bias standard deviation in (m,0.001)

TABLES: Table 1.2.2 Cal Zone Bias and Standard Deviation (see Appendix B)

ALGORITHM:

```
1.  Check input values
          IF (ISTAT(25).LT.1 .OR. ISTAT(25).GT.2) GO TO 4
          IF (TMJD1.LT.0.0D+00 .OR. TMJD1.GT.1.0D+05) GO TO 4
2.  Select the proper table entry index
          DO 100 I = 2,5
          K = I - 1
          IF (DCZB(I) .GT. TMJD1) GO TO 3
      100 CONTINUE
          K = 5
3.  Set the cal zone bias and its standard deviation
          IFLAG       = ISTAT(25)
          WLEV1(126) = CZB(K,IFLAG)
          WLEV1(127) = SCZB(K,IFLAG)
          GO TO 6
4.  Input out of range - print warning message and all input variables
5.  WLEV1(126) = -9999.0
    WLEV1(127) = -9999.0
6.  End of algorithm
          RETURN
```

COMMENTS:

1. The following TYPE statements must be included in the code of this submodule:

```
DOUBLE PRECISION TMJD1,DCZB
COMMON /STATUS/ ISTAT(100)
COMMON /SENSOR/ WLEV1(300)
COMMON /T122/   DCZB(5),CZB(5,2),SCZB(5,2)
EQUIVALENCE (TMJD1,WLEV1(117))
```

2. This submodule will be accessed only once per day

CENTER OF GRAVITY OFFSET SUBMODULE A.A-1.2.3

TITLE: SUBROUTINE OFFSET

FUNCTION: To set the altitude center of gravity offset correction and its standard deviation. The correction depends upon the expenditure of onboard fuel and the position of each of the altimeters with respect to the center of gravity of the spacecraft.

REFERENCE: None

CONTROL: A.A-1.2 AL1COR level 1 altitude correction module

SUPPORT: None

ACCESS: CALL OFFSET

INPUTS:

ISTAT(25) = altimeter designation flag (1 or 2)
TMJD1 = modified julian date in (d,1.0×10^{-8})
DCGO = array of time from Table 1.2.3
CGO = array of C.G. offsets from Table 1.2.3
SCGO = array of standard deviations from Table 1.2.3

OUTPUTS:

WLEV1(128) = altitude C.G. offset in (m,0.001)
WLEV1(129) = C.G. offset standard deviation in (m,0.001)

TABLES: Table 1.2.3 C.G. Offset and Standard Deviation (see Appendix B)

ALGORITHM:

```
1.  Check input values
          IF (ISTAT(25).LT.1 .OR. ISTAT(25).GT.2) GO TO 4
          IF (TMJD1.LT.0.0D+00 .OR. TMJD1.GT.1.0D+05) GO TO 4
2.  Select the proper table entry index
          DO 100 I = 2,10
          K = I - 1
          IF (DCGO(I) .GT. TMJD1) GO TO 3
      100 CONTINUE
          K = 10
3.  Set the C.G. offset and standard deviation
          IFLAG      = ISTAT(25)
          WLEV1(128) = CGO(K,IFLAG)
          WLEV1(129) = SCGO(K,IFLAG)
          GO TO 6
4.  Input out of range - print warning message and all input variables
5.  WLEV1(128) = -9999.0
    WLEV1(129) = -9999.0
6.  End of algorithm
```

COMMENTS: 1. The following TYPE statements must be included in the code of this submodule:

```
DOUBLE PRECISION TMJD1,DCGO
COMMON /STATUS/ ISTAT(100)
COMMON /SENSOR/ WLEV1(300)
COMMON /T123/   DCGO(10),CGO(10,2),SCGO(10,2)
EQUIVALENCE (TMJD1,WLEV1(5))
```

2. This submodule is scheduled to be accessed only once per day. However, the C.G. offset might be required more frequently.

CAL MODE BIAS SUBMODULE A.A-1.2.4

TITLE: SUBROUTINE CMBIAS

FUNCTION: To set the altitude and AGC cal mode bias corrections and standard deviations. The bias corrections maximize the relative accuracies in order to maintain consistent output products.

REFERENCE: None

CONTROL: A.A-1.2 AL1COR level 1 altitude correction module

SUPPORT: None

ACCESS: CALL CMBIAS

INPUTS:
ISTAT(25) = altimeter designation flag (1 or 2)
TMJD1 = modified julian date in (d,1.0x10^{-8})
TCMB = array of times from Table 1.2.4
CMB = array of altitude biases from Table 1.2.4
SCMB = array of standard deviations from Table 1.2.4
AGCB = array of AGC biases from Table 1.2.4
SAGCB = array of AGC standard deviations from Table 1.2.4

OUTPUTS:
WLEV1(130) = altitude cal mode bias in (m,0.001)
WLEV1(131) = cal mode bias standard deviation in (m,0.001)
WLEV1(132) = AGC cal mode bias in (dB,0.01)
WLEV1(144) = AGC cal mode bias standard deviation in (dB,0.01)

TABLES: Table 1.2.4 Cal Mode Biases and Standard Deviations (see Appendix B)

ALGORITHM:

1. Check input values

```
IF (ISTAT(25).LT.1 .OR. ISTAT(25).GT.2 GO TO 4
IF (TMJD1.LT.0.0D+00 .OR. TMJD1.GT.1.0D+05) GO TO 4
```

2. Select the proper table entry index

```
DO 100 I = 2,5
K = I - 1
IF (DCMB(I) .GT. TMJD1) GO TO 3
```

```
      100 CONTINUE
          K = 5
3.  Set the cal mode biases and standard deviation
          IFLAG       = ISTAT(25)
          WLEV1(130) = CMB(K,IFLAG)
          WLEV1(131) = SCMB(K,IFLAG)
          WLEV1(132) = AGCB(K,IFLAG)
          WLEV1(144) = SAGCB(K,IFLAG)
          GO TO 6
4.  Input out of range - print warning message and all input variables
5.  WLEV1(130) = -9999.0
    WLEV1(131) = -9999.0
    WLEV1(132) = -9999.0
    WLEV1(144) = -9999.0
6.  End of algorithm
          RETURN
```

COMMENTS:

1. The following TYPE statements must be included in the code of this submodule:

```
          DOUBLE PRECISION TMJD1,DCMB
          COMMON /STATUS/ ISTAT(100)
          COMMON /SENSOR/ WLEV1(300)
          COMMON /T124/   DCMB(5),CMB(5,2),SCMB(5,2),AGCB(5,2),SAGCB(5,2)
          EQUIVALENCE (TMJD1,WLEV1(5))
```

2. This submodule will be accessed only once per day

CAL MODE MODULE A.A-1.3

TITLE: SUBROUTINE CALMOD

FUNCTION: To control the processing and evaluation of all calibration mode data, which is designed to detect changes in altitude, AGC, and other parameters due to aging, temperature and voltage fluctuations, etc. The cal mode data, normally commanded once per day, will last for 60 seconds, the first 44 seconds being Cal I and the last 16 seconds being Cal II.

REFERENCE: None

CONTROL: A.A-1.0 ALT1DR level 1 driver module

SUPPORT:

A.A-1.3.1	CAL1	Cal I processing submodule
A.A-1.3.2	CAL2	Cal II processing submodule
A.A-1.3.3	TREND	trend file processing submodule

ACCESS: CALL CALMOD (NFINAL)

INPUTS:

NFINAL = 0 for all but last Cal II entry
= 1 for last Cal II entry (wrap up)
ISTAT(22) = Cal I indicator
ISTAT(23) = Cal II indicator

OUTPUTS: None

TABLES: None

ALGORITHM:

```
1. Determine the Cal type (NCAL = 1 for Cal I, NCAL = 2 for Cal II,
   NCAL = -1 for improper input)
         IF (NFINAL .EQ. 1) GO TO 5
         NCAL = -1
         IF (ISTAT(22) .EQ. 1) NCAL = 1
         IF (ISTAT(23) .EQ. 1) NCAL = 2
         IF (NCAL .EQ. 1) GO TO 2
         IF (NCAL .EQ. 2) GO TO 3
         GO TO 7
2. Process Cal I data if present
         CALL CAL1 (0)
         MCAL = 1
         GO TO 8
3. Wrap up Cal I data if required
         IF (MCAL .EQ. 1) CALL CAL1 (1)
4. Process Cal II data
         CALL CAL2 (0)
         MCAL = 2
         GO TO 8
5. Wrap up Cal II data if required
         CALL CAL2 (1)
6. Process Trend File
         CALL TREND
         GO TO 8
7. Input out of range - print warning message and all input variables
8. End of algorithm
         RETURN
```

COMMENTS:

1. The following TYPE statement must be included in the code of this submodule:

```
COMMON /STATUS/ ISTAT(100)
```

2. MCAL is the type of calibration for the previous module entry (MCAL = 1 for Cal I or MCAL = 2 for Cal II)

CAL I PROCESSING SUBMODULE A.A-1.3.1

TITLE: SUBROUTINE CAL1

FUNCTION: To process all of the Cal I calibration mode data. Cal I is designed to measure transmitter/receiver power and RF pathlength changes and to verify the shape of the transmitted pulse. It will occur during the first 44 seconds of the calibration mode and will be divided into 11 steps of 4 seconds each.

REFERENCE: None

CONTROL: A.A-1.3 CALMOD cal mode monitor module

SUPPORT: A.A-1.3.1.1 STAT cal mode statistical accumulation submodule

ACCESS: CALL CAL1 (IEND)

INPUTS:

IEND = 0 for all but last Cal I entry
= 1 for last Cal I entry (wrap up)
ISTAT(1) = command out to SACU
ISTAT(11) = channel select (1 to 20)
ISTAT(25) = altimeter designator (1 or 2)
ALT = altitude in (m,0.001)
TMJD = modified julian date in (d,1.0x10^{-9})
WLEV1 = level 1 output products
XL = Cal I parameter lower edit limit array
XU = Cal I parameter upper edit limit array
XE = Cal I parameter expected value array
TL = Cal I parameter lower tolerance limit array
TU = Cal I parameter upper tolerance limit array
ST = Cal summary time array
SM = Cal summary mean array
SE = Cal summary engineering mean array

OUTPUTS: A.D-1.91 = Cal report file

TABLES: A table containing Cal I Parameter Edit Limits, Tolerances, and Nominal Values to be defined later.

ALGORITHM:

1. Check the value of the Cal I step number (NSTEP) against the previous Cal I step number (MSTEP). Normally, their difference should be either zero or one.

   a.
   ```
   K = ISTAT(25)
   IF (IEND .EQ. 1) GO TO 8
   NSTEP = ISTAT(1) - 4
   IF (NSTEP.LT.1 .OR. NSTEP.GT.11) GO TO (1-d)
   NTEST = NSTEP - MSTEP
   ```

```
        IF (NTEST .EQ. 0) GO TO 2
        IF (NTEST .EQ. 1) GO TO 8
    b.  Cal I step number out of order - print warning message
    c.  GO TO 8
    d.  Illegal value for Cal I step number - print warning message
    e.  GO TO 11
2.  Add previous record of subcom data to summations
    a.  M1 is indication of data present from subcom #1
        IF (M1 .EQ. 0) GO TO 100
        CALL STAT (2,X(M1),1,XL(M1,K),XU(M1,K),XE(M1,K),S1(M1),S2(M1),
      1           N1(M1),N2(M1),XMIN(M1),XMAX(M1),XM(M1),XV(M1),XS(M1))
    b.  M2 is indication of data present from subcom #2
    100 IF (M2 .EQ. 0) GO TO 3
        CALL STAT (2,X(M2),1,XL(M2,K),XU(M2,K),XE(M2,K),S1(M2),S2(M2),
      1           N1(M2),N2(M2),XMIN(M2),XMAX(M2),XM(M2),XV(M2),XS(M2))
3.  Save current record of subcom #1 data.  N is a counter based upon the
    channel select status (ISTAT(11)).  ISTAT(11) = 5 is skipped because
    subcom #1 contains no data for that value.
          N = ISTAT(11)
          IF (ISTAT(11) .EQ. 5) GO TO 200
          IF (ISTAT(11) .GT. 5) N = ISTAT(11) - 1
          M1 = N + 71
          X(M1) = WLEV1(N+18)
          GO TO 4
      200 M1 = 0
4.  Save current record of subcom #2 data.  N is a counter based upon the
    channel select status (ISTAT(11)).  ISTAT(11) = 14 and ISTAT(11) > 15
    are skipped because subcom #2 contains no data for those values.
          N = ISTAT(11)
          IF (ISTAT(11) .EQ. 14) GO TO 300
          IF (ISTAT(11) .GT. 15) GO TO 300
          IF (ISTAT(11) .EQ. 15) N = ISTAT(11) - 1
          M2 = N + 90
          X(M2) = WLEV1(N+37)
          GO TO 5
      300 M2 = 0
5.  Check for steady state condition (2 seconds elapsed since beginning
    of the Cal I step)
```

```
          TDIFF = (TMJD-TSTART) * 86400.0
          IF (TDIFF .LT. 2.0) GO TO 7
6.  Add previous record of non-subcom data to summations
          CALL STAT(2,X,71,XL(1,K),XU(1,K),XE(1,K),S1,S2,N1,N2
         1          XMIN,XMAX,XM,XV,XS)
7.  Save current record of non-subcom data
          DO 400 I = 1,63
          J = I + 159
      400 X(I)  = WLEV1(J)
          X(64) = ALT - ALTREF
          X(65) = WLEV1(7)
          X(66) = WLEV1(10)
          X(67) = WLEV1(15)
          X(68) = WLEV1(16)
          X(69) = WLEV1(17)
          X(70) = WLEV1(18)
          X(71) = WLEV1(9)
          GO TO 11
8.  Compute final statistics for previous step (non-subcom data)
    a.  If MSTEP = 0, no data present for statistical computations
          IF(MSTEP .EQ. 0) GO TO 10
          CALL STAT (3,X,71,XL(1,K),XU(1,K),XE(1,K),S1,S2,N1,N2,
         1          XMIN,XMAX,XM,XV,XS)
    b.  Print NSTEP and statistics (i.e., XM(I), XS(I), XMIN(I), and
        XMAX(I) for I=1 to 71)
    c.  Save means for summary and trend
          ST(MSTEP) = TSTART
          DO 500 I=1,71
      500 SM(I,MSTEP) = XM(I)
    d.  Determine if this is the final entry of Cal I data
          IF(IEND .EQ. 0) GO TO 10
9.  Compute final statistics for entire Cal I (subcom data)
    a.  CALL STAT (3,X(72),33,XL(72,K),XU(72,K),XE(72,K),S1(72),S2(72),
       1          N1(72),N2(72),XMIN(72),XMAX(72),XM(72),XV(72),XS(72))
    b.  Print statistics (i.e., XM(I), XS(I), XMIN(I), XMAX(I), and
        N1(I) for I=72 to 104)
    c.  Print summary of means, flagging parameters which lie outside
        tolerance limits
```

d. Save subcom means for trend

```
      DO 600 I = 1,33
      J = I + 71
  600 SE(I,1)=XM(J)
```

e.
```
   NSTEP = 0
   M1    = 0
   M2    = 0
```

10. Initialization section for next step (MSTEP = 0 for initial entry of entire calibration mode; N is the number of variables to be zeroed - 104 for initial entry of entire cal mode or 71 for initialization of each subsequent step)

```
      N = 71
      IF (MSTEP .GT. 0) GO TO 700
      N = 104
      DO 625 I = 1,12
      ST(I) = 0.0D+00
      DO 625 J = 1,71
  625 SM(J,I) = 0.0
      DO 650 I = 1,38
      DO 650 J = 1,2
  650 SE(I,J) = 0.0
  700 MSTEP  = NSTEP
      TSTART = TMJD
      CALL STAT (1,X,N,XL(1,K),XU(1,K),XE(1,K),S1,S2,N1,N2,
     1          XMIN,XMAX,XM,XV,XS)
```

11. End of algorithm

```
      RETURN
```

COMMENTS: 1. The following TYPE statements must be included in the code of this submodule:

```
      DOUBLE PRECISION TMJD,TSTART,ALT,ALTREF,ST
      DIMENSION X(104),S1(104),S2(104),N1(104),N2(104)
      DIMENSION XMIN(104),XMAX(104),XM(104),XV(104),XS(104)
      COMMON /STATUS/ ISTAT(100)
      COMMON /SENSOR/ WLEV1(300)
      COMMON /T131/ XL(104,2),XU(104,2),XE(104,2),TL(104,2),TU(104,2)
      COMMON /CALSUM/ ST(12),SM(71,12),SE(33,2)
      COMMON /ALTCAL/ ALTREF
      EQUIVALENCE (TMJD,WLEV1(117)) , (ALT,WLEV1(5))
```

```
DATA MSTEP/0/ , XM,XV,XS/312*0.0/ , N1,N2/208*0/
DATA M1,M2/2*0/
```

2. The previous record is used to accumulate the summations due to the possible occurrence of bad data in the last record of each step.
3. "Steady state" is presumed to be attained after the first two seconds of each step.

CAL MODE STATISTICAL ACCUMULATION SUBMODULE A.A-1.3.1.1

TITLE: SUBROUTINE STAT

FUNCTION: To calculate the mean, variance, standard deviation, minimum, and maximum of several data sets simultaneously, with the editing of spurious data.

REFERENCE: Not applicable

CONTROL: A.A-1.3.1.1 Cal I Processing Submodule
A.A-1.3.1.1 Cal II Processing Submodule

SUPPORT: None

ACCESS: CALL STAT (J,X,M,XL,XU,XE,S1,S2,N1,N2,XMIN,XMAX,XM,XV,XS)

The module is accessed in one of the three modes, depending upon the value of J:

J = 1 This mode is used to initialize constants. No data are supplied in this mode.

J = 2 This mode is used when data are supplied. The summations required for the calculation are accumulated.

J = 3 This mode is used to calculate the final statistics. Only the summations are supplied in this mode.

For example, when the Cal I submodule is processing cal mode data, it accesses SUBROUTINE STAT once with J = 1 at the beginning of each Cal I step. After the data has reached a steady state condition, SUBROUTINE STAT is accessed with J = 2 once for each input data record (approximately 40 times). At the end of each Cal I step, SUBROUTINE STAT is accessed once with J = 3 to calculate the final statistics.

The input variable M defines the number of separate data sets whose statistics are to be determined simultaneously. Each time that the module is accessed with J = 2, only one point for each of the M separate data sets is supplied.

INPUTS: J = 1 for initialization (no data are supplied)
= 2 for supplying data and accumulating statistics
= 3 for final wrap up (no data are supplied)

X = the vector of data (each element contains a data point for one of the M separate data sets)

M = the number of separate data sets for which statistics are to be calculated simultaneously and the dimension of X, XL, XU, XE, S1, S2, N1, N2, XMIN, XMAX, XM, XV, and XS

XL = the vector containing the lower edit limits

XU = the vector containing the upper edit limits

XE = the vector containing the expected mean values

OUTPUTS:

S1 = the vector containing the summation of the unedited points

S2 = the vector containing the summation of the squares of the unedited points

N1 = the vector containing the number of unedited points

N2 = the vector containing the number of edited points

XMIN = the vector containing the minimum unedited points

XMAX = the vector containing the maximum unedited points

XM = the vector containing the calculated mean values

XV = the vector containing the calculated variances

XS = the vector containing the calculated standard deviations

TABLES: None

ALGORITHM:

```
1.  Check input mode
    a.  IF (J .EQ. 1) GO TO 2
        IF (J .EQ. 2) GO TO 3
        IF (J .EQ. 3) GO TO 4
    b.  Erroneous value of J - print warning message
          GO TO 4
2.  Initialize parameters
          DO 100 I = 1,M
          XM(I) = 0.0
          XV(I) = 0.0
          XS(I) = 0.0
          S1(I) = 0.0
          S2(I) = 0.0
          XMIN(I) = +9999.0E+20
          XMAX(I) = -9999.0E+20
          N1(I) = 0
      100 N2(I) = 0
          GO TO 5
3.  Update the summations
          DO 200 I = 1,M
```

```
          IF (X(I) .LT. XL(I)) GO TO 150
          IF (X(I) .GT. XU(I)) GO TO 150
          CONS  = X(I) - XE(I)
          N1(I) = N1(I) + 1
          S1(I) = S1(I) + CONS
          S2(I) = S2(I) + CONS*CONS
          IF (X(I) .LT. XMIN(I)) XMIN(I) = X(I)
          IF (X(I) .GT. XMAX(I)) XMAX(I) = X(I)
          GO TO 200
      150 N2(I) = N2(I) + 1
      200 CONTINUE
          GO TO 5
```

4. Calculate the final statistics

```
          DO 300 I = 1,M
          IF (N1(I) .GE. 2) GO TO 250
          XM(I) = -9999.0
          XV(I) = -9999.0
          XS(I) = -9999.0
          GO TO 300
      250 CONS  = N1(I)
          XM(I) = S1(I)/CONS + XE(I)
          XV(I) = (CONS*S2(I)-S1(I)*S1(I)) / (CONS*(CONS-1.0))
          XS(I) = SQRT(XV(I))
      300 CONTINUE
```

5. End of algorithm

```
          RETURN
```

COMMENTS:

1. The following TYPE statements must be included in the code of this module:

```
          DIMENSION X(M),XL(M),XU(M),XE(M),S1(M),S2(M),N1(M),N2(M)
          DIMENSION XMIN(M),XMAX(M),XM(M),XV(M),XS(M)
```

2. The values of the S1, S2, N1, N2, XMIN, and XMAX vectors must not be altered by the controlling module until after SUBROUTINE STAT has been accessed with J = 3 since these vectors contain variables that are required for the final calculation of the statistical parameters.
3. The units of all of the input and output variables are consistent with the units of X. It should be noted however that imbedded in the calculations of the statistics is the sum of the squares of the unedited points. If the individual data points are large (in absolute value), then this summation could cause loss of accuracy due to truncation

error. To alleviate this potential problem, the expected mean value (XE) is subtracted from each data point prior to the calculation of the summations. Then, before the final calculation of the statistics, the summations are modified to remove the effects of subtracting out XE. This entire process is invisible to the controlling module. In fact, some elements of XE may be set equal to zero if truncation error is not a problem for that particular data set.

4. The XL, XU, and XE arrays are column vectors of doubly subscripted arrays which are dimensioned in the accessing module.

CAL II PROCESSING MODULE A.A-1.3.2

TITLE: SUBROUTINE CAL2

FUNCTION: To process all of the Cal II calibration mode data. Cal II data are designed to determine the aging characteristics of the system reference to noise. These data will occur during the last 16 seconds of the calibration mode.

REFERENCE: None

CONTROL: A.A-1.3 CALMOD cal mode monitor module

SUPPORT: A.A-1.3.1.1 STAT cal mode statistical accumulation submodule

ACCESS: CALL CAL2 (IEND)

INPUTS:
IEND = 0 for all but last Cal II entry
= 1 for last Cal II entry (wrap up)
ISTAT(11) = channel select (1 to 20)
ISTAT(25) = altimeter designator (1 or 2)
ALT = altitude in (m,0.001)
TMJD = modified julian date in (d,1.0×10^{-9})
WLEV1 = level 1 output products
YL = Cal II parameter lower edit limit array
YU = Cal II parameter upper edit limit array
YE = Cal II parameter expected value array
ZL = Cal II parameter lower tolerance limit array
ZU = Cal II parameter upper tolerance limit array
ST = Cal summary time array
SM = Cal summary mean array
SE = Cal summary engineering mean array

OUTPUTS: A.D-1.91 = Cal report file

TABLES: A table containing Cal II Parameter Edit Limits, Tolerances, and Nominal Values to be defined later.

ALGORITHM:

1. Check for first Cal II record; if first record (NS=0) then initialize summations

```
     IF (IEND .EQ. 1) GO TO 8
     IF (NS .EQ. 1) GO TO 2
     K = ISTAT(25)
     CALL STAT(1,Y,104,YL(1,K),YU(1,K),YE(1,K),S1,S2,N1,N2,
    1          YMIN,YMAX,YM,YV,YS)
     TSTART = TMJD
     NS = 1
```

2. Add previous record of subcom data to summations
   a. M1 = 0 is indication of data present in subcom #1

```
     IF (M1 .EQ. 0) GO TO 100
     CALL STAT (2,Y(M1),1,YL(M1,K),YU(M1,K),YE(M1,K),S1(M1),S2(M1),
    1          N1(M1),N2(M1),YMIN(M1),YMAX(M1),YM(M1),YV(M1),YS(M1))
```

   b. M2 = 0 is indication of data present in subcom #2

```
 100 IF (M2 .EQ. 0) GO TO 3
     CALL STAT (2,Y(M2),1,YL(M2,K),YU(M2,K),YE(M2,K),S1(M2),S2(M2),
    1          N1(M2),N2(M2),YMIN(M2),YMAX(M2),YM(M2),YV(M2),YS(M2))
```

3. Save the current record of subcom #1 data. N is a counter based upon the channel select status (ISTAT(11)). ISTAT(11) = 5 is skipped because subcom #1 contains no data for that value.

```
     N = ISTAT(11)
     IF (ISTAT(11) .EQ. 5) GO TO 200
     IF (ISTAT(11) .GT. 5) N = ISTAT(11) - 1
     M1 = N + 71
     Y(M1) = WLEV1(N+18)
     GO TO 4
 200 M1 = 0
```

4. Save current record of subcom #2 data. N is a counter based upon the channel select status (ISTAT(11)). ISTAT(11) = 14 and ISTAT(11) > 15 are skipped because subcom #2 contains no data for those values.

```
     N = ISTAT(11)
     IF (ISTAT(11) .EQ. 14) GO TO 300
     IF (ISTAT(11) .GT. 15) GO TO 300
     IF (ISTAT(11) .EQ. 15) N = ISTAT(11) - 1
     M2 = N + 90
     Y(M2) = WLEV1(N+37)
```

```
          GO TO 5
      300 M2 = 0
5.  Check for steady state condition
          TDIFF = (TMJD-TSTART) * 86400.0
          IF (TDIFF .LT. 6.0) GO TO 7
6.  Add previous record of non-subcom data to summations
          CALL STAT (2,Y,71,YL(1,K),YU(1,K),YE(1,K),S1,S2,N1,N2,
         1         YMIN,YMAX,YM,YV,YS)
7.  Save current record of non-subcom data
          DO 400 I = 1,63
          J = I + 159
      400 Y(I)  = WLEV1(J)
          Y(64) = ALT - ALTREF
          Y(65) = WLEV1(7)
          Y(66) = WLEV1(10)
          Y(67) = WLEV1(15)
          Y(68) = WLEV1(16)
          Y(69) = WLEV1(17)
          Y(70) = WLEV1(18)
          Y(71) = WLEV1(9)
          GO TO 9
8.  Compute final statistics for Cal II
    a.  CALL STAT (3,Y,104,YL(1,K),YU(1,K),YE(1,K),S1,S2,N1,N2,
       1         YMIN,YMAX,YM,YV,YS)
    b.  Print statistics (i.e., YM(I), YS(I), YMIN(I), and YMAX(I) for
        I = 1 to 104)
    c.  Save means for summary
        ST(12) = TSTART
        DO 500 I = 1,71
    500 SM(I,12) = YM(I)
        DO 600 I = 1,33
        J = I+71
    600 SE(I,2) = Y(J)
    d.  Print summary of means, flagging parameters which lie outside
        tolerance limits
    e.  M1 = 0
        M2 = 0
        NS = 0
```

9. End of algorithm

```
RETURN
```

COMMENTS: 1. The following TYPE statements must be included in the code of this submodule:

```
DOUBLE PRECISION TMJD,TSTART,ALT,ALTREF,ST
DIMENSION Y(104),S1(104),S2(104),N1(104),N2(104)
DIMENSION YMIN(104),YMAX(104),YM(104),YV(104),YS(104)
COMMON /STATUS/ ISTAT(100)
COMMON /SENSOR/ WLEV1(300)
COMMON /T132/ YL(104,2),YU(104,2)YE(104,2),ZL(104,2),ZU(104,2)
COMMON /CALSUM/ ST(12),SM(71,12),SE(33,2)
COMMON /ALTCAL/ ALTREF
EQUIVALENCE (TMJD,WLEV1(117)) , (ALT,WLEV1(5))
DATA M1,M2,NS/3*0/ , YM,YV,YS/312*0.0/ , N1,N2/208*0/
```

TREND FILE PROCESSING A.A-1.3.3

TITLE: SUBROUTINE TREND

FUNCTION: To identify long-range trends in the calibration mode data. A file of trends will be automatically maintained, and significant changes in altimeter parameters will be calculated and flagged. A human decision of the appropriate action to be taken to correct the flagged parameters will be required. The inputs to the module will be the Cal Report data and the trend file, and the outputs will be the updated trend file and a display of flagged parameters.

