AUTOMATIC WEATHER RADAR ECHO ASSESSMENT AND TRACKING

Robert K. Crane

Environmental Research & Technology, Inc.
696 Virginia Road
Concord, Massachusetts 01742

March 1979

Final Report
15 February 1978 - 30 September 1978

Approved for public release; distribution unlimited

AIR FORCE GEOPHYSICS LABORATORY
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
HANSCOM AFB, MASSACHUSETTS 01731

THIS DOCUMENT IS BEST QUALITY PRACTICABLE.
THE COPY FURNISHED TO DDC CONTAINED A
SIGNIFICANT NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.
DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DDC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.
Automata Weather Radar Echo Assessment and Tracking

Robert K. Crane

Environmental Research & Technology, Inc.
696 Virginia Road
Concord, Massachusetts 01742

Air Force Geophysics Laboratory
Hanscom AFB, Massachusetts 01731

Monitor A, Chmela/LYW

Approved for public release; distribution unlimited.

Previously developed algorithms for automatic radar cell detection and tracking were adapted for real-time use on the AFGL Echo Track and Significance Estimator. Additional significance estimation algorithms were developed to reduce the number of detected cells to a manageable number for display and interpretation.
TABLE OF CONTENTS

ACKNOWLEDGMENTS

1. INTRODUCTION
   1.1 Program Objectives
   1.2 Summary of Results
   1.3 Software Development
   1.4 Organization of the Report

2. BACKGROUND
   2.1 Overview of Automatic Processing Scheme
   2.2 Cell Detection
   2.3 Tracking

3. PROGRAMS FOR THE AFGL TRACKING AND SIGNIFICANCE ESTIMATOR
   3.1 Processing Options
   3.2 Cell Detection
   3.3 Tracking

4. PROGRAM STATUS AND RECOMMENDATIONS

5. REFERENCES

APPENDIX A - Operating Instructions, CSS Files, TET Files,
   Task Establishment Maps, Definition of Variables,
   List of Arrays, List of Common Blocks
   A-1

APPENDIX B - Flow Charts
   B-1

APPENDIX C - Real-Time Version
   C-1

APPENDIX D - Post-Mission Analysis Version
   D-1

APPENDIX E - Sample Output
   E-1

APPENDIX F - Options to Increase:
   Number of Significant Cells,
   Number of Active Cells
   F-1
ACKNOWLEDGMENTS

The results described in this report are the culmination of a sequence of contracts with the Air Force Geophysics Laboratory: "Parameterization of Weather Radar Data for Use in the Prediction of Storm Motion and Development", F19628-76-C-0264; "Development of Techniques for Short-Range Precipitation Forecasts", F19628-77-C-0058; and the current contract F19628-78-C-0076. Data used for the refinement of the algorithms were obtained by ERT under contracts with the Bureau of Reclamation, U.S. Department of the Interior, Contract No. 14-06-D-7673, and the Federal Aviation Administration, Amendment Agreement No. 4 to the Bureau of Reclamation contract.

Mr. A. Koscielny was the computer programmer operator for the Joint Agency Doppler Technology Tests in Norman, Oklahoma. Messieurs J. Leslie and G. Gustafson provided the programming support required to prepare the Interdata 7-32 computer programs.
1. INTRODUCTION

1.1 Program Objectives

The goal of the work reported herein is the real time operation of the cell detection and tracking algorithms previously developed by Environmental Research & Technology, Inc. (ERT) for the Air Force Geophysics Laboratory (AFGL). Specifically, the work included the following tasks: (1) encoding the automatic echo assessment and forecasting algorithms developed under Contract No. F19628-77-C-0058 on the Echo Track and Significance Estimator (ETSE) Interdata Model 7-32 Computer; (2) testing the algorithms for real time operation; (3) operating the computer and writing assembly level programs on the Interdata 7-32 computer during the Joint Agency Doppler Technology tests at the National Severe Storms Laboratory, Norman, Oklahoma, March through June 1978; and (4) refining the automatic assessment and forecasting algorithms based upon a critical radar meteorological analysis of the use of the algorithms.

The cell detection and tracking algorithms were developed to automatically process weather radar data to provide real time identification of severe weather and short range (0-20 minute) forecast of regions of potential hazard to aircraft operation.

1.2 Summary of Results

The cell detection and tracking algorithms previously developed under Air Force Contract (Crane, 1978; Crane, 1977) were designed for real time operation on a medium scale computer based on our experience with an extremely limited data sample (five consecutive azimuth scans) and with the CDC 6600 computation facility at AFGL. The subsequent development of a viable real time weather radar data processing system for use by the Air Force, however, required access to a significantly larger set of weather radar data.

The initial development of the cell detection technique had been undertaken for the Federal Aviation Administration (FAA) (Crane, 1976) for use in automatic air traffic hazard detection. Their continued interest in the processing scheme and its evaluation resulted in a contract between ERT and the FAA to process several hours of radar observations made simultaneously with aircraft penetrations. ERT was also under
contract with the Bureau of Reclamation, Department of Interior (BuRec) during the period of performance of this contract for the acquisition and analysis of significant amounts of radar data for the development of precipitation augmentation strategies for the High Plains (HIPLEX). As a part of that contract, the cell detection and tracking algorithms were installed on the CYBER-74 Computer System in Denver and used to obtain statistical data on the spatial organization of precipitation production within high plains storms.

The three concurrent programs, the development of real time techniques for the Air Force, hazard detection algorithm evaluation for the FAA, and the spatial organization analysis for BuRec provided the experience with a significantly larger data base needed for the refinement of the tracking algorithms (Task 4) and the development of a viable real time processing program.

The work under this contract was organized in the four tasks listed in Section 1.1: (1) encoding the algorithms for real time operation on the Interdata 7-32 computer; (2) testing the algorithms for real time operation; (3) operating the ETSE in Norman, Oklahoma during the Joint Agency Doppler Technology tests; and (4) refining the algorithms for improved tracking and short range forecasting. ERT provided a programmer operator for the 1978 Joint Agency Doppler Technology tests in fulfillment of Task 3. The detection and tracking algorithms refined as a result of Task 4 have been coded and installed on the Interdata 7-32 computer in fulfillment of Task 1. Their description is the subject of this report. Program listings and a copy of the operating instructions are included in Appendix C and D. The algorithms have been tested in compliance with Task 2 and, as coded and operated, performed in real time.

The algorithms operate in real-time on the Interdata 7-32 computer as required. Real-time operation on the Interdata 7-32 in the manner used in non-real-time analysis, however, requires additional programs to fetch and store the raw radar data. For real-time operation at the level of performance of the previously developed non-real-time program, a new operating system is required for the Interdata 7-32 computer which utilizes the real-time interrupt capability of the machine to run the cell detection and tracking programs as background programs with the data averaging and storage programs in foreground. Such an undertaking was
beyond the scope of this contract, and a simpler program has been specified which allows operation in real-time on alternate scans; one to fetch and store the data, the second to process the data. This program, a modification to the TSE program provided by Raytheon (Boak et al., 1977), is listed in Appendix C.

1.3 Software Development

The previously developed cell detection and tracking programs were extensively modified for use on the Interdata 7-32 computer. The tracking program described in the final report of the previous contract (Crane, 1978) was completely rewritten starting with a new set of algorithms. Experience with the larger volume of radar data available from BuRec forced the program revision. Two major problems existed with the initial tracking program; excessive computer storage requirements for the large numbers of cells encountered in practice and an inherent inability of the program logic to separately establish individual smoothed track velocities for each cell. The new tracking program develops the volume cell attributes discussed in the previous contract report, establishes the existence of cell clusters, provides an estimate of cell significance, and maintains both instantaneous and smoothed velocity estimates for each cell.

The cell detection program encoded for use on the Interdata 7-32 computer is a streamlined version of the original cell detection program. The fixed contour outlines are developed not as line segments enclosing the contoured area but as azimuth strobes within the echo region in conformity with the ETSE display scheme. Attributes are not generated for the fixed contours. The cell detection subroutine operates as before. Addressing in the arrays used in the subroutine has been extensively altered to increase operating speed.

The programs can operate over a wide range of reflectivity thresholds but should be used at a relatively high reflectivity threshold, processing only data with reflectivity values above, say, 40 dBZ. This threshold was selected to reduce the number of cells being processed in the computer. The reduction in the number of cells being processed improves operating speed and matches the output requirements. Experience with storms in the high plains indicates that severe storms produce large
numbers of cells. Cell counts for a 20 dBZ reflectivity threshold are over 150 during the active period of a storm; the total number of separate cells observed during a storm often exceeds several thousand. By way of contrast, the output requirements established by the remote display system (communicated by AFGL personnel), are for not more than 12 cells at any one time. The more than an order of magnitude reduction in active cell count can best be accomplished by increasing the threshold reflectivity value.

In addition to reflectivity threshold selection, significance is established using the integrated tangential shear of the radial velocity for each cell. Tangential shear cell detection as previously coded (Crane, 1975) is not attempted in the streamlined version of the program. The tangential shear data are used to develop a shear attribute for each cell detected on the basis of reflectivity alone. The program operates by detecting all cells that occur above the processing threshold but saving for tracking and output only the 12 cells having the highest reflectivity, integrated tangential shear product. Internally, the program processes 31 volume cells but only the 12 volume cells with the highest reflectivity, shear product are output after each volume scan. The program can be modified to process and output data for more cells by changing array sizes and test limits.

The object of the work reported in this contract was to streamline the original version of the cell detection and tracking programs for real-time use on the Interdata 7-32 computer with the operating system provided by Interdata. Many of the features of the original program, such as the generation of fixed contour attributes and the independent detection of tangential shear peaks, were removed to establish the real-time program. These features may be recovered only if the original version of the program, the program operating on the AFGL CR-6600 computer, is installed on the Interdata 7-32 for non-real-time processing.

1.4 Organization of the Report

This report considers only the software developed for use on the FTSP, documentation for Tasks 1, 2 and 4. Task 3 covered the programmer computer operator for the 1978 measurements in Norman, Oklahoma. The results of that task were reported in the quarterly reports and will not be considered further.
Background material and algorithm refinement based upon results from the FAA and BuRec programs are considered in Section 2. Section 3 documents the program for the ETSF. Section 4 summarizes program status and makes recommendations for future work. Program listings, flow charts, variable definition, and operating instructions will be found in the appendices.
2. BACKGROUND

2.1 Overview of Automatic Processing Scheme

Conventional weather radars produce large amounts of data - a significant fraction of which is highly redundant. Doppler radars produce even larger amounts of data. Significant weather events may be imbedded in the mass of redundant data. It is the goal of the automatic processing scheme to extract the relevant information from the mass of data to (1) reduce the data transmission requirements for the communication of weather data obtained from a radar, (2) to screen the data prior to display to meteorologists, (3) to preprocess the data for automatic hazard detection, and (4) to prepare the data for use in objective short range forecasting.

The processing scheme is structured to use the cell detection algorithm in on-radar-site computers to perform the bulk of the data reduction. The cell data are then communicated to regional computers (or to a second program in a stand alone radar data processor) for tracking and interpretation. For a national network of weather radars, the tracking program would accomplish the task of netting different radars and developing a single, best estimate description of the current weather for use in displays, hazard detection and warning, and short range forecast.

For this contract the cell detection and tracking algorithms are operated in a single computer; the final output is track data for the 12 most significant cells. The track data include smoothed cell velocities which are used in the tracking program for data association and may be used externally for short range forecast.

The output cell and track data are for significant features in the larger mass of radar observations. Significance is defined in an ad hoc manner using cell intensity, area, vertical development, and tangential shear data. Parameters that are intuitively associated with significant events such as severe hail, severe thunderstorm turbulence, and tornadoes have been selected for the determination of a significant cell.Operational experience with the processing algorithms and a large sample of data is required before the values of the thresholds used to establish cell significance can be refined. The current algorithm has been
partially tested using aircraft penetration data. For the measurements currently available, a positive correlation has been obtained between the location of significant cells and aircraft turbulence. An example of this association is presented in Figures 1 and 2 (output obtained from the work at ERT sponsored by the FAA). The significant cells are indicated by the tightly clustered symbols for reflectivity values greater than 40 dBZ. The time marks are 6 km apart along the aircraft track. The aircraft was within a typical cell diameter of two significant cells between 1640 and 1641, a time marked by the strongest acceleration fluctuations (turbulence) during the penetration. By way of contrast, the remainder of the penetration was quiet and did not show strong acceleration fluctuations and was not in the vicinity of significant cells.

The display in Figure 1 immediately identifies the locations of the significant cells and graphically presents the essential data contained in the radar observations. In contrast, a section of a conventional contour display and of the cell display are presented in Figure 3. The essence of the data is immediately evident in the cell display. The important 45 dBZ cell that results from the strong updraft depicted between 1640 and 1641 MDT in the aircraft data is observed in the cell display but not in the contour display. This cell produced the strongest turbulence. Even with a color display, this important region would not be evident although the higher reflectivity contours in Figure 3a would stand out more vividly, in the manner of the darkened symbols in Figure 3b.

A display such as Figure 3b is readily interpreted but is not the normal weather radar display familiar to trained radar meteorologists. The cell display provides the important details but at a scale that is smaller than used by most meteorologists and since many cells may be observed at any single time, a display of all the cells may prove to be confusing. The number of cells active at any one time with reflectivities above 15 dBZ and the number of significant cells for a set of observations of showers in Kansas (output from data processed by ERT for BuRec) are depicted in Figure 4 as a function of time and volume scan sequence. The total number of cells present within a 25 to 150 km annulus of Goodland, Kansas approached 200 during the most active part
PENETRATION #2

TIME
○ 1639:00 - 1640:30
△ 1640:30 - 1642:00
□ 1642:00 - 1643:30

INTENSITY (dBZ)
△ <40
△ 40-50
△ >50

Figure 1  Aircraft Penetration 22 July 1976 as Observed by the Grover Radar
Figure 3  Contour and Cell Data, Grover Radar Data
Figure 4  Active Cells and Total Water Production, Goodland, Kansas
10 June 1976

Water Production Rate (Km/hr)

TIME (MINUTES)

150
100
50
0

1976 KANSAS DAY 162

PRODUCTION RATE

ALL CELLS

SIGNIFICANT CELLS

NUMBER OF ACTIVE CELLS

VOLUME SCAN

200
100
0
of the storm. The number of significant cells was as high as 30. A display of all the cells would be very difficult to interpret. A display of only the significant cells, though readily interpreted by a computer, may still be difficult for a meteorologist to interpret. A further reduction in the number of significant cells is required. The program developed under this contract utilizes tangential shear data in addition to reflectivity information to further reduce the number of cells for display as the most significant cells. The utility of this algorithm for significant cell selection still needs verification.

The small cells detected and tracked by the algorithms developed under this and prior contracts are well behaved in time and space. The cells show vertical development, persist, and have average velocities that approximate a steering level wind. Summary statistics for the June 10, 1976 storm displayed in Figure 4 are presented in Figures 5 to 9. These data provide statistical summaries of several cell characteristics representing either the values averaged over the lifetime of a cell (average) or the peak value obtained during the cell lifetime (peak). Lifetime data are presented in Figure 10. The data in Figure 5 depict the statistics of the highest reflectivity values reached by a cell during its lifetime. The data are drawn from a sample of 900 cells whose lifetimes exceeded 10 minutes. These data show that reflectivity alone was not used as a criterion for significance. Over 77 cells had a peak reflectivity in excess of 50 dBZ while only 8 significant cells had a peak reflectivity in excess of 50 dBZ. In the processing used to obtain these data, significance was defined based upon the vertical structure and horizontal dimension of the cell as well as its reflectivity. A high reflectivity echo that was observed only at one elevation angle was not considered to be a significant cell.

The cells detected by the processing scheme have relatively small areas. The average cell area for all cells and for significant cells do not differ as indicated in Figures 6 and 7. The peak cell diameter is of the order of 3 km. The cells tend to be vertical structures. Statistically, the area of a cell at the lowest elevation angle does not differ from the area of the cell at the height at which it has a maximum reflectivity value. Note that the area at any height is defined by a contoured region 3 dB below its peak reflectivity value at that height. At another
Figure 5  Reflectivity Statistics, Goodland, Kansas
10 June 1976

PEAK REFLECTIVITY
1976 DAY 162

NUMBER OF OCCURRENCES

1000-

100-

10-

LARGEST CELL IN A CLUSTER

SIGNIFICANT CELLS

ALL CELLS

10 20 30 40 50 60 70

PEAK REFLECTIVITY (dBZ)
Figure 6  Cell Area Statistics, Goodland, Kansas - 10 June 1970

SURFACE PEAK Z

AVG. DIA. = 2.3 km  2.3 km
PEAK DIA. = 2.9 km  2.9 km

- AVERAGE SURFACE AREA
- PEAK SURFACE AREA
- AVERAGE PEAK Z AREA
- PEAK PEAK Z AREA
Figure 7  Significant Cell Area Statistics, Goodland, Kansas
10 June 1976

SIGNIFICANT CELLS
DAY 162  1976

SURFACE  PEAK Z

AVG. DIA.  2.3km  2.2km
PEAK DIA.  3.0km  3.0km

- AVG. SURFACE AREA
- PEAK SURFACE AREA
- AVG. PEAK Z AREA
- PEAK PEAK Z AREA

NUMBER OF OCCURRENCES

AREA (km$^2$)
Figure 8  Cell Velocity Statistics,
Goodland, Kansas - 10 June 1976

1976
DAY 162

NUMBER OF OCCURRENCES

ALL CELLS

SIGNIFICANT CELLS

EAST - AVERAGE VELOCITY (m/s)
Figure 9 Cell Velocity Statistics,
Goodland, Kansas - 10 June 1976

1976
DAY 162

NUMBER OF OCCURRENCES

1000-
100-
10-
1-

NORTH-AVG. VELOCITY (m/s)

-20 -10 0 10 20

ALL CELLS

SIGNIFICANT CELLS
Figure 10  Cell Lifetime Statistics, Goodland, Kansas - 10 June 1976

24
height, the contour used to define its area will have a different reflectivity value.

The tracking algorithms track each cell individually. The distributions of cell velocity averaged over the cell lifetime are depicted in Figures 8 and 9. These data show no significant differences between the velocities of all the cells and of significant cells. The individual track velocities may differ from the mean (or steering level wind) velocity by as much as 4 m/s (rms in each component of the wind). Observations show that the deviations are not entirely random. Larger scale convergence and divergence patterns are evident in the cell trajectories. On a smaller scale, it is evident that cells affect each others motion. A tendency has been observed for cells to follow each other along the same track even though they may not have initially developed along the track.

The cells persist for a range of lifetimes. On average, cells with reflectivities above 15 dB last for a little over 12 minutes. Significant cells last over 30 minutes on average. These results show that ideally, a full cycle of radar observations (a volume scan) should be acquired in between four and five minutes to get more than two observations of a cell during its lifetime. Practically, a longer time, six to seven minutes, is required for the processing algorithms as implemented under this contract. Processing speed can be increased by preparing a new operating system for the Interdata 7-32 computer but this was beyond the scope of the contract. The motion of significant cells, if detected early during its development, may be extrapolated for upwards of 20 minutes before they disappear indicating that short range forecasting of cell location is feasible.

2.2 Cell Detection

The cell detection algorithms have been previously defined (Crane, 1977, 1978) and will not be detailed again. Flow charts for the processing algorithm are presented in Appendix B. Briefly, a cell is a region within a contour, a fixed number of quantization steps below a local maxima that includes no other cells. For most observations, a quantization step of 1 dB and the use of contours 3 dB below a local reflectivity maxima seems to work best. The quantization step and contour threshold
were empirically established by Crane (1970) using data from Wallops Island, Virginia. The 1 dB step and 3 dB threshold produced a detection probability better than 0.95 on these scans and a false alarm rate of less than 2 per scan. By increasing the threshold, the false alarm rate was reduced but at the expense of a lowered detection probability.

Cell detection is performed for all localized reflectivity maxima that exceed a processing threshold. In the post mission processing versions of the cell detection program, attributes are obtained for all the cells detected above the lowest threshold fixed reflectivity processing contour (fixed contour or echo region). The streamlined edition of the cell detection program provides output only for the most significant cells.

2.5 Tracking

An entirely new tracking algorithm was developed during the period covered by this contract primarily for application to the post mission processing requirements of the FAA and BuRec contracts. This program was subsequently modified for real time application under this contract. The real time version generates volume cells and volume cell tracks.