REFERENCE: Not applicable

CONTROL: A.A-1.3 CALMOD cal mode module

SUPPORT: None

ACCESS: CALL TREND

INPUTS: Cal Report data (A.D-1.91) and trend file (A.D-1.71).

OUTPUTS: Updated trend file (A.D-1.71) and trend report (A.D-1.92).

TABLES: A table of tolerances (not presently available). The maximum number of entries will be 100 for each of two altimeters.

ALGORITHM: The input Cal Report statistics will be compared with previous calibration mode summaries. A curve-fit technique will be employed to determine if any long-range trends are present. If significant long-range trends are found, the parameters involved will be flagged and displayed for human interpretation and reaction. This module is run once per calibration mode command, which is normally once per day.

COMMENTS:

1. The altimeter has internal calibration modes to detect changes in altitude, AGC, and other parameters attributable to aging, temperature, voltage fluctuation, etc. This mode will be employed for 60 seconds about once per day. The cal mode monitor will process the calibration mode data and generate calibration reports of comparisons with preflight nominal calibrations. These reports will require Mission Contractor interpretation when flagged changes are significant enough to justify updates to the cal mode bias submodule tables. The trend file processor is designed to identify long-range (one month to one year) trends in the calibration mode data. To this end, a file of trends will be automatically maintained by the altimeter processing software, and significant changes in altimeter parameters will be calculated and flagged. A human decision of the appropriate action to be taken to correct the flagged parameters will be required. It may take two to three weeks to verify that the characteristic is valid and, if valid, to implement a correction. Any faster changes will be detected by health status monitoring.
2. This is only a functional description. The algorithms will be defined at a later date.

ADAPTIVE RESOLUTION A.A-1.4

TITLE: SUBROUTINE ADPRES

FUNCTION: To process all adaptive resolution mode data. Adaptive resolution is a mode in which the altimeter has detected surface slope changes and automatically switched to a wider pulse width and different track constants to maintain lock. The input to the module will be the altimeter engineering units data and the output will be modified altitudes, AGC, and time tags.

REFERENCE: Not applicable

CONTROL: A.A-1.0 ALT1DR level 1 driver module

SU?PORT: None

ACCESS: CALL ADPRES

INPUTS: Altimeter engineering unit data from module A.A-1.1

OUTPUTS:
WLEV1(133) = modified altitude in (m,0.001)
WLEV1(135) = modified AGC in (dB,0.01)
WLEV1(117) = modified altitude time tag in (s,0.0001)

WLEV1(146) = total height correction in (m,0.001)
WLEV1(148) = AGC correction (dB,0.01)
WLEV1(119) = time-tag correction in (s,0.0001)

TABLES: Tables for each resolution step similar to those in A.A-1.2. Maximum entries are 30 for each of two altimeters.

ALGORITHM: Correct altitude, AGC, and time tags for offsets due to the selected pulse width and tracker characteristics. Generally these will be in the form of table lookups indexed by pulse width (five possible steps). This module will also process data from the CW mode.

Adaptive resolution will occur primarily over ice sheets and land. The corrections made by this module will make level 1 output products consistent with the normal track mode processed by A.A-1.2.

COMMENTS: 1. This is only a functional description.

LOCATION CLASSIFICATION MODULE A.A-1.5

TITLE: SUBROUTINE LOCATE

FUNCTION: To act as the driver module for the ephemeris interpolation software. This routine interpolates the spacecraft latitude, longitude, height above the reference ellipsoid, and attitude to the altimeter data time tag. The data is also catalogued for surface type classification.

REFERENCE: Not applicable

CONTROL: A.A-1.0 ALT1DR level 1 driver module

SUPPORT: A.A-1.5.1 PLHINT spacecraft ephemeris interpolation submodule
A.A-1.5.2 CLSIFY subsatellite point classification submodule

ACCESS: CALL LOCATE

INPUTS: None

OUTPUTS: None

TABLES: None

ALGORITHM:

```
1. Interpolate the spacecraft ephemeris and attitude
        CALL PLHINT
2. Determine if subsatellite point is water, land, or ice
        CALL CLSIFY
3. End of algorithm
        RETURN
```

COMMENTS: None

SPACECRAFT EPHEMERIS INTERPOLATION SUBMODULE A.A-1.5.1

TITLE: SUBROUTINE PLHINT

FUNCTION: To determine the spacecraft latitude, longitude, height above the reference ellipsoid, and attitude by interpolating the ephemeris file, A.D(M)-1.82, to the proper time.

REFERENCE: None

CONTROL: A.A-1.5 LOCATE location classification module

SUPPORT: None

INPUTS:
A.D(M)-1.82 = ephemeris file
TMJD = modified Julian date in $(d, 1.0 \times 10^{-9})$

OUTPUTS:
WLEV1(136) = spacecraft attitude
WLEV1(151) = geodetic latitude in (deg, 0.000001)
WLEV1(152) = longitude in (deg, 0.000001)
HGT = height above the reference ellipsoid in (m,0.001)
ISTAT(43) = subtrack direction flag

TABLES: None

ALGORITHM: TBD

COMMENTS:

1. The following TYPE statements must be included in the code of this submodule:

```
DOUBLE PRECISION TMJD,HGT
COMMON /STATUS/ ISTAT(100)
COMMON /SENSOR/ WLEV1(300)
EQUIVALENCE (TMJD,WLEV1(117)) , (HGT,WLEV1(137))
```

2. The spacecraft attitude is computed from onboard sensors and should not be confused with the attitude calculated in the Waveform Module, which is computed from analysis of waveform data.
3. The specific design of this algorithm for now is being left to the mission contractor.

SUBSATELLITE POINT CLASSIFICATION SUBMODULE A.A-1.5.2

TITLE: SUBROUTINE CLSIFY

FUNCTION: To classify the subsatellite point as water, land, or ice based upon the world surface map

REFERENCE: None

CONTROL: A.A-1.5 LOCATE location classification module

SUPPORT: None

ACCESS: CALL CLSIFY

INPUTS:
A.D(M)-1.83 = world surface map file
TMJD = modified Julian date in (d,$1.0\text{x}10^{-9}$)
WLEV1(151) = geodetic latitude in (deg,0.000001)
WLEV1(152) = longitude in (deg, 0.000001)

OUTPUTS: ISTAT(44) = zone flag (1 for water, 2 for land or ice)

TABLES: None

ALGORITHM: TBD

COMMENTS:

1. The following TYPE statements must be included in the code of this submodule:

```
DOUBLE PRECISION TMJD
COMMON /STATUS/ ISTAT(100)
COMMON /SENSOR/ WLEV1(300)
EQUIVALENCE (TMJD,WLEV1(117))
```

2. The specific design of this algorithm for now is being left to the mission contractor.

HEALTH/STATUS MONITOR DRIVER MODULE A.A-1.6

TITLE: SUBROUTINE HSMNTR

FUNCTION: To monitor critical instrument parameters and turn on system alarms when a potentially damaging or dangerous condition exists.

REFERENCE: Not applicable

CONTROL: A.A-1.0 ALT1DR level 1 driver module

SUPPORT:

A.A-1.6.1	HSDAYW	health/status monitor 1-day wrap up submodule
A.A-1.6.2	HSMINW	health/status monitor n-minute wrap up submodule
A.A-1.6.3	HSSTAT	health/status monitor status processing submodule
A.A-1.6.4	HSNONS	health/status monitor non-subcom processing submodule
A.A-1.6.5	HSSUB1	health/status monitor subcom #1 processing submodule
A.A-1.6.6	HSSUB2	health/status monitor subcom #2 processing submodule
A.A-1.6.7	HSSUB3	health/status monitor subcom #3 processing submodule

ACCESS: CALL HSMNT

INPUTS: None

OUTPUTS: None

TABLES: None

ALGORITHM:

1. Finalize statistics for 1-day averaging period if required

```
CALL HSDAYW
```

```
2. Finalize statistics for n-minute averaging period if required
        CALL HSMINW
3. Process status words
        CALL HSSTAT
4. Process non-subcom data
        CALL HSNONS
5. Process subcom #1
        CALL HSSUB1
6. Process subcom #2
        CALL HSSUB2
7. Process subcom #3
        CALL HSSUB3
8. End of algorithm
        RETURN
```

COMMENTS: 1. The variables contained in COMMON /HSCOM/ , which is used in the submodules supporting this module, are described in Tables IX(a) and IX(b).

HEALTH/STATUS MONITOR 1-DAY WRAP UP SUBMODULE A.A-1.6.1

TITLE: SUBROUTINE HSDAYW

FUNCTION: To check to see if the end of a 1-day averaging period has been reached. If it has, final statistics for the period are calculated and printed on the Altimeter Diagnostics File (A.D-1.93)

REFERENCE: Not applicable

CONTROL: A.A-1.6 HSMNTR health/status monitor driver module

SUPPORT: A.A-1.6.1.1 STATZ health/status monitor statistical submodule

ACCESS: CALL HSDAYW

INPUTS:
TMJD = modified Julian date in (d,1.0×10^{-9})
N1D = number of unedited points (1-day averaging)
N2D = number of edited points (1-day averaging)
ZMIND = minimum unedited points (1-day averaging)
ZMAXD = maximum unedited points (1-day averaging)
ZMD = means (1-day averaging)
ZSD = standard deviations (1-day averaging)

OUTPUTS:
A.D-1.93 = Altimeter Diagnostics Files
NEWDAY = 0 for same day
= 1 for new day

TABLES: None

ALGORITHM:

1. Check for a new 1-day averaging period by comparing the current modified Julian date (TMJD) with the last modified Julian date (TMJDL). For each modified Julian date, 1/2 day must be added since the Julian day changes at noon and not at midnight. If the modified Julian date does not cross midnight, then NEWDAY is set equal to zero. If the modified Julian date does cross midnight, then NEWDAY is set equal to one, unless the last modified Julian date was -9999, which indicates that the current record is the first record to be processed.

```
NEWDAY = 0
MJDL = TMJDL + 0.5D+00
MJD = TMJD + 0.5D+00
IF (MJDL .EQ. MJD) GO TO 5
IF (MJDL .LT. 0) GO TO 4
NEWDAY = 1
```

2. New 1-day averaging period - compute final statistics for the previous 1-day averaging period for each altimeter (J = 1 and 2).

```
CALL STATZ (3,1,75,1,1)
CALL STATZ (3,1,75,1,2)
```

3. Print final statistics for the previous 1-day averaging period. Included in this print will be the number of unedited points (N1D), the number of edited points (N2D), the minimum unedited points (ZMIND), the maximum unedited points (ZMAXD), the means (ZMD), and the standard deviations (ZSD) for each of the 43 parameters and for each altimeter. (See explanation of parameters in COMMON /HSCOM/ located in module A.A-1.6.)

4. Initialize statistics for next 1-day averaging period for each altimeter (J = 1 and 2)

```
CALL STATZ (1,1,75,1,1)
CALL STATZ (1,1,75,1,2)
```

5. End of algorithm

```
TMJDL = TMJD
RETURN
```

COMMENTS:

1. The following TYPE statements must be included in the code of this submodule:

```
DOUBLE PRECISION TMJD,TMJDL
COMMON /HSCOM/ Z(75),ZL(75),ZU(75),ZE(75),
1     S1D(75,2),S2D(75,2),N1D(75,2),N2D(75,2),
2     ZMIND(75,2),ZMAXD(75,2),ZMD(75,2),ZVD(75,2),ZSD(75,2),
```

```
     3     S1M(75,2),S2M(75,2),N1M(75,2),N2M(75,2),
     4     ZMINM(75,2),ZMAXM(75,2),ZMM(75,2),ZVM(75,2),ZSM(75,2),
     5     NEWDAY,ZAL(75,2),ZAU(75,2),NCNT(25)
       COMMON /SENSOR/ WLEV1(300)
       EQUIVALENCE (TMJD,WLEV1(117))
       DATA TMJDL/-9999.0D+00/
```

HEALTH/STATUS MONITOR STATISTICAL SUBMODULE A.A-1.6.1.1

TITLE: SUBROUTINE STATZ

FUNCTION: To calculate the mean, variance, standard deviation, minimum, and maximum of several data sets simultaneously, with the editing of spurious data

REFERENCE: Not applicable

CONTROL:
A.A-1.6.1 HSDAYW health/status monitor 1-day wrap up submodule
A.A-1.6.2 HSMINW health/status monitor n-minute wrap up submodule
A.A-1.6.3 HSSTAT health/status monitor status processing submodule
A.A-1.6.4 HSNONS health/status monitor non-subcom processing submodule
A.A-1.6.5 HSSUB1 health/status monitor subcom #1 processing submodule
A.A-1.6.6 HSSUB2 health/status monitor subcom #2 processing submodule

SUPPORT: None

ACCESS: CALL STATZ (J,K1,K2,L,N)

INPUTS:
J = 1 to initialize constants (no data are supplied)
= 2 to accumulate summations for later calculation of statistics
= 3 to calculate final statistics (no data are supplied)
K1 = starting value of Z array first index
K2 = ending value of Z array first index
L = 1 for 1-day averaging
= 2 for n-minute averaging
N = altimeter number
Z = data to be averaged
ZL = lower edit limits
ZU = upper edit limits
ZE = expected values

OUTPUTS:
S1D = summations of unedited points (1-day averaging)
S2D = summations of unedited points squared (1-day averaging)
N1D = number of unedited points (1-day averaging)
N2D = number of edited points (1-day averaging)
ZMIND = minimum unedited points (1-day averaging)

ZMAXD = maximum unedited points (1-day averaging)

ZMD = means (1-day averaging)

ZVD = variances (1-day averaging)

ZSD = standard deviations (1-day averaging)

S1M = summations of unedited points (n-minute averaging)

S2M = summations of unedited points squared (n-minute averaging)

N1M = number of unedited points (n-minute averaging)

N2M = number of unedited points (n-minute averaging)

ZMINM = minimum unedited points (n-minute averaging)

ZMAXM = maximum unedited points (n-minute averaging)

ZMM = means (n-minute averaging)

ZVM = variances (n-minute averaging)

ZSM = standard deviations (n-minute averaging)

TABLES: None

ALGORITHM:

```
1.  Check input values
          IF (J.LT.1 .OR. J.GT.3) GO TO 5
          IF (K1.LT.1 .OR. K1.GT.75) GO TO 5
          IF (K2.LT.1 .OR. K2.GT.75) GO TO 5
          IF (L.LT.1 .OR. L.GT.2) GO TO 5
          IF (N.LT.1 .OR. N.GT.2) GO TO 5
          IF (J. EQ. 1) GO TO 2
          IF (J. EQ. 2) GO TO 3
          GO TO 4
2.  Initialize parameters
          DO 100 I = K1,K2
          ZMD(I,N)   = 0.0
          ZVD(I,N)   = 0.0
          ZSD(I,N)   = 0.0
          S1D(I,N)   = 0.0
          S2D(I,N)   = 0.0
          ZMIND(I,N) = +9999.0E+20
          ZMAXD(I,N) = -9999.0E+20
          N1D(I,N)   = 0
          N2D(I,N)   = 0
          ZMM(I,N)   = 0.0
          ZVM(I,N)   = 0.0
          ZSM(I,N)   = 0.0
          S1M(I,N)   = 0.0
          S2M(I,N)   = 0.0
```

```
          ZMINM(I,N) = +9999.0E+20
          ZMAXM(I,N) = -9999.0E+20
          N1M(I,N)   = 0
      100 N2M(I,N)   = 0
          GO TO 6
3.  Update the summations
    a.  DO 200 I = K1,K2
        IF (Z(I) .LT. ZL(I)) GO TO 3-d
        IF (Z(I) .GT. ZU(I)) GO TO 3-d
        CONS = Z(I) - ZE(I)
        IF (L .EQ. 2) GO TO 3-c
    b.  1-day averaging
        N1D(I,N) = N1D(I,N) + 1
        S1D(I,N) = S1D(I,N) + CONS
        S2D(I,N) = S2D(I,N) + CONS*CONS
        IF (Z(I) .LT. ZMIND(I,N)) ZMIND(I,N) = Z(I)
        IF (Z(I) .GT. ZMAXD(I,N)) ZMAXD(I,N) = Z(I)
        GO TO 200
    c.  n-minute averaging
        N1M(I,N) = N1M(I,M) + 1
        S1M(I,N) = S1M(I,M) + CONS
        S2M(I,N) = S2M(I,M) + CONS*CONS
        IF (Z(I) .LT. ZMINM(I,N)) ZMINM(I,N) = Z(I)
        IF (Z(I) .GT. ZMAXM(I,N)) ZMAXM(I,N) = Z(I)
        GO TO 200
    d.  Edited point
        IF (L .EQ. 1) N2D(I,N) = N2D(I,N) + 1
        IF (L. EQ. 2) N2M(I,N) = N2M(I,N) + 1
    e.  End of loop
    200 CONTINUE
        GO TO 6
4.  Calculate the final statistics
    a.  DO 300 I=K1,K2
        IF (L. EQ. 2) GO TO 4-c
    b.  1-day averaging
        IF (N1D(I,N) .GE. 2) GO TO 225
        ZMD(I,N) = -9999.0
        ZVD(I,N) = -9999.0
        ZSD(I,N) = -9999.0
```

```
    GO TO 300
225 CONS = N1D(I,N)
    ZMD(I,N) = S1D(I,N)/CONS + ZE(I)
    ZVD(I,N) = (CONS*S2D(I,N) - S1D(I,N)**2) / (CONS*(CONS-1.0))
    ZSD(I,N) = SQRT(ZVD(I,N))
    GO TO 300
```

c. n-minute averaging

```
    IF(N1M(I,N) .GE. 2) GO TO 250
    ZMM(I,N) = -9999.0
    ZVM(I,N) = -9999.0
    ZSM(I,N) = -9999.0
    GO TO 300
250 CONS = N1M(I,N)
    ZMM(I,N) = S1M(I,N)/CONS + ZE(I)
    ZVM(I,N) = (CONS*S2M(I,N)-S1M(I,N)**2) / (CONS*(CONS-1.0))
    ZSM(I,N) = SQRT(ZVM(I,N))
```

d. End of loop

```
300 CONTINUE
    GO TO 6
```

5. Input out of range - print warning message and all input variables
6. End of algorithm

```
      RETURN
```

COMMENTS:

1. The following TYPE statements must be included in the code of this submodule:

```
      COMMON /HSCOM/ Z(75),ZL(75),ZU(75),ZE(75),
     1      S1D(75,2),S2D(75,2),N1D(75,2),N2D(75,2),
     2      ZMIND(75,2),ZMAXD(75,2),ZMD(75,2),ZVD(75,2),ZSD(75,2),
     3      S1M(75,2),S2M(75,2),N1M(75,2),N2M(75,2),
     4      ZMINM(75,2),ZMAXM(75,2),ZMM(75,2),ZVM(75,2),ZSM(75,2),
     5      NEWDAY,ZAL(75,2),ZAU(75,2),NCNT(25)
```

2. The units of all of the input and output variables are consistent with the units of Z. It should be noted however that imbedded in the calculations of the statistics is the sum of the squares of the unedited points. If the individual data points are large (in absolute value), then this summation could cause loss of accuracy due to truncation error. To alleviate this potential problem, the expected mean value (ZE) is subtracted from each data point prior to the calculation of the summations. Then, before the final calculation of the statistics, the summations are modified to remove the effects of subtracting out

ZE. This entire process is invisible to the controlling module. In fact, some elements of ZE may be set equal to zero if truncation error is not a problem for that particular data set.

HEALTH/STATUS MONITOR N-MINUTE WRAP UP SUBMODULE A.A-1.6.2

TITLE: SUBROUTINE HSMINW

FUNCTION: To check to see if the end of an n-minute averaging period has been reached, where the number of minutes in the averaging period is a program input (nominally 25). If it has, final statistics for the period are calculated and printed on the Altimeter Diagnostics File (A.D-1.93).

REFERENCE: Not applicable

CONTROL: A.A-1.6 HSMNTR health/status monitor driver module

SUPPORT: A.A-1.6.1.1 STATZ health/status monitor statistics submodule

ACCESS: CALL HSMINW

INPUTS:
TMJD = modified Julian date in (d,1.0x10^{-9})
IEXEC(4) = interval for n-minute averaging period
N1M = number of unedited points (n-minute averaging)
N2M = number of edited points (n-minute averaging)
ZMINM = minimum unedited points (n-minute averaging)
ZMAXM = maximum unedited points (n-minute averaging)
ZMM = means (n-minute averaging)
ZSM = standard deviations (n-minute averaging)

OUTPUTS: A.D-1.93 = Altimeter Diagnostics File

TABLES: None

ALGORITHM:

1. Check for a new n-minute averaging period by comparing the current Julian date (TMJD) converted to minutes past midnight (MIN) with the last modified Julian date (TMJDL) converted to minutes past midnight (MINL). If the difference is greater than the averaging period (IEXEC(4)), then a new n-minute averaging period has begun (unless the last modified Julian date was -9999, which indicates that the current record is the first record to be processed).

```
MINL = DMOD(TMJDL+0.5D+00,1.0D+00) * 1440.0D+00
MIN  = DMOD(TMJD+0.5D+00,1.0D+00) * 1440.0D+00
MDIFF = MIN - MINL
IF (MDIFF .LT. 0) MDIFF = MDIFF + 1440
IF (MDIFF .LT. IEXEC(4)) GO TO 5
IF (TMJDL .LT. 0.0D+00) GO TO 5
```

2. New n-minute averaging period - compute final statistics for the previous n-minute averaging period for each altimeter (J = 1 and 2).

   ```
   CALL STATZ (3,1,75,2,1)
   CALL STATZ (3,1,75,2,2)
   ```

3. Print the final statistics for the previous n-minute averaging period. Included in this print will be the number of unedited points (N1M), the number of edited points (N2M), the minimum unedited points (ZMINM), the maximum unedited points (ZMAXM), the means (ZMM), and the standard deviations (ZSM) for each of the 43 parameters and for each altimeter. (See explanation of parameters in COMMON /HSCOM/ located in module A.A-1.6.)

4. Initialize statistics for next n-minute averaging period for each altimeter (J = 1 and 2).

   ```
   CALL STATZ (1,1,75,2,1)
   CALL STATZ (1,1,75,2,2)
   ```

5. End of algorithm

   ```
   TMJDL = TMJD
   RETURN
   ```

COMMENTS: 1. The following TYPE statements must be included in the code of this submodule:

```
       DOUBLE PRECISION TMJD,TMJDL
       COMMON /HSCOM/ Z(75),ZL(75),ZU(75),ZE(75),
      1     S1D(75,2),S2D(75,2),N1D(75,2),N2D(75,2),
      2     ZMIND(75,2),ZMAXD(75,2),ZMD(75,2),ZVD(75,2),ZSD(75,2),
      3     S1M(75,2),S2M(75,2),N1M(75,2),N2M(75,2),
      4     ZMINM(75,2),ZMAXM(75,2),ZMM(75,2),ZVM(75,2),ZSM(75,2),
      5     NEWDAY,ZAL(75,2),ZAU(75,2),NCNT(25)
       COMMON /EXECUT/ IEXEC(100)
       COMMON /SENSOR/ WLEV1(300)
       EQUIVALENCE (TMJD,WLEV1(117))
       DATA TMJDL/-9999.0D+00/
```

HEALTH/STATUS MONITOR STATUS PROCESSING SUBMODULE A.A-1.6.3

TITLE: SUBROUTINE HSSTAT

FUNCTION: To monitor critical status bits for changes and potentially damaging or dangerous conditions

REFERENCE: Not applicable

CONTROL: A.A-1.6 HSMNTR health/status monitor driver module
SUPPORT: A.A-1.6.1.1 STATZ health/status monitor statistical submodule
ACCESS: CALL HSSTAT
INPUTS: TMJD = modified Julian date in (d,$1.0\mathrm{x}10^{-9}$)
ISTAT(2) = HV on
ISTAT(3) = HV ready
ISTAT(4) = TWT fault
ISTAT(7) = rain processing enable
ISTAT(9) = mode command
ISTAT(14) = ACQ/TRK
ISTAT(18) = chirp/CW
ISTAT(20) = TWTA fault reset
ISTAT(21) = trigger kill
ISTAT(24) = TWT heater ON/OFF
ISTAT(25) = altimeter designator
ISTAT(26) = program version
ISTAT(27) = tracker type
ISTAT(28) = resolution step
WLEV1(139) = status #1
OUTPUTS: A.D-1.93 = Altimeter Diagnostics File
A.D-CRT = Master Control CRT File
NCNT(1) = HV on counter
NCNT(2) = HV off counter
NCNT(3) = altimeter on counter
NCNT(4) = standby counter
NCNT(5) = calibrate counter
NCNT(6) = trigger kill counter
NCNT(7) = track 1 counter
NCNT(8) = track 2 counter
NCNT(9) = track 3 counter
NCNT(10) = track 4 counter
NCNT(11) = TWT fault reset counter
NCNT(12) = test mode 1 (CW) counter
NCNT(13) = test mode 2 counter
NCNT(14) = test mode counter
NCNT(15) = test mode counter
NCNT(16) = adapt. resolution counter
NCNT(17) = TBD counter
NCNT(18) = TBD counter

NCNT(19) = TBD counter

NCNT(20) = rain processing counter

NCNT(21) = ACQ/TRK counter

NCNT(22) = chirp mode counter

NCNT(23) = CW mode counter

NCNT(24) = trigger kill counter

NCNT(25) = TWTA fault reset counter

TABLES: None

ALGORITHMS:

```
1. Process the Cal Atten/SACU Status
    a. Check HV ON for changes
          IF (ISTAT(2) .EQ. 1) NCNT(1) = NCNT(1) + 1
          IF (ISTAT(2) .EQ. 0) NCNT(2) = NCNT(2) + 1
          IF (NFIRST .EQ. 1) GO TO 1-b
          IF (ISTAT(2) .NE. JSTAT(1)) Print HV ON status change notice
    b. Check HV ready for changes
          IF (ISTAT(3) .EQ. 1) GO TO 1-c
          IF (JSTAT(2) .EQ. 0) GO TO 100
          TMJDX = TMJD
          NCNT(3) = NCNT(3) + 1
      100 IF ((TMJD-TMJDX)*1440.0D+00 .GT. 3.0) Print and display
            HV ready alarm
    c. Check TWT fault
          IF (ISTAT(4) .EQ. 1) Print and display TWT fault alarm
2. Process status #1 - accumulate time for each mode and look for changes
          N = ISTAT(9)
          NCNT(N+4) = NCNT(N+4) + 1
          IF (ISTAT(7) .EQ. 1) NCNT(20) = NCNT(20) + 1
          NSTAT1 = WLEV1(139)
          IF (NFIRST .EQ. 1) GO TO 3
          IF (NSTAT1 .NE. JSTAT(3)) print change of status #1 notice
3. Process status #3 - count changes in the ACQ/TRK status
          IF (ISTAT(14) .NE. JSTAT(4)) NCNT(21) = NCNT(21) + 1
4. Process status #4 - accumulate times and check for alarms
          IF (ISTAT(9) .LT. 3 .OR. ISTAT(9) .GT. 6) GO TO 100
          IF (ISTAT(18) .EQ. 0) NCNT(22) = NCNT(22) + 1
          IF (ISTAT(18) .EQ. 1) NCNT(23) = NCNT(23) + 1
          IF (ISTAT(21) .EQ. 1) NCNT(24) = NCNT(24) + 1
      100 IF (ISTAT(20) .EQ. 1) NCNT(25) = NCNT(25) + 1
```

```
          IF (ISTAT(20) .EQ. 1) print and display TWTA fault reset alarm
          IF (ISTAT(24) .EQ. 0) print and display TWT heater ON/OFF alarm
5.  Process status #5 - report changes
          IF (ISTAT(25) .NE. JSTAT(5)) print change of altimeter notice
          IF (ISTAT(26) .NE. JSTAT(6)) print change of program version
            notice
          IF (ISTAT(14) .NE. 1) GO TO 6
          IF (ISTAT(27) .NE. JSTAT(7)) print change of tracker type notice
          IF (ISTAT(28) .NE. JSTAT(8)) print change of resolution step
            notice
6.  Reset all test words so that the next data record can be compared
    with this data record
          NFIRST = 0
          JSTAT(1) = ISTAT(2)
          JSTAT(2) = ISTAT(3)
          JSTAT(3) = NSTAT1
          JSTAT(4) = ISTAT(14)
          JSTAT(5) = ISTAT(25)
          JSTAT(6) = ISTAT(26)
          JSTAT(7) = ISTAT(27)
          JSTAT(8) = ISTAT(28)
7.  End of algorithm
          RETURN
```

COMMENTS:

1. The following TYPE statements must be included in this submodule:

```
      DOUBLE PRECISION TMJD,TMJDX
      DIMENSION JSTAT(8)
      COMMON /HSCOM/ Z(75),ZL(75),ZU(75),ZE(75),
     1     S1D(75,2),S2D(75,2),N1D(75,2),N2D(75,2),
     2     ZMIND(75,2),ZMAXD(75,2),ZMD(75,2),ZVD(75,2),ZSD(75,2),
     3     S1M(75,2),S2M(75,2),N1M(75,2),N2M(75,2),
     4     ZMINM(75,2),ZMAXM(75,2),ZMM(75,2),ZVM(75,2),ZSM(75,2),
     5     NEWDAY,ZAL(75,2),ZAU(75,2),NCNT(250
      COMMON /STATUS/ ISTAT(100)
      COMMON /SENSOR/ WLEV1(300)
      EQUIVALENCE (TMJD,WLEV1(117))
      DATA NFIRST/1/ , JSTAT/8*-9999/
```

2. "Print" refers to printing on the Altimeter Diagnostics File (A.D-1.93) and "display" refers to displaying on the Master Control CRT File (A.D-CRT).

HEALTH/STATUS MONITOR NON-SUBCOM PROCESSING SUBMODULE A.A-1.6.4

TITLE: SUBROUTINE HSNONS

FUNCTION: To monitor non-subcom critical instrument parameters. In particular, this submodule

1. accumulates statistics for selected non-subcom parameters over 1-day and n-minute averaging periods, and
2. sets off system alarms when potentially damaging or dangerous conditions are detected in selected non-subcom parameters.

REFERENCE: Not applicable

CONTROL: A.A-1.6 HSMNTR health/status monitor driver module

SUPPORT: A.A-1.6.1.1 STATZ health/status monitor statistical submodule

ACCESS: CALL HSNONS

INPUTS:

TMJD = modified Julian date in (d,1.0×10^{-9})
WLEV1(4) = altitude rate in (m/s,0.01)
WLEV1(7) = altitude error in (m,0.01)
WLEV1(8) = SWH in (m,0.01)
WLEV1(9) = AGC word in (dB,0.01)
WLEV1(10) = AGC gate amplitude in (v,0.1)
WLEV1(15) = noise gate amplitude in (v,0.1)
WLEV1(18) = transmit power in (kw,0.1)
WLEV1(136) = attitude (spacecraft) in (deg,0.01)
ISTAT(9) = operate status
ISTAT(12) = AFC mode
ISTAT(14) = ACQ/TRK status
ISTAT(18) = chirp/CW status
ISTAT(19) = high voltage ON/OFF status
ISTAT(25) = altimeter designator (1 or 2)
ISTAT(44) = zone flag

OUTPUT:

A.D-1.93 = Altimeter Diagnostics File
A.D-CRT = Master Control CRT
Z(1) = spacecraft attitude (HV ON only)
Z(2) = altitude rate (HV ON only)
Z(3) = altitude error (HV ON only)
Z(4) = SWH (HV ON only)
Z(5) = AGC word (HV ON only)
Z(6) = AGC gate amplitude (HV ON only)
Z(7) = noise gate amplitude (HV ON only)

Z(8) = transmit power (chirp mode; HV ON only)

Z(9) = transmit power (CW mode; HV ON only)

TABLES: None

ALGORITHM:

```
1.  Check for steady-state condition (i.e., high voltage must have been on
    for at least 2 seconds).  If high voltage is on, then NHVON is set
    equal to one.  If not, NHVON is set equal to zero.  When high voltage
    is turned from off to on, the modified Julian date is saved in THVON.
    a.  IF (ISTAT(19) .EQ. 0) GO TO 1-b
        IF (NHVON .EQ. 0) THVON = TMJD
        NHVON = 1
        GO TO 2
    b.  NHVON = 0
        GO TO 14
2.  Check for open ocean (ACQ/TRK status equal to 1 and zone flag equal to
    1) and for in track mode (operate status between 3 and 6).
          IF (ISTAT(14) .EQ. 0) GO TO 14
          IF (ISTAT(44) .NE. 1) GO TO 14
          IF (ISTAT(9) .LT. 3) GO TO 14
          IF (ISTAT(9) .GT. 6) GO TO 14
3.  Store spacecraft attitude and check for alarm condition
    a.  J = ISTAT(25)
        Z(1) = WLEV1(136)
        IF (Z(1).GT.ZAL(1,J) .AND. Z(1).LT.ZAU(1,J)) GO TO 4
    b.  Print and display the spacecraft attitude alarm
4.  Determine if mode is chirp or CW
          IF (ISTAT(18) .EQ. 0) GO TO 12
5.  Chirp mode - store altitude rate and check for alarm condition
    a.  Z(2) = WLEV1(4)
        IF (Z(2).GT.ZAL(2,J) .AND. Z(2).LT.ZAU(2,J)) GO TO 6
    b.  Print and display the altitude rate alarm
6.  Store altitude error and check for alarm condition
    a.  Z(3) = WLEV1(7)
        IF (Z(3).GT.ZAL(3,J) .AND. Z(3).LT.ZAU(3,J)) GO TO 7
    b.  Print and display the altitude error alarm
7.  Store SWH and check for alarm condition
    a.  Z(4) = WLEV1(8)
        IF (Z(4).GT.ZAL(4,J) .AND. Z(4).LT.ZAU(4,J)) GO TO 8
    b.  Print and display the SWH alarm
```

C-2

```
 8.  Store the AGC word and check for alarm condition
     a.  Z(5) = WLEV1(9)
         IF (Z(5).GT.ZAL(5,J) .AND. Z(5).LT.ZAU(5,J)) GO TO 9
     b.  Print and display the AGC word alarm
 9.  Store the AGC gate amplitude and check for alarm condition
     a.  Z(6) = WLEV1(10)
         IF (Z(6).GT.ZAL(6,J) .AND. Z(6).LT.ZAU(6,J)) GO TO 10
     b.  Print and display the AGC gate amplitude alarm
10.  Store the noise gate amplitude and check for alarm condition
     a.  Z(7) = WLEV1(15)
         IF (Z(7).GT.ZAL(7,J) .AND. Z(7).LT.ZAU(7,J)) GO TO 11
     b.  Print and display the noise gate amplitude alarm
11.  Chirp mode - store transmit power and check for alarm condition
     a.  Z(8) = WLEV1(18)
         IF (Z(8).GT.ZAL(8,J) .AND. Z(8).LT.ZAU(8,J)) GO TO 13
     b.  Print and display the chirp mode transmit power alarm
         GO TO 13
12.  CW mode - store transmit power and check for alarm condition
     a.  Z(9) = WLEV1(18)
         IF (Z(9).GT.ZAL(9,J) .AND. Z(9).LT.ZAU(9,J)) GO TO 13
     b.  Print and display the CW mode transmit power alarm
13.  Update statistics for both 1-day and n-minute averaging
     a.  All non-subcom data except transmit power (only attitude for CW
         mode)
           K2 = 7
           IF (ISTAT(18).EQ. 0) K2 = 1
           CALL STAT (2,1,K2,1,J)
           CALL STAT (2,1,K2,2,J)
     b.  Transmit power (N = 8 for chirp mode, N = 9 for CW mode)
           IF (ISTAT(18) .EQ. 1) N = 8
           IF (ISTAT(18) .EQ. 0) N = 9
           CALL STAT (2,N,N,1,J)
           CALL STAT (2,N,N,2,J)
14.  End of algorithm
           RETURN
```

COMMENTS: 1. The following TYPE statements must be included in the code of this submodule:

```
DOUBLE PRECISION TMJD,THVON
COMMON /HSCOM/ Z(75),ZL(75),ZU(75),ZE(75),
```

```
     1     S1D(75,2),S2D(75,2),N1D(75,2),N2D(75,2),
     2     ZMIND(75,2),ZMAXD(75,2),ZMD(75,2),ZVD(75,2),ZSD(75,2),
     3     S1M(75,2),S2M(75,2),N1M(75,2),N2M(75,2),
     4     ZMINM(75,2),ZMAXM(75,2),ZMM(75,2),ZVM(75,2),ZSM(75,2),
     5     NEWDAY,ZAL(75,2),ZAU(75,2),NCNT(25)
      COMMON /STATUS/ ISTAT(100)
      COMMON /SENSOR/ WLEV1(300)
      EQUIVALENCE (TMJD,WLEV1(117))
```

2. "Print" refers to printing on the Altimeter Diagnostics File (A.D-1.93) and "display" refers to displaying on the Master Control CRT File (A.D-CRT).

HEALTH/STATUS MONITOR SUBCOM #1 PROCESSING SUBMODULE A.A-1.6.5

TITLE: SUBROUTINE HSSUB1

FUNCTION: To monitor subcom no. 1 parameters. In particular, this submodule

1. accumulates statistics for subcom #1 parameters over 1-day and n-minute averaging periods, and
2. sets off system alarms when potentially damaging or dangerous conditions are detected.

REFERENCE: Not applicable

CONTROL: A.A-1.6 HSMNTR health/status monitor driver module

SUPPORT: A.A-1.6.1.1 STATZ health/status monitor statistical submodule

ACCESS: CALL HSSUB2

ISTAT(2) = HV ON

INPUTS: ISTAT(11) = channel select status (1 to 20)

ISTAT(12) = ATU mode

ISTAT(18) = chirp/CW (0 or 1)

ISTAT(25) = altimeter designator (1 or 2)

WLEV1(19) = TWT beam current

WLEV1(20) = TWT cathode voltage

WLEV1(21) = TWT HVPS temperature

WLEV1(22) = TWT collector temperature

WLEV1(23) = receiver temperature

WLEV1(24) = noise gate amplitude

WLEV1(25) = plateau gate amplitude

WLEV1(26) = attitude gate amplitude

WLEV1(27) = transmit power

WLEV1(28) = UCFM temperature
WLEV1(29) = DDL temperature
WLEV1(30) = DDL ASSY temperature
WLEV1(31) = HSWS temperature
WLEV1(32) = DFB temperature no. 1
WLEV1(33) = AT no. 1 temperature
WLEV1(34) = AT no. 2 temperature
WLEV1(35) = ICU temperature
WLEV1(36) = SACU temperature
WLEV1(37) = LVPS temperature

OUTPUTS: A.D-1.93 = Altimeter Diagnostics File
A.D-CRT = Master Control CRT
Z(10) = TWT beam current (HV ON only)
Z(11) = TWT cathode voltage (HV ON only)
Z(12) = TWT HVPS temperature (HV ON only)
Z(13) = TWT collector temperature (HV ON only)
Z(14) = receiver temperature (HV ON only)
Z(15) = noise gate amplitude (HV ON only)
Z(16) = plateau gate amplitude (HV ON only)
Z(17) = attitude gate amplitude (HV ON only)
Z(18) = transmit power (chirp mode; HV ON only)
Z(19) = transmit power (CW mode; HV ON only)
Z(20) = UCFM temperature (HV ON only)
Z(21) = DDL temperature (HV ON only)
Z(22) = DDL ASSY temperature (HV ON only)
Z(23) = HSWS temperature (HV ON only)
Z(24) = DFB temperature no. 1 (HV ON only)
Z(25) = AT no. 1 temperature (HV ON only)
Z(26) = AT no. 2 temperature (HV ON only)
Z(27) = ICU temperature (HV ON only)
Z(28) = SACU temperature (HV ON only)
Z(29) = LVPS temperature (HV ON only)
Z(44) = TWT HVPS temperature (HV OFF only)
Z(45) = TWT collector temperature (HV OFF only)
Z(46) = receiver temperature (HV OFF only)
Z(47) = noise gate amplitude (HV OFF only)
Z(48) = plateau gate amplitude (HV OFF only)
Z(49) = attitude gate amplitude (HV OFF only)
Z(50) = transmit power (chirp mode; HV OFF only)

Z(51) = transmit power (CW mode; HV OFF only)
Z(52) = UCFM temperature (HV OFF only)
Z(53) = DDL temperature (HV OFF only)
Z(54) = DDL ASSY temperature (HV OFF only)
Z(55) = HSWS temperature (HV OFF only)
Z(56) = DFB temperature no. 1 (HV OFF only)
Z(57) = AT no. 1 temperature (HV OFF only)
Z(58) = AT no. 2 temperature (HV OFF only)
Z(59) = ICU temperature (HV OFF only)
Z(60) = SACU temperature (HV OFF only)
Z(61) = LVPS temperature (HV OFF only)

TABLES: None

ALGORITHM:

```
1.  Set up indices and branch to appropriate subcom word
    a.  Set indices and check for HV ON.  N is the channel select (N = 5
        is not used) and M is the accumulation array index (see Tables 6.1
        and 6.2)
          J = ISTAT(25)
          N = ISTAT(11)
          M = N + 9
          IF (N.GE.6 .AND. N.LE.10) M = N + 8
          IF (ISTAT(2) .EQ. 1) GO TO 1-c
    b.  HV OFF (skip N = 1 and N = 2)
          IF (N .LE. 2) GO TO 23
          M = M + 32
    c.  Branch to appropriate subcom word
          GO TO 2,3,4,...,20,21), N
2.  TWT beam current
    a.  Z(M) = WLEV1(19)
        IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 22
    b.  Print and display the TWT beam current alarm
    c.  GO TO 22
3.  TWT cathode voltage
    a.  Z(M) = WLEV1(20)
        IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J) GO TO 22
    b.  Print and display the TWT cathode voltage alarm
    c.  GO TO 22
4.  TWT HVPS temperature
    a.  Z(M) = WLEV1(21)
        IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 22
```

```
    b.  Print and display the TWT HVPS temperature alarm
    c.  GO TO 22
5.  TWT collector temperature
    a.  Z(M) = WLEV1(22)
        IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 22
    b.  Print and display the TWT collector temperature alarm
    c.  GO TO 22
6.  No data
          GO TO 23
7.  Receiver temperature
    a.  Z(M) = WLEV1(23)
        IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 22
    b.  Print and display the receiver temperature alarm
    c.  GO TO 22
8.  Noise gate amplitude
    a.  Z(M) = WLEV1(24)
        IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 22
    b.  Print and display the noise gate amplitude alarm
    c.  GO TO 22
9.  Plateau gate amplitude
    a.  Z(M) = WLEV1(25)
        IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 22
    b.  Print and display the plateau gate amplitude alarm
    c.  GO TO 22
10. Attitude gate amplitude
    a.  Z(M) = WLEV1(26)
        IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 22
    b.  Print and display the attitude gate amplitude alarm
    c.  GO TO 22
11. Transmit power (chirp mode)
    a.  IF (ISTAT(18) .EQ. 0) GO TO 11-d
        Z(M) = WLEV1(27)
        IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 22
    b.  Print and display the transmit power (chirp mode) alarm
    c.  GO TO 22
    Transmit power (CW mode; accumulate only in Test Mode 1)
    d.  M  =  M + 1
        IF (ISTAT(12) .NE. 8) GO TO 24
```

```
            Z(M) = WLEV1(27)
            IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 22
        e.  Print and display the transmit power (CW mode) alarm
        f.  GO TO 22
12.     UCFM temperature
        a.  Z(M) = WLEV1(28)
            IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 22
        b.  Print and display the UCFM temperature alarm
        c.  GO TO 22
13.     DDL temperature
        a.  Z(M) = WLEV1(29)
            IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 22
        b.  Print and display the DDL temperature alarm
        c.  GO TO 22
14.     DDL ASSY temperature
        a.  Z(M) = WLEV1(30)
            IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 22
        b.  Print and display the DDL ASSY temperature alarm
        c.  GO TO 22
15.     HSWS temperature
        a.  Z(M) = WLEV1(31)
            IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 22
        b.  Print and display the HSWS temperature alarm
        c.  GO TO 22
16.     DFB temperature no. 1
        a.  Z(M) = WLEV1(32)
            IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 22
        b.  Print and display the DFB temperature no. 1 alarm
        c.  GO TO 22
17.     AT no. 1 temperature
        a.  Z(M) = WLEV1(33)
            IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 22
        b.  Print and display the AT no. 1 temperature alarm
        c.  GO TO 22
18.     AT no. 2 temperature
        a.  Z(M) = WLEV1(34)
            IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 22
        b.  Print and display the AT no. 2 temperature alarm
        c.  GO TO 22
```

```
19. ICU temperature
    a. Z(M) = WLEV1(35)
        IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 22
    b. Print and display the ICU temperature alarm
    c. GO TO 22
20. SACU temperature
    a. Z(M) = WLEV1(36)
        IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 22
    b. Print and display the SACU temperature alarm
    c. GO TO 22
21. LVPS temperature
    a. Z(M) = WLEV1(37)
        IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 22
    b. Print and display the LVPS temperature alarm
22. Update the statistics for 1-day and n-minute averaging
          CALL STATZ (2,M,M,1,J)
          CALL STATZ (2,M,M,2,J)
23. End of algorithm
          RETURN
```

COMMENTS: 1. The following TYPE statements must be included in the code of this submodule:

```
      COMMON /HSCOM/ Z(75),ZL(75),ZU(75),ZE(75),
     1     S1D(75,2),S2D(75,2),N1D(75,2),N2D(75,2),
     2     ZMIND(75,2),ZMAXD(75,2),ZMD(75,2),ZVD(75,2),ZSD(75,2),
     3     S1M(75,2),S2M(75,2),N1M(75,2),N2M(75,2),
     4     ZMINM(75,2),ZMAXM(75,2),ZMM(75,2),ZVM(75,2),ZSM(75,2),
     5     NEWDAY,ZAL(75,2),ZAU(75,2),NCNT(25)
      COMMON /STATUS/ ISTAT(100)
      COMMON /SENSOR/ WLEV1(300)
```

2. "Print" refers to printing on the Altimeter Diagnostics File (A.D-1.93) and "display" refers to displaying on the Master Control CRT File (A.D-CRT).

HEALTH/STATUS MONITOR SUBCOM # 2 PROCESSING SUBMODULE A.A-1.6.6

TITLE: SUBROUTINE HSSUB2

FUNCTION: To monitor subcom #2 parameters. In particular, this submodule

1. accumulates statistics for subcom #2 parameters over 1-day and n-minute averaging periods, and
2. sets off system alarms when potentially damaging or dangerous conditions are detected.

REFERENCE: Not applicable
CONTROL: A.A-1.6 HSMNTR health/status monitor driver module
SUPPORT: A.A-1.6.1.1 STATZ health/status monitor statistical submodule
ACCESS: CALL HSSUB2
INPUTS: ISTAT(2) = HV ON
ISTAT(11) = channel select status (1 to 20)
ISTAT(25) = altimeter designator (1 or 2)
WLEV1(38) = LVPS 38V current
WLEV1(39) = +28V S/C bus isolated
WLEV1(40) = +28V
WLEV1(41) = +15V
WLEV1(42) = -15V
WLEV1(43) = +7V
WLEV1(44) = -9V
WLEV1(45) = +5V
WLEV1(46) = -5.2V
WLEV1(47) = +1.00 REF
WLEV1(48) = 0.657V REF
WLEV1(49) = SACU PLO LOCK
WLEV1(50) = MTU temperature
WLEV1(51) = DFB temperature no. 2
OUTPUTS: A.D-1.93 = Altimeter Diagnostics File
A.D-CRT = Master Control CRT
Z(30) = LVPS 38V current
Z(31) = +28V S/C bus isolated
Z(32) = +28V
Z(33) = +15V
Z(34) = -15V
Z(35) = +7V
Z(36) = -9V
Z(37) = +5V
Z(38) = -5.2V
Z(39) = +1.00 REF
Z(40) = 0.657V REF
Z(41) = SACU PLO LOCK

Z(42) = MTU temperature
Z(43) = DFB temperature no. 2
Z(62) = LVPS 38V current (HV OFF only)
Z(63) = +28V S/C bus isolated (HV OFF only)
Z(64) = +28V (HV OFF only)
Z(65) = +15V (HV OFF only)
Z(66) = -15V (HV OFF only)
Z(67) = +7V (HV OFF only)
Z(68) = -9V (HV OFF only)
Z(69) = +5V (HV OFF only)
Z(70) = -5.2V (HV OFF only)
Z(71) = +1.00V REF (HV OFF only)
Z(72) = 0.657V REF (HV OFF only)
Z(73) = SACU PLO LOCK (HV OFF only)
Z(74) = MTU temperature (HV OFF only)
Z(75) = DFB temperature no. 2 (HV OFF only)

TABLES: None

ALGORITHM:

```
1. Set up indices and branch to appropriate subcom word
   a. Set indices and check for HV ON.  N is the channel select (N = 14
      and N > 15 are not used) and M is the accumulation array index
      (see Tables 6.1 and 6.2).
        J = ISTAT(25)
        N = ISTAT(11)
        M = N + 29
        IF (N.EQ.14 .OR. N.GT.15) GO TO 17
        IF (N .EQ. 15) M = N + 28
        IF (ISTAT(2) .EQ. 1) GO TO 1-c
   b. HV OFF
        M  =  M + 32
   c. Branch to appropriate subcom word
        GO TO (2,3,4,....,14,15), N
2. LVPS 38V current
   a. Z(M) = WLEV1(38)
      IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,U)) GO TO 16
   b. Print and display the LVPS 38V current alarm
   c. GO TO 16
3. +28V S/C bus isolated
   a. Z(M) = WLEV1(39)
      IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 16
```

```
        b.  Print and display the +28V S/C bus isolated alarm
        c.  GO TO 16
    4.  +28V
        a.  Z(M) = WLEV1(40)
            IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 16
        b.  Print and display the +28V alarm
        c.  GO TO 16
    5.  +15V
        a.  Z(M) = WLEV1(41)
            IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 16
        b.  Print and display the +15V alarm
        c.  GO TO 16
    6.  -15V
        a.  Z(M) = WLEV1(42)
            IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 16
        b.  Print and display the -15V alarm
        c.  GO TO 16
    7.  +7V
        a.  Z(M) = WLEV1(43)
            IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 16
        b.  Print and display the +7V alarm
        c.  GO TO 16
    8.  -9V
        a.  Z(M) = WLEV1(44)
            IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 16
        b.  Print and display the -9V alarm
        c.  GO TO 16
    9.  +5V
        a.  Z(M) = WLEV1(45)
            IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 16
        b.  Print and display the +5V alarm
        c.  GO TO 16
   10.  -5.2V
        a.  Z(M) = WLEV1(46)
            IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 16
        b.  Print and display the -5.2V alarm
        c.  GO TO 16
   11.  +1.00V REF
        a.  Z(M) = WLEV1(47)
```

```
        IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 16
    b.  Print and display the +1.00V REF alarm
    c.  GO TO 16
12. 0.657V REF
    a.  Z(M) = WLEV1(48)
        IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 16
    b.  Print and display the 0.657V REF alarm
    c.  GO TO 16
13. SACU PLO LOCK
    a.  Z(M) = WLEV1(49)
        IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 16
    b.  Print and display the SACU PLO LOCK alarm
    c.  GO TO 16
14. MTU temperature
    a.  Z(M) = WLEV1(50)
        IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 16
    b.  Print and display the MTU temperature alarm
    c.  GO TO 16
15. DFB temperature no. 2
    a.  Z(M) = WLEV1(51)
        IF (Z(M).GT.ZAL(M,J) .AND. Z(M).LT.ZAU(M,J)) GO TO 16
    b.  Print and display the DFB temperature no. 2 alarm
16. Update statistics for 1-day and n-minute averaging
          CALL STATZ (2,M,M,1,J)
          CALL STATZ (2,M,M,2,J)
17. End of algorithm
          RETURN
```

COMMENTS:

1. The following TYPE statements must be included in this submodule

```
      COMMON /HSCOM/ Z(75),ZL(75),ZU(75),ZE(75),
     1     S1D(75,2),S2D(75,2),N1D(75,2),N2D(75,2),
     2     ZMIND(75,2),ZMAXD(75,2),ZMD(75,2),ZVD(75,2),ZSD(75,2),
     3     S1M(75,2),S2M(75,2),N1M(75,2),N2M(75,2),
     4     ZMINM(75,2),ZMAXM(75,2),ZMM(75,2),ZVM(75,2),ZSM(75,2),
     5     NEWDAY,ZAL(75,2),ZAU(75,2),NCNT(25)
      COMMON /STATUS/ ISTAT(100)
      COMMON /SENSOR/ WLEV1(300)
```

2. "Print" refers to printing on the Altimeter Diagnostics File (A.D-1.93) and "display" refers to the displaying on the Master Control CRT File (A.D-CRT).

HEALTH/STATUS MONITOR SUBCOM #3 PROCESSING SUBMODULE A.A-1.6.7

TITLE: SUBROUTINE HSSUB3

FUNCTION: To monitor changes in the status words contained in subcom #3 and to print a report when any changes occur.

REFERENCE: Not applicable

CONTROL: A.A-1.6 HSMNTR health/status monitor driver module

SUPPORT: None

ACCESS: CALL HSSUB3

INPUTS:
ISTAT(11) = channel select status (1 to 20)
ISTAT(30) = LVPS current
ISTAT(31) = AT number
ISTAT(32) = acquisition constant index
ISTAT(33) = track constant index
ISTAT(34) = AGC threshold index
ISTAT(35) = α, β, and AGC acquisition
ISTAT(36) = α, β, and AGC track index
ISTAT(37) = ΔH gate width index
ISTAT(38) = L_6-E_6 track index
ISTAT(39) = height error index
ISTAT(40) = L_6-E_6 acquisition index
ISTAT(41) = waveheight curve offset
ISTAT(42) = acquisition height offset

OUTPUTS:
A.D-1.93 = Altimeter Diagnostics File
A.D-CRT = Master Control CRT File

TABLES: None

ALGORITHM:

1. Branch to appropriate subcom word (NCHG = 0 means no changes, NCHG = 1 means changes have occurred, MSTAT(I) = 0 means that no change has occurred in that particular status, MSTAT(I) = 1 means that a change has occurred in that particular status)

```
      J = ISTAT(25)
      N = ISTAT(11)
      IF (ISTAT(11) .GT. 5) GO TO 9
      GO TO (2,3,4,5,6), N
```

2. Check relay status for changes

```
      IF (ISTAT(29) .NE. JSTAT(1)) MSTAT(1) = 1
      IF (ISTAT(30) .NE. JSTAT(2)) MSTAT(2) = 1
      IF (ISTAT(29) .EQ. 0) print and display the TWTA fault
        override alarm
```

```
          IF (ISTAT(30) .EQ. 1) print and display the LVPS current
            override alarm
          IF ((MSTAT(1)-1)*(MSTAT(2)-1) .EQ. 0) NCHG = 1
          GO TO 9
    3. Check parameter select no. 1 for changes
          IF (ISTAT(32) .NE. JSTAT(3)) MSTAT(3) = 1
          IF (ISTAT(33) .NE. JSTAT(4)) MSTAT(4) = 1
          IF (ISTAT(34) .NE. JSTAT(5)) MSTAT(5) = 1
          IF ((MSTAT(3)-1)*(MSTAT(4)-1)*(MSTAT(5)-1) .EQ. 0) NCHG = 1
          GO TO 9
    4. Check parameter select no. 2 for changes
          IF (ISTAT(35) .NE. JSTAT(6)) MSTAT(6) = 1
          IF (ISTAT(36) .NE. JSTAT(7)) MSTAT(7) = 1
          IF ((MSTAT(6)-1)*(MSTAT(7)-1) .EQ. 0) NCHG = 1
          GO TO 9
    5. Check parameter select no. 3 for changes
          IF (ISTAT(37) .NE. JSTAT(8)) MSTAT(8) = 1
          IF (ISTAT(38) .NE. JSTAT(9)) MSTAT(9) = 1
          IF (ISTAT(39) .NE. JSTAT(10)) MSTAT(10) = 1
          IF (ISTAT(40) .NE. JSTAT(11)) MSTAT(11) = 1
          IF ((MSTAT(8)-1)*(MSTAT(9)-1)*(MSTAT(10)-1)*(MSTAT(11)-1)
         1     .EQ. 0) NCHG = 1
          GO TO 9
    6. Check parameter select no. 4 for changes
          IF (ISTAT(41) .NE. JSTAT(12)) MSTAT(12) = 1
          IF (ISTAT(42) .NE. JSTAT(13)) MSTAT(13) = 1
          IF ((MSTAT(12)-1)*(MSTAT(13)-1) .EQ. 0) NCHG = 1
          IF (NCHG .EQ. 0) GO TO 8
    7. Changes have occurred - print all subcom #3 statuses with an asterisk
       next to those statuses that have changed (MSTAT(I) = 1)
    8. Reset all test words so that the next data record can be compared
       with this data record
          DO 100 I = 1,13
          MSTAT(I) = 0
      100 JSTAT(I) = ISTAT(I+29)
          NCHG = 0
    9. End of algorithm
          RETURN
```

COMMENTS:

1. The following TYPE statements must be included in the code of this submodule

```
DIMENSION JSTAT(13),MSTAT(13)
COMMON /STATUS/ ISTAT(100)
DATA JSTAT,MSTAT/26*-9999/ , NCHG/0/
```

2. "Print" refers to printing on the Altimeter Diagnostic File (A.D-1.93) and "display" refers to displaying on the Master Control CRT File (A.D-CRT).