The same tracking algorithm is used to generate the 3-dimensional cell from the successive azimuth scans within a volume scan sequence and to track the volume cell in time. Each new volume cell is identified using cell data from a single azimuth scan that cannot be associated with other, previously established tracks. Association is attempted with the newly detected volume cell on each successive azimuth scan until the track is terminated. A track is terminated when no new data are obtained for a volume cell during a complete volume scan cycle.

Association is established using the location of the cell on a azimuth scan as compared with the volume cell location extrapolated to the time of the azimuth scan together with the differences in the heights of the last observed data and the current data, differences in cell areas, and differences in reflectivities. A measure of the goodness of an association between a cell and track, is established for each possible track, cell combination. The final cell track pairings are those that minimize the sum of the measures (maximize the goodness of association) for all the cells and tracks that may be possibly paired during the association.
process. The set of cells and tracks that may possibly be paired are defined as a cluster.

The volume cell track is the primary entity maintained by the tracking program and successive radar observations are associated with the expected position of the cell along its track at the time of the radar observation. The tracking algorithm was developed in this manner to allow the use of data from more than one radar system since all that is needed for the association algorithm is the location and time of the cell centroid as reported by each radar together with other attributes such as reflectivity, tangential shear, etc.

Two velocities are maintained for each cell track - an instantaneous velocity, the difference in cell centroid location divided by the time interval between observations, and a smoothed velocity obtained by low pass digital filtering of the raw velocity data. The coefficients in the filter were established by trial and error using a large data sample. The initialization of the track velocity for each cell is important due to the extrapolation process used for tracking. Experiments with the tracking program show that adequate results are obtained if a zero velocity is used for the initial velocity estimate but better results can be obtained if an estimate of the steering level wind is used for the initial tracking velocity. The program automatically updates the initial velocity estimate after processing sufficient data to establish a stable estimate. The measure of success of the tracking program was taken as the rate required for the smoothed track velocities to stabilize.

The cell detection and tracking programs were initially developed to process a large number of cells, up to 512 active cell tracks at any one time and to calculate upwards of 30 attributes for each cell, cluster, and fixed contour (echo) region. Processing this amount of data is not possible in a real time environment with storage and display limitations. The basic algorithms for cell detection and tracking have been maintained. The number of tracks to be processed has been reduced by increasing the reflectivity threshold for processing and by incorporating the tangential shear information in the decision process for saving the most important 12 to 16 cells of 30 or more cells that exceed the reflectivity processing threshold. Further storage savings have been accomplished by reducing the number of attributes for each cell.
The cell detection and tracking programs were not coded in efficient manner for operation on the ETSI Interdata 7-32 computer. Extensive program revision was performed to reduce the number of subroutine calls and to revise the addressing procedure to reduce the time required to fetch or store a variable. The result is a streamlined cell detection and tracking program that will handle a reasonable number of active cells during the time required for an azimuth scan. Specifically, over 100 cells can be detected and processed in less than 5 seconds using the Interdata 7-32 programs. The processing time can be reduced further by dynamically varying the reflectivity processing threshold to maintain fewer than say 20 detected cells but this has not been necessary.

The programs to fetch the raw radar data and prepare the data for use in the cell detection program are included in Appendix C and D. The real program is designed to permit data gathering and cell detection on alternate scans. Real-time processing using the cell detection algorithms on every azimuth scan is possible with the Interdata 7-32 computer but will not occur unless considerable effort is expended to develop a new operating system for the computer tailored to use the interrupt and background/foreground processing capabilities of the computer to provide quasi-real-time cell detection and simultaneous real-time radar data acquisition, averaging and storage. Sufficient time is available for all the programs to operate on all the data from a volume scan within the time of a volume scan but the processing of the data from the lower elevation angles will lag behind data acquisition and only catch up on the higher elevation angles.
3. PROGRAMS FOR THE AGEI TRACKING
AND SIGNIFICANCE ESTIMATOR

3.1 Processing options

Two programs exist for operation on the IBM. A post mission proces-
sing program is available called CRAM that reads previously prepared
data from the disk and performs cell detection and tracking. This pro-
gram is not intended for real time operation and threshold levels may be
reduced to allow detection of a large number of cells. The program is
intended for post mission data analysis when time is not at a premium and
larger amounts of output can be handled by the user. The input data
must be prepared for storage on the disk using a modification to the IBM
programs generated by Raytheon. The program is called KRI and is listed
in Appendix B.

The operational program uses the same cell detection and tracking
subroutines but is called from a modified version of the Raytheon pro-
vided IBM programs. These programs store the data on one arithmetic scan
and process the data on the next. Data from the first, third, and fifth
scans are processed during the second, fourth, and sixth. During the
seventh scan the displays are prepared and the programs are reinitialized
for the next cycle of the seven scan sequence.

All the programs generated under this contract for use on the Inter
data '337' computer are listed in Appendix C and D. CRAM is the post mission
main program which calls CONDOR and TRACK. The output from this program
is stored on disk for subsequent listing. This program requires the use
of a preprocessing program PPRDFD (modified for IBM) written by MD
personnel and listed in Appendix D. The real time program includes
modified versions of the IBM programs supplied by Raytheon. The modified
programs are TSEMAIX and the subroutines RMAP and PPRDFD. The CONDOR
and TRACK subroutines are called from RMAP. These programs are presented
in Appendix C.

3.2 Cell Detection

The cell detection subroutines CONDOR and PLWP were substantially
modified to reduce processing time spent in addressing the data arrays.
The principal modification was to change all the arrays to single dimen
sional arrays and to explicitly perform the address calculations in the program. In this way, multiple references to the same array location would not involve the time consuming recalculation of the address for each array reference.

CONTOR was extensively modified to remove the fixed contour attribute generation algorithm. Contour data are still prepared but in the azimuth-strobe format used by the TSI display program. No radial-to-radial association is required for this streamlined version of the program significantly decreasing the length and complexity of the program.

PIAKD has been changed only to accept the modified system of addressing and to select only the 16 most significant cells for further processing. Reference to the fixed contour identification tag was also removed from PIAKD since the tags were produced in the association, attribute generation logic of the CONTOR subroutine which was removed for the Interdata version of the program.

4 Tracking

The new tracking program consists of the subroutine TRACK which calls COMPARE to perform the cell to track associations. The subroutine COMPARE searches the track list and the cell list from the last scan, and finds all possible pairs for which the goodness of fit measure does not exceed a preselected threshold. The measure is given by

\[
M = L \times \frac{[X_c - X_t]^2 + [Y_c - Y_t]^2 + [W_c - W_t]^2 + [H_c - H_t]^2 + W_t^2]}{[W_c]^2 + [W_t]^2 + W_t^2}
\]

where \( M \) is the measure, the subscripts \( c \) refer to cell and \( t \) to track and \( X \) is reflectivity, \( Y \) area, \( H \) height and \( W \) are centroid locations. The weights \( W \) were set by trial and error. The current values are listed in Table 1. The best cell-track pairing has the lowest measure; pairings with a measure greater than \( M \) are not allowed.

Several pairings are possible in a cluster of cells. Subroutine RESORT selects the best set of cell-track pairs in a cluster. The attributes are updated in ATRAK which is called from COMPARE if there is no cell cluster or from RESORT if a cluster exists. The subroutine ATRAK
is also called to store cell data in the VR array each time a track is updated. This information is used in calculating the measure M. The VR array data are either from the lowest elevation angle on which the track was observed or from the last elevation angle at which it was observed. The measure used to evaluate the cell, track pairing is the minimum measure obtained using either the last or lowest elevation angle data.

At the end of a volume scan cycle, STRAK is called to calculate the attributes and to output the track data. Only 12 tracks are output from STRAK in the operational version although a maximum of 32 tracks are maintained at any one time. The list of the 16 attributes maintained for each track is given in Table 2.

<table>
<thead>
<tr>
<th>TABLE 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRACKING, WEIGHTS, MAXIMUM MEASURE AND VELOCITY FILTER</td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$W_X$</td>
<td>-</td>
<td>$1/10$</td>
</tr>
<tr>
<td>$W_A$</td>
<td>-</td>
<td>$1/25$</td>
</tr>
<tr>
<td>$W_B$</td>
<td>-</td>
<td>$1/2$</td>
</tr>
<tr>
<td>$W_L$</td>
<td>-</td>
<td>$1/5$</td>
</tr>
</tbody>
</table>

$M_p$ = 5.

Velocity Filter:

$$(V)_N = a \frac{Ax}{At} + b (V)_N + c V_N$$

$$(V)_E = a \frac{Ay}{At} + b (V)_E + c V_E$$

$V =$ velocity; $N,E$ refer to Northward and Eastward ($v,x$) components

$Ax,Ay =$ change in position between volume scans

$At =$ time between scans

$\bar{V} =$ average velocity, all cells

$a = .3$  $b = .3$  $c = .3$
## TABLE 2

**TRACK ATTRIBUTES**

<table>
<thead>
<tr>
<th>No.</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Observation Time</td>
<td>(seconds from start of year)</td>
</tr>
<tr>
<td>2.</td>
<td>East Location</td>
<td>(km)</td>
</tr>
<tr>
<td>3.</td>
<td>North Location</td>
<td>(km)</td>
</tr>
<tr>
<td>4.</td>
<td>Average Reflectivity</td>
<td>(dBZ)</td>
</tr>
<tr>
<td>5.</td>
<td>Volume</td>
<td>(km³)</td>
</tr>
<tr>
<td>6.</td>
<td>Peak Reflectivity</td>
<td>(dBZ)</td>
</tr>
<tr>
<td>7.</td>
<td>Height of Reflectivity Peak</td>
<td>(km)</td>
</tr>
<tr>
<td>8.</td>
<td>Reflectivity at Lowest Elevation Angle</td>
<td>(dBZ)</td>
</tr>
<tr>
<td>9.</td>
<td>Area of Cell on Lowest Elevation Angle</td>
<td>(km²)</td>
</tr>
<tr>
<td>10.</td>
<td>Height of Cell at Lowest Elevation Angle</td>
<td>(km)</td>
</tr>
<tr>
<td>11.</td>
<td>Reflectivity at Cell Top</td>
<td>(dBZ)</td>
</tr>
<tr>
<td>12.</td>
<td>Height of Cell Top</td>
<td>(km)</td>
</tr>
<tr>
<td>13.</td>
<td>Track Identifier</td>
<td></td>
</tr>
<tr>
<td>14.</td>
<td>Smoothed Track Velocity - East</td>
<td>(m/s)</td>
</tr>
<tr>
<td>15.</td>
<td>Smoothed Track Velocity - North</td>
<td>(m/s)</td>
</tr>
<tr>
<td>16.</td>
<td>Integrated Tangential Shear</td>
<td>(m/s/km)</td>
</tr>
</tbody>
</table>
4. PROGRAM STATUS AND RECOMMENDATIONS

The processing programs are operational on the Interdata 7-32 computer. Experience must now be gained in using the system for the observation of weather. A number of parameters (Table 1) were set in the program on the basis of our experience with the data obtained from our contracts with the FAA and BuRec. It is anticipated that a different radar system operating in a different environment may need a different set of parameters. These parameters, such as the tracking measure weights and the constants in the velocity smoothing filter, are readily changed in the program. Experience with a larger data set is required to obtain the best estimate values for the parameters.

It is recommended that the cell detection and tracking programs be used on available data to develop the required operational experience to adequately use the new displays. The new output attributes are in a form that may be readily adapted to objective warning and forecasting systems.
REFERENCES


APPENDIX A

Operating Instructions*
CSS Files*
TET Files*
Task Establishment Maps*
Definition of Variables
List of Arrays
List of Common Blocks

*Listings for Real-Time and Post-Mission Programs
REAL-TIME PROGRAM OPERATING INSTRUCTIONS

TO EXECUTE REAL-TIME CELL DETECTION AND TRACKING PROGRAM (ERT)

1. Compile each subroutine file:
   DCOMPILE ERT1:File Name*

2. Create an object file for each of the compiled subroutine files:
   DCALOBJ ERT1:File Name*

3. Delete old task file:
   DE ERT1:CRANE.TSK

4. Create a new task file:
   ESTAB ERT1:GARY

5. Dispose I/O devices by editing the CSS file:
   EDITR CREAL.CSS

6. Execute the program:
   CREAL

*List of subroutine file names:

TSEMAIN  ATRAK
TSEDATA  BTRAK
REALTM   COMPAR
INPARM   RESOLV
CONTOR   COMBIN
PEAKD    STRAK
TRACK
**Program Segments:**

<table>
<thead>
<tr>
<th>SEG</th>
<th>TYPE</th>
<th>NAME</th>
<th>SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>IMPU</td>
<td></td>
<td>143.75K</td>
</tr>
<tr>
<td>3</td>
<td>TCOM</td>
<td>2STORE TCM</td>
<td>4.25K</td>
</tr>
<tr>
<td>5</td>
<td>TCOM</td>
<td>EXTRA TCM</td>
<td>1.25K</td>
</tr>
<tr>
<td>15</td>
<td>RTL</td>
<td>TSERTL RTL</td>
<td>7.75K</td>
</tr>
</tbody>
</table>

**Program Labels:**

```
000100 TSEMAIN 000620 TSEDATA 003836 ETRMAP 010A80 ICLOCK
01BB70 COS 010CB0 SIN 010DF0 I 010EE8 R
011000 R 011110 SORT 011210 ALOG10 011268 ALOG
011438 EXP 011588 FLOAT 011588 IFIX 0115F8 MOD2
011640 ABS 011678 IABS2 011698 H 011708 S
011798 .01 0117A0 .0 011D48 .MES
```

**Task Entry-Points:**

```
000664 PPRDSC 00367C RMAP 004204 CONTOR 0049A4 PEAKD
009BEC TRACK 0089A4 ATRAK 00897C BTRAK 00A8F4 COMPAR
008FAC RESOLV 00EFCC STRAK 010764 COMBIN 010882 ICLOCK
010872 COS 01088A COS 010CBA SIN 010CD2 SIN
010DFA .I 010EFA .A 011002 .R 011112 SORT
01112A SORT 011212 ALOG10 01122A ALOG10 0112A ALOG
0112F2 ALOG 01131A LOGRT2 011432 EXP 0114A8 EXP
01158A FLOAT 0115AA IFIX 0115FA MOD2 011642 ABS
01167A IABS2 01169A H 01170A S 0117A2 O
0117B2 .01 01192E .0I 011D4A MES
```

**Local Common Blocks:**

```
011D68 CALB 011D78 MUSIG 011D80 READZ 011D88 SECTOR
011D90 REFL 0129B0 CAL 0129C0 RUNSUM 0139D0 TLIS
0139E8 CALR 016208 SWITCH 020098 AZ2 0200B0 AZM
0200C8 PNTRS 0200D8 INTL 0200E8 ZLOOK 020258 ECONST
020260 MAPPAR 020278 CNT 020288 DATA1 020610 DATA2
021198 DATR3 0214A0 NVLIS 0214B0 FILTER 0214C0 KTA
0214C8 CDRAYS 0222D0 CONST 022300 VPARM 022308 DVAL
022310 CNTRY 022320 FLGS 022330 TMX 022338 PHDRK
0223E8 FIXED 0223F8 PRSTOR 023AE8 THRESH 023AE8 CONPK
023AF0 RSLY 023C78 COMB
```

**Library Entries:**

```
0F0002 .U 0F0052 .V 0F006A .P 0F00FA .O
0F0182 .OR 0F02CB .DH 0F17E2 CONMSG 0F186A FLOAT2
```

**Task Common Blocks:**

```
030000 2STORE 050000 EXTRA
```

END
POST-MISSION PROGRAM OPERATING INSTRUCTIONS

TO EXECUTE POST-MISSION CELL DETECTION AND TRACKING PROGRAM (CRANE)

1. Compile each subroutine file:
   DCOMPILE ERT1:File Name*

2. Create an object file for each of the compiled subroutine files:
   DCALORJ ERT1:File Name*

3. Delete old task file:
   DE ERT1:CRANE.TSK

4. Create a new task file:
   ESTAB ERT1:CRANE

5. Dispose I/O devices by editing the CSS file:
   EDITR CRANE,CSS

6. Execute the program:
   CRANE

*List of subroutine file names:

<table>
<thead>
<tr>
<th>CRANE</th>
<th>RESOLVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>INPARM</td>
<td>COMBIN</td>
</tr>
<tr>
<td>CONTOR</td>
<td>STRAK</td>
</tr>
<tr>
<td>PFAKD</td>
<td></td>
</tr>
<tr>
<td>TRACK</td>
<td></td>
</tr>
<tr>
<td>ATRAK</td>
<td></td>
</tr>
<tr>
<td>BTRAK</td>
<td></td>
</tr>
<tr>
<td>COMPARE</td>
<td></td>
</tr>
</tbody>
</table>
POST-MISSION CSS FILE

**Listing of the CRANF.CSS File**

$JOB
SF PA 1 140,2 0.3 0
T 11B,FRT1;TS,FRT1,RT1
T CRANF,FRT1;CRANF
T CRANF
AS2,NULL:
AS3,PR:
AS4,NULL:
AS5,FRT1:User Defined Output File*
AS6,FRT1:User Defined Calibration File**
AS7,FRT1:User Defined Input Data File***
ST
$TERMJOB
$EXIT

*File to which program output is to be sent

**Disk file containing values for the parameters: IOUT, PRB, MAV, MAXS, SLOPE, OLDATA

***Disk file generated by program "FRT"
TET/32 R02-03
JOB ERT
REMOTE
ES TASK
MXXSPACE 2000
OPTIONS F
GET 400
PRIORITY 10.10
TCOM TASK/2/RW
TCOM ZSTORE/3/RW
TCOM DONE/4/RW
TCOM EXTRA/5/RW
IN ERT1:CRANE
IN ERT1:INPARM
IN ERT1:CONTOR
IN ERT1:PEAKD
IN ERT1:TRACK
IN ERT1:ATRAK
IN ERT1:ASTRAK
IN ERT1:COMPAR
IN ERT1:RESOLV
IN ERT1:STRAK
IN ERT1:COMBIN
RESOLVE ERT1:TSERTL
EDIT FVRL
BU TASK, ERT1:CRANE
MAP
OS12MT TASK-ESTABLISHMENT LOAD MAP

DATE 07/19/79  TIME 13:10:27

JOB:ERT
**** CTOP=01E5FE  UTOP=01E1F8  MIN PARTITION= 121 50K  ****

PROGRAM SEGMENTS:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SEG</td>
<td>TYPE</td>
<td>NAME</td>
</tr>
<tr>
<td>0</td>
<td>IMPU</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>TCOM ZSTORE TCM</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>TCOM EXTRA TCM</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>RTL TSERTL RTL</td>
<td></td>
</tr>
</tbody>
</table>

PROGRAM LABELS:

| 000100 | ETRMAP | 00E950 COS | 00E998 SIN | 00EAD0 1 |
| 00EBC8 .A | 00EC0 R | 00EDF0 SORT | 00EF0 ALOG10 |
| 00EFBA ALOG | 00F110 EXP | 00F268 FLOAT | 00F288 IFIX |
| 00F2DA ABS | 00F310 IABS2 | 00F330 .S | 00F3C0 0 |
| 00F968 .MES |

TASK ENTRY-POINTS:

| 001FD0 | CONTOR | 002774 PEAKD | 0079BE TRACK | 007E74 BTRAK |
| 00646C BTRAK | 008BC4 COMPAR | 009D7C RESOLY | 00CD9C STRAK |
| 00E983 COMBIN | 00E952 COS | 00E86A COS | 00E99A SIN |
| 00E9B2 SIN | 00EADA .A | 00EBCA .A | 00EC0 R |
| 00EDF2 SORT | 00EE0A SORT | 00EF0 ALOG10 | 00EF0 ALOG10 |
| 00EFBA ALOG | 00EFDA ALOG | 00EFFA LOGRT2 | 00F112 EXP |
| 00F12A EXP | 00F26A FLOAT | 00F28A IFIX | 00F2DA ABS |
| 00F312 IABS2 | 00F332 .S | 00F3C2 0 | 00F3D2 01 |
| 00F54E 01 | 00F96A .MES |