LEVEL 2 DRIVER MODULE A.A-2.0

TITLE: SUBROUTINE ALT2DR

FUNCTION: Driver module for the altimeter level 2 software

REFERENCE: Not applicable

CONTROL:	A.A-0.0	ALTMTR	altimeter main driver module
SUPPORT:	A.A-2.1	CONTAM	contamination processing module
	A.A-2.2	COMPRS	data compression module
	A.A-2.3	WAVFRM	waveform module
	A.A-2.4	ATMOS	atmospheric module
	A.A-2.5	WNDSIG	wind speed/backscatter coefficient module
	A.A-2.6	AL2COR	level 2 altitude correction module
	A.A-2.7	ICE	ice sheet height module
	A.A-2.8	SEAICE	sea ice module
	A.A-2.9	QUALTY	quality control module

ACCESS: Call ALT2DR (IFLAG)

INPUTS:
IEXEC = array containing executive parameters
ISTAT = array containing status words
WLEV1 = array containing level 1 output products

OUTPUTS:
ISTAT = array containing status words
WLEV2 = array containing level 2 output products
IFLAG = 0 if level 2 output to be written
= 1 if level 2 output not to be written

TABLES: None

ALGORITHM:

1. Classify the subsatellite point

```
CALL CONTAM
```

2. Compress the data

```
CALL COMPRS (IFLAG)
IF (IFLAG .EQ. 1) GO TO 11
```

```
3.  Process waveform data
          CALL WAVFRM
4.  Process atmospheric data
          CALL ATMOS
5.  Calculate the wind speed
          CALL WNDSIG
6.  Calculate the level 2 altitude corrections
          CALL AL2COR
7.  Check subsatellite point for ice or currents
          IF (ISTAT(42) .EQ. 1) GO TO 8
          IF (ISTAT(42) .EQ. 2) GO TO 9
          GO TO 10
8.  Ice sheet data
          CALL ICE
          GO TO 10
9.  Sea ice data
          CALL SEAICE
10. Monitor quality control
          CALL QUALTY (IFLAG)
11. End of algorithm
          RETURN
```

COMMENTS: 1. The following TYPE statements must be included in the code of this module:

```
COMMON /EXECUT/ IEXEC(100)
COMMON /STATUS/ ISTAT(100)
COMMON /SENSOR/ WLEV1(300)
COMMON /GEOPHY/ WLEV2(325)
```

CONTAMINATION PROCESSING A.A-2.1

TITLE: SURBOURINTE CONTAM

FUNCTION: To classify the nature of the subsatellite point as either land, water, or ice. This classification, which is derived from analysis of the altimeter data with a resolution of approximately 1 km, should not be confused with the classification performed in the location classification module (A.A-1.5), which is derived from a world surface map and has a resolution of approximately 10 km. The module will estimate the time and location of of land/water and ice/water interfaces. In addition, a flag will be set to indicate if the data are appropriate for processing by the waveform module.

REFERENCE: Martin, C. F. and R. L. Taylor, "Report on Boundary Detection Criteria for Satellite Altimeters," NASA CR-156880, September 1981.

CONTROL: A.A-2.0 ALT2DR level 2 driver module

SUPPORT: None

ACCESS: CALL CONTAM

INPUTS:
ISTAT(28) = adaptive resolution step flag
ISTAT(44) = subsatellite point classification flag from A.A-1.5
WLEV1(7) = altitude error in (m,0.01)
WLEV1(135) = AGC in (dB,0.01)
WLEV1(16) = plateau gate in (mv,0.01)
WLEV1(17) = attitude gate in (mv,0.01)
WLEV1(8) = significant wave height in (m,0.01)
WLEV1(114) = altitude acceleration in (m/s^2,0.01)

OUTPUTS: WLEV2(15) = subsatellite point classification flag (1 for water, 2 for ice, 3 for land, 4 for unknown)

TABLES: None

ALGORITHM: TBD

COMMENTS: 1. Only a functional description is given.

DATA COMPRESSION MODULE A.A-2.2

TITLE: SUBROUTINE COMPRS

FUNCTION: To compress the level 1 output to a selectable rate. Nonproduction data modes, such as cal mode, trigger kill, and standby will be edited. The module also calculates standard deviations of selected parameters for later quality analysis.

REFERENCE: Not applicable

CONTROL: A.A-2.0 ALT2DR level 2 driver module

SUPPORT: None

ACCESS: CALL COMPRS (IFLAG)

INPUTS:
IEXEC(3) = orbit number
IEXEC(5) = compression period in hundredths of a second
ISTAT(9) = mode command
ISTAT(11) = channelselect
ISTAT(25) = altimeter designator

ISTAT(44) = zone flag
WLEV1 = level 1 products

OUTPUTS: IFLAG = 0 if compression period is complete
= 1 if compression period is not complete
WLEV2 = level 2 output array

TABLES:
Table 2.2a Parameters For Which Means Only Are Calculated
Table 2.2b Parameters For Which Means and Standard Deviations Are Calculated
Table 2.2c Parameters Which Are Linearly Fit Without Computation of Standard Deviations
Table 2.2d Parameters Which Are Linearly Fit With Computation of Standard Deviations
Table 2.2e Special Parameters To Be Compressed

ALGORITHM:

1. Skip all but track mode data

```
      IFLAG = 1
      IF (ISTAT(9).LT.3 .OR. ISTAT(9).GT.6) GO TO 16
```

2. Check to make sure that the current time (TMJD) is greater than the last time (TL) and that it is less than the ending time of the compression period (T2)

```
      IF (TMJD .LT. TL) GO TO 9
      IF (TMJD .GT. T2) GO TO 10
```

3. Process new data for those parameters listed in Table 2.2e
   a. Altimeter number (ISTAT(25) is the altimeter number and LALT is the previous altimeter number)

```
      IF ISTAT(25) .NE. LALT) GO TO 10
      WLEV2(2) = ISTAT(25)
```

   b. Orbit number (TMJD is the current time, and TE is the evaluation time of the compression period, and IEXEC(3) is the orbit number).

```
      IF (TMJD .LE. TE) WLEV2(1) = IEXEC(3)
```

   c. Zone flag. If ISTAT(44) changes, then the zone flag is set to zero, indicating an undefined area. (LFLAG is the zone flag at the beginning of the compression period.)

```
      IF (ISTAT(44) .NE. LFLAG) LFLAG = 0
```

4. Add new data to those parameters listed in Table 2.2a. N1 is the total number of parameters, I1 is the array containing the WLEV1 indices, S1 is the array containing the summations, and M1 is the array containing the number of samples used in the summations.

```
      DO 100 N = 1, N1
      IF (N .LE. 8) GO TO 50
```

```
      IF (N.EQ.9 .AND. ISTAT(11).NE.13) GO TO 100
      IF (N.EQ.10 .AND. ISTAT(11).NE.15) GO TO 100
      IF (N.EQ.11 .AND. ISTAT(11).NE.4) GO TO 100
   50 I = I1(N)
      M1(N) = M1(N) + 1
      S1(N) = S1(N) + WLEV1(I)
  100 CONTINUE
```

5. Add new data to those parameters listed in Table 2.2b. N2 is the total number of parameters, I2 is the array containing the WLEV1 indices, S2A and S2B are the arrays containing the summations, and M2 is the array containing the number of samples used in the summations.

```
      DO 200 N = 1, N2
      I = I2(N)
      M2(N) = M2(N) + 1
      S2A(N) = S2A(N) + WLEV1(I)
  200 S2B(N) = S2B(N) + WLEV1(I)*WLEV1(I)
```

6. Add new data to those parameters listed in Table 2.2c. N3 is the total number of parameters, I3 is the array containing the WLEV1 indices, S3A, S3B, S3C, and S3E are the arrays containing the summations, M3 is the array containing the number of samples used in the summations and DT is the number of seconds elapsed since the beginning of the compression period. The assumption is made that the orbit is retrograde and that the longitude is always between 0 and 360 degrees.

```
      DT = (TMJD-T1) * 86400.0D+00
      DO 300 N = 1, N3
      I = I3(N)
      DX = WLEV1(I)
      IF (N.EQ.2 .AND. DX.GT.RLON) DX = DX - 360.0
      IF (N .EQ. 3) DX = HGT - HGTO
      M3(N) = M3(N) + 1
      S3A(N) = S3A(N) + DT
      S3B(N) = S3B(N) + DX
      S3C(N) = S3C(N) + DT*DX
  300 S3E(N) = S3E(N) + DT*DT
```

7. Add new data to those parameters listed in Table 2.2d. N4 is the total number of parameters, I4 is the array containing the WLEV1 indices, S4A, S4B, S4C, S4D, and S4E are the arrays containing the summations, and M4 is the array containing the number of samples used in the summations.

```
      DO 400 N = 1, N4
      I = I4(N)
      DX = WLEV1(I)
      IF (N .EQ. 1) DX = H - HO
      M4(N) = M4(N) + 1
      S4A(N) = S4A(N) + DT
      S4B(N) = S4B(N) + DX
      S4C(N) = S4C(N) + DT*DX
      S4D(N) = S4D(N) + DX*DX
  400 S4E(N) = S4E(N) + DT*DT
```

8. Save the current time and then skip to the end of the module.

```
      TL = TMJD
      GO TO 16
```

9. Current time (TMJD) is either less than the previous time (TL) print warning message and skip this record.

```
      GO TO 16
```

10. Check for the first entry to the module (IEXEC(5) is the compression period in hundredths of a second.).

```
      IF (TE .GT. 0.0) GO TO 11
      T2 = TMJD
      CP = DFLOAT(IEXEC(5)) / 86400.0D+02
      WLEV2(4) = 0.01 * IEXEC(5)
      GO TO 15
```

11. Compute the means of those parameters listed in Table 2.2a. N1 is the total number of parameters, J1 is the array containing WLEV2 indices, S1 is the array containing the summations, and M1 is the array containing the number of samples used in the summations.

```
      IFLAG = 0
      WLEV2(14) = LFLAG
      DO 500 N = 1, N1
      J = J1(N)
      WLEV2(J) = -9999.0
      IF (M1(N) .EQ. 0) GO TO 500
      WLEV2(J) = S1(N) / M1(N)
  500 CONTINUE
```

12. Compute the means and standard deviations about the means for those parameters listed in Table 2.2b. N2 is the total number of parameters, J2A is the array containing the WLEV2 mean indices, J3B is the array containing the WLEV2 standard deviation indices, S2A and

S2B are the arrays containing the summations, and M2 is the array containing the number of samples used in the summations.

```
      DO 600 N = 1, N2
      JA = J2A(N)
      JB = J2B(N)
      WLEV2(JA) = -9999.0
      WLEV2(JB) = -9999.0
      IF (M2(N) = EQ. 0) GO TO 600
      WLEV2(JA) = SA(N) / M2(N)
      IF (M2(N) .EQ. 1) GO TO 600
      WLEV2(JB) = SQRT((SB(N)*M2(N)-SA(N)*SA(N))/(M2(N)*(M2(N)-1)))
  600 CONTINUE
```

13. Compute the linear fit of those parameters listed in Table 2.2c. N3 is the total number of parameters, J3 is the array containing the WLEV2 indices, S3A, S3B, S3C, and S3E are the arrays containing the summations, M3 is the array containing the number of samples used in the summations and TE is the evaluation time.

```
      DO 700 N = 1, N3
      J = J3(N)
      WLEV2(J) = -9999.0
      IF (M3(N) .EQ. 0) GO TO 700
      D = S3A(N)*S3A(N) - S3E(N)*M3(N)
      A = (S3A(N)*S3B(N) - S3C(N)*M3(N)) / D
      B = (S3A(N)*S3C(N) - S3B(N)*S3E(N)) / D
      WLEV2(J) = A*TE + B
  700 CONTINUE
      IF (WLEV2(8) .LT. 0.0) WLEV2(8) = WLEV2(8) + 360.0
      HGTC = WLEV2(9) + HGTO
      IF (M3(3) .EQ. 0) HGTC = -9999.0D+00
```

14. Compute the linear fit and the standard deviation about the fit for those parameters listed in Table 2.2d. N4 is the total number of parameters, J4A is the array containing the WLEV2 mean indices, J4B is the array containing the WLEV2 standard deviation indices, S4A, S4B, S4C, S4D, and S4E are the arrays containing the summations, M4 is the array containing the number of samples used in the summations and TE is the evaluation time.

```
      DO 800 N = 1, N4
      JA = J4A(N)
      JB = J4B(N)
```

```
    WLEV2(JA) = -9999.0
    WLEV2(JB) = -9999.0
    IF (M4(N) .EQ. 0) GO TO 800
    D = S4A(N)*S4A(N) - S4C(N)*M4(N)
    A = (S4A(N)*S4B(N)-S4C(N)*M4(N)) / D
    B = (S4A(N)*S4C(N)-S4B(N)*S4E(N)) / D
    WLEV2(JA) = A*TE + B
    IF (M4(N) .EQ. 1) GO TO 750
    SH = S4D(N) - S4B(N)*S4B(N)/M4(N)
    WLEV2(JB) = SQRT(SH/(M4(N)*(M4(N)-1)))
800 CONTINUE
    WLEV2(13) = M4(1)
    HC = WLEV2(17) + HO
    IF (M4(1) .EQ. 0) HC = -9999.0D+00
```

15. Reset parameters for next compression period

   a. Set time parameters (T1 is the beginning time of the compression period, T2 is the ending time of the compression period, TE is the evaluation time within the compression period, and CP is the length of the compression period expressed in days).

```
900 IF (TMJD.GE.T1 .AND. TMJD.LT.T2) GO TO 910
    T1 = T2
    T2 = T1 + CP
    GO TO 900
910 TE = T1 + CP/2.0
```

   b. Reset the counters and summations

```
    DO 920 N = 1, N1
    M1(N) = 0
920 S1(N) = 0.0
    DO 930 N = 1, N2
    M2(N) = 0
    S2A(N) = 0.0
930 S2B(N) = 0.0
    DO 940 N = 1, N3
    M3(N) = 0
    S3A(N) = 0.0
    S3B(N) = 0.0
    S3C(N) = 0.0
940 S3E(N) = 0.0
    DO 950 N = 1, N4
```

```
      M4(N) = 0
      S4A(N) = 0.0
      S4B(N) = 0.0
      S4C(N) = 0.0
      S4D(N) = 0.0
  950 S4E(N) = 0.0
```

c. Save the initial conditions for longitude, spheroid height, and altitude

```
      RLON = WLEV1(152)
      HGTO = HGT
      HO = H
      LALT = ISTAT(25)
      LFLAG = ISTAT(44)
      GO TO 4
```

16. End of module

```
      RETURN
```

COMMENTS: 1. The following TYPE statements must be included in the code of this module:

```
      DOUBLE PRECISION TMJD,TE,TL,T1,T2,CP,HGT,HGTO,HGTC,H,HO,HC
      DIMENSION M1(11),S1(11),M2(93),S2A(93),S2B(93),M3(3),
     1          S3A(3),S3B(3)
      DIMENSION S3C(3),S3E(3),M4(5),S4A(5),S4B(5),S4C(5),S4D(5),
     1          S4E(5)
      COMMON /T22A/ N1,I1(11),J1(11)
      COMMON /T22B/ N2,I2(93),J2A(93),J2B(93)
      COMMON /T22C/ N3,I3(3),J3(3)
      COMMON /T22D/ N4,I4(5),J4A(5),J4B(5)
      COMMON /STATUS/ ISTAT(100)
      COMMON /EXECUT/ IEXEC(100)
      COMMON /SENSOR/ WLEV1(300)
      COMMON /GEOPHY/ WLEV2(325)
      EQUIVALENCE (H, WLEV1(5)) , (HGT, WLEV1(137))
      EQUIVALENCE (HC,WLEV2(17)) , (HGTC,WLEV2(9))
      EQUIVALENCE (TMJD, WLEV1(117))
      EQUIVALENCE (TE, WLEV2(5))
      DATA TE/-9999.0D+00/ , T2/-9999.0D+00/ , LALT/-9999/
```

WAVEFORM MODULE A.A-2.3

TITLE: SUBROUTINE WAVFRM

FUNCTION: Driver module for the level 2 waveform processing

REFERENCE: Not applicable

CONTROL: A.A-2.0 ALT2DR level 2 driver module

SUPPORT:
A.A-2.3.1 SWHFIT waveform processor driver submodule
A.A-2.3.2 WAVALT waveform altitude correction submodule
A.A-2.3.3 SWHCOR waveform SWH correction submodule
A.A-2.3.4 SWHBIAS waveform SWH bias submodule
A.A-2.3.5 WFPROD waveform products submodule

ACCESS: CALL WAVFRM

INPUTS: None

OUTPUTS: None

TABLES: None

ALGORITHM:

```
1.  Access waveform processor submodule
          CALL SWHFIT (NLO,NUP,WTY)
2.  Access waveform altitude correction submodule
          CALL WAVALT
3.  Access waveform SWH correction submodule
          CALL SWHCOR
4.  Access waveform SWH bias submodule
          CALL SWHBIAS
5.  Access waveform products submodule
          CALL WFPROD
6.  End of algorithm
          RETURN
```

COMMENTS: 1. The following TYPE statements must be included in the code of this submodule:

```
      DIMENSION WTY(63)
      DATA NLO/1/,NUP/63/ , WTY/63*0.01
```

These values are SEASAT specific.

WAVEFORM PROCESSOR DRIVER SUBMODULE A.A-2.3.1

TITLE: SUBROUTINE SWHFIT

FUNCTION: Driver submodule of the waveform modeled parameter recovery. This group of submodules fits a model waveform to the waveform data to obtain estimates of the parameters characterizing the modeled waveform. The particular modeled radar waveform used for SEASAT data analysis is characterized by six parameters: 1) amplitude; 2) time origin or track point; 3) ocean surface rms roughness; 4) noise baseline; 5) ocean surface skewness; and 6) attitude or off-nadir angle. These are also expected to characterize the NOSS waveform.

The time origin parameter is the location of the actual mean return waveform relative to the altimeter's altitude-tracker-positioned waveform sample set, and the time origin is thus directly interpretable as an altitude correction to be applied to the real-time altitude output. The ocean surface rms roughness provides a revised estimate of the significant waveheight (SWH). The ocean surface skewness parameter provides additional information about the surface elevation probability density function and possibly also about the ocean wave spectrum. The amplitude parameter may be used to revise the altimeter-estimated surface backscattering cross-section, and the attitude angle also leads to a correction to the backscatter. The noise baseline parameter is of relatively little direct interest but must be included as one of the fitted waveform parameters because the waveform samples measure radar signal <u>plus</u> noise.

REFERENCE: Hayne, G. S., "Radar Altimeter Waveform Modeled Parameter Recovery," NASA TM-73294, April 1981

CONTROL: A.A-2.3 WAVFRM waveform module

SUPPORT: A.A-2.3.1.1 FILLD derivative submodule
A.A-2.3.1.2 SYMINV matrix inversion submodule
A.A-2.3.1.1.1 FILLV convolution submodule

ACCESS: CALL SWHFIT (NLO,NUP,WTY)

INPUTS:
NLO = first waveform sample used in the fit (≥ 1)
NUP = last waveform sample used in the fit (≤ 63)
WTY = input waveform sample weights
WLEV2(21) = mean compressed SWH
WLEV2(101-163) = mean compressed waveform samples

OUTPUTS:
WLEV2(227) = rss of fit
WLEV2(228) = kurtosis estimate
WLEV2(229) = waveform processor convergence flag
1 for normal convergence
-1 for increasing sum of squared errors
-2 for matrix inversion failure
-10 for waveform sample value outside acceptable limits

2 for amplitude estimate outside acceptable limits
3 for time origin estimate outside acceptable limits
4 for risetime estimate outside acceptable limits
5 for baseline estimate outside acceptable limits
6 for skewness estimate outside acceptable limits
7 for attitude estimate outside acceptable limits
8 for kurtosis estimate outside acceptable limits

WLEV2(230) = significant waveheight estimate
WLEV2(231) = attitude estimate
WLEV2(232) = skewness estimate
WLEV2(233) = amplitude estimate
WLEV2(234) = altitude correction from waveform processor
WLEV2(235) = baseline estimate

TABLES: Table 2.3.1. Time Location and Indexing for the 63 SEASAT Waveform Samples

ALGORITHM: 1. Set initial values, limits

Lower and upper waveform sample numbers (63 is the number of SEASAT waveform samples)

```
IF (NLO .LT. 1) NLO = 1
IF (NUP .GT. 63) NUP = 63
```

Fractional change limit for satisfactory convergence

```
ERLIM = 0.001
```

Maximum iteration limit

```
LIMIT = 30
```

Iteration counter

```
ITER = 0
```

Set GUESS(3) to real-time SWH estimate

```
GUESS(3) = WLEV2(21) / 0.6
```

Minimum sum of the squared errors

```
SQMIN = 1.0E+06
```

Working sum of the squared errors

```
SERSQ = SQMIN
```

Output parameters

```
IER = -10
WLEV2(227) = -8888.0
WLEV2(228) = -8888.0
WLEV2(229) = IER
WLEV2(230) = -8888.0
WLEV2(231) = -8888.0
```

```
      WLEV2(232) = -8888.0
      WLEV2(233) = -8888.0
      WLEV2(234) = -8888.0
      WLEV2(235) = -8888.0
```

2. Check that input waveform sample values (YIN) are within allowed limits (return with no further SWH work if not within limits), and sum input weighting factors for normalization

```
      WTI = 0.0
      DO 50 I = NLO,NUP
      WTI = WTI + WTY(I)
      YI = YIN(I)
```

Expected nominal range of SEASAT waveform samples

```
      IF (YI.LT.-25.0 .OR. YI.GT.500.0) GO TO 20
   50 CONTINUE
      DO 100 I = NLO,NUP
  100 WT(I) = WTY(I) / WTI
```

3. Set initial fit parameter estimates equal to GUESS(7) which provides initial estimates of parameters not being fitted as well as those being fitted. GUESS(7) has been previously defined in COMMON /SSM4N/.

```
      DO 105 I = 1,7
  105 A(I) = GUESS(I)
```

4. Set the inverse-variance constraints to be added to the on-diagonal terms below at Step #8. CNSTR(7), which are the a priori estimates of the fit parameter standard deviations, limit the step size of the parameter changes in the fitting iteration and have been previously defined in COMMON /SSM4N/. Test to avoid and input standard deviation estimates less than 0.033. JORDR is the fitting order of the fit parameters. AKEEP always contains the best fit parameters before the current iteration. NA is the number of parameters used in the estimation process. (JORDR, AKEEP, and NA have all been previously defined in COMMON /SSM4N/.)

```
      IJK = 0
      DO 107 II = 1,NA
      I = JORDR(II)
      YI = CNSTR(I)**2
      IF (YI .LT. 0.001) YI = 0.001
      CNSTI(I) = 1.0 / YI
  107 AKEEP(I) = A(I)
```

5. FILLV gets the 63 modeled waveform samples from the FFT-convolution program. These are in ascending order in the independent variable

   ```
    60 CALL FILLV (VAL)
       IER = 1
   ```

   Compute residuals by subtracting the modeled waveform from the actual waveform. Then compute the sum of the weighted squares of the residuals.

   ```
       EOLD = 0.0
       DO 109 I = NLO,NUP
       YI = YIN(I) - VAL(I)
   109 EOLD = EOLD + YI*YI*WT(I)
       SQMIN = EOLD
   ```

6. Iteration restart (each iteration is restarted at this point). Zero upper part, symmetric matrix

   ```
   110 CONTINUE
       DO 200 I = 1,NA
       DO 150 J = 1,I
   150 XMAT(J,I) = 0.0
   200 BCOLM(I) = 0.0
   ```

   Limit the size of the correction in the parameter space by use of a fractional multiplier on the first three iterations. This method of preventing divergence may not be acceptable in the context of realtime processing.