LOCAL COMMON BLOCKS:

| 00F988 TLIS | 00F9A0 CONST | 00F9D0 SWITCH | 019860 A72 |
| 019878 AZM | 019890 REFL | 01A400 PNTR5 | 01A4C0 INTL |
| 01A400 ZLOOK | 01A440 ECONST | 01A648 MAPPAR | 01A660 CNT |
| 01A670 DATAL | 01A9F8 DATA2 | 01B580 DATA3 | 01B888 NVLIS |
| 01B988 FILTER | 01B9AB KTA | 01B880 CDRAVS | 01C6B6 HPARM |
| 01C6C0 DVAL | 01C6CB CNTS | 01C6D8 FLGS | 01C6L6 TRAX |
| 01C6F0 PWORK | 01C700 FIXED | 01C708 P3RSTOR | 01DE98 THRESH |
| 01DEA0 CONPK | 01DEB0 RSLV | 01E030 COMB |

LIBRARY ENTRIES:

| 0F0002 .U | 0F0052 .V | 0F006A F | 0F00FA 0 |
| 0F0182 OR | 0F02CB 0H | 0F186A FLOAT2 |

TASK COMMON BLOCKS:

| 030000 ZSTORE | 050000 EXTRA |
| END |
VARIABLE LIST

CRANE & REALISM
*TL1 - lower threshold (dBZ)
*TL2 - higher threshold (dBZ)
*RQUANT - threshold quantization factor (dBZ)
*IL - threshold - intermediate values in dBZ conversion
*HFOFF - reflectivity offset to insure positive values
*PB - calibration constant
*BITVEL - constant used in calculation of velocity
*BITVAR - constant used in calculation of variance
*ESWORD - square of variance
*ELXVAT - raw elevation (encoder units)
*ELXAA - elevation in degrees
*ELSN - elevation in degrees
*T - time (seconds from start of year)
*AC - raw azimuth (deg)
*AFT - azimuth in degrees
*A - azimuth in radians
*ACHK - azimuth in deg. + 360
*K - radial counter
*B3NA - start azimuth (deg)
*ENDA - stop azimuth (deg)
*S1NA - sin(A)
*COSA - cos(A)
*NA - check on first azimuth of new scan
*SAC - offset multiplier for arrays in CRONT & PEAKD
*DAY - DAY - Julian date data collected
*HR - hour
*MIN - minute
*SEC - second
*DELTACL - azimuth increment (radians)
*PHI - elevation angle (radians)
*COSPHI - cos(PHI)
*SINPHI - sin(PHI)
*EARTH - 6.4857E+5 (km⁻¹)
*COSPHI*EARTH
*AIZLAST - azimuth of previous radial (deg)

* - INTEGER*2

* - REAL.
*IFLAG - intermediate print flag
*TLS - lower threshold (TL1 in CRANE)
*TATRMN - test on area
*NEMC - array addressing offset
*NCEL - cell counter
*NEMAX, *JMAX - array limits
*NEM1 - address variable
*NCL - maximum number of positions
*TL1 - lower threshold
*HEM - event counter on first threshold
*HEM2 - event counter on second threshold
*IPB - peak start location
*IP - peak counter
*EVENT - event number

* - INTEGER*2

* - REAL
PEAKD

+NBADR,+NCADR,+NBKA,+NCKA - address variables
+NAX,+NA - radial counter
+LM - number of variables in UP (Array)
+LMDP,+NAN,+NAN1,+LAM,+IDX,+NCLM,+LDBM,+LDX,+NPDP,+ID2 - address variables
+FQUANT - threshold quantization factor
+KOFST,+LIMT,+NIDP,+MXTR,+KMAX - Array limits
+IE - event number
+IEM - number of events on radial
+IES,+K1E,+KIEM - address variables on IEM
+ICEST - event start position
+ICESP - event end position
+JEB,+KA,+KB - address variable on event and peak
+IP - peak start location
+IP - peak stop location
+NTHRRES - threshold counter for peak
+LDB - dB below peak value used to define peak
+IR1 - range to peak
+IR - dBZ value at IR1
+IT - dBZ above threshold at IR1 for LDB thresholds
+.JMIXM - limit on IT
+KA - address variable on peak thresholds
+KPT - number of thresholds associated with peak
+.JR - limit on KPT
+IBGN - first position in event
+IND - last position in event
+III - reflectivity values within event
+IKA - address variable on peak threshold values within event
+.IMXJX - limit on number of contours per radial
+IREG - start or stop range of contour
+IPE - contour counter
+IADDR & +IEQL - address variables on contour threshold and number
+KC,+KA,+KZ - address variables on contour thresholds
+TCVL,+TCVM,+TCVLB - threshold values on a peak
+NPC - number of contours by threshold on this radial
+NPL - contour counter by threshold
+NHBM - start range of contour
+TIB - TIBM + 1
+THD - stop range of contour
+K2,KY,KZ - address variables on next threshold
+NPE - number of contours, this threshold, on radial
+NPL - contour counter, this threshold
+NPC1L - ID number for possible cell, this threshold
+TATC - ID TCTR(NPCEL) points to
+IE1I - first event previous radial
+IE2L - last event previous radial
+JEM - address variable on events, previous radial
+IPB - number of peak thresholds
+KB,+KKB,+KKB,B,KRC - address variables on contour thresholds
+TRVL - threshold value on a peak
+NP2 - number of contours on a threshold, previous radial
+NPL - contour counter, by threshold, previous radial
* - INTEGER*2 * - RFAI
PEAKD (continued)

+LPCEL - ID number for possible cell, previous radial
+TEQL - ID TATR(LPCEL) points to
+NPC - equals NPCEL if associated
+JEQL - next higher threshold
+JN1, +JN2, +JN3, +JN4, +JN5 - address variable on peak
+IST - start range of contour
+ISP - stop range of contour
*R - area per azimuth degree at peak range
+II - reflectivity at that range
+RU - reflectivity weighted area
*SAZ - sin(azimuth)
*CAZ - cos(azimuth)
*KXN, *KN - address variable on NPCEL
+IMDX - address variable on second threshold of peak
+IND, +INDX, +LNX - address variable on LPCEL.
+IN, +HEQL - address variable on LPCEL
+JEQL - area address of NPCEL pointed to by area of LPCEL
+IPTT - number of peak thresholds
+KTI, +KTA, +KTB - address variables on contour threshold
+NPCT - number of contours, this threshold
+IEQL - threshold value
+INDXT - address variable on NPCEL
+NIMN - number of possible cells tested so far
+NIDP - limit on number of possible cells
+IE - event number
+IPT - number of peak thresholds in event
+NPC - number of thresholds this peak
+NCVM - threshold value on peak
+NPL - contour counter

* - INTEGER*2

* - REAL
PEAKD (continued)

*I - each position out the radial
+IA - address variable on I
+IEQL - ID value at each position
+J - event number on previous radial
+JA - address variable on J
+IPB - number of peak thresholds in event
+KA, +KAP, +KAM - address variables on event and threshold
+NP - number of contours, this threshold, on previous radial
+ITERM - code for eliminating possible cell
+MG - address in UP (array) for measure of cell significance
+LMT - limit on number of cells to be carried in order of significance
+JK1 - address variable on previous radial
+JKL - address variable on current radial

* - INTEGER*2
*VKM - \( \cos(\text{elevation}) \times \text{conversion m to km} \)
*SAVM4 - unit area convert from \( \text{m}^2 \) to \( \text{km}^2 \)
+MA - address in ECL (array) of measure of cell significance
+M - cell counter
+M1, +M2, +M3, +M4, +M5, +M6, +M7 - address of each attribute in ECL (array) by cell
+KOST, +NAN2 - array addressing offset
*VKME - convert reflectivity weighted line of sight distance to horizontal dist.
*FNSN, +NSCAN - scan counters
+KTL - time
+JDAY, +IDAY - Julian date
+JHR, +IHR
+JMIN, +IMIN - start time of first scan in sequence
+JSEC, +ISEC
+NC - cell number (current scan)
+NC, +NCBG, +NCB - NC to pass through common
+NVM1, +NVMX, +NCMX - number of significant cells detected this scan
+IELSN - current elevation angle
+IESNL - elevation angle last scan

ATRAK

+NCEC - address offset on NC
+NVC - address offset on NV
+NC - cell counter for ECL (array)
+NV - cell counter for VCL (array)
+NCA - address variable on NC
+NVA - address variable on NV
+IZ - cell reflectivity
*X - cell position east [(-)west]
*Y - cell position north [(-)south]
*H - cell height
+IZL - offsetted reflectivity
*Z - reflectivity
*HL - height of last cell
*ZP - peak reflectivity

BTRAK

+NCEC - address offset on NC
+NVR - address offset on NV
+NC - cell counter for ECL (array)
+NV - cell counter for VR (array)
+NVA - address variable on NV
+NCA - address variable on NC
+ - INTEGER*2
* - REAL

A-15
+NCMX - number of cells detected this scan
+JM, +JM - array limits
+NC - cell numbers detected this scan
+NCEC - address offset on NC
+(NC1 to NC6) - address variables on NC
+NV - cell numbers tracked from previous scans
+NVVC - address offset on NV for VCL (array)
+NVVR - address offset on NV for VR (array)
+NLR - address variable on NV

A TEST - estimate of cell NV's movement from last scan to this scan

DELT - time since last scan

DELX, DELY - distance on X & Y coordinates between cell NV and NC

DELVW - a measure of the association between NV and NC using reflectivity, location, area, height

1O - overflow of D and ID (arrays)

NSCAN - scan number

1O - overflow of IHV (arrays)

DX - DELW of a previous association with this NC cell

NV - NV previously associated with this NC cell

NC - NC previously associated with this NV cell

DX - DELW of a previous association with this NV cell

VMX - number of active cell tracks

NCR - NC cell to test NV cells against

NVR - NV previously associated

NVR - NC previously associated

HCT - update height

+ - INTEGER*2

* - REAL
RESO
STRAK

*NVA - address offset on NV
*(NV1 to NV41) - address variables on NV for VCL (array)
*NVR - address offset on NV
*(NR41 to NR6) - address variables on NV for VR (array)
*NV - cell counter
*VXT - velocity east [(-)west]
*VYT - velocity north [(-)south]
*DELTN - time since last scan
*I2VAL - reflectivity
*IDCT - percent of scans cell was detected
*VXC - sum of eastward velocity components
*VYC - sum of northward velocity components
*NFN - number of scans processed
*NSN - number of velocity values summed in VXC and VYC
*VX - average eastward velocity of all cells
*VY - average northward velocity of all cells
*NVSCN - number of volume scans
*KTL - time of last scan

* - INTEGER*2

* - RFAL
LIST OF ARRAYS - THEIR SIZE AND CONTENT

In CRANE

ANC(1028) - header information - INTEGER*4
ZEE(1024) - raw data - INTEGER*4
IREF(1024) - reflectivity - INTEGER*2
IVEL(1024) - radial velocity - INTEGER*2
IVAR(1024) - variance - INTEGER*2
RE(1025) - decoded reflectivity - INTEGER*2
IVELL(256) - decoded velocity - INTEGER*2
IDVEL(256) - tangential shear - INTEGER*2
ZARY(91) - convert reflectivity to dBZ - REAL

In CONTOR

ICL(44)* - start position of event - INTEGER*2
IC2(44)* - stop position of event - INTEGER*2
IDC(22) - number of peaks in each event - INTEGER*2
IPRNG(34) - location of peaks - INTEGER*2
IC21(22) - start position of event on second threshold - INTEGER*2
IC22(22) - stop position of event on second threshold - INTEGER*2

*array contains indicated parameter(s) on the current radial, offsetted from the same parameter(s) on the previous radial
In PEAKD

T(80) - all possible thresholds a peak may have - INTEGER*2
TC(1980)* - thresh of each peak - INTEGER*2
IPTC(44)* - number of thresholds in each event - INTEGER*2
IPCNT(1980)* - contour counter - INTEGER*2
IPCI(5400)* - start range of contour segment - INTEGER*2
IPC2(5400)* - end range of contour segment - INTEGER*2
IPC3(5400)* - number of peaks within the segment - INTEGER*2

TATR(1400) - temporary attribute array - stores peak attributes until a
cell is detected or peak discarded - REAL
IACT(70) - overflow (too many peaks) - INTEGER*2

UP(6) - cell attributes - REAL
  1 - area
  2 - reflectivity
  3 - location in km east of radar
  4 - location in km north of radar
  5 - tangential shear
  6 - a measure of relating cell significance = const * area + tan shear

*array contains indicated parameter(s) on the current radial, offsetted
from the same parameter(s) on the previous radial
### PEAKD & TRACK: ECL (7x16x2) \( (I,N,NAN) \)

<table>
<thead>
<tr>
<th>( I )</th>
<th>( 1 )</th>
<th>( 2 )</th>
<th>( 3 )</th>
<th>( 4 )</th>
<th>( 5 )</th>
<th>( 6 )</th>
<th>( 7 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>variable</td>
<td>area surf</td>
<td>dBZ</td>
<td>east</td>
<td>north</td>
<td>range</td>
<td>height</td>
<td>tang shear</td>
</tr>
<tr>
<td>real</td>
<td>real</td>
<td>real</td>
<td>real</td>
<td>real</td>
<td>real</td>
<td>real</td>
<td>real</td>
</tr>
</tbody>
</table>

Defined in TRACK.

### ATRAK: VCL (23x5x2) \( (I,N) \)

| \( I \) | \( 1 \) | \( 2 \) | \( 3 \) | \( 4 \) | \( 5 \) | \( 6 \) | \( 7 \) | \( 8 \) | \( 9 \) | \( 10 \) | \( 11 \) | \( 12 \) | \( 13 \) | \( 14 \) | \( 15 \) | \( 16 \) | \( 17 \) | \( 18 \) | \( 19 \) | \( 20 \) | \( 21 \) | \( 22 \) | \( 23 \) | \( 24 \) | \( 25 \) |
| variable | east \( (X) \) | north \( (Y) \) | dBZ | area surf | time | height | range | NT; track ID | \( N \) # scans each det. | Y | Z | ZZ | ZXY | ZYX | \( H = H-L \times \) Area | \( H \) summit \( (H-T \) last) | dBZ peak | peak | \( X \) | \( Y \) | time | Vel x | Vel y | \( \Sigma \) tang shear |
| real | real | real | int*4 | real | int*4 | real | real | int*4 | int*4 | real | real | real | real | real | real | int*4 | int*4 | real | real | int*4 | real | real | real |

Set in ATRAK. Def in CRANE. Lowest elevation angle only. Updated each scan. Updated each vol scan.

Value updated each scan.
<table>
<thead>
<tr>
<th></th>
<th>variable</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>time</td>
<td>int*4</td>
</tr>
<tr>
<td>2</td>
<td>$X$</td>
<td>real</td>
</tr>
<tr>
<td>3</td>
<td>$Y$</td>
<td>real</td>
</tr>
<tr>
<td>4</td>
<td>$Z$</td>
<td>real</td>
</tr>
<tr>
<td>5</td>
<td>$(H-H_0)\times Area$</td>
<td>real</td>
</tr>
<tr>
<td>6</td>
<td>(not used)*</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>$\Delta \Sigma$ peak</td>
<td>int*4</td>
</tr>
<tr>
<td>8</td>
<td>$H$ peak</td>
<td>real</td>
</tr>
<tr>
<td>9</td>
<td>(not used)</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>$\Delta \Sigma$</td>
<td>int*4</td>
</tr>
<tr>
<td>11</td>
<td>area surf</td>
<td>real</td>
</tr>
<tr>
<td>12</td>
<td>height</td>
<td>real</td>
</tr>
<tr>
<td>13</td>
<td>(not used)</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>(not used)</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>$\Delta \Sigma$ summit</td>
<td>int*4</td>
</tr>
<tr>
<td>16</td>
<td>$H$ summit</td>
<td>real</td>
</tr>
<tr>
<td>17</td>
<td>$\text{ABS}(NT) \times \text{TRAKID}$</td>
<td>int*4</td>
</tr>
<tr>
<td>18</td>
<td>(not used)</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>(not used)</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>$\Delta X/AT$</td>
<td>real</td>
</tr>
<tr>
<td>21</td>
<td>$\Delta Y/AT$</td>
<td>real</td>
</tr>
</tbody>
</table>

*Array locations not used in the current version of the program
track cells from previous scan (NV) to cells in current scan (NC)

1UC1(16) - points to NV each NC is associated with - INTEGER*2
  i.e. 1VC1(NC) = NV

1C(16) - stores the "measure" of the above association - INTEGER*2

1UC2(16) - if one NC is associated with more than one NV then 1VC2(NC) ≠ 0 - INTEGER*2

1UV1(32) - points to NC each NV is associated with - INTEGER*2

1V(32) - stores the "measure" of the above association - INTEGER*2

1UV2(32) - if one NV is associated with more than one NC then 1UV2(NV) ≠ 0 - INTEGER*2

1C(32x10) - if 1UC2(NC) ≠ 0 - INTEGER*2
  IC(7,1) = number of conflicts
  IC(7,2-10) = the NV's associated with

C(32x9) - stores the measures of each NC-NV association - REAL

ID(32x10) - same as the IC(array) for conflicts on NV - INTEGER*2

D(32x9) - stores the measures of each NV-NC association - REAL

In RESOLVE

IV(32,7) - ordered IC(array) or ID(array) - INTEGER*2

V(384) - stores the measures of the tested associations - REAL
## List of Common Blocks and Their Associated Routines

<table>
<thead>
<tr>
<th>Block Name/Size</th>
<th>CRANE DATA</th>
<th>CONTOR PEAKD TRACK</th>
<th>ATRAK Btrak</th>
<th>COMPAR RESOLV</th>
<th>STRAK</th>
<th>STRAK COMBIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>AZM (4.5)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AZ2 (6)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>CDRAYS (897)</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CLST (257)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CNT (3)</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>CNTRST (3)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>CONPIC (11)</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>CONST (12)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>DATA1 (226)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>DATA2 (737)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>DATA3 (193)</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>DVAL (1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ECONST (2)</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>EXTRA (597.5)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>FILTER (3)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FIXED (2.5)</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FLGS (3.5)</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>INTL (3)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>KTA (2)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>KTB (1)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAPPAR (4.5)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NVL1S (41)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PNTNS (4)</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PRSTOR (1465)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PWORK (43.5)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REFL (775)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RSLV (72.5)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>SWITCH (1047.5)</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>THRESH (.5)</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TLIS (5)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>TMAX (.5)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>VPARM (2)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>ZLOOK (91.5)</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>ZSTORE (1028)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

*Size is number of 32 bit words
Program CRANE
Initialize Tables, Read And Decode Radar Data

Subroutine CONTOR
Find Fixed Contours & Calculate Attributes

Subroutine TRACK
Initialize Track Tables

Subroutine COMPAR
Associate Current To Previous Scan

Subroutine STRAK
Output Track Attributes

Subroutine PEAKD
Find Peak Referenced Contours And Calculate Attributes

Subroutine RESOLV
Resolve Conflicts In Compar

Subroutine ATRAK
Define Lowest Elevation Attributes
Subroutine BTRAK
Define Last Elevation Attributes

END

COMPUTER PROGRAM STRUCTURE
Loop on Volume Scan

Loop on Azimuth Scan (Fixed Elevation)

Loop on Azimuth (Single Radial)

Read & Scale

Reformat Data to Reflectivity, Velocity & Velocity Variance Estimates

Fixed Contour

Detect Larger Echo Regions Using Fixed Threshold Contours

Detect Cells

Detect Cells Using Variable Threshold Contours

Sum Attributes

Accumulated Intermediate Results to Calculate Attributes

Associate Cells & Echoes

Determine Height Dependences of Cells & Echoes by Identifying the Same Cells on Successive Azimuth Scans

Associate Cells & Echoes

Determine Cell Tracks by Identifying the Same Cell in Successive Volume Scans

Output Cell Track Data

Overall Processing Scheme
REAL TIME PROCESSING SCHEME

START

Loop On
Volume Scan

Loop On
Odd Numbered Scans

Loop On
Azimuth

PPRDSG
Input Radar Data
By Azimuth
Decode Radar Data
By Azimuth
Output Decoded Data

End of scan

No

RMAP
Process This Scan
While New Scan
Being Made

Yes

Next Scan
Last Scan in The Sequence

No

TSEMAIN
Display Data
Initialize For Next Cycle

Yes
Event Identification
Subroutine CONTOR
Peak Detection
Subroutine PLAN

START
Loop on Peaks within Event (Lowest Threshold)
Calculate Thresholds LDB Units Below Peak Value
Loop on Range within Event
Loop on Threshold
Contour to Identify Segments at Each Threshold
Loop on Threshold
Loop on Segment
Loop on Segment next Higher Threshold C Radial
Associate Nextset C Radial Segments
Loop on Threshold B Radial
Loop on Segments B Radial
Associate B to C Radial
Detect Cell if not Updated the Radial and C Radial values are lower than Cell Threshold
STOP
Event Association

Subroutine PEAAM

START

Loop on Events this Radial (C - Radial)

Loop on Events Prior Radial (B - Radial)

Stop Range, B < Start, C

Start, B > Stop, C

Any Prior B Radial Association

Yes

No

Combine B & C

Combine B with Prior B & C

Any B Radial Association

New End Region Initialize Attributes

STOP
INPUT AFTER EACH SCAN

Loop on cell this scan

Loop on track

Compare

Yes

Store association in UV, UC arrays

No

Loop on all

Single

Compare

No

Find all cells and tracks in cluster

Yes

Update Attributes

End of volume scan

Yes

Output Track Attributes

No

Exit

CELL TRACKING SCHEME
APPENDIX C

REAL-TIME VERSION

C-1
* 07/19/79 16:31:48
***LISTING FOR RT1:TSEMAIN.FTN
$IN
$ASSM NORX3
TSEMAIN PROG
$FORT
$TITL FILE TSEMAIN - MAIN PROGRAM FOR ETSE
IMPLICIT INTEGER*2 (A-Z)
REAL DB, BITVEL
INTEGER*2 RHO, TFSO, GRND, I, ZERO, T, TSY, STOP, ZTH, TFS, K, BETA
INTEGER*2 INDCCTR, ANG(7), DISPLA, MMU, SIGMA, ELEV(2)
INTEGER*4 TIME, RSAVE(16), SECOND, NRCEAD
INTEGER*4 NEXTIM, ITIME
REAL PCTMIN, RRAREA
COMMON /CALB/ DB, BITVEL, NRC
COMMON /MUSIG/PCTMIN, MMU, SIGMA
COMMON /READ2/NRCEAD, OLD
COMMON /SECTOR/INDCTR
COMMON /EXTRA/RHO, GRND, ZTH, BETA, K, RRAREA(12, 24), RHO2, ZTH2, BETA2,
PCT2, MMU2, SIGMA2, MINUT2, ANG(6), CHANGE
EQUIVALENCE (DISPLA, ANG(7)), (DEMAND, ANG(6)), (ELEV(1), ANG(4))
EQUIVALENCE (ANGLE, ANG<2)), (OLDATA, ANG(7>)
DATA DISPLA/4, NLEVEL/3/
REWIND 4
INDCTR=0
10000 CONTINUE
$ASSM FREZE
COPY SVC1
STM 0, RSAVE
READ SVC 1, READBLOK READ IN DISPLAY, ANGLES
LH R0:READBLOK+SVC1 STA READ IN DEVICE STATUS
BNZ IOERR BRANCH IF NOT ZERO
LH R0:ANGLE LOAD IN BEGINNING ANGLE
CHI R0: 360 LESS THAN 360?
BNL ALLD NO, FULL SCAN
LIS R0:1
STH R0: INDCCTR INDCTR=1
ALLD LM 0, RSAVE
$FORT READ<7, 333> IOUT, DBB, MAXV, MAXS, SLOPE
333 FORMAT(13)
BITVEL=MAXV
DB=65. 28
READ<5, 110> RHO, ZTH, BETA, IPCTMN, MMU, SIGMA, MINUTE
110 FORMAT(13)
CHANGE=0
PCTMIN=IPCTMN*0. 01
SECOND=MINUTE*60
OLD=OLDATA
50 CONTINUE
CALL ICLOCK(2, TIME)
NEXTIM=TIME+SECOND
SVELE=RNG(5)
ANG(5)=ANG(4)+5
DO 51 N=1, NLEVEL
CALL PPRDSC(ANG)
CALL RMAP
ANG(5)=SVELE
51 CONTINUE
CALL PPRDSC(ANG)
CALL TSEPLT(DISPLA)
IF(CHANGE)55, 55, 52
CHANGE=0

52 IF(RHO2. NE. (-1))RHO=RHO2
    IF(PCT2. NE. (-1))PCTMN=PCT2
    IF(MMU2. NE. (-1))MMU2=MMU2
    IF(SIGMA2 NE. (-1))SIGMA=SIGMA2
    IF(ZTH2. NE. (-1))ZTH=ZTH2
    IF(BETA2. NE. (-1))BETA=BETA2
    IF(MINUT2. NE. (-1))MINUTE=MINUT2
DO 54 I=1, 6
    IF(ANG2(I). NE. (-1))ANG(I)=ANG2(I)
    CONTINUE

PCTMN=PCTMN*0.01
SECOND=60*MINUTE

55 IF(Demand.LE.56,75,50

56 CALL ICLOCK2,itime)
    IF(ITIME. GT. TIME)GO TO 58
    IF(NEXTIM. LE. 86400)GO TO 50
    NEXTIM=NEXTIM-86400
58 IF(NEXTIM. LT. ITIME)GO TO 50

CONTINUE

$ASSM
R0 EQU 0
R1 EQU 1
R2 EQU 2
R3 EQU 3
R4 EQU 4
STM R0, RSAVE
L R1, NEXTIM
ST R1, SVCSTIM STORE TIME OF DAY
SVC 2, TIMBLK TIME OF DAY WAIT
LM R0, RSAVE RETURN TO FORTRAN

$FORT
GO TO 50

75 PAUSE 0
GO TO 50

$ASSM
ALIGN 4
IOERR EXBR R1, R0
NHI R1, X'00FF'
CHI R1, X'88'
BE ALLD YES, FULL SCAN, DEFAULT DISPLAY
CHI R1, X'90'
BE ALLD YES, FULL SCAN, DEFAULT DISPLAY
SVC 2, ERRCODE PRINT OUT ERR MSG TO CONSOLE
SVC 2, PAUSE PAUSE
B READ TRY AGAIN
ALIGN 4
READBLOK EQU *
DB X'58' READ BINARY AND WAIT
DB 4
DC H'0' STATUS
DC A(DISPLA) BEGINNING OF BUFFER
DC A(OLDATA+1) END OF BUFFER
DSF 3

ALIGN 4
ERRCODE EQU *
DB 0 4 CHRS OF ASCII HEX
DB 6
DC H'0'  TO ERROR MSG
DC A(ERCD)
ALIGN 4
ERRBLOK EQU *
(   DB   7
   DC  'H'30'    30 CHR'S
ERCD  DC  C'
DC  C'I/O ERROR IN RTN TSEMAIN'
   ALIGN 4
PAUSE  EQU  *
   DB  0.1
   ALIGN 4
TIMBLK  EQU  *
   DB  0
   DB  10    CODE 10
   DB  0.0
SVCTIM  DC  'F0'
$FORT
   TIME OF DAY
END
*   ")
**LISTING FOR ERT1: TSEDATA.FTN**

`$N`  
`$ASSM`  
`SCRT`  
`SQUEZ 3`  
`TSEDATA PROG`  
`$FORT`  
`$TITL FILE 'TSEDATA' - DATA INPUT SUBROUTINE FOR ETSE`  
`SUBROUTINE PPRDSC(ANG)`  
`IMPLICIT INTEGER*2 (A-Z)`  
`INTEGER*4 PPR1(1028), PPR2(1028), RSAVE(16)`  
`+ REAL DB, BITVEL`  
`+ INTEGER*4 IREF(1024), IVEL(1024), IVEL(1024),`  
`+ IREF2(1024), IVEL2(1024)`  
`INTEGER*4 PPRANG`  
`INTEGER*2 ANG(6), RE(1025), HR(258), TL1, TL2, ROUANT`  
`EQUIVALENCE (PPRANG, PPRIC3)), (PP, PPRIC2))`  
`EQUIVALENCE (DECOD(5), IREF(1)), (DECOD(517), IVEL(1))`  
`+ (DECOD(5), IREF(1)), (DECOD2(517), IVEL2(1))`  
`COMMON /REFL/ RE, HR, NCL, NIDP, INCL, IMX.`  
`+ IMN, TL1, TL2, ROUANT, IDVEL(258)`  
`COMMON /CAL/ DB, BITVEL, NRC`  
`COMMON /SECTOR/ INDCTR`  
`COMMON /ZSTORE/ PPR1`  
`COMMON /RUNSUM/ PPR2`  
`CALL CONMSG('6. 'PPRDSC')`  
`REWIND 9`  
`NAZ=0`  
`ELEV1=ANG(4)*11.37778`  
`ELEV2=ANG(5)*11.37778`  
`IF(INDCTR.EQ.2) INDCTR = 0`  
`IF (INDCTR.EQ.0) GO TO 1`  
`CW = ANG(3) - ANG(2)`  
`IF (CW.GT.180) CW = CW - 360`  
`IF (CW.LT.(-180)) CW = 360 + CW`  
`IF (CW.LT.0) CW = 0`  
`IF (CW.GT.0) CW = 15`  
`BNRA = 11.37778 * ANG(2)`  
`ENDA = 11.37778 * ANG(3)`  
`BNRA=MOD(BNRA, 4096)`  
`ENDA=MOD(ENDA, 4096)`  
`IF(ENDA.GT.4086)ENDA=0`  
`STPFLG = 0`  
`NITG=4`  
`NRC=768`  
`IF (CW.GT.0 .AND. ENDA.LT. BNRA) STPFLG = 1`  
`IF (CW.EQ.0 .AND. ENDA.GT. BNRA) STPFLG = 1`  
`IF (BNRA.GT.10 .AND. BNRA.LT.4086)GO TO 1`  
`BNRA=0`  
`STPFLG=2`  
`1 CONTINUE`  
`10000 CONTINUE`  
`$ASSM`  
`CROZE`  
`COPY SVC1`  
`R0 EQU 0`  
`R1 EQU 1`  
`R2 EQU 2`  
`R3 EQU 3`
C
R5  EQU  5
R6  EQU  6
R14 EQU  14
R15 EQU  15
R13 EQU  13
R6  EQU  6
STM  0.RSAVE
    L  R4.WAITREAD+SVC1.SAD
    AIS R4.15
    ST R4.WAITREAD+SVC1.EAD
    BAL R13.WREAD
    LIS R1.4
    L  R0.PPRI(R1)
    NHI R0.3
    AIS R0.1
    SLLS R0.8
    AIS R0.4
    SLLS R0.2
    SIS R0.1
    L  R4.PPRIBLK+SVC1.SAD GET BEGINNING ADDRESS
    AR R4.80 COMPARE END ADDRESS
    ST R4.PPRIBLK+SVC1.EAD STORE IN EAD
    L  R4.WAITREAD+SVC1.SAD
    AR R4.80
    ST R4.WAITREAD+SVC1.EAD SAME NUMBER
    L  R4.OUTBLK+SVC1.SAD
    AHI R4.1028*4-1
    ST R4.OUTBLK+SVC1.EAD
    L  R4.PPRIBLK2+SVC1.SAD GET NEXT BEGINNING ADDR
    AR R4.80
    ST R4.PPRIBLK2+SVC1.EAD STORE IN SVC BLOCK
    L  R4.OUTBLK2+SVC1.SAD
    AHI R4.1028*4-1
    ST R4.OUTBLK2+SVC1.EAD
    LDAI R15.PPRI
    LDAI R14.PPRI2

DETECT
    LCS R6.1 SET R6 TO -1 FOR COUNTER
    BAL R13.WREAD READ IN AN AZIMUTH
    LH R1.8CR15 LOAD IN AZIMUTH DATA
    NHI R1.X'FFF' AND OUT UNWANTED BITS
    BAL R13.WREAD GET ANOTHER AZIMUTH
    LH R2.8CR15 GET AZIMUTH DATA
    NHI R2.X'FFF' AND OUT UNWANTED BITS
    CR R1.R2 COMPARE TWO AZIMUTHS
    BL CWISE IF R1<R2, RADAR IS GOING CWISE
    LIS R3.0 DIRECTION FLAG
    B  WHATIZIT CONTINUE

CWISE
    LIS R3.15 DIR 15LAG = CW
    LH R4.INDCTR SECTOR SCAN OR FULL CIRCLE?
    BZ EDTECT FULL CIRCLE
    CH R3.CH IS DIRECTION OF ROTATION CORRECT?
    BNE DETECT WRONG DIRECTION, WAIT
    OR R3.R3 WHICH DIRECTION IS IT?
    BZ CCW COUNTER COUNTER CWISE
    LH R5.STPFLG

THI R5.2 CASE 2?

BNZ CWCASE2 CW CASE 2
    CH R2.BGNA ANGLE < BGNA?
    BL WAIT YES, GET READY
    B DETECT NO, TRY AGAIN

CWCase2
    CH R2.X'800' ANGLE > 180?
    BL CMP1 NO ALL OK
    SHI R2.X'1000' YES, SUBTRACT 360
    B CMP1

CCH
    LH R5.STPFLG (-6)
CCW CASE2

CHI R2.*,X'0800'

BNL CMP2

B CMP2

ANGLE < 180?

THI R2.*,X'1000'

BL CMP3

SHI R2.*,X'1000'

NOW CHECK FOR BEGINNING

YES, ADD 360

NO, ALL OK

YES, SUBTRACT 360

READ IN NEW AZIMUTH

READ IN ELEVATION

AND OUT UNWANTED BITS

READ2

BAL R13, WREAD

LH R2, 12(R15)

NHI R2, X'0FF'

CHI R2, 681

BP ZER01

CPEV CH R2, ELEV1

BL EDETECT

CH R2, ELEV2

BP EDETECT

LH R2, PPRANG

NHI R2, X'FFF'

CHI R2, 6

BL FUDGE

WITHIN RANGE?

YES, GET AZIMUTH

READ2

BAL R13, GOREAD

LH R4, PPRANG

NHI R4, X'FFF'

NXT OR R3, R3

BZ CCLK04

CR R4, R2

BL NXT2

B READ2

ZERO1 LIS R2, 0

B CPEV

FUDGE LIS R2, 6

B READ2

NXT2 BAL R13, GOREAD

LH R4, PPRANG

NHI R4, X'FFF'

OR R3, R3

BZ CCLK04

CR R4, R2

BP DONE

B NXT2

CCW4

CR R4, R2

BP NXT2

B READ2

DONE LH R15, PPR1

NHI R15, X'0FF'

STH R15, PPR1

SVC 1, OUTBLK

LH R0, OUTBLK+SRC1, STA

BNZ ERROR

LM 0, RSERVE

#FORT RETURN

#ASSM

WREAD SVC 1, WAITREAD

LH R0, WAITREAD+SRC1 STA READ RETURNED STATUS

BNZ ERROR

IF NOT ZERO, ERROR

BR R13

GOREAD SVC 1 PPRIBLK2

READ IN ONE AZIMUTH

LM 0 RSERVE

C-8
10 DECOD(I)=PPRI(I)
JSIZ=NRC/NITG
IMX=JSIZ+1
K=5
DO 101 I=1,JSIZ
REF=0
VEL=0
DO 20 J=1,NITG
RAW=PPRI(K)
K=K+1
$ASSM
ST R0.RSAVE
L R0.RAW GET PACKED SOURCE WORD
EXHR R0,R0 SHIFT POWER TO LOW HALF
NHI R0.X'1FF' NINE BITS
AHM R0.REF INTEGRATE REF
L R0.RAW GET LOW HALF
SRHA R0.8 SHIFT DOPPLER TO LOW ORDER
AHM R0.VEL INTEGRATE DOPPLER
L R0.RSAVE RESTORE REGISTER
$FORT
10 CONTINUE IF(REF. LE. 0) GO TO 111
REF=REF+.0390625-DB
IF(REF. LT. -39) REF=REF+100
IF(REF(I)=REF
VEL=VEL+BITVEL/128.
VEL(I)=VEL-VEL
GO TO 101
111 IREF(I)=0
VEL(I)=0
VEL(I)=0
101 CONTINUE
$ASSM
STM 0.RSAVE OUTPUT LAST AZIMUTH
SVC 1.OUTBLK WAIT FOR FIRST READ TO FINISH
SVC 1.WAITBLK READ NEXT AZIMUTH
SVC 1.PPRIBLK
LM 0.RSAVE
$FORT
DO 40 I=1,4
40 DECOD2(I)=PPRI2(I)
JSIZ=NRC/NITG
IMX=JSIZ+1
K=5
DO 201 I=1,JSIZ
REF=0
VEL=0
DO 400 J=1,NITG
RAW=PPRI2(K)
K=K+1
$ASSM
ST R0.RSAVE GET PACKED SOURCE WORD
L R0.RAW SHIFT POWER TO LOW HALF
EXHR R0.0 R0 NINE BITS
NHI R0.X'1FF' INTEGRATE REF
L R0.RAW GET LOW HALF
SRHA R0.8 SHIFT DOPPLER TO LOW ORDER
AHM R0.VEL INTEGRATE DOPPLER
L R0.RSAVE RESTORE REGISTER
$FORT
400 CONTINUE IF(REF. LE. 0) GO TO 411
IF (REF. LT. -39) REF = REF + 100
IREF2(I) = REF
VEL = VEL + BITVEL / 128.
IREF2(I) = VEL - IVEL(I)
IVEL(I) = IVEL(I) - VEL
GO TO 201
411 IREF2(I) = 0
VEL2(I) = 0
IVEL(I) = 0
201 CONTINUE

*ASSM
STM 0, RSAN
SVC 1, OUTBLK2
SVC 1, WAITBLK
LH R0, PPRI2BLK2 + SVC1, STA READ STATUS
BNZ ERROR IF NOT ZERO, ERROR
LH R0, PPRI2BLK + SVC1, STA
BNZ ERROR
LH R0, OUTBLK + SVC1, STA
BNZ ERROR
LH R0, OUTBLK2 + SVC1, STA
BNZ ERROR
LH R6, NAZ
AIS R6, 2
STH R6, NAZ
CHI R6, 446
BLR F:13
LIS F:6, 2
STH R6, INDCTR
B DONE
ERROR SVC 2, ERRCODE
SVC 2, ERRBLOK
SVC 2, PAUSE
LM 0, RSAN

***$P2 IS FORTRAN STMT NO 2
B $P2
PAUSE
ALIGN 4
DB 0, 1
PAUSE
ERRCODE EQU *
OPT DB 0
DB 6
DC H'0'
DC A(ERCD)
ALIGN 4
ERRBLOK EQU *
DB 0, 7
DC H'15'
DC C'
DC C'I/O ERROR'
ALIGN 4
WAITBLK EQU *
DB X'06'
DB 10
DB 0, 0
DSF 5
ALIGN 4
WAITREAD EQU *
DB X'59'
DB 10
DB 0, 0
DC A(PPRI)
DC R(PPRI)
DSF 3
ALIGN 4
PPRIBLK2 EQU *
DB X'51' READ
DB 10 LU 10
DB 0.0
DC A(PPRI2)
DC A(PPRI2)
DSF 3
ALIGN 4
OUTBLK2 EQU *
DB X'31'
DB 9 LU 9
DB 0.0
DC A(DECOD2)
DC A(DECOD2)
DSF 3
ALIGN 4
PPRIBLK EQU *
DB X'51' READ
DB 10 LU 10
DB 0.0
DC A(PPRI)
DC A(PPRI)
DSF 3
ALIGN 4
OUTBLK EQU *
DB X'31' WRITE
DB 9 LU 9
OUTST DB 0.0
DC A(DECOD)
DC A(DECOD)
DSF 3
$FORT RETURN
END
*
SUBROUTINE RMAP