   ```
       XFRCT = 1.0
       IF (ITER .LT. 4) XFRCT = (1.0+FLOAT(ITER)) / 5.0
       ELIM = ERLIM * XFRCT
       SERSQ = 0.0
   ```

7. FILLD sets up the (63,7) derivative array (DRV) by making steps in the values of the parameters A(7); the step sizes taken are carried by STPRM(7), and the order of the derivatives in DRV(63,7) is set by JORDR. The order of STPRM is the same as A. FILLV must have been called before FILLD; the held value in VAL(63) from the call to FILLV is used in filling DRV(63,7). Incidently, the numerical derivative is not taken in case of the amplitude and baseline, so the values set in STPRM(1) and STPRM(4) are irrelevant

   ```
       CALL FILLD (VAL,DRV,STPRM)
   ```

   "DO 300" loop fills upper half of the symmetric matrix, also the column vector (see Hayne, general discussion)

   ```
       DO 300 JP = NLO,NUP
   ```

```
    WTI = WT(JP)
    DY = YIN(JP) - VAL(JP)
    DO 250 JA = 1,NA
    PVECT(JA) = DRV(JP,JA)
250 CONTINUE
    DO 300 I = 1,NA
    YI = PVECT(I) * WTI
    DO 275 J = 1,I
275 XMAT(J,I) = XMAT(J,I) + YI * PVECT(J)
300 BCOLM(I) = BCOLM(I) + YI * DY
```

8. Add on-diagonal constraint elements to the symmetric matrix; these are from the a priori information on variation expected in the parameters to be fitted. The constraint is the inverse of the input variance estimate, minimum allowed variance of 0.001

```
    DO 520 I = 1,NA
    JJ = JORDR(I)
520 XMAT(I,I) = XMAT(I,I) + CNSTI(JJ)
```

9. Symmetric matrix inversion (upper triangular portion only computed)

```
    CALL SYMINV (XMAT,NA,IFAIL,7,PSYM,QSYM,MSYM)
    IF (IFAIL .NE. 0) GO TO 1000
    DO 660 I = 1,NA
    II = JORDR(I)
```

Compute correction in the parameter space before fractional multiplier

```
    ACLM = 0.0
    DO 620 J = 1,NA
    IF (J .LT. I) GO TO 600
    ACLM = ACLM + XMAT(I,J)*BCOLM(J)
    GO TO 620
600 ACLM = ACLM + XMAT(J,I)*BCOLM(J)
620 CONTINUE
```

The II = 6 parameter is pointing angle; following treatment is ad hoc and specific to SEASAT-1 case

```
    IF (II .NE. 6) GO TO 640
    IF (ABS(A(6)) .LT. 0.025) ACLM = ACLM / 5.0
    A(6) = A(6) + XFRCT*ACLM
    IF (A(6).LT.-4.0 .OR. A(6).GT.4.0) A(6) = A(6) / 10.0
    GO TO 660
640 A(II) = A(II) + XFRCT*ACLM
660 CONTINUE
```

The following avoids negative risetime

```
      IF (A(3) .LT. 1.0E-06) A(3) = 1.0E-06
```

10. Recalculate values of the sampled waveform function for the new, updated estimates of the parameters A(7) and recalculate the sum of the weighted squared errors

```
  700 CALL FILLV (VAL)
      SERSQ = 0.0
      DO 775 I = NLO,NUP
      YI = YIN(I) - VAL(I)
  775 SERSQ = SERSQ + YI*YI*WT(I)
      ITER = ITER + 1
```

11. Check that we keep coefficients producing minimum sum squared errors but do not retain solution if sum was not smaller

```
      IF (SERSQ .GE. SQMIN) GO TO 785
      DO 780 I = 1,NA
      DO 779 J = 1,I
  779 XKEEP(J,I) = XMAT(J,I)
      II = JORDR(I)
  780 AKEEP(II) = A(II)
      SQMIN = SERSQ
  785 CONTINUE
```

12. Check to see if more iterations are required. Do not try for absurdly small residuals about fit (4.4E-05 is an empirically derived SEASAT constant)

```
      IF (SERSQ .LT. 4.4E-05) GO TO 3000
```

Do not exceed iteration limit

```
      IF (ITER .GE. LIMIT) GO TO 2500
```

Do not allow errors to increase

```
      IF (SERSQ .GT. EOLD) GO TO 800
```

Check for fractional error convergence

```
      IF ((EOLD-SERSQ)/EOLD .LE. ELIM) GO TO 3000
```

Best solution has not been obtained, try again

```
  790 EOLD = SERSQ
      GO TO 110
```

13. Check the fractional change in the sum of the squared errors; value less than 0.0099 is an individual waveform sample standard deviation of about 0.015, which is assumed to be an adequate lower limit to the SEASAT situation when using only last 45 waveform samples. For the SEASAT waveform case, check if the sum of the squared errors

has increased; don't make error exit if (fractional) increase is less than 10 times limits.

```
  800 IF (SERSQ .LE. 0.0099) GO TO 3000
      IF (ABS((SERSQ-EOLD)/EOLD) .LE. 10.0*ERLIM) GO TO 190
```

14. Sum errors **2 increased

```
      IER = -1
      GO TO 2500
```

15. Matrix inversion failure (singular matrix)

```
 1000 IER = -2
      GO TO 2500
```

16. Iteration count exceeded. From SEASAT experience, if ITER > 2, figure that some sort of solution exists, so set IER = 1 and retrieve the minimum value producing set of A(.) and the resulting XMAT(.,.) values if necessary

```
 2500 IF (ITER .LE. 2) GO TO 4500
      IER = 1
      IF (SERSQ .LT. SQMIN) GO TO 3000
      DO 2510 I = 1,NA
      DO 2505 J = 1,I
 2505 XMAT(J,I) = XKEEP(J,I)
      II = JORDR(I)
 2510 A(II) = AKEEP(II)
      SERSQ = SQMIN
 3000 CONTINUE
```

17. Use values from XMAT (at last iteration and after inversion) to find correlations which will then be set into array CORRL(21) in order: 2,1 3,1 3,2 4,1 4,2 4,3 etc. Note that first the square roots of diagonal elements will be taken, for convenience. Also note that order in this correlation array is in terms of the order in which the parameters were fitted, not the order in which they are in A(.)

```
      DO 3001 I = 1,NA
 3001 CORRL(I) = 0.0
      IJ = 0
      DO 3005 J = 2,NA
      JM = J - 1
      DO 3005 I = 1,JM
      IJ = IJ + 1
      CORRL(IJ) = XMAT(I,J) / (XMAT(I,I)*XMAT(J,J))
 3005 CONTINUE
```

The following statement is reached when linefit converged, produced parameter estimates

```
 3010 CONTINUE
```

18. Check linefit parameters against edit limits, signal by IER > 1

```
      DO 3012 II = 1,NA
      I = JORDR(II)
      YI = Y(I)
 3012 IF (YI.LT.AEDIT(1,I) .OR. YI.GT.AEDIT(2,I)) GO TO 4000
      GO TO 4500
 4000 IER = 1 + I
```

19. Store final output estimates in COMMON /GEOPHY/

```
 4500 WLEV2(227) = SQRT(SERSQ/(NUP-NLO)
      WLEV2(228) = A(7)
      WLEV2(229) = IER
      WLEV2(230) = A(3) * 0.6
      WLEV2(231) = A(6)
      WLEV2(232) = A(5)
      WLEV2(233) = A(1)
      WLEV2(234) = A(2) * 0.149896
      WLEV2(235) = A(4)
```

20. End of algorithm

```
      RETURN
```

COMMENTS: 1. The following TYPE statements must be included in the code of this submodule:

```
      DIMENSION YIN(63),WTY(63),XMAT(7,7)BCOLM(7),PVECT(7)
      DIMENSION WT(63),PSYM(7),QSYM(7),MSYM(7),CNSTI(7)
      DIMENSION AKEEP(7),XKEEP(7,7),VAL(63),DRV(63,7),STPRM(7)
      DATA STPRM /1.0,0.25,0.2,1.0,0.05,0.005,0.05/
      COMMON /SYSTM/ SYS(514),NSYS,NSCTR,SMSYS
      COMMON /SSM4N/ A(7),XCNST(7),NA,ITER,SERSQ,
     $     CORRL(21),GUESS(7),CNSTR(7),JORDR(7),AEDIT(2,7)
      COMMON /GEOPHY/ WLEV2(235)
      EQUIVALENCE (WLEV2(101),YIN(1))
```

TITLE: SUBROUTINE FILLD

FUNCTION: To fill an array DRV(63,7) containing 63 sample values of up to 7 derivatives. The order of the derivative terms in DRV is set by JORDR(7). FILLD is a companion to FILLV (A.A-2.3.1.1); it requires that FILLV has been called already.

REFERENCE: Hayne, G. S., "Radar Altimeter Waveform Modeled Parameter Recovery," NASA TM-73294, April 1981

CONTROL: A.A-2.3.1 SWHFIT waveform processor driver submodule

SUPPORT: A.A-2.3.1.1.1 FILLV convolution submodule

ACCESS: CALL FILLD (VAL,DRV,STPRM)

INPUTS:
VAL = modeled waveform
STPRM = step size of the fit parameters

OUTPUTS:
DRV = partial derivatives of the modeled waveform with respect to the fit parameters

TABLES: None

ALGORITHM:

```
1.  Loop to 400 for the NA derivates needed
          DO 400 K = 1,NA
          J = JORDR(K)
          GO TO (2,4,4,3,4,4,4), J
2.  Amplitude derivative
          TMP = AMPLI
          IF (TMP .LT. 1.0E-05) TMP = 1.0E-05
          DO 110 I = 1,63
      110 DRV(I,K) = (VAL(I)-BSLIN) / TMP
          GO TO 5
3.  Baseline derivative
          DO 210 I = 1,63
      210 DRV(I,K) = 1.0
          GO TO 5
4.  Numerical derivative for all except amplitude and baseline
          ATMP = A(J)
          STEP = STPRM(J)
          A(J) = A(J) + STEP
    Compute new estimates for the Jth parameter
          CALL FILLV (TMP)
          DO 320 I = 1,63
```

```
320 DRV(I,K) = (TMP(I)-VAL(I)) / STEP
    A(J) = ATMP
```

5. End of loop

```
400 CONTINUE
```

6. End of algorithm

```
    RETURN
```

COMMENTS: 1. The following TYPE statements must be included in the code of this submodule:

```
DIMENSION VAL(63),DRV(63,7),STPRM(7),TMP(63)
COMMON /SSM4N/ A(7),XCNST(7),NA,ITER,SERSQ
$ CORRL(21),GUESS(7),CNSTR(7),JORDR(7),AEDIT(2,7)
EQUIVALENCE (AMPLI,A(1)) , (BSLIN,A(4))
```

CONVOLUTION SUBMODULE A.A-2.3.1.1.1

TITLE: SUBROUTINE FILLV

FUNCTION: To evaluate waveforms using FFT techniques to perform convolution of: 1) system point target response, 2) sea surface elevation distribution, and 3) flat sea response. This submodule is set up for a 512-point transform and uses FFT submodules FFA and FFS and their associated submodules.

REFERENCE: Hayne, G. S., "Radar Altimeter Waveform Modeled Parameter Recovery," NASA TM-73294, April 1981

CONTROL:	A.A-2.3.1	SWHFIT	waveform processor driver submodule
	A.A-2.3.1.1	FILLD	derivative submodule
SUPPORT:	A.A-2.3.1.1.1.1	GTFSR	flat sea response submodule
	A.A-2.3.1.1.1.2	GTSEA	surface elevation distribution submodule
	A.A-2.3.1.1.1.3	FFA	fast Fourier transform submodule
	A.A-2.3.1.1.1.4	FFS	fast Fourier synthesizing submodule

ACCESS: CALL FILLV (VAL)

INPUTS: None

OUTPUTS: VAL = modeled waveform

TABLES: None

ALGORITHM: 1. Initialization

```
IF (I1ST .NE. 0) GO TO 3
I1ST = 1
```

2. Transform the system impulse response, SYS, which has been previously defined in COMMON /SYSTEM/

```
CALL FFA (SYS,NNP)
```

3. Set up the sea surface elevation distribution and perform transform, replacing the input distribution

```
      CALL GTSEA (NNP,SEA,SMSEA)
      CALL FFA (SEA,NNP)
```

4. Set up flat sea response and perform transform, replacing the input distribution

```
      CALL GTFSR (NNP,FSR,SMFSR)
      CALL FFA (FSR,NNP)
```

5. Form amplitude normalization ANORM, then set up phase multiplier delta factor DPHI. XNCTR is included as a possible time bias to be defined at a later date.

```
      XNCTR = 0.0
      PHI = 0.0
```

   The constant 1.5625 is the SEASAT 1/2 waveform sample interval

```
      DPHI = -( (XNCTR-24.5-FLOAT(NSCTR)) + TIMO/1.5625)*
     $        6.2831853/FLOAT(NNP)
```

   SMSYS is the sum of the system impulse response and has been set previously in COMMON /SYSTEM/

```
      ANORM = 1.0 / (SMSYS*SMSEA)
      CTRES(1) = CMPLX(ANORM,0.0) * CTSYS(1) * CTSEA(1) * CTFSR(1)
      DO 500 I = 2,NC2
      PHI = PHI + DPHI
      CPHAS = CMPLX(ANORM*COS(PHI),ANORM*SIN(PHI))
  500 CTRES(I) = CPHAS * CTSYS(I) * CTSEA(I) * CTFSR(I)
```

6. Inverse transform to get final convolution result (RES and CTRES are equivalenced)

```
      CALL FFS (RES,NNP)
      IF (ABS(BSLIN) .LE. AMPLI*0.1E-05) GO TO 7
      DO 530 I = 1,NNP
```

   Add the baseline back in

```
  530 RES(I) = RES(I) + BSLIN
```

7. Transfer data to final output array (512 array is at SEASAT 1/2 waveform sample spacing; therefore pick every other one). Output waveform sample #1 matches resulting waveform estimate #1 by the selection of DPHI.

```
      DO 610 I = 1,29
      J = I + I
      VAL(I+34) = RES(J+61)
  610 VAL(I) = RES(J-1)
```

```
      DO 620 I = 30,34
  620 VAL(I) = RES(I+28)
```

8. End of algorithm

```
      RETURN
```

COMMENTS: 1. The following TYPE statements must be included in the code of this submodule:

```
      DIMENSION VAL(63),SEA(514),FSR(514),RES(514)
      COMPLEX CTSYS(257),CTSEA(257),CTFSR(257),CTRES(257),CPHAS
      COMMON /SYSTM/ SYS(514),NSYS,NSCTR,SMSYS
      COMMON /SSM4N/ A(7),XCNST(7),NA,ITER,SERSQ,
     $  CORRL(21),GUESS(7),CNSTR(7),JORDR(7),AEDIT(2,7)
      DATA I1ST/0/ , NNP/512/ , NP2/514/ , NC2/257/
      EQUIVALENCE (SYS(1),CTSYS(1)) , SEA(1),CTSEA(1))
      EQUIVALENCE (FSR(1),CTFSR(1)) , (RES(1),CTRES(1))
      EQUIVALENCE (A(1),AMPLI) , (A(2),TIMO) , (A(3),SIGMA)
      EQUIVALENCE (A(4),BSLIN) , (A(5),XLMDA) , (A(6),XIDEG)
      EQUIVALENCE (A(7),XKURT)
```

FLAT SEA RESPONSE SUBMODULE A.A-2.3.1.1.1.1

TITLE: SUBROUTINE GTFSR

FUNCTION: To determine the flat sea response using I_0 term only from Brown's expansion (see references). A power series from Abramowitz and Stegun is used to evaluate I_0. This version of GTFSR uses 230 non-zero values of the flat sea response and assumes that NNP>231.

REFERENCES: Hayne, G. S., "Radar Altimeter Waveform Modeled Parameter Recovery," NASA TM-73294, April 1981.

Brown, G. S., "The Average Impulse Response of a Rough Surface and Its Applications," IEEE Trans. Antennas and Propagation, Vol. AP-25, No. 1, January 1977.

Abramowitz, A. and Stegun, I. A., NBS Handbook of Mathematical Functions, Dover Books, 1972.

CONTROL: A.A-2.3.1.1.1 FILLV convolution submodule

SUPPORT: None

ACCESS: CALL GTFSR (NNP, FSR, SMFSR)

INPUTS: NNP = number of points in the flat sea response array (at least 2 less than the dimension of FSR)

OUTPUTS: FSR = flat sea response array

SMFSR = flat sea response normalization sum

TABLES: None

ALGORITHM:

1. Initialization (NSPR = 320 is SEASAT specific and related to using a 512 point transform).

```
      NSFR = 230
      SMFSR = 0.0
```

SEASAT 1/2 waveform sample interval

```
      DT = 1.5625
      T = -DT/2.0
```

2. Test for (impossible) negative angle; if present choose branch which effectively increases the DLTA at < zero degrees pointing

```
      IF (XIDEG .GT. 0.0) GO TO 3
```

Brown's equation do not allow for negative angle. Therefore approximate extrapolation by exponental. The following statement causes DLTA to increase by a factor of two for one degree (fictitious) negative angle.

```
      DLTA = 2.66496E-03 * (1.0-XIDEG)
      DO 15 J = 1,NFSR
      T = T + DT
      Z = AMPLI * EXP(-DLTA*T)
      FSR(J) = Z
   15 SMFSR = SMFSR + Z
      GO TO 4
```

3. Fill in surface response by proper power series for $I_0(Z)$. The constants used in the calculation of BETA and DLTA (see Hayne, equations 19, 20, and 22) are SEASAT related and dependent upon beam width and altitude.

```
      X2RAD = XIDEG / 28.64789
      BETA = 4.35331 * SIN(X2RAD)
      DLTA = 2.66496E-03 * COS(X2RAD)
      DO 30 J = 1,NFSR
      T = T + DT
      Z = BETA * SQRT(T)
```

Select which of two series to use for $I_0(Z)$ from Abramowitz and Stegun

```
      IF (Z .GT. 3.75) GO TO 23
      Z = Z * Z / 14.0625
      A = 1.0 + Z*(3.515623+Z*(3.089942+Z*(1.206749
```

```
    $ +Z*(0.2659732+Z*(0.0360768+Z*0.0045813)))))
      GO TO 27
   23 A = EXP(Z) / SQRT(Z)
      Z = 3.75 / Z
      A = Z*(0.3989423-Z*(0.03988024+Z*(0.00362018
    $ -Z*(0.00163801-Z*(0.01031555-Z*(0.02282967-Z*(0.02895312
    $ -Z*(0.01787654-Z*(0.00420059))))))))
   27 Z = AMPLI * EXP(-DLTA*T) * A
      FSR(J) = Z
   30 SMSFR = SMSFR + Z
```

4. Fill rest of the array with zeroes

```
      K = NFSR + 1
      NP2 = NNP + 2
      DO 40 J = K,NP2
   40 FSR(J) = 0.0
      I1ST = 1
```

5. End of algorithm

```
      RETURN
```

COMMENTS: 1. The following TYPE statements must be included in the code of this submodule:

```
      DIMENSION FSR(2)
      COMMON /SSM4N/ A(7),XCNST(7),NA,ITER,SERSQ,
    $  CORRL(21),GUESS(7),CNSTR(7),JORDR(7),AEDIT(2,7)
      EQUIVALENCE (A(1),AMPLI) , (A(2),TIMDO) , (A(3),SIGMA)
      EQUIVALENCE (A(4),BSLIN) , (A(5),XLMDA) , (A(6),XIDEG)
      EQUIVALENCE (A(7),XKURT)
```

SURFACE ELEVATION DISTRIBUTION SUBMODULE A.A-2.3.1.1.1.2

TITLE: SUBROUTINE GTSEA

FUNCTION: To fill array with a skewed Gaussian surface elevation distribution centered on the sample number 86. Zeroes entered in all other elements than in interval 1 - 171. Assumes NNP > 171. It is intended for use in the 512-point FFT processes.

REFERENCE: Hayne, G. S., "Radar Altimeter Waveform Modeled Parameter Recovery," NASA TM-73294, April 1981.

CONTROL: A.A-2.3.1.1.1 FILLV convolution submodule

SUPPORT: None

ACCESS: CALL GTSEA (NNP,SEA,SMSEA)

INPUTS: NNP = number of points in the surface elevation distribution array (at least 2 less than the dimension of SEA)

OUTPUTS: SEA = surface elevation distribution array

SMSEA = surface elevation distribution normalization sum

TABLES: None

ALGORITHM:

1. Convert sea sigma to # gate intervals (zero width is not allowed). The constant 1.5625 is the SEASAT 1/2 waveform sample interval.

```
      XNGTS = SIGMA / 1.5625
      IF (XNGTS .LT. 0.001) XNGTS = 0.001
      WGTS = 0.0
```

2. Establish center at 86th gate

```
      NCTR = 86
      SEA(NCTR) = 1.0
      SMSEA = 1.0
      K = NCTR - 1
      X6 = XLMDA / 6.0
      XNDX = 0.0
```

3. Fill non-zero elements of the array

```
      DO 20 J = 1,K
      XNDX = XNDX + 1.0
      WGTS = XNDX / XNGTS
      Z = -WGTS * WGTS / 2.0
      IF (Z .LT. -80.0) GO TO 10
      A = EXP(Z)
      Z = X6 * WGTS * (WGTS*WGTS-3.0)
      A1 = A * (1.0-Z)
      A2 = A * (1.0+Z)
      GO TO 15
   10 A1 = 0.0
      A2 = 0.0
   15 SEA(NCTR-J) = A1
      SEA(NCTR+J) = A2
   20 SMSEA = SMSEA + A1 + A2
```

4. Fill zero elements of the array

```
      K = 2 * NCTR
      NP2 = NNP + 2
      DO 25 J = K,NP2
   25 SEA(J) = 0.0
```

```
5. End of algorithm
      RETURN
```

COMMENTS: 1. The following TYPE statements must be included in the code of this submodule:

```
      DIMENSION SEA(2)
      COMMON /SSMAN/ A(7),XCNST(7),NA,ITER,SERSQ,CORRL(21)
     $  GUESS(7),CNSTR(7),JORDR(7),AEDIT(2,7)
      EQUIVALENCE (A(1),AMPLI) , (A(2),TIMO) , (A(3),SIGMA)
      EQUIVALENCE (A(4),BSLIN) , (A(5),XLMDA) , (A(6),XIDEG)
      EQUIVALENCE (A(7),XKURT)
```

FAST FOURIER TRANSFORM SUBMODULE A.A-2.3.1.1.1.3

TITLE: SUBROUTINE FFA

FUNCTION: To replace the real vector B(K), (K=1,2,...,N), with its finite discrete Fourier transform. The DC term is returned in location B(1) with B(2) set to 0. Thereafter, the J-TH harmonic is returned as a complex number stored as B(2*J+1) + i B(2*J+2). Note that the N/2 harmonic is returned in B(N+1) with B(N+2) set to 0. Hence, B must be dimensioned to size N+2. The subroutine is called as FFA (B,N) where N = 2M and B is an N term real array (for SEASAT, M = 9). A real-valued, radix 8 algorithm is used with in-place reordering and the trig functions are computed as needed.

REFERENCE: "Programs for Digital Signal Processing," ed. by Digital Signal Processing Committee of the IEEE ASSP, Published by IEEE Press, November 1979.

CONTROL: A.A-2.3.1.1.1 FILLV convolution submodule

SUPPORT: FFA-FFS package (see COMMENTS)

ACCESS: Call FFA (B,NFFT)

INPUTS:
B = the real vector to be transformed
NFFT = the dimension of the B vector

OUTPUTS: B = the finite discrete Fourier transform of the input vector

TABLES: None

ALGORITHM: See above reference

COMMENTS: 1. The submodule is one of the set of routines for fast Fourier transform of real data sequences as described in the above reference. These routines are collectively the "FFA-FFS package" which includes: FFA, FFS, R2TR, R4TR, R8TR, R4SYN, R8SYN, ORD1 and ORD2.

FAST FOURIER SYNTHESIZING SUBMODULE A.A-2.3.1.1.1.4

TITLE: SUBROUTINE FFS

FUNCTION: To synthesize the real vector B(K), where K=1,2,...,N. The initial Fourier coefficients are placed in the B array of size N+2. The DC term is in B(1) with B(2) equal to 0. The J-TH harmonic is stored as B(2*J+1) + i B(2*J+2). The N/2 harmonic is in B(N+1) with B(N+2) equal to 0. The subroutine is called as FFS (B,N) where N = 2M (for SEASAT, M = 9) and B is the N term real array discussed above.

REFERENCE: "Programs for Digital Signal Processing," ed. by Digital Signal Processing Committee of the IEEE ASSP, published by IEEE press, November 1979.

CONTROL: A.A-2.3.1.1.1 FILLV convolution submodule

SUPPORT: FFA-FFS package (see COMMENTS)

ACCESS: CALL FFS (B,NFFT)

INPUTS:
B = the Fourier coefficients
NFFT = the dimension of B vector

OUTPUT: B = the real vector

TABLES: None

ALGORITHM: See above reference

COMMENTS:

1. The submodule is one of the set of routines for fast Fourier transform of real data sequences as described in the above reference. These routines are collectively the "FFA-FFS package" which includes: FFA, FFS, R2TR, R4TR, R8TR, R4SYN, R8SYN, ORD1, and ORD2.

MATRIX INVERSION SUBMODULE A.A-2.3.1.2

TITLE: SUBROUTINE SYMINV

FUNCTION: To compute the inverse of a symmetric matrix

REFERENCE:

CONTROL: A.A-2.3.1 SWHFIT waveform processor driver submodule

SUPPORT: None

ACCESS: CALL SYMINV (A,N,IFAIL,NROW,P,Q,M)

INPUTS:
A = symmetric matrix to be inverted
N = order of the matrix to be inverted
NROW = row dimension of A
P = dummy vector of dimension N

Q = dummy vector of dimension N

M = dummy vector of dimension N

OUTPUTS: A = inverted matrix

IFAIL = 0 for successful inversion

= 1 for unsuccessful inversion

TABLES: None

ALGORITHM:

```
1.  Initialization
          IFAIL = 0
          DO 10 I = 1,N
       10 M(I) = 1
2.  Search for pivot
          DO 140 I = 1,N
          BIG = 0.0
          DO 40 J = 1,N
          TEST = ABS(A(J,J))
          IF (TEST-BIG) 40,40,20
       20 IF (M(J)) 150,40,30
       30 BIG = TEST
          K = J
       40 CONTINUE
3.  Preparation for elimination step
          M(K) = 0
          Q(K) = 1.0 / A(K,K)
          P(K) = 1.0
          A(K,K) = 0.0
          KP1 = K + 1
          KM1 = K - 1
          IF (KM1) 150,80,50
       50 DO 70 J = 1,KM1
          P(J) = A(J,K)
          Q(J) = A(J,K) * Q(K)
          IF (M(J)) 150,70,60
       60 Q(J) = -Q(J)
       70 A(J,K) = 0.0
       80 IF(K-N) 90,130,150
       90 DO 120 J = KP1,N
          P(J) = A(K,J)
          IF (M(J)) 150,100,110
      100 P(J) = -P(J)
```

```
       110 Q(J) = -A(K,J) * Q(K)
       120 A(K,J) = 0.0
4. Elimination proper
       130 DO 140 J = 1,N
           DO 140 K = J,N
       140 A(J,K) = A(J,K) + P(J)*Q(K)
           GO TO 6
5. Error exit
       150 IFAIL = 1
6. End of algorithm
           RETURN
```

COMMENTS:

1. The following TYPE statement must be included in the code of this submodule:

```
DIMENSION A(NROW,1),P(1),Q(1),M(1)
```

2. This submodule uses only the upper triangular portion of A as input and returns only the upper triangular portion of the matrix inverse.

WAVEFORM ALTITUDE CORRECTION A.A-2.3.2

TITLE: SUBROUTINE WAVALT

FUNCTION: To compute the altitude correction for attitude, SWH, and $\ddot{h}$ when the waveform processor fails to converge. This module is used only as a backup to the waveform processor and is not called when convergence is achieved by the waveform processor. The inputs to the module are the spacecraft estimate of attitude and the real-time SWH and the software estimate of $\ddot{h}$. The output is the altitude correction for attitude, SWH, and $\ddot{h}$.

REFERENCE: "Seasat Algorithm Development Facility Altimeter Sensor Algorithm Specification," Jet Propulsion Laboratory, PD 622-202, Revision A, March 1980.

CONTROL: A.A-2.3 WAVFRM waveform module

SUPPORT: None

ACCESS: CALL WAVALT

INPUTS:
WLEV2(12) = attitude in (deg,0.01)
WLEV2(21) = significant wave height in (m,0.01)
WLEV2(236) = height acceleration in (m/s^2,0.01)
WLEV2(229) = waveform processor flag
ISTAT(9) = track-mode flag

OUTPUTS:
WLEV2(238) = height correction for attitude, SWH, and $\ddot{h}$ in (m,0.001)
WLEV2(240) = standard deviation of $\delta h_{2,6}$ in (m,0.001)

TABLES: A table of sea states and attitudes.

ALGORITHM: Whenever the waveform processor does not converge:

(1) Compute the altitude correction for attitude and SWH by table lookup.

(2) Compute altitude correction for $\ddot{h}$.

(3) Output the total correction and its standard deviation.

COMMENTS: 1. This is only a functional description. At a later date it will be defined or deleted if not needed to backup the waveform parameter recovery.

WAVEFORM SWH CORRECTION A.A-2.3.3

TITLE: SUBROUTINE SWHCOR

FUNCTION: To compute the SWH correction for attitude and SWH when the waveform processor fails to converge. This module is used only as a backup to the waveform processor and is not called when convergence is achieved by the waveform processor. The inputs to the module are the spacecraft estimate of attitude and the real-time SWH. The output is the SWH correction for attitude and SWH.

REFERENCE: "SEASAT Algorithm Development Facility Altimeter Sensor Algorithm Specifications," Jet Propulsion Laboratory, PD 622-202, Revision A.

CONTROL: A.A-2.3 WAVFRM waveform module

SUPPORT: None

ACCESS: CALL SWHCOR

INPUTS:
WLEV2(12) = attitude in (deg,0.01)
WLEV2(21) = significant wave height in (m,0.01)
WLEV2(229) = waveform processor convergence flag

OUTPUTS: WLEV2(237) = SWH correction for attitude and SWH in (m,0.001)

TABLES: A table consisting of two entries for each of 16 attitudes and 20 sea states (640 entries).

ALGORITHM: If the waveform processor did not converge, this module will compute the SWH correction for attitude and SWH by table lookup.

COMMENTS: 1. This is only a functional description. At a later date this module will be defined or deleted if not needed as a backup.

WAVEFORM SWH CAL ZONE BIAS A.A-2.3.4

TITLE: SUBROUTINE SWHBIAS

FUNCTION: To calculate the SWH cal zone bias. The correction will be based on comparisons of the calculated SWH with ground-truth measurements from buoys and aircraft and will therefore be computed after launch. The input to the module will be Julian date and the output will be the SWH cal bias.

REFERENCE: "SEASAT Algorithm Development Facility Altimeter Sensor Algorithm Specifications," Jet Propulsion Laboratory, PD-622-202, Revision A.

CONTROL: A.A-2.3 WAVFRM waveform module

SUPPORT: None

ACCESS: CALL SWHBIAS

INPUTS:
WLEV2(15) = classification flag
WLEV2(5) = time as Julian date in (days, 1 x10^{-8})
WLEV2(21) = significant waveheight (m,0.01)

OUTPUTS: WLEV2(241) = SWH cal zone bias in (m,0.001)

TABLES: A table of ΔSWH as a function of SWH (not presently available). There will be a maximum of 20 table entries of SWH and ΔSWH for each of a maximum of five T_{JD} (100 total entries).

ALGORITHM: For data over water a table lookup of SWH will be made. Linear interpolation and extrapolation may be used to calculate the output.

COMMENTS:
1. This is only a functional description.
2. The table will not be available until postlaunch inflight data are compared and analyzed against ground-truth data. This table may be updated during the mission.

WAVEFORM PRODUCTS SUBMODULE A.A-2.3.5

TITLE: SUBROUTINE WFPROD

FUNCTION: To compute the upper ocean dynamics estimates of significant slope and dominant wavelength.

REFERENCE: Huang, N. E., and S. R. Long, "A Study of the Waveheight Probability Distribution and Statistics of Wind Generated Waves," submitted for publication Journal of Fluid Mechanics.