IMPLICIT INTEGER*2 (I-N)
INTEGER*2 NRC, VAR, STORE(10), PLACE, OLDDATA
INTEGER*4 ANC(1028), IDTIME, ITZ, IZB, IZS
+ SRC(1028), DST(1028)
INTEGER*2 RHO2, ZTH2, BETA2, PCT2, SIGMA2, ANG2, CHANGE
INTEGER*2 TWENTY, ELEVEN, DAY, HOUR, MINUTE, SECOND, TP, ELEVAT, AZ
INTEGER*2 AZIM, TC, TA, T
INTEGER*2 Y, I, THETA1, NRC1, II, THETA, RHO, STOP, GRND
INTEGER*2 MEAN, POWER, SIGMA, TP2, TP3, SEGNO, Q, J, ZTH, MMU
INTEGER*2 BEGIN, SUM1, SUM2, JMIN, M, K, L, I
INTEGER*2 Y, ZERO, TWO, BETA
INTEGER*4 RSAVE(16), RISAVE
INTEGER*2 JREF(1024), JDVEL(1024)
INTEGER*2 RE, HR, TL1, TL2, ROQUANT
REAL RNN, RRAREA
REAL PCTMIN, AZT, ELEVAR, BGNA, AZCHK, ENDA
INTEGER*2 FLAG, IQ, IB, IE, IM
INTEGER*2 DBB, SLOPE
INTEGER*4 NRCEAD
COMMON /TLIS/ T, TA, JDAY, JHR, JMIN, JSEC, IDAY, IHR, IMIN, ISEC
COMMON /CALR/ SRC, DST, IREF(1024)
COMMON /SWITCH/ IC1(44), IC2(44), TC(1920), IPC1(5400),
+ IPC2(5400), IPC3(5400), IPCNT(1920), IPTC(44),
+ NEMB, NEMC, NAC
COMMON /AZ2/ SINA, COSA, DELTAZ, ISCANF, NEL, RI, SA
COMMON /AZ2/ AZLAST, AZLAST, NA, ELEVRA
COMMON /REL/ RE(1985), HR(258), NCL, NID, NIDP, INCL
X, IMX, IIN, TLL, TL2, ROQUANT, IDVEL(258)
COMMON /PNTRS/ NCMX, NVMIN, NNUM, IELSNS, NSCAN, IESNL, NVSCN, NT
COMMON /INTL/ MSBN, MSN, HM, FNSN
COMMON /EXTRA/ RHO, GRND, ZTH, BETA, K, RRAREA(12, 24), RH02, ZTH2, BETA2,
PCT2, MMU2, SIGMA2, MINUT2, ANG2(6), CHANGE
COMMON /ZSTORE/ ANC
COMMON /ZLOOK/ ZLOOK2, ZARY(91)
COMMON /ZCONST/ EARTH, VMK
COMMON /MAPPAR/ DAY, HOUR, MINUTE, SECOND, DBB, MAXV, MAXS, IOUT, SLOPE
COMMON /CNT/ COSPH1, SIRPH1, COSPH2
COMMON /DATA1/ ECL(224), NOFST, KOST, ICAD, NAM1
COMMON /DATA2/ VCL(736), MVVC, MVC1
COMMON /DATA3/ VI(192), MVVR, NVR1
COMMON /NVLIS/ NVARM, NCARM, NVO, ICO, IO, JO, JVR, KTL
COMMON /FILTER/ TATRMN, AREARM, DAZM
EQUIVALENCE(ANC(5), JREF(2)), (ANC(517), JDVEL(2))
DATA TWENTY/16/
DATA TWO/2/
DATA PCTMIN/ .05/, MMU/0/, SIGMA/0/, OLDDATA/0/
DATA ZERO/0/

C ********** INITIALIZE ARRAY

C
T=0
AZLAST=-999
DO 901 J=1, MVVC
901 VCL(J)=0.
C-12
DO 902 J=1,10
VRC(J)=0.
RE(1)=0
ICC=0
CALL CONMSG(7,'TSERMAP')
TL1=TL1/ROUANT
TL2=TL2/ROUANT
REWIND 4
REWIND 6
DO 3 IX=1,91
II=IX-I2OFF
ZX=FLOAT(II)/10.
3 ZARY(IX)=10.**ZX
00001 CONTINUE
00750 FORMAT(1X,'ENTER PRF, 0=768, 1=922, 2=1075, 3=1229')
RCKM=.075
FLAG=0
333 FORMAT(I3)
READ(7,333)OLDATA
01002 FORMAT(1X,'AZIM ELEV RAN S( REF VEL STD)')
REWIND 9
K=0
10000 CONTINUE
ASSM
R0 EQU 0
R1 EQU 1
R2 EQU 2
R3 EQU 3
R4 EQU 4
R5 EQU 5
R6 EQU 6
R7 EQU 7
R8 EQU 8
R9 EQU 9
R10 EQU 10
R11 EQU 11
R12 EQU 12
R13 EQU 13
R14 EQU 14
R15 EQU 15
STM R0, RSAVE
FREZE
COPY SVC1.
L R3, ANCBLK+SVC1. SAD ANCBLK START
AIS R3, 15 READ ANCELLARY ONLY
ST R3, ANCBLK+SVC1. EAD
SVC 1, ANCBLK THE READ
LH R0, ANCBLK+SVC1. STA
BNZ ERROR
LIS R1, 0
STH R1, STOP
LIS R5, 0 WORKING REGISTER
LIS R9, 10 MULTIPRICAND
LHI R11, 100 MULTIPRICAND
L R0, ANC(R1) DATA
STBR R0, R5 HOUR
LIS R10, 15 MASK
NR R10, R5 1 HOUR
SRLS R5, 4 10 HOUR
NHI R5, 2
MHR R5, R9 (+10)
AMR R5, R10 TOTAL
STH R5, HOUR HOURS
EXBR R0, R0
STBR R0, R5 MINUTES
LIS R10, 15 MASK (-13)
NR  R10.R0  1 MINUTE
SRLS R5, 4   10 MINUTE
NHI  R5, 7   MASK
MHR  R5, R9  (+10)
AHR  R5, R10  TOTAL MINUTES
STH  R5, MINUTE

EXHR R0, R0
STBR R0, R5
LIS  R10.15  DAYS
NR  R10,R5  1 DAY
SR. 5 R5, 4   10 DAY
MHR  R5, R9  (+10)
AHR  R5, R10
SRLS R0, 8   100 DAYS
NHI  R0,X"F"  MASK
MHR  R0, R11  (+100)
AHR  R5, R0  TOTAL DAYS
STH  R5, DAY  DAYS
AIS  R1, 4   NEW DATA
L  R0, ANC(R1)
LHI R10, 256
STH R10, NRC1

LHI  10, 1028+4-1
R  10, NRCBLK+SVC1.SAD
ST  10, NRCBLK+SVC1.EAD  STORE END ADDRESS FOR READ
ST  10, NRCREAD  STORE AWAY FOR SECOND READ
NHI  R5,X"CO"  MASK TP
SRLS R5, 6
STH R5, TP
EXHR R0, R0
STBR R0, R5
LIS  R10.15  SECONDS
NR  R10, R5  1 SECOND
SRLS R5, 4
NHI  R5, 7  MASK 10 SECONDS
MHR  R9, R5  (+10)
AHR  R9, R10  TOTAL SECONDS
STH R9, SECOND
AIS R1, 8  MORE DATA
LHL F0, ANC(R1)  ELEVATION
NHI F0, X"FFF"  ANGLE
STH F0, ELEVAT
LM  R0, RSAVE

$FORT
NRC=NRC1
GRND=PTMIN+NRC
REWIND 9
17 ELEVEN=0
TA=T
DR=(2**TP)*RCKM
K=K+1
NRLI=NRC
10001 CONTINUE

$ASM
STH R0, RSAVE
SVC 1, NRCBLK
LH  R0, NRCBLK+SVC1 STA
LIS  R1, 0
LH  R0, ANC(R1)
BM MINUS
LIS  R1, 1
STH R1, STOP
MINUS LIS  R1, 8
LH  R0, ANC(R1)
NHI  R0, X"FFF"  C-14  MASK
DIM R1, R2, R3, R4, R5, R6
LHL R0, ANCEVLO
NHI R0, X'FFF'
STH R0, ELEVAT
STORE IT

$FORT
DO 101 I=IMN, IMX
RE(I)=JREF(I)
IDVEL(I)=JDVEL(I)
101 CONTINUE
00500 CONTINUE
ICC=ICC+1
ELEVA=ELEVAT/11.37778
IELSN=IFIX(ELEVA)
T= ((DAY*24+HOUR)+60+MINUTE)*60+SECOND
AZT=AZ/11.37778
FLAG=FLAG+1
A=AZT* .01743
AZCHK=AZT
IF(K. NE. 1) GO TO 105
BGNA=AZT
ENDA=AZT+359.
105 CONTINUE
IF(K. LT. 180) GO TO 106
IF(AZT. GT. 180.) GO TO 106
AZCHK=AZT+359.
106 CONTINUE
IF(AZCHK. GT. ENDA) K=1
SINA=SIN(A)
COSA=COS(A)
DELTZ=0.019187
NA=K
RE(258)=0
IF(NA. EQ. 1)NAC=1
NAC=NAC+1.
IF(NAC. GT. 1)NAC=0
IF(NA. NE. 1. OR. AZLAST. LT. -990.) GO TO 224
IDAY=DAY
IHR=HOUR
IPIN=MINUTE
ISEC=SECOND
PHI=ELEVA+DAZM
COSPHI=COS(PHI)
SINPHI=SIN(PHI)
COSPH2=COSPHI*COSPHI*EARTH
CALL TRACK
224 CALL CONTOR
AZLAST=AZT
IF(STOP. NE. 1) GO TO 17
800 CALL STRAK
STOP

$ASSM
ALIGN 4
ERROR SVC 2. ERRCODE DECODE ERROR BITS
SVC 2. ERRBLOK OUTPUT ERROR MSG TO CONSOLE
SVC 2. PAUSE
LM 0. RSAVE RETURN TO FORTRAN

$FORT
IF(K. EQ. 0) GO TO 10000
IF(K. EQ. 1) GO TO 10001
GO TO 1
10003 CONTINUE
$ASSM
ALIGN 4
ERRCODE EQU *
DB 0
DB 6
DC H'0'
DC A(ERCD)
ALIGN 4
ERRBLOK EQU *
DB 0
DB 7
DC H'28'
ERCD DC C'
DC C'I/O ERROR IN RTN RANGE'
ALIGN 4
PAUSE EQU *
DB 0.1
ALIGN 4
ANCBLK DB X'59'
DB 9
DB 0.0
DC A(ANC) START ADDRESS
DC A(ANC) END ADDRESS
DSF 3
NRCBLK DB X'59'
DB 9
DB 3.0
DC A(ANC) START ADDRESS
DC A(ANC) END ADDRESS
DSF 3
$FORT END
APPENDIX D

POST-MISSION ANALYSIS VERSION

(Note BLOCK DATA, CONTOR, PEAKD, TRACK, ATRAK, STRAK, COMPAR, RESOLV, COMBIN and STRAK are identical in both versions)
SQUEZ

TSEDATA PROG

FILE TSEDATA - DATA INPUT SUBROUTINE FOR ETSE

SUBROUTINE PPRDSC(ANG)

IMPLICIT INTEGER*2 (A-Z)
INTEGER*4 PPRI1(1028),PPRI2(1028),RSAVE(16)
INTEGER*4 OUT(260),OUT2(260)
INTEGER*4 PPRANG
INTEGER*2 ANG(6)
EQUIVALENCE (PPRANG,PPRI1(2)),(PP,PPRI2(2))
EQUIVALENCE (OUT(1),PPRI1(1)),(OUT2(1),PPRI2(1))

COMMON /SECTOR/INDCTR
COMMON /ZSTORE/PPRI1
COMMON /RUNSUM/PPRI2
AVEN=3
CALL CONMSG(6,'PPRDSC')

REWIND 9
ELEV1=ANG(4)*11.37778
ELEV2=ANG(5)*11.37778
IF (INDCTR.EQ.2)INDCTR = 0
IF (INDCTR.EQ.0) GO TO 1
CW = ANG(3) - ANG(2)
IF (CW.GT.180) CW = CW - 360
IF (CW.LT.(-180)) CW = 360 + CW
IF (CW.EQ.0) CW = 0
IF (CW.GT.0) CW = 15
BGNA = 11.37778 * ANG(2)
ENDA = 11.37778 * ANG(3)
BGNA=MOD(BGNA,4096)
ENDA=MOD(ENDA,4096)
IF (ENDA.GT.4096) ENDA=0
STPFLG = 0
IF (CW.GT.0 .AND. ENDA.LT. BGNA) STPFLG = 1
IF (CW.EQ.0 .AND. ENDA.GT. BGNA) STPFLG = 1
IF (BGNA.GT.10 .AND. BGNA.LT. 4086) GO TO 1
BGNA=0
STPFLG=2
1 CONTINUE
10000 CONTINUE

$ASSM

SCRAZ TSEDATA
SQUEZ 0

$ASSM

FREIZE
CROSS
COPY SVC1.
R0 EQU 0
R1 EQU 1
R2 EQU 2
R3 EQU 3
R4 EQU 4
R5 EQU 5
R6 EQU 6
R14 EQU 14
R15 EQU 15
R12 EQU 13
R7 EQU 7
R8 EQU 8
R9 EQU 9
R10 EQU 10
R11 EQU 11
R12 EQU 12
STM 0.RSAVE
L R4.WAITREAD+SVC1.SAD
AIS R4.15
ST R4.WAITREAD+SVC1.EAD
BAL R13.WREAD
LIS R1.4
L R0.PPRL(R1)
NHI R0.3
AIS R0.1
SLLS R0.8
AIS R0.4
SLLS R0.2
SIS R0.1
L R4.PPRLBLK+SVC1.SAD GET BEGINNING ADDRESS
AR R4.R0 COMPUTE END ADDRESS
ST R4.PPRLBLK+SVC1.EAD STORE IN EAD
ST R4.WAITREAD+SVC1.EAD SAME NUMBER

* LEVS ONLY 256 WORDS OF VIDEO OUT

SI R4.2048
ST R4.OUTBLK+SVC1.EAD
L R4.PPRLBLK2+SVC1.SAD GET NEXT BEGINNING ADDR
AR R4.R0 COMPUTE END ADDRESS
ST R4.PPRLBLK2+SVC1.EAD STORE IN SVC BLOCK

* LEVS ONLY 256 WORDS OF VIDEO OUT

SI R4.2048
ST R4.OUTBLK2+SVC1.EAD
LDI R15.PPRL
LDI R14.PPRL2
LCS R6.1 SET R6 TO -1 FOR COUNTER
DTECT BAL R13.WREAD READ IN AN AZIMUTH
LH R1.8(R15) LOAD IN AZIMUTH DATA
NHI R1.X'FFF' AND OUT UNWANTED BITS
BAL R13.WREAD GET ANOTHER AZIMUTH
LH R2.8(R15) GET AZIMUTH DATA
NHI R2.X'FFF' AND OUT UNWANTED BITS
CR R1.R2 COMPARE TWO AZIMUTHS
BL CWISE IF R1>R2, RADAR IS GOING CWISE
LIS R3.0 DIRECTION FLAG
B WHATIZIT CONTINUE
LIS R3.15 DIR 15LAG = CW
WHATIZIT LH R4.INDCTR SECTOR SCAN OR FULL CIRCLE?
B2 EDETET FULL CIRCLE
WH R3.CW IS DIRECTION OF ROTATION CORRECT?
BNE DETECT WRONG DIRECTION, WAIT
OP R3.R3 WHICH DIRECTION IS IT?
B2 CCW COUNTER CLOCKWISE
LH R5.STPFLG
THI R5.2 CASE 2?
CMP1 CH R2.BGNA ANGLE < BGNA?
BL WAIT YES, GET READY
B DETECT NO, TRY AGAIN
CWCASE2 CHI R2.X'800' ANGLE > 180?
BL CMP1 NO, ALL OK
SHI R2.X'1000' YES, SUBTRACT 360
B CMP1
CCW LH R5.STPFLG
THI R5.2 CHECK FOR CASE 2
CMP2 CH R2.BGNA ANGLE > BGNA?
BL DETECT NO, TRY AGAIN
WAIT PAI R13.WREAD YES, GET READY
GET NEXT AZIMUTH AND OUT UNWANTED BITS

CASE 2?

INCREMENT COUNTER

DO IT TWICE TO BE SURE!

BRANCH IF COUNTER CLOCKWISE

CASE 1?

* WANT SEVERAL PPIS

B READ

ANGLE > ENDA?

YES, ALL DONE

NO, KEEP READING

59.5 DEG = 0

ANGLE < BGNA?

YES, KEEP READING

* WANT SEVERAL PPIS

B READ

NO, ARE WE DONE YET?

YES

NO, KEEP READING

CHECK FOR CASE 1

ANGLE < ENDA?

YES, ALL DONE

NO, CONTINUE

ANGLE > BGNA?

NO, KEEP READING

YES, CHECK FOR FINISHED

OK

NOT YET

B READ1

B CMP4

B WWCASE2

RN R2 X '1000'

P CMP4
CCW CASE2

Chi R2,X'000'

BNL CMP2

NO. ALL OK

AHI R2,X'1000'

YES. ADD 360

B CMP2

NOW CHECK FOR BEGINNING

WCW CASE2

Chi R2,X'000'

BL CMP3

NO. ALL OK

SHI R2,X'1000'

YES. SUBTRACT 360

B CMP3

EDETECT

BAL R13,WREAD

READ IN NEW AZIMUTH

LH R2,12(R15)

READ IN ELEVATION

NHI R2,X'FFF'

AND OUT UNWANTED BITS

CHI R2,681

BP ZERO1

CPEV

CH R2,ELEV1

BL EDETECT

CH R2,ELEV2

BP EDETECT

WITHIN RANGE?

LH R2,PPRANG

YES. GET AZIMUTH

NHI R2,X'FFF'

CHI R2,6

BL FUDGE

READ2

BAL R13,GOREAD

START READING

LH R4,PPRANG

NXT OR R3,R3

BZ CCLOK4

COUNTERCLOCKWISE

CR R4,R2

BL NXT2

B READ2

ZERO1 LIS R2,0

CPEV

B FUDGE

LIS R2,6

IF 0. MAKE IT 6

B READ2

NXT2 BAL R13,GOREAD

GET NEW AZIMUTH

LH R4,PPRANG

NHI R4,X'FFF'

AND OUT UNWANTED BITS

OR R3,R3

BZ CCW4

* WANT SEVERAL PPIS

B NXT2

CR R4,R2

ANG > ENDA?