CONTROL: A.A-2.3 WAVFRM waveform module

SUPPORT: None

ACCESS: CALL WFPROD

INPUTS: WLEV2(230) = refined SWH estimate
WLEV2(232) = skewness estimate

OUTPUTS: WLEV2(245) = significant slope
WLEV2(246) = dominant wavelength
WLEV2(247) = dominant frequency
WLEV2(248) = dominant phase speed
WLEV2(249) = dominant wave number

TABLES: None

ALGORITHM:

```
1.  Compute the significant slope
          WLEV2(245)  =  WLEV2(232) / (8.0*PI)
2.  Compute the dominant wavelength
          WLEV2(246)  =  WLEV2(230) / (4.0*WLEV2(245))
3.  Compute the dominant frequency
          WLEV2(247)  =  SQRT(2.0*PI*G/WLEV2(246))
4.  Compute the dominant phase speed
          WLEV2(248)  =  SQRT(G*WLEV2(246) / (2.0*PI))
5.  Compute the dominant wave number
          WLEV2(249)  =  2.0 * PI / WLEV2(246)
6.  End of algorithm
          RETURN
```

COMMENTS: 1. The following TYPE statements must be included in the code of this submodule:

```
COMMON /GEOPHY/ WLEV2(325)
DATA PI/3.14159265358979/
DATA G/9.80621/
```

ATMOSPHERIC CORRECTIONS A.A-2.4

TITLE: SUBROUTINE ATMOS

FUNCTION: To calculate the atmospheric corrections to the spacecraft altitude and the radar backscatter coefficient. The module will also process data from the rain gate. The altitude correction from this module consists of the combined effects of ionospheric refraction, wet and dry tropospheric refraction, and atmospheric pressure. The radar backscatter coefficient correction accounts for the effects of the atmosphere on return power, and the rain-gate processing detects the presence of rain at the subsatellite point. All inputs and outputs are processed by submodules.

REFERENCE: Not applicable

CONTROL: A.A-2.0 ALT2DR level 2 main driver

SUPPORT: A.A-2.4.1 RAIN rain estimate submodule
A.A-2.4.2 BARTRD barotropic effects submodule
A.A-2.4.3 IONO ionospheric refraction
A.A-2.4.4 DRYTRO dry tropospheric refraction
A.A-2.4.5 WETTRO wet tropospheric refraction
A.A-2.4.6 SIGCOR radar backscatter correction

ACCESS: CALL ATMOS

INPUTS: TBD

OUTPUTS: TBD

TABLES: None

ALGORITHM: Each submodule is called in turn:
(1) Rain gate (A.A-2.4.1)
(2) Barotropic effects (A.A-2.4.2)
(3) Ionospheric refraction (A.A-2.4.3)
(4) Dry tropospheric refraction (A.A-2.4.4)
(5) Wet tropospheric refraction (A.A-2.4.5)
(6) Sigma-naught correction (A.A-2.4.6)

COMMENTS: 1. This is only a functional description.

RAIN GATE A.A-2.4.1

TITLE: SUBROUTINE RAIN

FUNCTION: To provide nadir rain-rate estimates. In addition, two flags, indicating the quantity and quality of the rainfall estimate, will be set to facilitate processing of possible altitude rain-rate correction.

REFERENCE: Walsh, E. J., "Altimeter Rain Detection," NASA TM-73791, July 1981.

CONTROL: A.A-2.4 ATMOS atmospheric correction module

SUPPORT: None

ACCESS: CALL RAIN

INPUTS: TBD

OUTPUTS: WLEV2(251) = rain rate in (mm/hr,0.1)
WLEV2(252) = rain-rate quantity flag:
1 for light $(0 < R \le 2)$
2 for mild $(2 < R \le 6)$
3 for medium $(6 < R \le \text{TBD})$
4 for heavy $(\text{TBD} < R)$

WLEV2(253) = rain-rate estimate quality flag:
0 for good
1 for questionable

TABLES: TBD

ALGORITHM: TBD

COMMENTS: 1. This algorithm will be defined at a later date.

BAROTROPIC EFFECTS A.A-2.4.2

TITLE: SUBROUTINE BARTRO

FUNCTION: To compute the altitude correction due to the effects of atmospheric pressure.

REFERENCE: "Seasat Geophysical Algorithm Specification," Jet Propulsion Laboratory, PD622-226, December 1980.

CONTROL: A.A-2.4 ATMOS atmospheric correction module

SUPPORT: None

ACCESS: CALL BARTRO

INPUTS: WLEV2(256) = sea-surface atmospheric pressure (from FNOC) in (mb,0.1)
WLEV2(259) = FNOC data-present flag

OUTPUTS: WLEV2(260) = atmospheric pressure altitude correction in (m,0.001)
WLEV2(261) = standard deviation of corrections in (m,0.001)

TABLES: A table of monthly averages of atmospheric pressure, temperature, and vapor pressure for each 5 degrees of latitude (not presently available). The size of the table will be approximately 3 by 12 by 30.

ALGORITHM: If the surface atmospheric pressure (P) is not available from FNOC, the table will be linearly interpolated for calculating P.

P = WLEV2(256)
WLEV2(260) = -0.009948 (P-1013.3)
WLEV2(261) = 0 (for default)

COMMENTS: 1. The detailed algorithm will be defined at a later date.

IONOSPHERIC REFRACTION A.A-2.4.3

TITLE: SUBROUTINE IONO

FUNCTION: To compute the altitude correction necessitated by the effects of ionospheric refraction.

REFERENCE: Nesturczuk, G., "Ionospheric Propagation Correction Modeling for Satellite Altimeters," NASA CR-156881, Contract NAS6-3075, November 1981.

CONTROL: A.A-2.4. ATMOS atmospheric correction module

SUPPORT: None

ACCESS: CALL IONO

INPUTS:
WLEV2(5) = time expressed as Julian date in $(d, 1\times10^{-8})$
WLEV2(7) = latitude of subsatellite point in (deg,0.001)
WLEV2(8) = longitude of the subsatellite point in (deg,0.001)
Ionospheric data file, A.D-2.82

OUTPUTS:
WLEV2(267) = ionospheric refraction altitude correction in (m,0.001)
WLEV2(268) = standard deviation of $\delta h_{2,2}$ in (m,0.001)

TABLES: TBD

ALGORITHM: TBD

COMMENTS: 1. This algorithm will be defined at a later date.

DRY TROPOSPHERIC REFRACTION A.A-2.4.4

TITLE: SUBROUTINE DRYTRO

FUNCTION: This module computes the altitude correction due to the effects of dry tropospheric refraction.

REFERENCE: "Seasat Geophysical Algorithm Specification," Jet Propulsion Laboratory, PD-622-226, December 1980.

CONTROL: A.A-2.4 ATMOS atmospheric correction module

SUPPORT: None

ACCESS: CALL DRYTRO

INPUTS:
WLEV2(256) = surface atmospheric pressure (from FNOC) in (mb,0.1)
WLEV2(259) = FNOC data-present flag
WLEV2(7) = spacecraft latitude (from ephemeris) in (deg,0.001)

OUTPUTS:
WLEV2(269) = dry tropospheric refraction altitude correction in (m,0.001)
WLEV2(270) = standard deviation of $\delta h_{2,3}$ in (m,0.001)

TABLES: A table of monthly averages of atmospheric pressure, temperature, and vapor pressure for each 5 degrees of latitude (not presently available). The size of the table will be approximately 3 by 12 by 30.

ALGORITHM: If the surface atmospheric pressure (P) is not available from FNOC, a table will be linearly interpolated for calculating P.

$$\mathrm{WLEV2}(269) = P(2.277-0.011\cos\phi)\times10^{-3}$$

$$\mathrm{WLEV2}(270) = 0 \text{ (for default)}$$

COMMENTS: 1. This is only a functional description.

WET TROPOSPHERIC REFRACTION A.A-2.4.5

TITLE: SUBROUTINE WETTRO

FUNCTION: This module computes the altitude correction due to the effects of wet tropospheric refraction. It uses either rain data, LAMMR level 2 data file or FNOC data.

REFERENCE: "Seasat Geophysical Algorithm Specification," Jet Propulsion Laboratory, PD-622-226, December 1980.

CONTROL: A.A-2.4 ATMOS atmospheric correction module

SUPPORT: TBD

ACCESS: CALL WETTRO

INPUTS:
WLEV2(5) = time
WLEV2(7) = spacecraft latitude
WLEV2(251) = rain rate
WLEV2(252) = rain rate quality flag
WLEV2(253) = rain rate quality flag
WLEV2(257) = FNOC sea surface water-vapor pressure (e)
WLEV2(258) = FNOC sea surface atmospheric temperature (T)
WLEV2(259) = FNOC data present flag
LAMMR = level 2 data file A.D(L)-2.83

OUTPUTS:
WLEV2(271) = wet tropospheric refraction altitude correction
WLEV2(272) = standard deviation of WLEV2(271)
WLEV2(273) = LAMMR data quality flag

TABLES: A table of monthly averages of temperature, and vapor pressure for each 5 degrees of latitude (not presently available).

ALGORITHM:

1. If rain data is of good quality then use it to compute WLEV2(271).
2. If not use LAMMR data if available.
3. If LAMMR not available then set e = WLEV2(257) or from the table if FNOC data not present.
4. Assign standard deviation and LAMMR quality flag.

COMMENTS:

1. This is only a functional description.

RADAR BACKSCATTER CORRECTION A.A-2.4.6

TITLE: SUBROUTINE SIGCOR

FUNCTION: To compute the atmospheric correction to sigma naught. The correction will be computed using rain data if it indicates a rate greater than 2 mm/hr, otherwise LAMMR data is used.

REFERENCE: Goldhirsh, J. and E. Walsh, "Precipitation Measurements from Space Using a Modified Seasat Type Radar Altimeter," JHU/APL S1R81U-022, May 1981.
CONTROL: A.A-2.4 ATMOS atmospheric correction module
SUPPORT: TBD
ACCESS: CALL SIGCOR
INPUTS: WLEV2(5) = time expressed as Julian date in $(d, 1\times10^{-8})$
WLEV2(251) = rain rate from A.A-2.4.1 in (mm/hr,0.1)
WLEV2(252) = rain-rate quantity flag from A.A-2.4.1
WLEV2(253) = rain-rate quality flag from A.A-2.4.1
LAMMR level 2 T_B data file (A.D(L)-2.84)
OUTPUTS: WLEV2(274) = sigma-naught atmospheric correction in (dB,0.01)
TABLES: None
ALGORITHM:
1. Compute correction using rain gate data if good quality or else 2.
2. Process according to S.A.(S)-2.4 (Ref. 6).

COMMENTS:
1. This is only a functional specification.

WIND SPEED AND RADAR BACKSCATTER COEFFICIENT DRIVER MODULE A.A-2.5

TITLE: SUBROUTINE WINDSIG
FUNCTION: To act as the driver module for the calculation of the wind speed and the radar backscatter coefficient
REFERENCE: Not applicable
CONTROL: A.A-2.0 ALT2DR level 2 driver module
SUPPORT: A.A-2.5.1 SIGZRO radar backscatter coefficient submodule
A.A-2.5.2 WIND wind speed submodule
ACCESS: CALL WINDSIG
INPUTS: None
OUTPUTS: None
TABLES: None
ALGORITHM:

```
1.  Determine the radar backscatter coefficient
          CALL SIGZRO
2.  Determine the wind speed (19½ meters altitude)
          CALL WIND
3.  End of algorithm
          RETURN
```

TITLE: SUBROUTINE SIGZRO

FUNCTION: To estimate the radar backscatter coefficient, which is an indication of the reflectance properties of the ocean surface.

REFERENCE: "SEASAT Gulf of Alaska Workshop," Vol. 1, JPL 622-101, April 1979

CONTROL: A.A-2.5 WINDSIG wind speed and radar backscatter coefficient module

SUPPORT: None

ACCESS: CALL SIGZRO

INPUTS:

ALT2 = corrected altitude in (m,0.001)
WLEV1(9) = AGC in (dB,0.01)
WLEV2(231) = altimeter off nadir angle in (deg,0.01)
WLEV2(274) = radar backscatter coefficient atmospheric correction in (dB,0.01)
AGCI = array of AGC attenuator values from Table 2.5.1-a
CALK = array of cal attenuator values from Table 2.5.1-a
ATT = array of attitude values from Table 2.5.1-b
AL = array of attitude loss values from Table 2.5.1-b
ALTLO = altitude lower limits in (m,1.0)
ALTHI = altitude higher limits in (m,1.0)

OUTPUTS: WLEV2(276) = radar backscatter coefficient in (dB,0.01)

TABLES:

Table 2.5.1-a Cal Attenuator and AGC (see Appendix B)
Table 2.5.1-b Attitude Loss (see Appendix B)

ALGORITHM:

```
1.  Check input values
        IF (ALT2.LT.ALTLO .OR. ALT2.GT.ALTHI) GO TO 6
        IF (WLEV1(9).LT.0.0 .OR. WLEV1(9).GT.60.58) GO TO 6
        IF (WLEV2(274).LT-10.0 .OR. WLEV2(274).GT.10.0) GO TO 6
        IF (WLEV2(231).LT.0.0 .OR. WLEV2(231).GT.0.75) GO TO 6
2.  Determine cal attenuator value
        DMIN = 9999.0
        DO 100 I = 1,8
        TEST = ABS(AGCI(I)-WLEV1(9))
        IF (TEST .GT. DMIN) GO TO 3
        K = I
    100 DMIN = TEST
3.  Determine the proper attitude loss table entry index
        DO  200 I = 2,16
        K2 = I
```

```
      IF (ATT(I) .GT. WLEV2(231)) GO TO 4
  200 CONTINUE
```

4. Determine the attitude loss

   ```
   K1    = K2 - 1
   FACT  = (AL(K2)-AL(K1)) / (ATT(K2)-ATT(K1))
   ALOSS = AL(K1) + FACT*(WLEV2(231)-ATT(K1))
   ```

5. Determine the radar backscatter coefficient

   ```
    CORR       = 30.0 * ALOG10(SNGL(ALT2)/796440.0)
    WLEV2(276) = 38.33 - DMIN - CALK(K) + ALOSS
   1  + CORR + WLEV2(274)
    GO TO 8
   ```

6. Input out of range - print warning message and all input variables
7. WLEV2(276) = -9999.0
8. End of algorithm

   ```
   RETURN
   ```

COMMENTS:

1. The following TYPE statements must be included in the code of this submodule

   ```
   DOUBLE PRECISION ALT2
   COMMON /SENSOR/ WLEV1(300)
   COMMON /GEOPHY/ WLEV2(325)
   COMMON /ALTLIM/ ALTLO,ALTHI
   COMMON /T251A/  AGCI(8),CALK(8)
   COMMON /T251B/  ATT(16),AL(16)
   EQUIVALENCE (ALT2,WLEV2(9))
   ```

2. The original (SEASAT) algorithm used WLEV2(274) = 0. The NOSS algorithm determines WLEV2(274) in the radar backscatter coefficient atmospheric correction submodule (A.A-2.4.6) using data that are supplied by the LAMMR and the CZCS instruments.
3. The radar backscatter coefficient bias used in Step #5 (38.33 dB) was changed from the original (SEASAT) value of 39.93 dB on the recommendation of L. Fedor, based upon results of the SEASAT Gulf of Alaska Workshop (Ref. 7).
4. The altimeter off-nadir angle, WLEV2(231), is determined by the waveform processor submodule (A.A-2.3.1) if that submodule converges. If not, the instrument-computed value of the pointing angle is used.

WIND SPEED SUBMODULE A.A-2.5.2

TITLE: SUBROUTINE WIND

FUNCTION: To calculate the wind speed at the subsatellite point and altitudes of 19½ meters and 10 meters

REFERENCE:
1. "SEASAT Geophysical Data Record Users Handbook, Altimeter," JPL 622-97 Rev. A, July 1980
2. Private communication, L. Clarke (FNOC), November 1980

CONTROL: A.A-2.5 WINDSIG wind speed and radar backscatter coefficient driver module

SUPPORT: None

ACCESS: CALL WIND

INPUTS:
WLEV2(276) = radar backscatter coefficient in (dB,0.01)
A = array of coefficients from Table 2.5.2-a
B = array of coefficients from Table 2.5.2-a
C = array of coefficients from Table 2.5.2-b

OUTPUTS:
WLEV2(277) = wind speed at 10 meters in (m/s,0.1)
WLEV2(278) = wind speed at 19½ meters in (m/s,0.1)

TABLES:
Table 2.5.2-a Wind A and B Coefficients (see Appendix B)
Table 2.5.2-b Wind Polynomial Coefficients (see Appendix B)

ALGORITHM:

```
1.  Check input values
          IF (WLEV2(276).LT.0.0 .OR. WLEV2(276).GT.15.0) GO TO 5
2.  Determine the proper table entry index
          I = 1
          IF (WLEV2(276) .GT. 10.12) I = 2
          IF (WLEV2(276) .GT. 10.90) I = 3
3.  Determine the wind speed at 10 meters altitude
          X = -0.1 * (WLEV2(276)+2.1)
          Y = EXP( (10.0**X-B(I)) / A(I) )
          IF (Y .GT. 16.0) WTEN = Y
          IF (Y .LE. 16.0) WTEN = C(1)*Y + C(2)*Y**2
         1  + C(3)*Y**3 + C(4)*Y**4 + C(5)*Y**5
          WLEV2(277) = WTEN
4.  Determine the wind speed at 19½ meters altitude
          W1          = 0.66783E-02 * WTEN
          W2          = ALOG(1.0 / (1.38E-15*WTEN) )
          WLEV2(278) = WTEN + W1/W2
          GO TO 7
5.  Input out of range - print warning message and all input variables
```

```
6. WLEV2(277) = -9999.0
   WLEV2(278) = -9999.0
7. End of algorithm
         RETURN
```

COMMENTS:

1. The following TYPE statements must be included in the code of this submodule:

```
COMMON /SENSOR/ WLEV1(300)
COMMON /GEOPHY/ WLEV2(325)
COMMON /T252A/  A(3),B(3)
COMMON /T252B/  C(5)
```

2. The A and B coefficients were determined empirically from comparisons between GEOS-3 wind speed estimates and ground truth wind speed measurements

LEVEL 2 ALTITUDE CORRECTION MODULE A.A-2.6

TITLE: SUBROUTINE AL2COR

FUNCTION: To act as the driver module for the calculation of the geophysical related corrections to the altitude (except for atmospheric corrections)

REFERENCE: Not applicable

CONTROL: A.A-2.0 ALT2DR level 2 driver module

SUPPORT:

A.A-2.6.1	GEOID	geoid height submodule
A.A-2.6.2	TIDE	tide height submodule
A.A-2.6.3	SETIDE	solid earth tide height submodule
A.A-2.6.4	EMBIAS	EM bias submodule
A.A-2.6.5	SSHRES	sea surface height submodule

ACCESS: CALL AL2COR

INPUTS: None

OUTPUTS: None

TABLES: None

ALGORITHM:

```
1. Determine the geoid height
         CALL GEOID
2. Determine the tide height
         CALL TIDE
3. Determine the solid earth tide height
         CALL SETIDE
4. Determine the EM bias
         CALL EMBIAS
```

```
5. Determine the sea surface height
         CALL SSHRES
6. End of algorithm
         RETURN
```

GEOID HEIGHT SUBMODULE A.A-2.6.1A

TITLE: SUBROUTINE GEOID

FUNCTION: To compute the geoid height at the subsatellite point. Any geoid model may be used as long as the geoid heights are supplied for a 1°x1° grid.

REFERENCES: None

CONTROL: A.A-2.6 AL2COR level 2 altitude correction module

SUPPORT: A.A-2.6.1.1 BILINE bilinear interpolation submodule

ACCESS: CALL GEOID

INPUTS:
A.D-2.71 = geoid data file (see Comment #4)
WLEV1(151) = geodetic latitude in (deg,0.000001)
WLEV1(152) = longitude in (deg,0.000001)
ISTAT(43) = subtrack direction flag (0 for south to north, 1 for north to south)

OUTPUTS: WLEV2(281) = geoid height in (m,0.001)

TABLES: None

ALGORITHM:

```
1. Check for input variables out of range
         IF (ISTAT(43).LT.0 .OR. ISTAT(43).GT.1) GO TO 9
         IF (WLEV1(151).LT.-90.0 .OR. WLEV1(151).GT.90.0) GO TO 9
         IF (WLEV1(152).LT.0.0 .OR. WLEV1(152).GE.360.0) GO TO 9
2. Compute the corner point coordinates of the 1°x1° rectangle enclosing
   the subsatellite point
         I0 = WLEV1(151)
         J0 = WLEV1(152)
         IF (WLEV1(151) .LT. 0.0) I0 = I0 - 1
         X1 = J0
         Y1 = I0
         X2 = J0 + 1
         Y2 = I0 + 1
3. Determine if the geoid file must be read
         NTEST = J0 - N
         IF (NTEST.GE.0 .AND. NTEST.LE.4) GO TO 8
```

```
            NTEST = J0 - N + 360
            IF (NTEST.GE.0 .AND. NTEST.LE.4) GO TO 8
 4.  Geoid file must be read - determine which records (current longitude
     through 5 degrees longitude down track)
            N = J0 - 4
            IF (N .LT. 0) N = N + 360
            NSTOP = N + 5
 5.  Determine the latitude range to be used (current latitude through 30
     degrees latitude down track)
            IF (ISTAT(43) .EQ. 0) L1 = I0
            IF (ISTAT(43) .EQ. 1) L1 = I0 - 29
            IF (L1 .LT. -90) L1 = -90
            IF (L1 .GT. 60) L1 = 60
            L2 = L1 + 30
 6.  Read the geoid file
     a.  DO 100 I = N,NSTOP
         NREC = I + 1
         IF (NREC .GT. 360) NREC = NREC - 360
     b.  Read record #NREC from the random access geoid file into the array
         called V
 7.  Store the geoid data inside the computed latitude range
            K = N - I + 1
            DO 100 L = L1,L2
            J = L + 91
            M = L1 - L + 1
        100 G(M,K)  = V(J)
 8.  Interpolate the geoid
            K1 = I0 - L1 + 1
            K2 = K1 + 1
            J1 = J0 - N + 1
            J2 = J1 + 1
            CALL BILINE (X1,Y1,Y2,WLEV1(152),WLEV1(151),G(J1,K1),
           1             G(J1,K2),G(J2,K1),G(J2,K2),WLEV2(281))
            GO TO 11
 9.  Input out of range - print warning message and all input variables
10.  WLEV2(281) = -9999.0
11.  End of algorithm
            RETURN
```

COMMENTS:

1. The following TYPE statements must be included in the code of this submodule:

```
DIMENSION V(181),G(31,6)
COMMON /STATUS/ ISTAT(100)
COMMON /SENSOR/ WLEV1(300)
COMMON /GEOPHY/ WLEV2(325)
DATA N/-9999/
```

2. The input longitude of the subsatellite point,WLEV1(152), must be in the range

   $0 \leq \mathrm{WLEV1}(152) < 360$

3. The geoid heights are read and stored into an array 6° in longitude by 31° in latitude
4. The geoid data file (A.D-2.71) consists of 360 records (one for each degree of longitude), each containing 181 words (one for each degree of latitude). The first record contains data for 1.0° longitude.

GEOID AND TIDE HEIGHT SUBMODULE A.A-2.6.1B

TITLE: SUBROUTINE GEOTID

FUNCTION: To compute the tide and geoid heights at the subsatellite point. The tide model used is the Schwiderski tide model. Any geoid model may be used as long as the geoid heights are supplied for a 1°x1° grid.

REFERENCES: Schwiderski, E. W., "Detailed Ocean Tide Models of (N2, M2, S2, K2) and (K1, P1, O1, Q1) Including an Atlas of Tidal Charts and Maps," IUGG General Assembly XXII, Canberra, Australia, 1979.

Schwiderski, E. W., "Global Ocean Tides, Part I: A Detailed Hydrodynamical Interpolation Model," Marine Geodesy, No. 3, 1980.

Schwiderski, E. W., "On Charting Global Ocean Tides," Reviews of Geophysics and Space Physics, 1980.

CONTROL: A.A-2.6 AL2COR level 2 altitude correction module

SUPPORT: A.A-2.6.1.1 BILINE bilinear interpolation submodule

ACCESS: CALL GEOTID

INPUTS:
A.D-2.71 = geoid and tide data file
TMJD = modified julian date in $(d, 1.0 \times 10^{-9})$
WLEV1(1) = day of year
WLEV1(2) = seconds past midnight
WLEV1(120) = year - 1900
WLEV1(151) = geodetic latitude in (deg,0.000001)

WLEV1(152) = longitude in (deg,0.000001)

ISTAT(43) = subtrack direction flag (0 for south to north, 1 for north to south)

OUTPUTS: WLEV2(281) = geoid height in (m,0.001)

WLEV2(282) = tide height in (m,0.001)

TABLES: None

ALGORITHM:

```
1. Check for input variables out of range
        IF (ISTAT(43).LT.0 .OR. ISTAT(43).GT.1) GO TO 12
        IF (WLEV1(151).LT.-90.0 .OR. WLEV1(151).GT.90.0) GO TO 12
        IF (WLEV1(152).LT.0.0 .OR. WLEV1(152).GE.360.0) GO TO 12
        IF (TMJD.LT.0.0D+00 .OR. TMJD.GT.1.0D+05) GO TO 12
2. Compute the corner point coordinates of the 1°x1° rectangle enclosing
   the subsatellite point
        IO = WLEV1(151)
        JO = WLEV1(152)
        IF (WLEV1(151) .LT. 0.0) IO = IO - 1
        X1 = JO
        Y1 = IO
        X2 = JO + 1
        Y2 = IO + 1
3. Determine if the geoid/tide file must be read
        NTEST = JO - N
        IF (NTEST.GE.0 .AND. NTEST.LE.4) GO TO 8
        NTEST = JO - N + 360
        IF (NTEST.GE.0 .AND. NTEST.LE.4) GO TO 8
4. Geoid/tide file must be read - determine which records (current
   longitude through 5 degrees longitude down track)
        N = JO - 4
        IF (N .LT. 0) N = N + 360
        NSTOP = N + 5
5. Determine the latitude range to be used (current latitude through 30
   degrees latitude down track)
        IF (ISTAT(43) .EQ. 0) L1 = IO
        IF (ISTAT(43) .EQ. 1) L1 = IO - 29
        IF (L1 .LT. -90) L1 = -90
        IF (L1 .GT. 60) L1 = 60
        L2 = L1 + 30
6. Read the geoid/tide file
   a. DO 100 I = N,NSTOP
```

```
NREC = I + 1
IF (NREC .GT. 360) NREC = NREC - 360
```

b. Read record #NREC from the random access geoid/tide file into the 181x13 array called V

7. Store the geoid and tide data inside the computed latitude range

```
      K = N - I + 1
      DO 100 L = L1,L2
      J = L + 91
      M = L1 - L + 1
      G(M,K)  = V(J,1)
      SM(M,K) = V(J,2)
      CM(M,K) = V(J,3)
      SS(M,K) = V(J,4)
      CS(M,K) = V(J,5)
      SN(M,K) = V(J,6)
      CN(M,K) = V(J,7)
      SK(M,K) = V(J,8)
      CK(M,K) = V(J,9)
      SO(M,K) = V(J,10)
      CO(M,K) = V(J,11)
      SP(M,K) = V(J,12)
  100 CP(M,K) = V(J,13)
```

8. Interpolate the geoid (data is now available for interpolation)

```
      K1 = I0 - L1 + 1
      K2 = K1 + 1
      J1 = J0 - N + 1
      J2 = J1 + 1
      CALL BILINE (X1,Y1,Y2,WLEV1(152),WLEV1(151),G(J1,K1),
     1            G(J1,K2),G(J2,K1),G(J2,K2),WLEV2(281))
```

9. Interpolate the tide coefficients

```
      CALL BILINE (X1,Y1,X2,Y2,WLEV1(152),WLEV1(151),SM(J1,K1),
     1            SM(J1,K2),SM(J2,K1),SM(J2,K2),SM2)
      CALL BILINE (X1,Y1,X2,Y2,WLEV1(152),WLEV1(151),CM(J1,K1),
     1            CM(J1,K2),CM(J2,K1),CM(J2,K2),CM2)
      CALL BILINE (X1,Y1,X2,Y2,WLEV1(152),WLEV1(151),SS(J1,K1),
     1            SS(J1,K2),SS(J2,K1),SS(J2,K2),SS2)
      CALL BILINE (X1,Y1,X2,Y2,WLEV1(152),WLEV1(151),CS(J1,K1),
     1            CS(J1,K2),CS(J2,K1),CS(J2,K2),CS2)
```

```
       CALL BILINE (X1,Y1,X2,Y2,WLEV1(152),WLEV1(151),SN(J1,K1),
      1            SN(J1,K2),SN(J2,KL),SN(J2,K2),SN2)
       CALL BILINE (X1,Y1,X2,Y2,WLEV1(152),WLEV1(151),CN(J1,K1),
      1            CN(J1,K2),CN(J2,K1),CN(J2,K2),CN2)
       CALL BILINE (X1,Y1,X2,Y2,WLEV1(152),WLEV1(151),SK(J1,K1),
      1            SK(J1,K2),SK(J2,K1),SK(J2,K2),SK1)
       CALL BILINE (X1,Y1,X2,Y2,WLEV1(152),WLEV1(151),CK(J1,K1),
      1            CK(J1,K2),CK(J2,K1),CK(J2,K2),CK1)
       CALL BILINE (X1,Y1,X2,Y2,WLEV1(152),WLEV1(151),SO(J1,K1),
      1            SO(J1,K2),SO(J2,K1),SO(J2,K2),SO1)
       CALL BILINE (X1,Y1,X2,Y2,WLEV1(152),WLEV1(151),CO(J1,K1),
      1            CO(J1,K2),CO(J2,K1),CO(J2,K2),CO1)
       CALL BILINE (X1,Y1,X2,Y2,WLEV1(152),WLEV1(151),SP(J1,K1),
      1            SP(J1,K2),SP(J2,K1),SP(J2,K2),SP1)
       CALL BILINE (X1,Y1,X2,Y2,WLEV1(152),WLEV1(151),CP(J1,K1),
      1            CP(J1,K2),CP(J2,K1),CP(J2,K2),CP1)
       SK2 = 0.29 * (SS2*0.99939083-CS2*0.0348995)
       CK2 = 0.29 * (CS2*0.99939083+SS2*0.0348995)
10.  Determine the phase of the tide
     a.  IDAY = WLEV1(1)
         SEC  = WLEV1(2)
         IYR  = WLEV1(120)
         IDB  = IDAY + 365*(IYR-75) + (IYR-77)/4
         T    = (T0+T1*IDB) / 36525.0
         S    = S0 + S1*T + S2*T**2 + S3*T**3
         H    = H0 + H1*T + H2*T**2
         P    = P0 + P1*T + P2*T**2 + P3*T**3
         DTR  = 3.1415926535 / 180.0
     b.  XM2  = 2.0 * (H-S) * DTR
         XS2  = 0.0
         XN2  = (2.0*H-3.0*S+P) * DTR
         XK2  = 2.0 * H * DTR
         XK1  = (H+90.0) * DTR
         XO1  = (H-2.0*S-90.0) * DTR
         XP1  = (-H-90.0) * DTR
     c.  PM2  = SM2*SEC + XM2
         PS2  = SS2*SEC + XS2
         PN2  = SN2*SEC + XN2
         PK2  = SK2*SEC + XK2
```

```
        PK1  = SK1*SEC + XK1
        PO1  = SO1*SEC + XO1
        PP1  = SP1*SEC + XP1
11.  Calculate the height of each constituent of the tide and sum
          HM2 = CM2*COS(PM2) + SM2*SIN(PM2)
          HS2 = CS2*COS(PS2) + SS2*SIN(PS2)
          HN2 = CN2*COS(PN2) + SN2*SIN(PN2)
          HK2 = CK2*COS(PK2) + SK2*SIN(PK2)
          HK1 = CK1*COS(PK1) + SK1*SIN(PK1)
          HO1 = CO1*COS(PO1) + SO1*SIN(PO1)
          HP1 = CP1*COS(PP1) + SP1*SIN(PP1)
          WLEV2(282) = HM2 + HS2 + HN2 + HK2 + HK1 + HO1 + HP1
          GO TO 14
12.  Input out of range - print warning message and all input variables
13.  WLEV2(281) = -9999.0
     WLEV2(282) = -9999.0
14.  End of algorithm
          RETURN
```

COMMENTS:

1. The following TYPE statements must be included in the code of this submodule:

```
DOUBLE PRECISION TMJD
DIMENSION SM(31,6),SS(31,6),SN(31,6),SK(31,6),SO(31,6),
DIMENSION CM(31,6),CS(31,6),CN(31,6),CK(31,6),CO(31,6)
DIMENSION SP(31,6),CP(31,6),V(181,13),G(31,6)
COMMON /STATUS/ ISTAT(100)
COMMON /SENSOR/ WLEV1(300)
COMMON /GEOPHY/ WLEV2(300)
COMMON /JUNK/   WORK(2500)
EQUIVALENCE (V(1,1),WORK(1)) , (TMJD,WLEV1(5))
DATA N/-9999/
DATA T0,T1/27392.500528,1.0000000356/
DATA S0,S1,S2,S3/270.434358,481267.88314137,-0.001133,0.0000019/
DATA H0,H1,H2/279.69668,36000.768930485,0.000303/
DATA P0,P1,P2,P3/334.329653,4069.034032957,-0.010325,-0.000012/
```

2. The input longitude of the subsatellite point,WLEV1(152), must be in the range

   $0 \leq \mathrm{WLEV1}(152) < 360$

3. The geoid and tide coefficients are read and stored into arrays 6° in longitude by 31° in latitude

BILINEAR INTERPOLATION SUBMODULE A.A-2.6.1.1

TITLE: SUBROUTINE BILINE

FUNCTION: To linearly interpolate a three-dimensional function Z = f(X,) given the four X and Y coordinates of the corners of a rectangle and the value of Z at each of the corners.

REFERENCE: Not applicable

CONTROL: A.A-2.6.1 GEOID geoid height submodule
A.A-2.6.2 TIDE tide height submodule

SUPPORT: None

ACCESS: `CALL BILINE (X1,Y1,X2,Y2,XE,YE,Z11,Z12,Z21,Z22,ZE)`

INPUTS:
X1 = first value of X
Y1 = first value of Y
X2 = second value of X
Y2 = second value of Y
XE = X evaluation point
YE = Y evaluation point
Z11 = f(X1,Y1)
Z12 = f(X1,Y2)
Z21 = f(X2,Y1)
Z22 = f(X2,Y2)

OUTPUTS: ZE = f(XE,YE)

TABLES: None

ALGORITHM:

```
1.  Compute interpolation constants
          A1 = 0.0
          A2 = 0.0
          IF (X1 .NE. X2) A1 = (XE-X1) / (X2-X1)
          IF (Y1 .NE. Y2) A2 = (YE-Y1) / (Y2-Y1)
2.  Linearly interpolate f(X,Y) along Y = Y1
          B1 = Z11 + A1*(Z21-Z11)
3.  Linearly interpolate f(X,Y) along Y = Y2
          B2 = Z12 + A1*(Z22-Z12)
4.  Linearly interpolate f(X,Y) along X = XE
          ZE = B1 + Z2*(B2-B1)
5.  End of algorithm
          RETURN
```

COMMENTS: None

TIDE HEIGHT SUBMODULE A.A-2.6.2

TITLE: SUBROUTINE TIDE

FUNCTION: To compute the tide height at the subsatellite point. The tide model used is the Schwiderski tide model.

REFERENCES: Schwiderski, E. W., "Detailed Ocean Tide Models of (N2, M2, S2, K2) and (K1, P1, O1, Q1) Including an Atlas of Tidal Charts and Maps," IUGG General Assembly XXII, Canberra, Australia, 1979.

Schwiderski, E. W., "Global Ocean Tides, Part I: A Detailed Hydrodynamical Interpolation Model," Marine Geodesy, No. 3, 1980.

Schwiderski, E. W., "On Charting Global Ocean Tides," Reviews of Geophysics and Space Physics, 1980.

CONTROL: A.A-2.6 AL2COR level 2 altitude correction module

SUPPORT: A.A-2.6.1.1 BILINE bilinear interpolation submodule

ACCESS: CALL TIDE

INPUTS:

A.D-2.72 = tide data file

TMJD = modified julian date in $(d, 1.0\times10^{-9})$

WLEV1(1) = day of year

WLEV1(2) = seconds past midnight

WLEV1(120) = year - 1980

WLEV1(151) = geodetic latitude in (deg,0.000001)

WLEV1(152) = longitude in (deg,0.000001)

ISTAT(43) = subtrack direction flag (0 for south to north, 1 for north to south)

OUTPUTS: WLEV2(282) = tide height in (m,0.001)

TABLES: None

ALGORITHM:

1. Check for input variables out of range

```
IF (ISTAT(43).LT.0 .OR. ISTAT(43).GT.1) GO TO 11
IF (WLEV1(151).LT.-90.0 .OR. WLEV1(151).GT.90.0) GO TO 11
IF (WLEV1(152).LT.0.0 .OR. WLEV1(152).GE.360.0) GO TO 11
IF (TMJD.LT.0.0D+00 .OR. TMJD.GT.1.0D+05) GO TO 11
```

2. Compute the corner point coordinates of the 1°x1° rectangle enclosing the subsatellite point

```
I0 = WLEV1(151) + 0.5
J0 = WLEV1(152) + 0.5
IF (WLEV1(151) .LT. 0.5) I0 = I0 - 1
X1 = J0 - 0.5
Y1 = I0 - 0.5
```

```
          X2 = J0 + 0.5
          Y2 = I0 + 0.5
3.  Determine if the tide file must be read
          NTEST = J0 - N
          IF (NTEST.GE.0 .AND. NTEST.LE.4) GO TO 8
          NTEST = J0 - N + 360
          IF (NTEST.GE.0 .AND. NTEST.LE.4) GO TO 8
4.  Tide file must be read - determine which records (current longitude
    through 5 degrees longitude down track)
          N = J0 - 4
          IF (N .LT. 0) N = N + 360
          NSTOP = N + 5
5.  Determine the latitude range to be used (current latitude through 30
    degrees latitude down track)
          IF (ISTAT(43) .EQ. 0) L1 = I0
          IF (ISTAT(43) .EQ. 1) L1 = I0 - 29
          IF (L1 .LT. -89) L1 = -89
          IF (L1 .GT. 60) L1 = 60
          L2 = L1 + 30
6.  Read the tide file
    a.  DO 100 I = N,NSTOP
        NREC = I + 1
        IF (NREC .GT. 360) NREC = NREC - 360
    b.  Read record #NREC from the random access tide file into the 180x12
        array called V
7.  Store the tide data inside the computed latitude range
          K = N - I + 1
          DO 100 L = L1,L2
          J = L + 90
          M = L1 - L + 1
          SM(M,K) = V(J,1)
          CM(M,K) = V(J,2)
          SS(M,K) = V(J,3)
          CS(M,K) = V(J,4)
          SK(M,K) = V(J,5)
          CK(M,K) = V(J,6)
          SO(M,K) = V(J,7)
          CO(M,K) = V(J,8)
          SN(M,K) = V(J,9)
```

```
      CN(M,K) = V(J,10)
      SP(M,K) = V(J,11)
  100 CP(M,K) = V(J,12)
      K1 = IO-L1+1
      K2 = K1+1
      J1 = JO-N+1
      J2 = J1+1
```

8. Interpolate the tide coefficients

```
      CALL BILINE (X1,Y1,X2,Y2,WLEV1(152),WLEV1(151),SM(J1,K1),
     1            SM(J1,K2),SM(J2,K1),SM(J2,K2),SM2)
      CALL BILINE (X1,Y1,X2,Y2,WLEV1(152),WLEV1(151),CM(J1,K1),
     1            CM(J1,K2),CM(J2,K1),CM(J2,K2),CM2)
      CALL BILINE (X1,Y1,X2,Y2,WLEV1(152),WLEV1(151),SS(J1,K1),
     1            SS(J1,K2),SS(J2,K1),SS(J2,K2),SS2)
      CALL BILINE (X1,Y1,X2,Y2,WLEV1(152),WLEV1(151),CS(J1,K1),
     1            CS(J1,K2),CS(J2,K1),CS(J2,K2),CS2)
      CALL BILINE (X1,Y1,X2,Y2,WLEV1(152),WLEV1(151),SN(J1,K1),
     1            SN(J1,K2),SN(J2,KL),SN(J2,K2),SN2)
      CALL BILINE (X1,Y1,X2,Y2,WLEV1(152),WLEV1(151),CN(J1,K1),
     1            CN(J1,K2),CN(J2,K1),CN(J2,K2),CN2)
      CALL BILINE (X1,Y1,X2,Y2,WLEV1(152),WLEV1(151),SK(J1,K1),
     1            SK(J1,K2),SK(J2,K1),SK(J2,K2),SK1)
      CALL BILINE (X1,Y1,X2,Y2,WLEV1(152),WLEV1(151),CK(J1,K1),
     1            CK(J1,K2),CK(J2,K1),CK(J2,K2),CK1)
      CALL BILINE (X1,Y1,X2,Y2,WLEV1(152),WLEV1(151),SO(J1,K1),
     1            SO(J1,K2),SO(J2,K1),SO(J2,K2),SO1)
      CALL BILINE (X1,Y1,X2,Y2,WLEV1(152),WLEV1(151),CO(J1,K1),
     1            CO(J1,K2),CO(J2,K1),CO(J2,K2),CO1)
      CALL BILINE (X1,Y1,X2,Y2,WLEV1(152),WLEV1(151),SP(J1,K1),
     1            SP(J1,K2),SP(J2,K1),SP(J2,K2),SP1)
      CALL BILINE (X1,Y1,X2,Y2,WLEV1(152),WLEV1(151),CP(J1,K1),
     1            CP(J1,K2),CP(J2,K1),CP(J2,K2),CP1)
      SK2 = 0.29 * (SS2*0.99939083-CS2*0.0348995)
      CK2 = 0.29 * (CS2*0.99939083+SS2*0.0348995)
```

9. Determine the phase of the tide

a.

```
    IDAY = WLEV1(1)
    SEC  = WLEV1(2)
    IYR  = WLEV1(120)
    IDB  = IDAY + 365*(IYR-75) + (IYR-77)/4
```

```
          T     = (T0+T1*IDB) / 36525.0
          S     = S0 + S1*T + S2*T**2 + S3*T**3
          H     = H0 + H1*T + H2*T**2
          P     = P0 + P1*T + P2*T**2 + P3*T**3
          DTR   = 3.1415926535 / 180.0
       b. XM2   = 2.0 * (H-S) * DTR
          XS2   = 0.0
          XN2   = (2.0*H-3.0*S+P) * DTR
          XK2   = 2.0 * H * DTR
          XK1   = (H+90.0) * DTR
          XO1   = (H-2.0*S-90.0) * DTR
          XP1   = (-H-90.0) * DTR
       c. PM2   = SM2*SEC + XM2
          PS2   = SS2*SEC + XS2
          PN2   = SN2*SEC + XN2
          PK2   = SK2*SEC + XK2
          PK1   = SK1*SEC + XK1
          PO1   = SO1*SEC + XO1
          PP1   = SP1*SEC + XP1
   10. Calculate the height of each constituent of the tide and sum
            HM2 = CM2*COS(PM2) + SM2*SIN(PM2)
            HS2 = CS2*COS(PS2) + SS2*SIN(PS2)
            HN2 = CN2*COS(PN2) + SN2*SIN(PN2)
            HK2 = CK2*COS(PK2) + SK2*SIN(PK2)
            HK1 = CK1*COS(PK1) + SK1*SIN(PK1)
            HO1 = CO1*COS(PO1) + SO1*SIN(PO1)
            HP1 = CP1*COS(PP1) + SP1*SIN(PP1)
            WLEV1(282) = HM2 + HS2 + HN2 + HK2 + HK1 + HO1 + HP1
            GO TO 13
   11. Input out of range - print warning message and all input variables
   12. WLEV2(282) = -9999.0
   13. End of algorithm
            RETURN
```

COMMENTS: 1. The following TYPE statements must be included in the code of this submodule:

```
DOUBLE PRECISION TMJD
DIMENSION SM(31,6),SS(31,6),SN(31,6),SK(31,6),SO(31,6),
DIMENSION CM(31,6),CS(31,6),CN(31,6),CK(31,6),CO(31,6)
DIMENSION SP(31,6),CP(31,6),V(180,12)
```

```
      COMMON /STATUS/ ISTAT(100)
      COMMON /SENSOR/ WLEV1(300)
      COMMON /GEOPHY/ WLEV2(300)
      COMMON /JUNK/   WORK(2500)
      EQUIVALENCE (V(1,1),WORK(1)) , (TMJD,WLEV1(5))
      DATA N/-9999/
      DATA T0,T1/27392.500528,1.0000000356/
      DATA S0,S1,S2,S3/270.434358,481267.88314137,-0.001133,0.0000019/
      DATA H0,H1,H2/279.69668,36000.768930485,0.000303/
      DATA P0,P1,P2,P3/334.329653,4069.034032957,-0.010325,-0.000012/
```

2. The input longitude of the subsatellite point, WLEV1(152), must be in the range

   $0 \leq \text{WLEV1}(152) < 360$

3. The tide coefficients are read and stored into an array 6° in longitude by 31° in latitude
4. The tide data file (A.D-2.72) consists of 360 records (one for each degree of longitude), each containing 180 words (one for each degree of latitude). The first record contains data for 0.5° longitude.
5. It was assumed that coefficients over land will cause this correction to be computed as zero. This is probably not the true case.

SOLID EARTH TIDE HEIGHT SUBMODULE A.A-2.6.3

TITLE: SUBROUTINE SETIDE

FUNCTION: To calculate the magnitude of the solid earth tide height at the subsatellite point.

REFERENCE: <u>SEASAT Altimeter Geophysical Algorithm Specifications</u>, JPL 622-226, December 1980

CONTROL: A.A-2.6 AL2COR level 2 altitude correction module

SUPPORT: A.A-2.6.3.1 SLINT solar/lunar ephemerides interpolation submodule

ACCESS: CALL SETIDE

INPUTS:

WLEV1(1) = day of year
WLEV1(2) = seconds past midnight
WLEV1(120) = year - 1900
WLEV2(7) = geodetic latitude in (deg,0.000001)
WLEV2(8) = longitude in (deg,0.000001)
RM = lunar inertial position vector in (m,1.0)
RS = solar inertial position vector in (m,1.0)

OUTPUTS: WLEV2(283) = solid earth tide height in (m,0.001)

TABLES: None

ALGORITHM:

1. Compute the modified Julian date (Julian date minus 2,400,000.5). Note that this modified Julian date is the modified Julian date defined by the GEOS and SEASAT software and is not consistent with the modified Julian date defined elsewhere in the NOSS algorithm specifications.

```
IDAY  = WLEV1(1)
IYEAR = WLEV1(120)
FODAY = WLEV1(2) / 86400.0
NYRM1 = IYEAR + 1899
IC    = NYRM1 / 100
MJD   = -678576 + 365*NYRM1 + NYRM1/4 - IC + IC/4
```

2. Convert latitude and longitude to inertial coordinates

```
TU    = (MJD-15019.5) / 36525.0
A     = 99.69098 + (36000.7689+0.00038708*TU)*TU
B     = A + 360.9856473*FODAY
C     = COS(A)
S     = SIN(A)
F     = 1.0 / FR
E     = 2.0*F - F*F
RLAT  = WLEV2(7) * PI / 180.0
RLON  = WLEV2(8) * PI / 180.0
P     = ATAN(TAN(RLAT*(1.0-E)))
XE1   = COS(P) * (C*COS(RLON)-S*SIN(RLON))
XE2   = COS(P) * (S*COS(RLON)+C*SIN(RLON))
XE3   = SIN(P)
```

3. Interpolate the inertial position vectors of the sun and the moon.

```
CALL SLINT
```

4. Calculate the deformation due to the moon

```
DM    = DSQRT(RM(1)**2+RM(2)**2+RM(3)**2)
TM    = ACOS((XE1*RM(1)+XE2*RM(2)+XE3*RM(3))/DM)
CTM   = COS(TM)
DHM   = H2 * RATM * AE**4 / DM**3 * (1.5*CTM**2-0.5)
```

5. Calculate the deformation due to the sun

```
DS    = DSQRT(RS(1)**2+RS(2)**2+RS(3)**2)
TS    = ACOS((XE1*RS(1)+XE2*RS(2)+XE3*RS(3))/DS)
CTS   = COS(TS)
DHS   = H2 * RATS * AE**4 / DS**3 * (1.5*CTS**2-0.5)
```

6. Compute the net deformation

```
WLEV2(283) = DHM + DHS
```

7. End of algorithm

```
RETURN
```

COMMENTS: 1. The following TYPE statements must be included in the code of this submodule

```
DOUBLE PRECISION RM(3),RS(3)
COMMON /SENSOR/ WLEV1(300)
COMMON /GEOPHY/ WLEV2(325)
EQUIVALENCE (WLEV2(311),RM(1)) , (WLEV2(317),RS(1))
DATA H2/0.612/ , AE/6378145.0/ , FR/298.257/
DATA RATM/0.01229997/ , RATS/332945.562/
DATA PI/3.1415926/
```

SOLAR/LUNAR EPHEMERIDES INTERPOLATION SUBMODULE A.A-2.6.3.1

TITLE: SLINT

FUNCTION: To determine the inertial position vectors of the sun and the moon by interpolating the ephemeris file, A.D(M)-2.85 to the proper time.

REFERENCE: None

CONTROL: A.A-1.5 LOCATE location classification module

SUPPORT: None

ACCESS: CALL SLINT

INPUTS:
A.D(M)-2.85 = solar/lunar ephemeris file
TMJD = modified Julian date in (d,1.0x10^{-9})

OUTPUTS:
RM = lunar inertial position vector in (m,1.0)
RS = solar inertial position vector in (m,1.0)

TABLES: None

ALGORITHM: TBD

COMMENTS: 1. The following TYPE statement must be included in the code of this submodule

```
DOUBLE PRECISION RM(3),RS(3)
COMMON /GEOPHY/ WLEV2(325)
EQUIVALENCE (WLEV2(311),RM(1) , (WLEV2(317),RS(1))
```

2. The specific design of this algorithm for now is being left to the mission contractor.

EM BIAS A.A-2.6.4

TITLE: SUBROUTINE EMBIAS

FUNCTION: To provide a correction for the electromagnetic (EM) bias effect in which the relative radar cross section tends to increase below mean sea level (MSL) and decrease above MSL in the presence of waves. Its effect is to shift the centroid of the radar return away from MSL toward the wave troughs, so that the altimeter tracks long. Recent experimental data from the surface contour radar (SCR) at 36 GHz, and the NRL 10-GHz adaptive radar altimeter indicate that the EM bias is in the range of 0 to 3 percent of the SWH by E. J. Walsh. The data indicate that the magnitude of the effect may increase with wave height. A theoretical development by Jackson (Ref. 8) using a one-dimensional model of the sea surface indicated that there should be a linear dependence of the EM bias on the skewness of the height distribution.

REFERENCE: See above.

CONTROL: A.A-2.6 AL2COR level 2 altitude correction module

SUPPORT: TBD

ACCESS: CALL EMBIAS

INPUTS:
WLEV2(230) = significant waveheight from A.A-2.3 (SWH)
WLEV2(232) = ocean-wave skewness from A.A-2.3 (λ_s)
WLEV2(277) = wind speed from A.A-2.5.2 (W_{10})

OUTPUTS:
WLEV2(284) = height correction from EM bias
WLEV2(285) = standard deviation of $\delta h_{2.5}$
WLEV2(286) = EM sea-state bias quality flag

TABLES: A table of constants, K_1, K_2, K_3, and K_4 (TBD).

ALGORITHM:
WLEV2(284) = $K_1 + K_2 SWH + K_3\lambda_s + K_4 W_{10}{}^{1/2}$ SWH
if less than zero then set to zero.
WLEV2(285) = TBD
WLEV2(286) = TBD

COMMENTS: 1. This is only a functional description.

SEA SURFACE HEIGHT SUBMODULE A.A-2.6.5

TITLE: SUBROUTINE SSHRES

FUNCTION: To compute the sea surface height and altitude residual. The sea surface height is the difference between the reference ellipsoid and the corrected altitude measurement and as such is an estimate of the altimeter geoid. The altitude residual is the difference between the corrected altitude measurement and the modeled altitude measurement and is used in estimation and orbit determination.

REFERENCE: None

CONTROL: A.A-2.6 AL2COR level 2 altitude correction module

SUPPORT: None

ACCESS: CALL SSHRES

INPUTS:

ALT2 = corrected altitude in (m,0.001)
HGT = height above the reference ellipsoid in (m,0.001)
WLEV2(281) = geoid height in (m,0.001)
WLEV2(282) = tide height in (m,0.001)
WLEV2(283) = solid earth tide height in (m,0.001)
ALTLO = altitude lower limit in (m,1.0)
ALTHI = altitude higher limit in (m,1.0)

OUTPUTS:

WLEV2(287) = sea surface height in (m,0.001)
WLEV2(288) = altitude residual in (m,0.001)

TABLES: None

ALGORITHM:

```
1.  Check input values
          IF (ALT2.LT.ALTOLO .OR. ALT2.GT.ALTHI) GO TO 4
          IF (HGT.LT.ALTLO .OR. HGT.GT.ALTHI) GO TO 4
          IF (WLEV2(281).LT.-150.0 .OR. WLEV2(281).GT.150.0) GO TO 4
          IF (WLEV2(282).LT.-10.0 .OR. WLEV2(282.GT.10.0) GO TO 4
          IF (WLEV2(283).LT.-10.0 .OR. WLEV2(283).GT.10.0) GO TO 4
2.  Determine the sea surface height
          WLEV2(287) = HGT - ALT2
3.  Determine the altitude residual
          WLEV2(288) = ALT2 - HGT + WLEV2(281)
         1  + WLEV2(272) + WLEV2(283)
          GO TO 6
4.  Input out of range - print warning message and all input variables
5.  WLEV2(287) = -9999.0
    WLEV2(288) = -9999.0
6.  End of algorithm
          RETURN
```

COMMENTS:

1. The following TYPE statements must be included in the code of this submodule:

```
DOUBLE PRECISION ALT2,HGT
COMMON /SENSOR/ WLEV1(300)
COMMON /GEOPHY/ WLEV2(325)
```

```
COMMON /ALTLIM/ ALTLO,ALTHI
EQUIVALENCE (ALT2,WLEV2(9)) , (HGT,WLEV1(137))
```

ICE SHEET HEIGHT A.A-2.7

TITLE: SUBROUTINE ICE

FUNCTION: To correct sea-surface height estimates over ice sheet (and possibly over all non-ocean surfaces) for non-ocean return characteristics.

REFERENCE: None

CONTROL: A.A-2.0 ALT2DR level 2 driver module

SUPPORT: TBD

ACCESS: CALL ICE

INPUTS:

WLEV2(287) = sea-surface height from A.A-2.6.5 in (m,0.001)
WLEV2(101-163) = waveform samples in (counts,0.1)
WLEV2(22) = AGC in (db,0.01)
WLEV2(19) = height rate in (m/s,0.01)
WLEV2(12) = spacecraft instrument attitude in (deg,0.01)
WLEV2(20) = height error in (m,0.001)
ISTAT(28) = adaptive resolution step size

OUTPUTS:

WLEV2(287) = corrected sea-surface height in (m,0.001)
WLEV2(290) = sea-surface height correction in (m,0.001)
WLEV2(291) = mean surface roughness in (dimensionless,0.1)
WLEV2(289) = mean surface slope in (deg,0.01)
WLEV2(292) = sea-surface height retrack estimate quality flag (0 for good, 1 for questionable)

TABLES: TBD

ALGORITHM: TBD

COMMENTS:

1. This module will be employed only over areas of interest. The altimeter data will be corrected for waveform shape changes that cause track-point shifts. This correction will be done by a software retracking process designed for ice-sheet processing, and then the surface height will be calculated.
2. This is only a functional description. The algorithms will be defined at a later date.

SEA ICE A.A-2.8

TITLE: SUBROUTINE SEAICE

FUNCTION: The sea ice related quantities, mean-squared slope, Fresnel power reflection and percent smooth area, are estimated.

REFERENCE: Stanley, H. R., and R. E. Dwyer, "NASA Wallops Flight Center GEOS-3 Altimeter Data Processing," NASA RP-1066, November 1980.

CONTROL: A.A-2.0 ALT2DR level 2 driver module

SUPPORT: None

ACCESS: CALL SEAICE

INPUTS:

WLEV2(12) = instrument attitude
WLEV2(22) = automatic gain control
WLEV2(37) = attitude gate
WLEV2(36) = plateau gate
WLEV2(99) = transmitted power
WLEV2(163) = 63 waveform samples

OUTPUTS:

WLEV2(242) = MSS (mean-square-slope)
WLEV2(243) = F (Fresnel power reflection coefficient)
WLEV2(244) = PCS (percent smooth area)

TABLES: None

ALGORITHM: These parameters were calculated on GEOS-3 (Ref. 9) and appeared to correlate with the Dwyer Godin index (Ref. 10). They are based on interpretation of the physics of radar scattering, whereas the Dwyer Godin index is an ad hoc ice index.