BP DONE

YES. FINISHED

B NXT2

NO. KEEP READING

CCLOK4 CR R4,R2

BP NXT2

B READ2

CCH4 CR R4,R2

BL DONE

B NXT2

DONE

LH R15,PPRI

NHI R15,X'0FF0'

STH R15,PPRI

SVC 1.OUTBLK

LH R0.OUTBLK+SVC1.STA

BNZ ERROR

LM 0. RSAVE

$FORT

RETURN

$ASSM

ALIGN 4

WREAD SVC 1.WAITREAD

LH R0.WAITREAD+SVC1.STA READ RETURNED STATUS

BNZ ERROR

IF NOT ZERO. ERROR

BR R13

GOREAD SVC 1.PPRIBLK2 READ IN ONE AZIMUTH

* THIS SURROGATES THREE DOGUE CELLS
LIS R16:12 COUNTER FOR MAIN BUFFER
LHI R4:16 COUNTER FOR NEW BUFFER
AVE LHS R8:0 ZERO POWER ACCUMULATOR
AVE LHS R10:0 ZERO VELOCITY ACCUMULATOR
AVE R1:3 AVERAGING COUNTER
AVE LHL R7:PPRI(R4) LOAD POWER HALFWORD
AVE LHL R7:PPRI(R4) LOAD VELOCITY HALFWORD
AVE NHI R7:’1FF’ GET ONLY THE POWER
AVE AR R8:R7 ADD TO ACCUMULATOR
AVE AR R10:R7 INCREMENT FOR NEXT HALFWORD
AVE AR R10:R7 INCREMENT FOR NEXT HALFWORD
AVE AR R10:R7 INCREMENT FOR NEXT HALFWORD
AVE AR R10:R7 INCREMENT FOR NEXT HALFWORD
AVE SIS R1:1 SUBTRACT FOR INNER LOOP
AVE BNZ AVER BRANCH IF NOT 3 ADDED UP
AVE DH R8:AVEN DIVIDE POWER
AVE NHI R9:’1FF’
AVE SLA R9:16 SHIFT POWER TO LEFT
AVE SRA R10:8 DIVIDE VELOCITY
AVE SLA R11:8
AVE NHI R11:’FFFF’ GET ONLY LOWER HALFWORD
AVE AR R9:R11 PUT TOGETHER
AVE AR R12:4 INCREMENT NEW ARRAY POINTER
AVE ST R9:PPRI(R12) PUT INTO ARRAY
AVE CLHI R4:3072
AVE BP AVEEND
AVE B AVE

AVEEND SVC 1.OUTBLK OUTPUT LAST AZIMUTH
AVE SVC 1.WAITBLK WAIT FOR FIRST READ TO FINISH
AVE SVC 1.PPRIBLK READ NEXT AZIMUTH
AVE SVC 2.OUTBLK2 OUTPUT LAST AZIMUTH
AVE SVC 1.WAITBLK WAIT FOR READ TO FINISH
AVE LH F0:PPRIBLK2+SVC1 STA READ STATUS
AVE BNZ ERROR IF NOT ZERO, ERROR
AVE LH R0:PPRIBLK+SVC1 STA
AVE BNZ ERROR
AVE LH F0:OUTBLK+SVC1 STA
AVE BNZ ERROR
AVE LH F0:OUTBLK2+SVC1 STA
AVE BNZ ERROR
AVE LH R6:NAZ
AVE AIS R6:2 ADD TWO TO AZIMUTH CTR
AVE STH R6:NAZ
AVE CHI R6:1760 TOO MANY AZIMUTHS?
AVE BLR R13 NO, KEEP GOING
AVE LIS R6:2 YES, INDCTR=2
AVE STH R6:INDCTR
AVE BNQ ONE QUIT
AVE SVC 2.ERRCODE CONVERT ERROR CODE
AVE SVC 2.ERRBLOK OUTPUT MSG TO CONSOLE
AVE SVC 2.PAUSE TASK PAUSED.
AVE LM 0.RSAVE RESTORE FORTRAN REGISTERS

***$P2 IS FORTRAN STMT NO 2
AVE B $P2 START OVER
AVE PAUSE EQU * PAUSE
AVE DB 0.1 PAUSE
AVE ERRCODE EQU * CODE 6
AVE OPT DB 6 CODE 6
AVE DC H’0’ CODE 6
AVE DC A(ERCD) DESTINATION
AVE ALIGN 4
```
ERRBLK EQU *  PRINT CONSOLE MSG
DB 0.7  
DC H'15'  CODE 7
DC C' '  PRINT 15 CHR
DC C'I/O ERROR'  ERROR CODE
ALIGN 4

WAITBLK EQU *  WAIT ONLY
DB X'08'  
DB 10  
DB 0.0  
DSF 5
ALIGN 4

WAITREAD EQU *  READ AND WAIT
DB X'59'  
DB 10  
DB 0.0  
DC A(PPRI)  
DC A(PPRI)  
DSF 3
ALIGN 4

PPRIBLK2 EQU *  READ
DB X'51'  
DB 10  
DB 0.0  
DC A(PPRI2)  
DC A(PPRI2)  
DSF 3
ALIGN 4

OUTBLK2 EQU *  
DB X'31'  
DB 9  
DB 0.0  
DC A(OUT2)  
DC A(OUT2)  
DSF 3
ALIGN 4

PPRIBLK EQU *  READ
DB X'51'  
DB 10  
DB 0.0  
DC A(PPRI)  
DC A(PPRI)  
DSF 3
ALIGN 4

OUTBLK EQU *  
DB X'31'  
DB 9  
OUTST  
DB 0.0  
DC A(OUT)  
DC A(OUT)  
DSF 3
$FORT  
RETURN  
END  
```
**$TITL**

FILE TSERMAP — PRINT OUT DATA FIELDS—CHANGED FOR ERT READ BY CLB

**$FORT**

**$SINAL**

$asm

**ETrMNP**

**$FORT**

**$SINAL**

LISTING FOR ERT1: CRANE. FTN

IMPLICIT INTEGER*2 (I-N)

INTEGER*2 NRC, VAR, STORE(10), PLACE, OLDATA

INTEGER*4 ZEE(1024), ANC(1024), IDTIME, IFZ, ITZ, IZB, IZS

INTEGER*2 TWENTY, ELEVEN, DAY, HOUR, MINUTE, SECOND, TP, ELEVAT, AZ

INTEGER*2 AZIM, TC, TA, T

INTEGER*2 Y, I, THETAI, NRCI, II, THETA, RHO, STOP, GRND

INTEGER*2 M, POWER, SIGMA, TP2, TP3, SEGNO, 0, J, ZTH, MMU

INTEGER*2 RRAEX, RRAEX2, RRAEX3, RRAEX4, RRAEX5

INTEGER*2 IREF(1024), IVEL(1024), IVAR(1024), IVELL(256)

INTEGER*2 RE, HR, TL1, TL2, RQUANT

REAL RNN, RRAREA, AXX

REAL PCTMIN, AZT, ELEVAA, BNAA, AZCHK, ENDA

INTEGER*2 FLAG, IO, IB, IE, IM

INTEGER*2 IOUT, DDB, MAXV, MAXS, SLOPE

INTEGER*4 NRCEAD

COMMON /TLIS/ T, TA, JDAY, JHR, JMIN, JSEC, IDAY, IHR, IMIN, ISEC

COMMON /CONST/ VMISW, DIV, VMAG, VMISWM, ZDIV, A1, A2, A3, B1, B2, HDIV

COMMON /SWITCH/ IC1(44), IC2(44), TC(1980), IPC1(5400)

+ IPC2(5400), IPC3(5400), IPCNT(1980), IPTC(44)

+ NMB, NMCN, NAC

COMMON/AZ2/SINA, COSA, DELTA, ISCAN, NEL, RI, SA

COMMON/AZM/AZT, AZLAST, AZSTAR, NA, ELEVAA

COMMON/REFl/RE(1025), HR(258), NCL, NID, NIDP, INCL

 COMMON /INTL/ MHSN, MNSN, HM, FNSN

COMMON/EXTRFL/RHO, GRND, ZTH, BETA, K, RRAREA(258), RHO2(16), AXX(258)

COMMON/ZSTORE/ANC

COMMON/ZLOOK/IZOFF, ZARY(91)

COMMON/ECONST/ EARTH, VMK

COMMON/MAPPAR/ DAY, HOUR, MINUTE, SECOND, DDB, MAXV, MAXS, IOUT, SLOPE

COMMON/CNT/ COSPHI, SINPHI, COSPH2

COMMON/DATA1/ ECL(224), NOFST, KOFST, ICLAD, NAN1

COMMON/DATA2/ VCL(736), MXVC, NVC1

COMMON/DATA3/ VR(192), MXVR, NVR1

COMMON/NVLS/ NVARM, NCARM, NVO, ICO, IO, JYR, KTL

COMMON/FILTER/ TATRMN, AREAMN, DAZM

EQUIVALENCE(RE(2), IREF(1)), (IDVEL(2), JVEL(1))

EQUIVALENCE(ANC(5), ZEE(1))

DATA TWENTY/18/

DATA TWO/2/

DATA PCTMIN/05/, MMU/0/, SIGMA/0/, OLDATA/0/

DATA ZERO/0/

C

C ---------- INITIALIZE ARRAY

C

T=0

AZLAST=-999.

DO 901 J=1, MXVC

901 VCL(J)=0.

DO 902 J=1, MXVR
RE(1) = 0
JOVEL(1) = 0
JDVEL(256) = 0

CALL CONMSG(7, 'TSERMAP')
TL1 = TL1/ROQUANT
TL2 = TL2/ROQUANT
REWIND 4
REWIND 8
DO 2 I = 1, 1024
   IREF(I) = 0
   IVEL(I) = 0
2   DO 3 IX = 1, 91
      II = IX - 1Z0FF
      ZX = FLOAT(II)/10.
3   ZARYV(IX) = 10. ** ZX
   CONTINUE
00001 CONTINUE
DB = 65. 28
750 FORMAT(IX, 'ENTER PRF, 0=768, 1=922, 2=1075, 3=1229')
RCKM = 075
FLAG = 0
READ(7, 333) IOUT, DBB, MAXV, MAXS, SLOPE
333 FORMAT(13)
READ(7, 333) OLDATA
01002 FORMAT(// 'AZIM ELEV RAN ', 8(< REF VEL STD'))
BITVEL = MAXV
BITVAR = MAXS
REWIND 9
K = 0
10000 CONTINUE

$ASSM

R0 EQU 0
R1 EQU 1
R2 EQU 2
R3 EQU 3
R4 EQU 4
R5 EQU 5
R6 EQU 6
R7 EQU 7
R8 EQU 8
R9 EQU 9
R10 EQU 10
R11 EQU 11
R12 EQU 12
R13 EQU 13
R14 EQU 14
R15 EQU 15
STM R0, RSAVE
FREZE
COPY SVC1.

L R3, ANCB + SVC1. SAD ANCBLK START
AIS R3, 15 READ ANCALLARY ONLY
ST R3, ANCB + SVC1. EAD
SVC 1. ANCBLK THE READ
LH R0, ANCB + SVC1. STA
BNZ ERROR
LIS R1, 0

STH R1, STOP
LIS R5, 0 WORKING REGISTER
LIS R9, 10 MULTIPICAND
LHI R11, 100 MULTIPICAND
L R8, ANC(R1) DATA
STBR R8, R5 HOUR 0
NR R10.R5 1 HOUR
SRLS R5, 4 10 HOUR
NHI R5. 3
MHR R5.R9 (*10)
AHR R5.R10 TOTAL
STH R5.HOUR HOURS
EXBR R0.R0
STBR R0.R5 MINUTES
LIS R10.15 MASK
NR R10.R5 1 MINUTE
SRLS R5, 4 10 MINUTE
NHI R5.7 MASK
MHR R5.R9 (*10)
AHR R5.R10 TOTAL MINUTES
STH R5.MINUTE
EXBR R0.R0
STBR F11.R5 DAYS
LIS R10.15 MASK
NR R10.R5 1 DAY
SRLS R5, 4 10 DAY
MHR R5.R9 (*10)
AHR R5.R10
STH R5.DAY DAYS
AIS R1.4 NEW DATA
L R0.ANC(R1)
LIS R5.0 JUST TO BE SURE
STBR R0.R5 NRC AND TP
LIS R10.3 MASK
NR R10.R5 NRC
AIS R10.1
LHI R12.256
MHR R10. R12 NRC=(NRC+1)*256
LHI R10.256
STH R10.NRC1 STORE
SLHLS 10.2 MULTIPLY BY 4 FOR BYTE COUNT
AIS R10.15
A 10.NRCLBLK+SVC1. SAD
ST 10.NRCLBLK+SVC1. EAD STORE END ADDRESS FOR READ
ST 10.NRCEAD STORE AWAY FOR SECOND READ
NHI R5. X'CO' MASK TP
SRLS R5. 6
STH R5.TP STORE(UNFIXED)
EXBR R0.R0
STBR R0. R5 SECONDS
LIS R10.15 MASK
NR R10.R5 1 SECOND
SRLS R5. 4
NHI R5.7 MASK 10 SECONDS
MHR R5.R9 (*10)
AHR R5.R10 TOTAL SECONDS
STH R5.SECOND STORE
AIS R1.8 MORE DATA
LHL R12. ANC(R1) ELEVATION
NHI R8. X'FFF' ANGLE
STH R8. ELEVAT STORE ANGLE
LM R8. RSAVE

#FORT
NRC=NRC1
GRND=PCTMIN+NRC
REWIND 9
17 ELEVEN=0
CONT
DATA STORING ROUTINE

REALLY LM RSAVE

$FORT

DO 101 I=1,256
   IF (IREF(I).LE.0) GO TO 111
   IREF(I)=IREF(I)*100./256.-DB
   IF(IREF(I).LT.-39) IREF(I)=IREF(I)+100
   IVEL(I)=IVEL(I)+BITVEL/128.
   IDVEL(I)=IVEL(I)-IVELL(I)
   IVEL(I)=IVEL(I)
   C F4WORD=IVAR(I)*BITVAR/256.
   C IVRI=SQRT<F4WORD
   GO TO 101
   00111 IREF(I)=0
   IVEL(I)=0
   IVAR(I)=0
101 CONTINUE
00500 CONTINUE
   ICC=ICC+1
   ELEVA=ELEVAT/11.37778
   IELSN=IFIX(ELEVA)
   T=((DAY*24+HOUR)*60+MINUTE)*60+SECOND
   AZT=AZ/11.37778
   FLAG=FLAG+1
   A=AZT+.01743
   AZCHK=AZT
   IF(K. NE. 1) GO TO 105
   BNGA=AZT
   ENDA=AZT+359.
   105 CONTINUE
   IF(K.LT.180) GO TO 106
   IF(AZT.GT.180.) GO TO 106
   AZCHK=AZT+359.
   106 CONTINUE
   IF(AZCHK.GT. ENDA) K=1
   SINA=SIN(A)
   COSA=COS(A)
   DELTAZ=0.0191987
   NA=K
   RE(258)=0
   IF(NA.EQ.1)NAC=1
   NAC=NAC+1
   IF(NA.GT.1)NAC=0
   IF(NA.EQ.1.OR.AZLAST.LT.-990.) GO TO 224
   IDAY=DAY
   IHR=HOUR
   IMIN=MINUTE
   ISEC=SECONI
   PHI=ELEVA+DAZM
   COSPHI=COS(PHI)
   SINPHI=SIN(PHI)
   COSPH2=COSPHI*COSPHI*EARTH
   CALL TRACK
   223 IF(NSCRN.EQ.4) GO TO 800
   224 CALL CONTOP
   AZLAST=AZT
   800 CALL STRAK
   STOP
SASMS
ALIGN 4
ERROR SVC 2, ERRCODE DECODE ERROR BITS
SVC 2, ERKBLUK
SVC 2, PAUSE
LM 0, RSAVE

*FORT

IF(K.EQ.0) GO TO 10000
IF(K.EQ.1) GO TO 10001
GO TO 1

10003 CONTINUE

*ASSM

ALIGN 4

ERRCODE EQU *
DB 0
DB 6
DC H'0'
DC A(ERCD)
ALIGN 4

ERRBLOK EQU *
DB 0
DB 7
DC H'28'

ERCD DC C'
DC C'I/O ERROR IN RTN RANGE'
ALIGN 4

PAUSE EQU *
DB 0.1
ALIGN 4

ANCBLK DB X'59'
DB 9
DB 0.0
DC A(ANC)
DC A(ANC)
DSF 3

NRCBLK DB X'59'
DB 9
DB 0.0
DC A(ANC)
DC A(ANC)
DSF 3

*FORT

END

*
***LISTING FOR ERT1: INPARM.FTN

$N

BLOCK DATA

C FOR PROGRAM EXTRAD  ERT NO. 162
C VERSION 4.0  LEVEL 780301
C JHW  IBM370
C

C IMPLICIT INTEGER*2 (I-N)
INTEGER*2 N, TLI, TL2, HR, RQUANT, T, TM

COMMON /CDRAYS/ IC(32,10), C(32,9), ID(32,10), D(32,9), IM, JM
COMMON /CONST/ VMISW, DIV, VMAG, VMISWM, ZDIV, ADIV, A1, A2, A3, B1, B2, HDIV
COMMON /VPARM/ VX, VY
COMMON /PNTS/ NCX, NVMIN, NUMX, IELS, NSCAN, IESNL, NVSCN, NT
COMMON /ECONST/ EARTH, VMK
COMMON /NLSTS/ NVAR, NCARM, NVO, ICO, IO, JYR, KTL
COMMON /ZLOOK/ IZOFF, ZARY(91)
COMMON /INTL/ MHSN, MNSN, HM, FNSN
COMMON /DPAR/ DELA

DATA ECL/224*0., VCL/736*0., VR/192*0.,
DATA TL1/3C0/, TL2/-60/,
DATA NOFST/i6/, KOFST/7/, ICLAD/12/,
DATA DRZM/0, 74533.,
DATA RI/0., SA/900.,
DATA NVARM/32/, MNSN/5/, MHSN/7/, HM/6./5/
DATA TM/0., NCARM/16/, VMK/1., E-3/
DATA IM/32/, JM/9/, NVO/0., ICO/0., IO/0., JYR/0.
DATA VMISW/5., DIV./2., VMAG/01., ZDIV./1., HDIV./5./
DATA ADIV./84./, A1./4., A2./3., A3./3., B1./7., B2./3./
DATA VX/0., VY/0., IESNL/0., IELS/0.,
DATA EARTH/6. 4857 E-5., NT/0., NSCAN/0., NVSCN/1.
END
SUBROUTINE CONTOR

**** SUBROUTINE CONTOR

JHN AFGL SUDBURY RADAR SUBROUTINE

VERSION 4.1 LEVEL 78117

FIND EVENTS ALONG SINGLE RADIAL

******************************************************************************

IMPLICIT INTEGER*2 (I-N)

INTEGER *2 W, TL1, TL2, HR, RQUANT, TC
+
, T, IC21(22), IC22(22)
+
IPC2(5400), IPC3(5400), IPCNT(1900), IPTC(44),
+
NEMB, NEMC, NAC

COMMON /SWITCH/ IC1(44), IC2(44), TC(1900), IPC1(5400),
+
IPC2(5400), IPC3(5400), IPCNT(1900), IPTC(44),
+
NEMB, NEMC, NAC

COMMON/AZM/ AZMUTH, AZLAST, AZSTAR, NA, ELEVAT

COMMON /AZ2/SINAZ, COSAZ, DELTAZ, ISCANF, NEL, RI, SA

COMMON/FILTER/TATRMN, AREAMN, DAZM

COMMON /FIXED/ NPA, IEMAX, NFC, IEM, JEM

COMMON /PRSTOR / NUP, TATR(1400), NUMAX, IACT(70),
+
IDC(22), IPNG(34), KAMAX, MXTR

COMMON/REFL/ W(1025), HR(258), NCL, NID, NIDP, INCL

COMMON /PWORK/ KMAX, TCBO), JMXDB, JMAX, IAMAX, JR, JMAX, JMAX, JMAX

COMMON /THRESH/LDV

COMMON /CONPIC/ NCEL, NMR

********************************************************************************

* DATA RPD/. 017453/

********************************************************************************

CALL CONMSG(6, 'CONTOR')

IF(IFLAG. EQ. I) WRITE(3,7)(W(IX), IX=1,102)

FORMAT(IX, 2015)

IF (NA.NE.1) GO TO 61

   INITIALIZE.

   61 CONTINUE
   DO 101 K=1, IEMAX
   IDC(K)=0
   101 CONTINUE
   DO 102 J=1, JMAX
   IPNG(J)=0
   102 CONTINUE
   NEMB=NEMC
   NEMC=NAC*IEMAX
   NEM1=NEMC+1
   IEM=0
   IEM2=0
   IP=0
   IPB=0

   FIND EVENTS

   DO 281 I=2, NCL
   IF(W(I).GT.TLS) GO TO 2311
   W(I)=0

D-15
2311 IF(RQUANT.GT.1)W(I)=W(I)/ROURNT
IF (W(I).GT.TL1) GO TO 131
GO TO 241
131 IF (W(I-1).LE.TL1) GO TO 141
GO TO 151
141 IEM=IEM+1
IEA=IEA+NEMC
IF(IEA.LT.IEMAX)GO TO 1411
WRITE(3,1412)IEMAX,K
1412 FORMAT(1X,39HEVENT COUNTER EXCEEDED MAX VALUE, IMAX=, I6, 5X, I4)
IEM=IEMAX
1411 IC1(IEA)=I-1
IC2(IEA)=0
C
C
PEAK DETECTION, LOCATE AND COUNT PEAKS.
C
151 IF (W(I)-W(I-1)) GT 171,181,161
161 IPB=I-1
GO TO 181
171 IF (IPB.EQ.0) GO TO 181
IP=IP+1
IF(IP.LE.JMAX)GO TO 1711
WRITE(3,1913)IP, IEVENT
1913 FORMAT(1X,17HN PEAKS EXCEEDED, 2I6)
IP=JMAX
GO TO 181
1711 IPRNG(IP)=(I+IPB)/2
IPB=0
181 CONTINUE
GO TO 282
241 IF (W(I-1).LE.TL1) GO TO 281
IC2(IEA)=I-1
C
C
KEEP COUNT OF PEAKS WITH EVENT.
C
IF (IPB.EQ.0) GO TO 251
IP=IP+1
IF(IP.LE.JMAX)GO TO 242
WRITE(6,1913)IP, IEM
IP=JMAX
GO TO 243
242 IPRNG(IP)=(I+IPB)/2
243 IPB=0
251 IDC(IEM)=IP
282 IF(W(I) LE TL2) GO TO 2412
IF(W(I-1) GT TL2) GO TO 281
IEM2=IEM2+1
IF(IEM2.GT.IEMAX)IEM2=IEMAX
IC21(IEM2)=I-1
IC22(IEM2)=0
GO TO 281
2412 IF(W(I-1) LE TL2) GO TO 281
IC22(IEM2)=I-1
281 CONTINUE
CALL PEAKD
C
C
STORE PRESENT PARAMETERS IN PREVIOUS PARAMETERS.
C
C
IF(IEM.EQ.1) GO TO 882
C
WRITE(6.1) AZMUTH, TL2, IEM2, (IC21(J), IC22(J), J=1, IEM2)
C
881 WRITE(8.2,218,(/1(216.3X)/1(216.3X))
C
JEM=IEM
RETURN
END
SUBROUTINE PEAKD

C ***************************************************************************************************************** C
C VERSION 5.0  LEVEL 780616
C JHW  AVCO  IBM360
C DETERMINES PEAK VALUES AND THEIR ATTRIBUTES.
C *****************************************************************************************************************

IMPLICIT INTEGER*2 (I-N)
REAL UP(6), TATR(1400), BUF(8)
INTEGER*4 RSAVE(16)
INTEGER *=2 TCVL, TBVL, TCVM, TATC, TATM
INTEGER *=2 HB, IACT(70), IDC(22), IPCRNG(34), TM
INTEGER *=2 W, RQUANT
INTEGER+2 TL2, T, TC
COMMON /DATA1/ ECL(224), NOFST, KOFST, ICLADJ, NANI
* COMMON /SWITCH/ IC1(44), IC2(44), TC(1980), IPC1(5400),
+ IPC2(5400), IPC3(5400), IPCNT(1980), IPTC(44),
+ NEMB, NEMC, NAC
COMMON /NWORK/ KMAX, T(80), JMXDB, JMAX, IAMAX, JR, IMXJMIX
COMMON/REFL/ W(1025), HB(258), NCL, NIDP, INCL,
X IMX, IMN, TM, TL2, RQUANT, IDVEL(258)
COMMON /FIXED/ NPA, IEMAX, NFC, IEM, JEM
COMMON/AZM/ AZMUTH, AZLAST, AZSTAR, NA, ELEVAT
COMMON /AZ2/ SAZ, CAZ, DAZ, SCANF, NEL, RI, SA
COMMON/FILTER/TATRMN, AREAMN, DAZM
COMMON /THRESH/ LDB
COMMON /PRSTOR/ NUP, TATR(1400), NUMAX, IACT(70),
+ IDC(22), IPCRNG(34), KAMAX, MXTR
COMMON /CONPK/ NCELL, NMX
COMMON /THRESH/ LDB
COMMON /FILTER/ TATRMN, AREAMN, DAZM

IEM IS NO. OF EVENTS IN C RADIAL.
INITIALIZE AND GENERATE HC ARRAY

NBADR=NCADR
NCADR=IAMAX+NAC
NBK=NCRA
NCKA=KAMAX+NAC
NAX=NA
ITY=0
IF(NA. NE. 1) GO TO 2109
LM=6
LMDP=LM*NIDP
NBADR=IAMAX+NAC
NBK=KAMAX+NAC
NAN=0
NANI=0
LMM=LM-1
IDX=LM+1
NIMN=1
FQUANT=RQUANT
NCLM=NCL-1
LDBM=LDB-1
NUMP=2+LM*LDB
LDX=NUMP-1
NPD=LDX*NIDP
ID2=1+(LDB-1)*LM
DO 17 1=1, KOFST
17 UP(I)=0
DO 18 I=1,LIMT
18 ECL(I)=0.
   DO 19 I=1,NCL
19 IDVEL(I)=0

NMX=1
   DO 2107 I=1,NIDP
2107 IACT(I)=0
   DO 2108 J=1,MXTR
2108 TATR(J)=0.
2109 NGM=0
   DO 23 K=1,KMAX
   NMX±I
2±07 II,
2±07 NIDP
2±07 IRCT(I)nS
2±08 J-MXTR
2±09 NGM=O
   DO 23 K-I, KMAX
   IF(IEM.LE.0)GO TO 
197
   DO 951 IE=I, IEM
951 IER=1+NEMC
   KIE=(IE-1)*KMAX+NCKA
   KIEM=KIE-KMAX
   IPTC(IER)=0
   ICST=ICST+IER
   ICESP=ICST+IER
   IF(IE.EQ.1)GO TO 232
   DO 233 K=1,KMAX
   KA=K+KIE
   KB=K+KIEM
   ICNCT(KA)=ICNCT(KB)
232 IPL=0
   IF(IE.GT.1)IPL=IDC(IE-1)
   IP=IDC(IE)
   IF(IP.LT.IPL)GO TO 951
   IPL=IPL+1
   JE1=0
   JE2=0
   FIND B EVENTS ASSOCIATED WITH C EVENTS.
   JEM IS NO. OF EVENTS IN PREVIOUS RADIAL.
   IF(JEM .EQ. 0) GO TO 41
   DO 31 JE=1,JEM
31 JEB=JE+NEMB
   IF(IC2(JEB).LT. ICST) GO TO 31
   IF(IC1(JEB).GT. ICESP) GO TO 41
   JE2=JE
   IF(JE1.EQ.0) JE1=JE
   CONTINUE

FIND THRESHOLDS FOR IE EVENT
   DO 51 J=1,JMXDB
51 T(J)=0
   NTHRES=1
   DO 71 L=IPL,IP
   IF(L.GT.JMAX)GO TO 71
   IR1=IPCRNG(L)
   IF(IR1.LT. ICST) GO TO 71
   IF(IR1.GT. ICESP) GO TO 712
   DO 711 K=1,LD
   IU=W(IR1)
   IT=IU-TH-K+1
   IF(IT.LE.0)GO TO 711
   IF(IT.GT.JMXDB)IT=JMXDB
   IF(IT.EQ.0)NTHRES=NTHRES+1
   T(IT)=1
711 CONTINUE
   CONTINUE
IF (NTHRES > KMAX) WRITE (6, 7878) NA, NTHRES, KMAX, IE
7878 FORMAT (2X, 13H NUMBER OF THRESHOLDS EXCEEDS KMAX, 41I10)
IF (NTHRES > KMAX) IPSRT = NTHRES - KMAX
IPT = 1
DO 91 L = 1, JMxDB
IF (T(L) LE. 0) GO TO 91
KA = IPT + KIE
TC(KA) = L + TM - 1
IPSRT = IPSRT - 1
IF (IPSRT GT. 0) GO TO 91
IPT = IPT + 1
91 CONTINUE
IF (IPT GT. JR) WRITE (6, 7879) NA, IPT, JR, IE
7879 FORMAT (2X, 13H NUMBER OF THRESHOLDS EXCEEDS JR, 41I10)
IF (IPT LE. JR) IPT = JR
IPSRT = IPT
IF (IPSRT LE. 0) GO TO 951
C LOOP ON RANGE IN IE EVENT TO FIND CONTOUR
IBGN = ICEST + 1
IND = ICESP + 1
DO 161 I = IBGN, IND
  C DO ON THRESHOLD
  C LOOP ON THRESHOLD
  DO 131 K = 1, IPT
    IU = W(I)
    KA = K + KIE
    IF (IU GT. TC(KA)) GO TO 111
    GO TO 141
  111 IU = W(I - 1)
    IF (IU LE. TC(KA)) GO TO 121
    GO TO 131
  C START RANGE FOR SEGMENT (CONTOUR)
  121 IPCNT(KA) = IPCNT(KA) + 1
    IF (IPCNT(KA) LE. IMXXM) GO TO 1211
    WRITE (6, 1212) ITY, K, IE
  1212 FORMAT (2X, 13H NUMBER OF SEGMENTS EXCEEDS IMX, 31I10)
    IPCNT(KA) = IMXJM0
  1211 IPE = IPCNT(KA)
    IREG = 1 - 1
    IADDR = IPE + (K - 1) * IMXJM0 + NCADR
    IPC1(IADDR) = IREG
    IPC3(IADDR) = 0
  131 CONTINUE
    GO TO 161
  C END RANGE FOR SEGMENT
  141 DO 151 KL = K, IPT
    IF (W(I - 1) EQ. -999) GO TO 161
    IU = W(I - 1)
    KA = KL + KIE
    IF (IU LE. TC(KA)) GO TO 161
    IPE = IPCNT(KA)
    IREG = 1 - 1
    IEOL = IPE + (KL - 1) * IMXJM0 + NCADR
    IPC2(IEOL) = IREG
  151 CONTINUE
  161 CONTINUE
C ASSOCIATE CELLS LOOP ON THRESHOLD HIGHEST TO LOWEST
DO 941 LC=1, IPT
   KL=IPT-LC+1
   KR=KC+KE
   KZ=KC+KIE
   IF(KC LE 0) GO TO 941
   TCVL=TC(KR)
   TCVM=TCVL+1
   TCVLB=TCVL+LDB
   NPC=IPCNT(KR)
   NPL=0
   IF(IE GT 1) NPL=IPCNT(KZ)
   IF(NPC LE NPL) GO TO 941
   NPL=NPL+1
   LOOP ON SEGMENTS
   DO 931 IPE=NPL, NPC
      JADDR=IPE+(KC-1)*IMXM+NCADR
      IHBM=IPC1(JADDR)
      IHD=IPC2(JADDR)
      K=KC+1
      KY=KA+1
      KX=KZ+1
      NPK=0
      TATM=0
      LPE=IPCNT(KY)
      LPL=0
      IF(IE GT 1) LPL=IPCNT(KX)
      LPL=LPL+1
      IF(LPE LT LPL OR K GT IPT) GO TO 193
   LOOP SEGMENTS, NEXT HIGHER THRESHOLD
   DO 191 L=LPL, LPE
      JADDR=L+(K-1)*IMXM+NCADR
      IF(IPC2(JADDR) LT IHBM) GO TO 191
      IF(IPC1(JADDR) GT IHD) GO TO 193
      NPCF=IPC3(JADDR)
      IF(NPCF LE 0) GO TO 1911
      TATC=TATR(NPCF)
      IF(TATC LT TATM) GO TO 231
      TATM=TATC
      NPK=NPCF
   IF(TATC LT 0) TATC=-TATC
   NPCEL IS FOR NEXT HIGHER (ENCLOSED) THRESHOLD ON C RADIAL
   IF(TATC GT TCVLB) GO TO 932
   CONTINUE
   GO TO 193
   932 NPK=-NPCEL
   GO TO 193
   1911 NPK=(-NIDF+1)
   ASSOCIATE CELLS ON B RADIAL, TOP DOWN
   MPK=0
   IF(NA EQ 1) GO TO 361
   TATM=0
   IF(JE2 EQ 0) GO TO 371
   DO 261 JE=JE1, JE2
   JEB=JE+NEMB
   ITATM=0
   IF(IC2(JEB) LT IHBM) GO TO 261
   IF(IC1(JEB) GT IHD) GO TO 3661
C
271  IPB=IPTC(JEB)
    IF(IPB.LE.0) GO TO 261
    DO 291 LB=1,IPB
         KB=IPB-LB+1
         KBB=(JE-1)+KMAX+NPKA
         KBA=KB+KBB
         KBC=KB+KBB-KMAX
         TBVL=TC(KBA)
         JEOL=TBVL+1
    IF(IPIB.LT.9)GO TO 261
    IF(IPB.LE.NPI=IPCNT(KBA)
       IF(IPB.LE.NP1)GO TO 291
       IF(IPB.GT.1)NP1=IPCNT(KBC)
       IF(NP2.LE.NP1)GO TO 291
       IF(NP2.GT.NP1)NP1=NP1+1
       DO 281 JPE=NP1,NP2
           IADDR=JPE+(KB-1)+IMXJMX+NBAJR
           IF(IPC2(IADDR).LT.IHEH)GO TO 281
           IF(IPC1(IADDR).GT.IHED)GO TO 291
           LPCEL=IPC3(IADDR)
           IF(LPCEL.LE.0)GO TO 281
           IF(TCVM.LE.TBVL)GO TO 282
           IEOL=TATR(LPCEL)
           IF(JEOIL.LT.IEOL)GO TO 261
       281 CONTINUE
    291 CONTINUE
    2911 IF(ITATM.EQ.0)GO TO 261
    291 CONTINUE
    261 CONTINUE
    3661 IF(MPK.EQ.0)GO TO 371
    IF(TATM.GT.TCVLB)MPK=-MPK
    GO TO 421
    371 DO 194 I=1,HB.1HD
       IF(IABS(HB(I)).EQ.-999)GO TO 194
       IF(IABS(HB(I)).LE.TC(KB)).GO TO 194
       IF(NPK.EQ.0)GO TO 366
       IF(NPK.GT.0)GO TO 366
       GO TO 3662
       194 CONTINUE
C
C HAVE B COMPARE WITHIN RANGE
C
C 361 CONTINUE
    IF(NPK.EQ.0)GO TO 631
C
C MPK=0 AND NPK=0 - NO COMPARE
C MPK=0 AND NPK NE 0 - NO B COMPARE
C NPK=0 AND MPK NE 0 - B COMPARE
C HIGHEST THIS RADIAL
C
C IF(NPK.LE.0. OR. NPK.GT.NMX)GO TO 3662
C
C NO PRIOR RADIAL FOR COMPARISON, INCREMENT NPCEL
C
C NPCEL=NPK
    359 INDX=TATR(NPCEL)-TCVM
    IF(INDX.GE.LDS OR. INDX.LE.0)GO TO 366
    392 IN=INDX+LM+1
    INX=INDEX+INDEX+LM
1 IF(IPG EQ 0) OR IF LE 1 GO TO 3664
  DO 3665 I=1,IE
  I=I+NEMC
  IPTT=IPTC(I)
  IF(IPTT LE 0) GO TO 3665
  DO 3666 KTI=1,IPPT
  KTI=I-KTI
  KEB=KTI-KMAX
  NPCT=IPCNT(KTA)
  IEOL=TC(KTA)+1
  NPCL=0
  IF(I GT 1) NPCL=IPCNT(KTB)
  IF(NPCT LE NPCL GO TO 3666
  NPCL=NPCL+1
  DO 3667 LP=NPCL NPCT
  IADDR=LP(KI+1)+lmx(mx+ncadr
  IF(LPCEL NE IPCS IADDR) GO TO 3667
  INDXT=TATR(NPCEL)-IEOL
  IF(INDXT LT LDB) GO TO 3668
  3669 IPCS(IADDR)=0
  GO TO 3667
  3668 IF(INDX GE LDB) GO TO 3669
  IPCS(IADDR)=NPCEL
  3667 CONTINUE
  3666 CONTINUE
  3665 CONTINUE
  IF=0
  3664 IF(INDX GE LDB) GO TO 365
  IACT(LPCEL)=NPCEL
  IEOL=LPCEL(INDX+LM+1)*NIDP
  TATR=IEOL-NPCEL
  IF(INDX NE 0) GO TO 365
  IACT(LPCEL)=NIDP-1
  LP=LPCEL+NIDP
  TATR=LIN=0
  365 CONTINUE
  GO TO 921
C C C C C
  COMBINE NPCEL AND LPCEL PEAK VALUES EQUAL
C C C C C
  COMBINE WITH F RADIAL CELLS
C C C C C
  421 IF(NPK LE 0) GO TO 422
  IF(NPK LT 0) GO TO 3662
  NGM=0
  LPCEL=LPK
  LNX(LPCEL+LMDF
  IEOL=TATR(LNX)
  KEBM=KEM+JEM+1)*KMAX+NKBK
  IF(I=IEOL EQ NA AND NPK EQ 0 AND TCYL GT + TC(EBM) GO TO 485
  INDEX=TATR(LPCEL)-TCY
  IMIXC=INDEX
  IF(NPK GT 0) INDEX=TATR(NPK)-TCY
  IF(INDEX LE INDEX) GO TO 4212
  NGM=1
  NPCEL=NPK
  IND=INDEX
  INDEX=INDEX
  IMNG=IND
  GO TO 4213
  4212 IF(INDEX LT 0) GO TO 461
  NPCEL(LPCEL
  IMNG=IND
  GO TO 4213
IF(INDX. GE. IND) GO TO 4221
INDX = INDX * LM
INL = (IN + 1) * NIDP
ILN = (IN + LM) * NIDP
IEQL = NPCEL + INL

IF(TATR(IEQL). NE. 0.) GO TO 5311
IEQL = NPCEL + IN1
JEQL = TATR(IEQL)
IF(JEQL. LE. 0. AND. NGM. EQ. 0) GO TO 4221
IF(NGM. NE. 1) GO TO 5312
IM = INDX * LM
IF(IM. LT. 0) GO TO 5311
IF(LPCEL. LE. 0. OR. LPCEL. GT. NMX) GO TO 4221
IEQL = LPCEL + (IN + LM) * NIDP
IF(TATR(IEQL). NE. 0.) GO TO 5311
IEQL = LPCEL + (IN + 1) * NIDP
IF(TATR(IEQL). GT. 0.) GO TO 5313
LPCEL = NPCEL
GO TO 4221

IF(LPCEL. EQ. NPCEL. OR. LPCEL. GT. NMX) GO TO 4221
IMDX = TATR(LPCEL) - TCVM
GO TO 5314

IEQL = NPCEL + IN1
NPCEL = TATR(IEQL)
IF(NPCEL. LE. 0. OR. NPCEL. GT. NMX) GO TO 4221
INDX = TATR(NPCEL) - TCVM
GO TO 5311

IPC3(JADDR) = NPCEL
IEQL = NPCEL + IN1
IF(TATR(IEQL). EQ. -999.) GO TO 8012
IST = IHB
ISP = IH2
IEQL2 = IEQL + NIDP
IEQL3 = IEQL2 + NIDP
IEQL4 = IEQL3 + NIDP
IEQL5 = IEQL4 + NIDP
DO 531 I = IST, ISP

R = RI + SA * (FLOAT(I - 1) - .5)
RU = R * FLOAT(IU * R
TATR(IEQL) = TATR(IEQL) + DAZ + R
TATR(IEQL2) = TATR(IEQL2) + RU
TATR(IEQL3) = TATR(IEQL3) + SA * R * RU
TATR(IEQL4) = TATR(IEQL4) + CAZ * R * RU
TATR(IEQL5) = TATR(IEQL5) + IDVEL(I)

CONTINUE

IF FIRST COMBINE, AREA=0, IF SECOND OR HIGHER, AREA=-1.
TEST AREA TO ESTABLISH NEW NUMBERS

INDEX = -INDEX
IND = NUMP - 1

COMBINE WITH B - RADIAL, C-LEVEL HIGHER
INS=2
IPG=0

TATR\(LPCEL)=TCVM
LMP=LPCEL\*NPDP
TATR\(LMP)=IE

IF\(INDEX\ GE\ LDB\)GO\ TO\ 482
IND=LDB-INDEX

DO \(4832\) I=INDEX, LDB
IEOL=LPCEL+(I+1)*LMDP
JEO\L=TATR\(IEO\L\)
IF\(JEOL\ EQ\ NA\)IPG=IPG+1

DO \(4832\) I=1, IND
DO \(4833\) J=I+LM
IN=LPCEL+(J+\(LDB-1\)+LM)+NIDP
IM=LPCEL+(J+\(INDEX-1\)+LM)+NIDP

\(TATR(IN)=TATR(IM)\)
IND=INDEX+LM+1
INDP=INDEX

DO \(4832\) I=1, LDB
IEOL=LPCEL+I\*LMDP
JEO\L=TATR\(IEO\L\)
IF\(JEOL\ EQ\ NA\)IPG=IPG+1

CONTINUE

DO \(4833\) I=INS, IND

IF\(IPG\ EQ\ 0\)OR\ IE\ LE\ 1\)GO\ TO\ 486
DO \(4831\) I=1, IE
IA=I+NEMC
ITP=IPTC\(IA\)
IF\(ITP\ LE\ 0\)GO\ TO\ 4831

DO \(4833\) KT=I, IPTT
KTP=(I-1)*KMAX+NCKA
KTA=KT+KTP
KTB=KT+KTP-KMAX
NPCT=IPCNT\(KTA\)
IEQL=TC\(KTA\)+1
NPCL=0
IF\(I\ GT\ 1\)NPCL=IPNCT\(KT\)
IF\(NPCT\ LE\ NPCL\)GO\ TO\ 4833
NPCL=NPCL+1