The Fresnel power reflection coefficient, F, is computed based on a relationship of the peak waveform amplitude and the mean square slope.

COMMENTS: 1. This is only a functional description.

QUALITY CONTROL A.A-2.9

TITLE: SUBROUTINE QUALTY

FUNCTION: To classify the quality of the level 2 output data A.D-2.91. The data will be flagged as being of questionable quality when prescribed standard deviation tolerances are exceeded, when the number of rejected points in the various smoothing algorithms exceeds acceptable limits, or when operational threshold limits are exceeded.

REFERENCES: None

CONTROL: A.A-2.0 ALT2DR level 2 driver module

SUPPORT: TBD

ACCESS: CALL QUALTY

INPUTS: Level 2 altimeter data.

OUTPUTS: The following data quality flags are set:

WLEV2(294) = quality flag for σ_h
WLEV2(295) = quality flag for height rate
WLEV2(296) = quality flag for radar backscatter
WLEV2(297) = quality flag for AGC standard deviation
WLEV2(298) = quality flag for attitude
WLEV2(299) = quality flag for MTU temperature
WLEV2(300) = quality flag for DFB temperature
WLEV2(301) = quality flag for noise gate
WLEV2(302) = quality flag for AGC gate
WLEV2(303) = quality flag for TWT collector temperature
WLEV2(304) = quality flag for SWH standard deviation
WLEV2(305) = EM sea-state bias quality flag
WLEV2(306) = quality flag for data validity

Values of 0 indicate acceptable quality; values of 1 indicate questionable quality.

TABLES: A table of preflight nominal tolerances to be used in setting the data quality flags (TBD).

ALGORITHM: Level 2 output parameters are compared with the table and output flags are set to the appropriate values. Limits on standard deviations are compared after a decay filter is used to remove spurious points.

COMMENTS:

1. As a final step in the level 2 processing, the altimeter software will analyze the contents of the level 2 output file in order to classify the quality of the data. The data will be flagged as being of questionable quality when: (a) prescribed standard deviation tolerances are exceeded, (b) the number of rejected points in the various smoothing algorithms exceeds acceptable limits, or (c) operational threshold limits are exceeded.
2. This is only a functional description.

OTHER CONSIDERATIONS

This section summarizes the additional considerations required for developing the software for the altimeter processing. The Program Limitations section lists known limitations that are built into the processing and Expected Types of Future Changes/Updates section identifies the status and charges needed to complete the algorithms.

Program Limitations

The following program limitations are repeated here from the Key Assumptions section:

(a) All data required from the input files (i.e., ephemeris files, FNOC file, LAMMR level 2 file, etc.) must be available and current as the altimeter processing software requires it. Failure to supply any of the input files is not to result in the abnormal termination of the altimeter processing software but to produce degraded output products, which will be flagged as such.

(b) The requirements associated with the possible processing of data from more than one spacecraft have not been addressed in this report (i.e., no tables allow for four altimeters).

Expected Types of Future Changes/Updates

Modules are logically grouped in this report. However, a multisensor processing system may require the processing order to be revised for some modules.

The following modules will require future updates:

(a) 1.1-Engineering Units Conversion - The conversion constants provided in this report are Seasat values. NOSS values will replace these constants as they become available.

(b) 1.1.8--Engineering Units Rain Subcom - This module has only been scoped and will be completed in the future.

(c) 1.1.9--EU Waveform, CW or Dump - Only waveform has been specified.

(d) 1.2.1--Time Tag Correction - The table entries in this module are based upon Seasat values. NOSS values will replace them as they become available.

(e) 1.2.2--Cal Zone Bias - The table of cal zone bias corrections will be provided after the launch of the spacecraft.

(e) 1.2.3--Center of Gravity Offset - The table of center of gravity offsets will be provided after the geometry of the spacecraft is defined and updated after launch as fuel is expended.

(f) 1.2.4--Cal Mode Bias - The table of altitude and AGC cal mode biases will be provided after the launch of the spacecraft.

(g) 1.3.1--Cal 1 Processor - The calibration mode data base constants will be supplied three months before the launch of the spacecraft.

(i) 1.3.2--Cal 2 Processor - The calibration mode data base constants will be supplied before the launch of the spacecraft.

(j) 1.3.3--Trend File Processor - The curve-fit technique and the display requirements will be specified prior to the launch of the spacecraft.

(k) 1.4--Adaptive Resolution - This module has only been scoped and will be completed in the future.

(l) 1.5.1--Spacecraft Ephemeris Interpolation - This module has only been scoped and will be completed in the future.

(m) 1.5.2--Subsatellite Point Calculation - This module has only been scoped and will be completed in the future.

(n) 2.1--Contamination Processing - This module has only been scoped and will be completed in the future.

(o) 2.3.2--Waveform Altitude Correction - This module has only been scoped and will be completed in the future.

(p) 2.3.3--Waveform SWH Correction - This module has only been scoped and will be completed in the future.

(q) 2.3.4--Waveform SWH Bias - This module has only been scoped and will be completed in the future.

(r) 2.4--Atmospheric Module - This module and its submodules has only been scoped and will be completed in the future.

(s) 2.5.2--Wind Speed - The algorithm depends on the necessity of correcting for the atmosphere for the best accuracy.

(t) 2.6.3.1--Solar/Lunar Ephemeris Interpolation - This module has only been scoped and will be completed in the future.

(u) 2.6.4--EM Bias This module has only been scoped and will be completed in the future.

(v) 2.7--Ice Sheet Height - This module has only been scoped and will be completed in the future.

(w) 2.8--Sea Ice - This module has only been scoped and will be completed in the future.

(x) 2.9--Quality Control - This module has only been scoped and will be completed in the future.

REFERENCES

1. Hancock, D. W., R. G. Forsythe and J. D. McMillan, "NOSS Altimeter Algorithm Specifications," NASA RP-1083, January 1982.
2. "Special Issue on the SEASAT-1 Sensors," IEEE Journal of Ocean Engineering, Vol. OE-5, No. 2, April 1980.
3. Townsend, W. F., "An Initial Assessment of the Performance Achieved by the SEASAT-1 Radar Altimeter," NASA TM-73279, February 1980.
4. Hayne, G. S., "Radar Altimeter Waveform Modeled Parameter Recovery," NASA TM-73294, August 1981.
5. Gibson, L. R., "Some Expansions for an Electromagnetic Wave Propagating Through a Spherically Symmetric Refracting Medium," Naval Surface Weapons Center, dl TR-3344, June 1975.
6. "NOSS Algorithm Freeze Report, Volume 4, SCATT," Goddard Space Flight Center, October 1980.
7. "Seasat Gulf of Alaska Workshop Report," Jet Propulsion Laboratory, PD622-101, April 1979.
8. Jackson, F. C., "The Reflection of Impulses from a Nonlinear Random Sea," Journal of Geophysical Research, Vol. 84, No. C8, August 1979.
9. Stanley, H. R., and R. E. Dwyer, "NASA Wallops Flight Center GEOS-3 Altimeter Data Processing," NASA RP-1066, November 1980.
10. Dwyer, R. E., and R. H. Godin, "Determining Sea-Ice Boundaries and Ice Roughness Using GEOS-3 Altimeter Data," NASA CR-156862, March 1980.

BIBLIOGRAPHY

Brown, G. S., "Estimation of Surface Wind Speeds Using Satellite-Borne Radar Measurements at Normal Incidence," Journal of Geophysical Research, Vol. 84, No. B8, July 1979.

Goldhirsh, J. and E. Walsh, "Precipitation Measurements from Space Using a Modified Seasat Type Radar Altimeter," JHU/APL S1R81U-022, May 1981.

Hayne, G. S., "Wallops Waveform Analysis of SEASAT-1 Radar Altimeter Data," NASA CR-156869, June 1980.

Huang, N. E., and S. R. Long, "A Study of the Waveheight Probability Distribution and Statistics of Wind Generated Waves," submitted for publication Journal of Fluid Mechanics.

Leitao, C. D., N. E. Huang, C. G. Parra, "Ocean Surface Measurement Using Elevations from GEOS-3 Altimeter," Journal of Spacecraft and Rockets, Vol. 15, No. 6, November 1978.

MacArthur, J. L., "SEASAT-A Radar Altimeter Design Description," Applied Physics Laboratory, SDL-5232, November 1978.

Martin, C. F. and R. L. Taylor, "Report on Boundary Detection Criteria for Satellite Altimeters," NASA CR-156880, September 1981.

McGoogan, J. T., and E. J. Walsh, "Real-Time Determination of Geophysical Parameters from a Multibeam Altimeter," AIAA 78-1735, November 1978.

Nesterczuk, G., "Ionospheric Propagation Correction Modeling for Satellite Altimeters," NASA CR-156881, Contract NAS6-3075, November 1978.

"NOSS Algorithm Development Plan," Goddard Space Flight Center, June 1980.

"NOSS Algorithm Freeze Report, Volume 1, ALT," Goddard Space Flight Center, October 1980.

"NOSS Algorithm Freeze Report, Volume 3, LAMMR," Goddard Space Flight Center, October 1980.

Schwiderski, E. W., "The Preliminary M_2-Tide Model (A Synoptic Description)," Naval Surface Weapons Center, February 1978.

"SEASAT Algorithm Development Facility Altimeter Sensor Algorithm Specifications," Jet Propulsion Laboratory, PD 622-202, Revision A, March 1980.

"Seasat Altimeter Geophysical Algorithm Specifications," Jet Propulsion Laboratory, PD 622-226, December 1980.

"SEASAT Interim Geophysical Data Record Users Handbook," Jet Propulsion Laboratory, PD 622-97, April 1979.

"SEASAT-A Instrument Data Processing System Capabilities and Operations Guide," Jet Propulsion Laboratory, PD 622-46.

"SEASAT-A Instrument Data Processing System Detail Functional Specification," Jet Propulsion Laboratory, PD 622-14, June 1977.

"SEASAT-A Sensor Data Record Tape Specification Interface Control Document and Telemetry Dictionary," Jet Propulsion Laboratory, PD 622-57, Rev. A, May 1979.

Stanley, H. R., R. L. Brooks, G. S. Brown, "Ice Freeboard Determination by Satellite Altimetry," International Workshop on Remote Estimation of Sea Ice Thickness, St. Johns, Newfoundland, September 1979.

Tapley, B. D., et al., "Accuracy Assessment of the SEASAT Orbit and Height Measurement," University of Texas at Austin, IASOM TR79-5, October 1979.

Townsend, W. F., et al., "SEASAT-1 Radar Altimeter Phase I Engineering Assessment Report," NASA Wallops Flight Center, December 1978.

Walsh, E. J., "Altimeter Rain Detection," NASA TM-73291, July 1981.

APPENDIX A

Table A.1

NOSS Altimeter Telemetry Mode 1 Content (20 records/sec.)

1. Time (GMT)
2. Time (GMT)
3. Time (GMT)
4. Time (GMT)
5. Altitude
6. Altitude
7. Altitude
8. Altitude Rate
9. Altitude Error
10. SWH
11. AGC word
12. AGC gate
13. Early gate
14. Late gate
15. Middle gate
16. Gate normalization factor
17. Noise gate
18. Plateau gate
19. Attitude gate
20. Transmit power
21. Cal atten/SACU status
22. Status 1
23. Status 2
24. Status 3
25. Status 4
26. Status 5
27. Engineering subcom #1 (20 deep)
28. Engineering subcom #2 (20 deep)
29. Engineering subcom #3 (20 deep)
30. Rain detection subcom (20 deep)
31. Waveform samples (average of 50 pulses)

⋮

93. Waveform Samples
94. Spare

⋮

100. Spare

<u>NOSS Altimeter Telemetry Mode 2 Content (CW)</u>--Words 1 to 30 will be the same as TM format 1. Words 31 to 93 will be nine groups of seven words containing T_x count, hit count, altitude and AGC.

<u>NOSS Altimeter Telemetry 3 Content (dump)</u>--Same content as TM 2, except words 31 to 93 will contain memory dump data.

Table A.1 (continued)

CAL Atten/SACU Status

MSB	10	Not used (=0)		
	9			
	8	MSB=48 dB	Calibrate attenuator "0"= insert attenuation (all zeroes is max value) 0 to 60 dB in 11 steps	command out to SACU
	7			
	6			
	5	LSB=6 dB		
	4	Spare		
	3	HV ON	SACU status "1" = true	
	2	HV ready		
	1	TWT fault		

Table A.1 (continued)

Status #1 (last Data Command Sent)

CMD word no.	10	9	8	7	6	5	4	3	Mode
1	0	0	0	0	0	0	0	0	Standy
2	1	0	0	0	0	0	0	1	Calibrate
3	1	0	0	0	0	0	1	0	Trigger kill
4	0	0	0	0	0	0	1	1	Track 1
5	1	0	0	0	0	1	0	0	Track 2
6	0	0	0	0	0	1	0	1	Track 3
7	0	0	0	0	0	1	1	0	Track 4
8	1	0	0	0	0	1	1	1	TWT fault reset
9	1	0	0	0	1	0	0	0	Test mode 1 (CW)
10	0	0	0	0	1	0	0	1	Test mode 2
11	0	0	0	0	1	0	1	0	Test mode 3
12	1	0	0	0	1	0	1	1	Test mode 4
13	0	0	0	0	1	1	0	0	Adapt. Resol.
14	1	0	0	0	1	1	0	1	TBD
15	1	0	0	0	1	1	1	0	TBD
16	0	0	0	0	1	1	1	1	TBD
	Parity	Memory dump	Rain processing enable	CAL I, II (add to track command to specify complete calibrate mode cycle to be run once every two hours)	Mode command				

Notes: Bits 2 & 1

1 0 → Execute bits 3 through 6 immediately

0 0 → First initialize the tracker, then execute bits 3 through 10

0 1 → Load memory dump control words, allows 256 data commands

1 1 → Load parameter select control words, allows 256 data command

Bits 3 through 10 used for parity. Commands sent with bits 1 and 2 either 01 or 11 will not appear in TM word 24

Table A.1 (continued)

Status #2 (Engineering Data Channel/ATU Mode)

MSB	10	MSB	Channel select (1-46)
	.		
	.		
	.		
	.		
	5	LSB	
	4		ATU mode - same as bits 6, 5, 4, and 3 of word 15 (last data command) except when in CAL III.
	.		
	.		
	.		
	1		

Status #3 (ATU Branch Status)

MSB	10	MSB	Gate width number (6=50 ns)	50, 25, 12.5, 6.25 3.125 normal spacing (early, middle and late gate continued) 3.125 close spacing (early and late gates overlapping middle gate)
	9			
	8	LSB		
	7	ACQ/TRK		
	6	Chirp ACQ step		
	5			
	4	Reacquire flag		
	3	$\Delta H > T_{\Delta H}$		
	2	Spare		
LSB	1			

Table A.1 (continued)

Status #4 (SACU Mode Command)

MSB	10	Not used (=0)
	9	
	8	Chirp/CW (chirp=1)
	7	High voltage ON/OFF (ON=1)
	6	TWTA fault reset (reset=1 for 50 ms)
	5	Trigger kill (=1)
	4	Calibrate mode I (1st 11 steps)
	3	Calibrate mode II (noise only)
	2	TWT heater ON/OFF (ON=1),(always ON if power applied)
LSB	1	Spare

Status #5 (ATU Control Status)

MSB	10	Altimeter designator
	9	Program version
	8	
	7	
	6	
	5	Tracker type
	4	
	3	Resolution step
	2	
LSB	1	

Table A.1 (continued)

Engineering Subcom #1

1. TWT beam current
2. TWT cathode voltage
3. TWT HVPS temperature
4. TWT collector temperature
5. No data
6. Receiver temperature
7. Noise gate amplitude
8. Plateau gate amplitude
9. Attitude gate amplitude
10. Transmit power amplitude
11. UCFM temperature
12. DDL temperature
13. DDL ASSY temperature
14. HSWS temperature
15. DFB temperature #1
16. AT #1 temperature
17. AT #2 temperature
18. ICU temperature
19. SACU temperature
20. LVPS temperature

Engineering Subcom #2

1. LVPS 38V current
2. +28V S/C bus isolated
3. +28V
4. +15V
5. -15V
6. +7V
7. -9V
8. +5V
9. -5.2V
10. +1.00V REF
11. 0.657V REF
12. SACU PLO LOCK
13. MTU temperature
14. No data
15. DFB temperature #2
16. Spare #1
17. Spare #2
18. Spare #3
19. Spare #4
20. Spare #5

Table A.1 (continued)

Engineering Subcom #3

1. Relay status (see below)
2. Bits 1 through 8 spare
3. Parameter select 1
4. Parameter select 2
5. Parameter select 3
6. Parameter select 4
7. Spare #6
8. Spare #7
9. Spare #8
10. Spare #9
11. Spare #10
12. Spare #11
13. Spare #12
14. Spare #13
15. Spare #14
16. Spare #15
17. Spare #16
18. Spare #17
19. Spare #18
20. Spare #19

Relay status

Bit #3
0 = TWTA fault override
1 = TWTA fault normal

Bit #2
0 = LVPS current normal
1 = LVPS current override

Bit #1
0 = AT #1
1 = AT #2

Table A.1 (continued)

Parameter select 1	
Bits 10 & 9	Index to select acquisition running average time constant
Bits 8 & 7	Index to select track running average time constant
Bits 6,5,4,&3	Index to select track AGC threshold
Parameter select 2	
Bits 10,9,8,&7	Index to select acquisition α, β, and AGC time constants
Bits 6,5,4,&3	Index to select track α, β, and AGC time constants
Parameter select 3	
Bits 10 & 9	Index to select minimum gate width for ΔH computation
Bits 8 & 7	Index to select average L_6-E_6 threshold for track (T_{TT})
Bits 6 & 5	Index to select height error threshold ($T_{\Delta H}$)
Bits 4 & 3	Index to select average L_6-E_6 threshold for chirp acquisition (T_{TA})
Parameter select 4	
Bits 10,9,&8	Offset for adjustment waveheight curves (ΔK_{L-E})
Bits 7,6,5,4,&3	Acquisition height offset, LSB=0.4 ms

Notes: Bits 2 & 1 are 1, 1 for all parameter words.

If no parameter selection is requested, then words 13-16 are all zero.

If parameters are selected, the 32 selectable bits will not alter the status of the tracker immediately but will be utilized with a subsequent track 4 command.

If the altimeter is placed in the OFF mode, or when any command is sent with bit 2 = 0, then the selection is lost and must be reloaded.

Table A.2

Altimeter-Related Spacecraft Engineering Data (1/sec)*

Name	Length (bytes)
Time (GMT)	8
Instrument attitude	8
Baseplate temperature	120
Instrument currents	16
Instrument voltages	8
Instrument heater status	8
Altimeter analog channels	96
	264

* Information is always available and must be continuously processed for both altimeters.

APPENDIX B

Table 0.0-a A.D-1.94 and A.D-2.91 Header Record

Word Number	Explanation
1	satellite ID
2	instrument ID
3	program version
4	year - 1900
5	TBD
.	.
.	.
.	.
100	TBD

Table 0.0-b A.D-1.94 Header Record #2

Word Number	Explanation	
1	EU(1,1,1)	
2	EU(2,1,1)	
.	.	
.	.	
.	.	
8	EU(8,1,1)	Altimeter #1
9	EU(1,2,1)	
.	.	
.	.	
.	.	
800	EU(8,100,1)	
801	EU(1,1,2)	
802	EU(2,1,2)	
.	.	
.	.	Altimeter #2
.	.	
1600	EU(8,100,2)	

Table 1.1.2 AGC Word Lookup Table

Atten. Setting	AGC Value	Atten. Setting	AGC Value	Atten. Setting	AGC Value
0	0.0	22	22.1	43	43.7
1	1.0	23	23.0	44	44.2
2	2.0	24	24.1	45	45.7
3	2.9	25	25.1	46	46.2
4	4.0	26	26.1	47	47.6
5	5.0	27	27.1	48	48.5
6	5.9	28	27.8	49	49.9
7	6.8	29	28.9	50	50.5
8	8.0	30	30.0	51	52.0
9	9.0	31	31.0	52	52.4
10	9.9	32	32.4	53	53.9
11	10.9	33	33.6	54	54.7
12	11.9	34	34.2	55	56.1
13	12.9	35	35.6	56	56.8
14	13.8	36	36.2	57	58.3
15	14.8	37	37.5	58	58.9
16	16.1	38	38.2	59	60.3
17	17.1	39	39.5	60	60.6
18	18.1	40	40.3	61	62.1
19	19.0	41	41.7	62	62.9
20	19.9	42	42.3	63	64.1
21	20.9				

Table 1.2.1 Constant Time Tag Correction

I	DTC(I,1)	ETC(I,2)
1	-0.147951	-0.147951
2	0.0	0.0
3	0.0	0.0
4	-0.147951	-0.147951

Table 1.2.2 Cal Zone Bias and Standard Deviation

I	DCZB(I)	CZB(I,1)	CZB(I,2)	SCZB(I,1)	SCZB(I,2)
1	0.0	0.0	0.0	0.0	0.0
2	9999.0	-9999.0	-9999.0	-9999.0	-9999.0
3	9999.0	-9999.0	-9999.0	-9999.0	-9999.0
4	9999.0	-9999.0	-9999.0	-9999.0	-9999.0
5	9999.0	-9999.0	-9999.0	-9999.0	-9999.0

Table 1.2.3 C.G. Offset and Standard Deviation

I	DCGO(I)	CGO(I,1)	CGO(I,2)	SCGO(I,1)	SCGO(I,2)
1	0.0	0.0	0.0	0.0	0.0
2	9999.0	-9999.0	-9999.0	-9999.0	-9999.0
3	9999.0	-9999.0	-9999.0	-9999.0	-9999.0
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
10	9999.0	-9999.0	-9999.0	-9999.0	-9999.0

Table 1.2.4 Cal Mode Biases and Standard Deviation

I	DCMB(I)	CMB(I,1)	CMB(I,2)	SCMB(I,1)	SCMB(I,2)	AGCB(I,1)	AGCB(I,2)	SAGCB(I,1)	SAGCB(I,2)
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	9999.0	-9999.0	-9999.0	-9999.0	-9999.0	-9999.0	-9999.0	-9999.0	-9999.0
3	9999.0	-9999.0	-9999.0	-9999.0	-9999.0	-9999.0	-9999.0	-9999.0	-9999.0
4	9999.0	-9999.0	-9999.0	-9999.0	-9999.0	-9999.0	-9999.0	-9999.0	-9999.0
5	9999.0	-9999.0	-9999.0	-9999.0	-9999.0	-9999.0	-9999.0	-9999.0	-9999.0

Table 2.2a. Parameters For Which Means Only Are Calculated

N	Parameter	I1(N) WLEV1 Array Location	J1(N) WLEV2 Array Location
1	instrument attitude	136	12
2	time correction	119	23
3	altitude C.G. offset	128	24
4	altitude cal zone bias	126	25
5	altitude cal mode bias	130	26
6	AGC cal mode bias	132	27
7	altitude adapt. res. correction	146	28
8	AGC adapt. res. correction	148	29
9	MTU temperature	50	38
10	DFB temperature #1	32	39
11	TWT collector temperature	22	40

N1 = 11

Table 2.2b. Parameters For Which Means and Standard Deviations Are Calculated

N	Parameter	I2(N) WLEV1 Array Location	J2A(N) WLEV2 Array Location of Mean	J2B(N) WLEV2 Array Location of S.D.
1	AGC gate amplitude	10	30	49
2	early gate amplitude	11	31	50
3	late gate amplitude	12	32	51
4	middle gate amplitude	13	33	52
5	altitude error	7	20	46
6	noise gate amplitude	15	35	54
7	plateau gate amplitude	16	36	55
8	attitude gate amplitude	17	37	56
9	rain subcom word #1	223	58	78
10	rain subcom word #2	224	59	79
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
28	rain subcom word #20	242	77	97
29	transmit power	18	99	100
30	waveform sample #1	160	101	164
31	waveform sample #2	161	102	165
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
93	waveform sample #63	222	163	226

N2 = 93

Table 2.2c. Parameters Which Are Linearly Fit Without Computation of Standard Deviations

N	Parameter	I3(N) WLEV1 Array Location	J3(N) WLEV2 Array Location
1	latitude	151	7
2	longitude	152	8
3	ellipsoid height	137 & 138	9, 10

N3 = 3

Table 2.2d. Parameters Which Are Linearly Fit With Computation of Standard Deviations

N	Parameter	I4(N) WLEV1 Array Location	J4A(N) WLEV2 Array Location of Fit	J4B(N) WLEV2 Array Location of S.D.
1	altitude	5 & 6	17 & 18	16
2	altitude rate	4	19	45
3	SWH	8	21	47
4	AGC word	9	22	48

N4 = 4

Table 2.2e. Special Parameters To Be Compressed

N	Parameter	Input Location	WLEV2 Array Location
1	orbit number	IEXEC(3)	1
2	altimeter number	ISTAT(25)	2
3	compression interval	IEXEC(5)	4
4	compressed time	WLEV1(117&118)	5&6
5	zone flag	ISTAT(44)	14

C-3

TABLE 2.3.1 TIME LOCATION AND INDEXING FOR THE 63 SEASAT WAVEFORM SAMPLERS

Index	Time, ns	SEASAT Waveform Sample #	Index	Time, ns	SEASAT Waveform Sample #
1	-92.1875	-30	33	1.5625	+ 1
2	-89.0625	-29	34	3.1250	+ 1½
3	-85.9375	-28	35	4.6875	+ 2
4	-82.8125	-27	36	7.8125	+ 3
5	-79.6875	-26	37	10.9375	+ 4
6	-76.5625	-25	38	14.0625	+ 5
7	-73.4375	-24	39	17.1875	+ 6
8	-70.3125	-23	40	20.3125	+ 7
9	-67.1875	-22	41	23.4375	+ 8
10	-64.0625	-21	42	26.5625	+ 9
11	-60.9375	-20	43	29.6875	+10
12	-57.8125	-19	44	32.8125	+11
13	-54.6875	-18	45	35.9375	+12
14	-51.5625	-17	46	39.0625	+13
15	-48.4375	-16	47	42.1875	+14
16	-45.3125	-15	48	45.3125	+15
17	-42.1875	-14	49	48.4375	+16
18	-39.0625	-13	50	51.5625	+17
19	-35.9375	-12	51	54.6875	+18
20	-32.8125	-11	52	57.8125	+19
21	-29.6875	-10	53	60.9875	+20
22	-26.5625	- 9	54	64.0625	+21
23	-23.4375	- 8	55	67.1875	+22
24	-20.3125	- 7	56	70.3125	+23
25	-17.1875	- 6	57	73.4375	+24
26	-14.0625	- 5	58	76.5625	+25
27	-10.9375	- 4	59	79.6875	+26
28	- 7.8125	- 3	60	82.8125	+27
29	- 4.6875	- 2	61	85.9375	+28
30	- 3.1250	- 1½	62	89.0625	+29
31	- 1.5625	- 1	63	92.1875	+30
32	0.0000	0			

Table 2.5.1a. Cal Attenuator and AGC

I	AGCI(I)	CALK(I)
1	16.58	43.7
2	24.15	36.1
3	30.30	30.3
4	35.67	24.2
5	42.27	18.2
6	48.07	12.2
7	54.52	6.2
8	60.58	0.0

Table 2.5.1b. Attitude Loss vs Attitude

I	ATT(I)	AL(I)
1	0.00	0.0161
2	0.05	0.0391
3	0.10	0.1081
4	0.15	0.2231
5	0.20	0.3842
6	0.25	0.5914
7	0.30	0.8449
8	0.35	1.1445
9	0.40	1.4904
10	0.45	1.8826
11	0.50	2.3213
12	0.55	2.8066
13	0.60	3.3386
14	0.65	3.9178
15	0.70	4.5430
16	0.75	5.2158

Table 2.5.2a. Wind A and B Coefficients

I	A(I)	B(I)
1	0.080074	-0.124651
2	0.039893	-0.031996
3	0.015950	0.017215

Table 2.5.2b. Wind Polynomial Coefficients

I	C(I)
1	2.087799
2	-0.3649928
3	0.04062421
4	-1.904952×10^{-3}
5	3.288189×10^{-5}