DO \(4834\) LP=NPCL\*NPCT
KADDR=LP+(I-1)+IMXJM\*NADDR
IF\(LPCEL\ NE\ IPC3\(KADDR\)\)GO\ TO\ 4834

INDX=TATR\(LPCEL\)-IEOL
IF\(INDX\ LT\ LDB\)GO\ TO\ 4834
IPC3\(KADDR\)=0

CONTINUE

CONTINUE

CONTINUE

IPG=0

DO \(488\) I=NIMN, NIDP
IF\(I\ GT\ 0\)GO\ TO\ 487
WRITE(6,644)
GO TO 931
487  LPCEL=I
     NIMN=I+1
     IACT(I)=1
     IF(NIMN GT NIDP) NIMN=NIDP
     IF(NMX LT NIMN) NMX=NIMN
     TATR(LPCEL)=TCVM
     LNP=LPCEL+NPDP
     TATR(LNP)=1E
     GO TO 488
422  LPCEL=MPK
     IF(LPCEL LT 0)LPCEL=-LPCEL
4221 IF(LPCEL GT NMX OR LPCEL LE 0)GO TO 3662
     INDX=TATR(LPCEL)-TCVM
     IF(INDX LT 0)GO TO 632
     DO 441 JE=JE1,JE2
     JEB=JE+NEMB
     IF(JC2(JEB) LT IHEM) GO TO 441
     IF(JC1(JEB) GT IHD) GO TO 632
     IPB=IPTC(JEB)
     IF(IPB LE 0)GO TO 441
     DO 471 LB=1,IPB
     KE=IPB-LB+1
     KBJ=(JE-1)KMAX+NPKA
     KBA=JE+KBJ
     KBC=JE+KBJ-KMAX
     IF(TC(KBA) NE TCVL) GO TO 471
     MPB=IPCNT(KBA)
     MPL=0
     IF(JE GT 1)MPL=IPCNT(KBC)
     IF(MPB LE MPL)GO TO 471
     MPL=MPL+1
     DO 461 JPE=MPL,MPB
     IADDR=JPE+(KB-1)IMXJMX+NPKADR
     IF(IPC2(IADDR) LT IHEM)GO TO 461
     IF(IPC1(IADDR) GT IHD)GO TO 471
     NPCEL=IPC3(IADDR)
     IF(NFCEL LE 0 OR NPCEL GT NMX)GO TO 461
     IF(LPCEL EQ NPCEL)GO TO 461
     COMBINE AT TB=TC LEVEL
     C
     C
502  INDX=TATR(NFCEL)-TCVM
     IF(INDX GE LDB)GO TO 461
     IF(INDX LT 0)GO TO 8511
     IF(INDX LT LDB)GO TO 861
851  ND=INDX*LM+1
     DO 852 I=1,LM
     IEQL=NPCEL+(ND+I-1)*NIDP
852  TATR(IEQL)=0
     IPC3(IADDR)=0
     GO TO 461
8511 IPC3(IADDR)=0
     DO 8512 J=1,LDX
     JN=NPCEL+J*NIDP
8512 TATR(JN)=0
     IACT(NPCEL)=-(NIDP+1)
     GO TO 461
861  LD=INDX*LM+1
     ND=INDX*LM+1
     LDR=LD*NIDP
     NDR=ND*NIDP
     IEQL=LPCEL+(LD+LMM)*NIDP
     JEQL=NPCEL+(ND+LMM)*NIDP
\[
\text{if (TATRL NE. 0. AND. TATRN NE. 0.) GO TO 8911}
\]
\[
\text{JEQL=LPCEL+LDA}
\]
\[
\text{TATRJ=TATR(JEQL)}
\]
\[
\text{if (TATRL EQ. 0. AND. TATRJ LE. 0.) GO TO 851}
\]
\[
\text{if (TATRL GT. 0.) GO TO 8912}
\]
\[
\text{LPCEL=TATRJ}
\]
\[
\text{if (LPCEL LE. 0. OR. LPCEL GT. NMX) GO TO 461}
\]
\[
\text{GO TO 4221}
\]
\[
8912 \text{IEQL=NPCEL+NDA}
\]
\[
\text{TATRJ=TATR(IEQL)}
\]
\[
\text{if (TATRN EQ. 0. AND. TATRJ LE. 0.) GO TO 8913}
\]
\[
\text{NPCEL=TATRJ}
\]
\[
\text{if (NPCEL LE. 0. OR. NPCEL GT. NMX. OR. NPCEL EQ. LPCEL) GO TO 461}
\]
\[
\text{IPCC3(IADDR)=NPCEL}
\]
\[
\text{GO TO 502}
\]
\[
8913 \text{DO 8914 I=1,LM}
\]
\[
\text{IEQL=LPCEL+(LD+I-1)*NIDP}
\]
\[
8914 \text{TATR(IEQL)=0.}
\]
\[
\text{IPCC3(IADDR)=0}
\]
\[
\text{GO TO 4221}
\]
\[
8911 \text{IBNDRY=0}
\]
\[
\text{IEQL=LPCEL+LDA}
\]
\[
\text{JEQL=NPCEL+NDA}
\]
\[
\text{if (TATR(IEQL) EQ. -999. OR. TATR(JEQL) EQ. -999.) X IBNDRY=1}
\]
\[
\text{DO 891 I=1,LM}
\]
\[
\text{IEQL=LPCEL+(LD+I-1)*NIDP}
\]
\[
\text{JEQL=NPCEL+(ND+I-1)*NIDP}
\]
\[
\text{if (IBNDRY EQ. 0) TATR(IEQL)=TATR(JEQL)+TATR(IEQL)}
\]
\[
\text{TATR(JEQL)=0.}
\]
\[
891 \text{CONTINUE}
\]
\[
\text{IEQL=LPCEL+LDA}
\]
\[
\text{if (IBNDRY EQ. 1) TATR(IEQL)=-999}
\]
\[
\text{IEQL=NPCEL+(ND+LMM)*NIDP}
\]
\[
\text{TATR(IEQL)=0.}
\]
\[
\text{IEQL=NPCEL+NDA}
\]
\[
\text{TATR(IEQL)=LPCEL}
\]
\[
\text{IACI(NPCEL)=-LPCEL}
\]
\[
\text{IPCC3(IADDR)=LPCEL}
\]
\[
461 \text{CONTINUE}
\]
\[
471 \text{CONTINUE}
\]
\[
441 \text{CONTINUE}
\]
\[
632 \text{IF (NPK. LE. 0) GO TO 3662}
\]
\[
\text{NPCEL=LPCEL}
\]
\[
\text{GO TO 366}
\]

C
C
C
C
631 \text{DO 642 J=NIMN, NIDP}
\]
\[
\text{IF (IACI(J) EQ. 0) GO TO 643}
\]
\[
642 \text{CONTINUE}
\]
\[
\text{WRITE(6,644)}
\]
\[
644 \text{FORMAT(5X,15H TOO MANY CELLS)}
\]
\[
\text{GO TO 931}
\]
\[
643 \text{NPCEL=J}
\]
\[
\text{NIMN=J}
\]
\[
\text{IF (NIMN GT. NIDP) NIMN=NIDP}
\]
\[
\text{IF (NMX. LT. NIMN) NMX=NIMN}
\]
\[
\text{IACI(J)=1}
\]
\[
\text{IPCC3(IADDR)=NPCEL}
\]
\[
\text{DO 671 I=1,NUMP}
\]
\[
\text{NMP=NPCEL+(I-1)*NIDP}
\]
\[
\text{TATR(NMP)=0. 0}
\]
\[
671 \text{CONTINUE}
\]
NMN=NPCEL+NPDP
TATR(NMN)=IE
IST=IHB
ISP=IHD
NP2=NPCEL+IDP
NP3=NP2+IDP
NP4=NP3+IDP
NP5=NP4+IDP
NP6=NP5+IDP
DO 621 I=IST, ISP
R=RI+SA*(FLOAT(I-1)-.5)
IU=W(I)
RU=R*FLOAT(IU)*DAZ
TATR(NP2)=DAZ*R+TATR(NP2)
TATR(NP3)=RU+TATR(NP3)
TATR(NP4)=SAZ*R+RU+TATR(NP4)
TATR(NP5)=TATR(NP5)+CAZ*R+RU
TATR(NP6)=TATR(NP6)+IDVEL(I)
CONTINUE
NIX=NPCEL+LMID
TATR(NIX)=NA
IF(IST .EQ. 2 .OR. ISP .EQ. IMX) TATR(NP2)=-999.
CONTINUE
CONTINUE
CONTINUE
C
CLEAN UP TATR AND IC ARRAYS - REMOVE IC POINTER
TO DELETED ARRAYS
C
DO 9512 I=1,NMX
IF(IACT(I).EQ.0)GO TO 9512
IF(IACT(I).GE.0)GO TO 9611
DO 9613 IE=1,IEM
IEA=IE+NEMC
KIE=(IE-1)*KMAX+NCKA
KIEM=KIE-KMAX
IPT=IPTC(IEA)
IF(IPT.LE.0)GO TO 9613
DO 9618 K=1,IPT
KA=K+KIE
KB=K+KIEM
NPC=IPCNT(KA)
TCVM=TC(KA)+1
NPL=0
IF(IE.GT.1)NPL=IPCNT(KB)
IF(NP.LE.NPL)GO TO 9618
NPL=NPL+1
DO 9619 IPE=NPL,NPC
IAADDR=IPE+(KC-1)*IMX+JMX+NCADR
IF(I .NE. IPC3(IADDR)) GO TO 9619
IF(IACT(I).LT.-NIDP)GO TO 9614
INDEX=TATR(I)-TCVM
IEQL=I+(INDEX+1)*LMID
JEQL=I+(INDEX+LM+1)*NIDP
IF(TATR(IEQL).NE.0)GO TO 9619
IEQL=TATR(JEQL)
IF(IEQL.NE.-IACT(I))GO TO 9614
IPC3(IADDR)=-IACT(I)
GO TO 9619
9614 IPC3(IADDR)=0
9619 CONTINUE
9618 CONTINUE
9613 CONTINUE
IF(IACT(I).GE.-NIDP)GO TO 9517
IACT(I)=0
IATR(I)=0.
GO TO 9512

9517 DO 9513 J=1,LDB
KEQL=I+(LM*(J-1)+1)*NIDP
JEQL=IATR(KEQL)
IEQL=I+J*LMDP
IEQL=IATR(IEQL)
IF( (JEQL).EQ.-IAC(T(I)).AND. IEQL.EQ.0 )
+ GO TO 9514

9513 CONTINUE
GO TO 9611

9514 TATR(KEQL)=0

9611 DO 9612 K=2,LDB
IEQL=I+K*LMDP
JEQL=I+(K-1)*LM+1)*NIDP
IF( TATR(IEQL).NE.0 .OR. TATR(IEQL).EQ.0 .)
+ TATR(IEQL)=0.

9612 CONTINUE
IACT(I)=1
9512 CONTINUE
IF(NA.EQ.1)GO TO 1030

C END OF ASSOCIATION LOOPS
C

952 DO 991 I=1,NMX
IA=I+(LDX-1)+*NIDP
IF(IACT(I).EQ.0)GO TO 991

961 IF( TATR(IA).EQ.0 )GO TO 9912
IEQL=TATR(IA)
IF(IEQL.LE.0)IEQ=-IEQL
IF(IEQL.EQ.NAX-1)GO TO 971
GO TO 991

C CHECK Background Coming Down
C

971 INBR=0
ITERM=1
DO 9716 J=1,LDBM
IEQL=I+(J-1)*LM+1)*NIDP
JEQL=IEQL+NIDP
IF( TATR(IEQL).LE.0 .OR. TATR(IEQL).EQ.0 )GO TO 9982

9716 CONTINUE
NMP=I+NPDP
IEQL=TATR(NMP)
J=1
IF(JM.LT.2)GO TO 968
DO 9711 J=1,JM

968 JA=J+NEMB
IF(IEQL.NE.J)GO TO 9711

9712 IPB=IP(TC(JA))
DO 9713 K=1,IPB
KA=(J-1)+KMAX+NBKA
KAP=K+KA
KAM=KA-KMAX
IEQL=IATR(I)
IEQL=IEQL-TC(KAP)
IF(IEQL.NE.LDB)GO TO 9713
NP=IPCNT(KAP)
NL=0
IF(J.GT.1)NL=IPCNT(KAM)
NL=NL+1
DO 9713 N=NL,NP
IEQL=N+(K-1)+IMXJM+NBADR
IF(I.NE.IPC3(IEQL))GO TO 9713
INBR=INBR+1
ITERM=1
IST=IPC1(IADDR)  
ISP=IPC2(IADDR)+1  
DO 9715 I=IST, ISP  
IU=W(L)  
IF(IU.EQ.-999)GO TO 9715  
IF(ABS(IU).GT.TC(KAP))GO TO 9982  
9715 CONTINUE  
9711 CONTINUE  
*TERM=3  
IF(INBR.EQ.0)GO TO 9982  
*I=I+ID2*NIDP  
IF(TATR(IID).LE.TATRMIN)GO TO 9982  
DO 981 J=1,LMM  
IEGL=I+(ID2+J-1)*NIDP  
981 UP(J)=TATR(IEQL)  
UP(2)=UP(2)/UP(1)  
UP(6)=A*UP(2)+UP(5)  
DO 985 M=1,NOST  
MG=6+(M-1)*KOFST+NAN1  
IF(UP(M).GT.ECL(MG))GO TO 986  
985 CONTINUE  
GO TO 989  
986 N=1+NAN+1  
IF(NAN.GT.1)NAN=0  
NAB=NAN1  
NAN1=NAN+ICLAD  
LMT=NOFST-1  
DO 988 J=1,LMT  
KJ=(J-1)*KOFST  
DO 987 K=1,LMM  
JK=K+KJ  
987 ECL(JK)=ECL(JK1)  
988 CONTINUE  
989 CONTINUE  
NC=1=NC+1  
*TERM=5  
GO TO 9982  
DO 9912 J=1,LDB  
INDP=I+J*LMDP  
IEGL=TATR(INDP1)  
IF(IEQL.LT.0)IEQL=-IEQL  
IF(IEQL.EQ.NA)GO TO 991  
IF(TATR(INDP1).LT.0).AND.TATR(INDP1).NE.-999.)GO TO 991  
991 CONTINUE  
993 CONTINUE  
*TERM=7  
9982 CONTINUE  
IF(I.LT.NIMN)NIMN=I  
DO 982 J=1,NUMP  
JA=I+(J-1)*NIDP  
982 TATR(JA)=0.  
IAC(I)=0  
991 CONTINUE  
1030 CONTINUE  
1031 CONTINUE  
037 FORMAT(3I15)  
1840 DO 1 J=1,1E6  
JR=J+NECM  
IST=IC1(JR)  
ISTOP=IC2(JR)
DO 2 I=ISTART, ISTOP
MH=W(I-1)
IF(W(I).GT.MH)MH=W(I)
IF(W(I+1).GT.MH)MH=W(I+1)
   HB(I)=MH
1 CONTINUE
RETURN
END
LISTING FOR ERT1: TRACK FTN

SUBROUTINE TRACK
IMPLICIT INTEGER*2 (I-N)
INTEGER*2 T, TA
COMMON /INTL/ MHSN, MNSN, HM, FNSN
COMMON /NLIS/ NVARM, NCARM, NVO, ICO, IO, JO, JVR, KTL
COMMON /TLIS/ T, TA, JDAY, JHR, JMIN, JSEC, IDAY, IHR, MIN, ISEC
COMMON /PNTRSL/ NCMX, NVMIN, NVMX, IELS, NSCAN, IESNL, NVSCN, NT
COMMON /DATA1/ ECL(224), NOFST, KOFST, ICLAD, NANI
COMMON /ECONST/ EARTH, VMK
COMMON /CNT/ COSPHI, SINPHI, COSPH2
COMMON /A22/ SINA, COSA, DELTAS, ISCANNF, NEL, RI, SA
COMMON /KTS/ NV, NC, UCN

C INITIALIZE

UCN=0
VMK=COSPHI*VMK
VKM2=VKM+VMK

SAVKN2=SA+VKM2
SAVKN=SA+VKM

NAN2=NAN1+1
DO 10 M=1, NOFST
MA=(NOFST-M)*KOFST+NAN2
IF(ECL(MA).LT.0.) GO TO 22

10 CONTINUE
GO TO 41

22 NCMX=NOFST-M+1
DO 30 M=1, NCMX
M1=1+(M-1)*KOFST+NAN1
M2=1+M1
M3=1+M2
M4=1+M3
M5=1+M4
M6=1+M5
M7=1+M6

WRITE(6, 50) M1, M2, M3, M4, M5, M6, COSPHI, VMK, ECL(M1), ECL(M2),

50 FORMAT(1X, 6I16, 4X, 3HCOS, F10.5, 4X, 3HVKN, F10.2, 5HECL-,


ECL(M5)=ECL(M5)+SAVKN
VKME=VKM/(ECL(M2)+ECL(M1))
ECL(M3)=ECL(M3)+VKME
ECL(M4)=ECL(M4)+VKME
ECL(M1)=ECL(M1)+SAVKN2
R2=ECL(M3)*ECL(M3)+ECL(M4)*ECL(M4)
ECL(M7)=ECL(M6)

30 ECL(M6)=(SQRT(R2))*SINPHI+R2*COSPH2
FNSN=FNSN+1
NSCAN=NSCAN+1
IF(NSCAN.NE.1) GO TO 41

KTL=T
JDAY=IDAY
JHR=IHR
JMIN=MIN
JSEC=ISEC
DO 40 NC=1, NCMX
NV=NC

CALL ATRAK

40 CALL BTRAK
NVNX = NCMX
NVMIN = NVMX
GO TO 45

41 CONTINUE
IF (IELSN LT. IESNL) CALL STRAK
IESNL = IELSN
CALL COMPAR

45 CONTINUE
RETURN
END
**SUBROUTINE ATRAK**

**IMPLICIT INTEGER*2 (1-N)**

**INTEGER*2 T, TA**

**LOGICAL PR1, PR2, PRIN2**

**INTEGER*4 IVCL(736)**

**COMMON /ZLOOK/, IZOFF, ZARY(91)**

**COMMON /TLIS/, T, TA, JDAY, JHR, JMIN, JSEC, IDAY, IHR, MIN, ISEC**

**COMMON /NVLIS/, NVARM, NCARM, NVO, ICO, IO, JO, JVR, KTL**

**COMMON /PNTRSS/, NCMX, NVMIN, NVMX, IELS, NSCAN, IESNL, NVSCN, NTT**

**COMMON /INTL/, MHSN, MNSN, HM, FNSN**

**COMMON /FLGS/, ITYPE, PR1, PR2, PRIN2**

**COMMON /DVAL/, DELA**

**COMMON /DATA1/, VCL(736), MXVC, NVC1**

**COMMON /DATA2/, ECL(224), NOFST, KOFST, ICLAD, NANI**

**COMMON /KTA/, NV, NC, DELW**

**EQUIVALENCE (VCL(1), IVCL(1))**

IF(NC LE 0. OR. NC GT NCMX) GO TO 100

IF(NV LE 0. OR. NV GT NVARM) GO TO 100

NCEC=(NC-1)*KOFST+NANI

NVVC=(NV-1)*NVC1

NCA=2+NCEC

IZ=ECL(NCA)

NCA=NCA+1

X=ECL(NCA)

NCA=NCA+1

Y=ECL(NCA)

NCA=6+NCEC

H=ECL(NCA)

NVA=9+NVVC

IF(IWCL(NVA) .GT. 0) GO TO 10

**DEFINE LOWEST ELEVATION VALUES**

NVA=1+NVVC

VCL(NVA)=:

NVA=NVA+1

VCL(NVA)=Y

NVA=NVA+1

IVCL(NVA)=IZ

NVA=NVA+1

NCA=NCEC+1

VCL(NVA)=ECL(NCA)

NVA=NVA+1

IVCL(NVA)=T

NVA=NVA+1

VCL(NVA)=H

NCA=3+NCEC

NCA1=4+NCEC

R2=ECL(NCA)+ECL(NCA)+ECL(NCA1)+ECL(NCA1)

R=SQRT(R2)

NVA=NVA+1

VCL(NVA)=R

NVA=NVA+1

IF(IWCL(NVA) .NE. 0) GO TO 10

NTT=NTT+1

VCL(NVA)=NTT
IZL=IZL+1
IF(IZL.LT.1) IZL=1
IF(IZL.GT.91) IZL=91
Z=ZARY(IZL)

NVA=9+NVVC
IVCL(NVA)=IVCL(NVA)+1
NCA=7+NCEC
NVA=23+NVVC
VCL(NVA)=VCL(NVA)+ECL(NCA)
NVA=10+NVVC
VCL(NVA)=VCL(NVA) + Z
NVA=NVA+1
VCL(NVA)=VCL(NVA) + Z*X
NVA=NVA+1
VCL(NVA)=VCL(NVA) + Z*Y
NVA=14+NVVC
HL=VCL(NVA)
IF(HL.GT.0 .OR. FNSN.LT.2.) GO TO 11
HL=H-DELA*R

NCA=NCEC+1
NVA=13+NVVC
VCL(NVA)=VCL(NVA)+Z*(H-HL)*ECL(NCA)

SUMMIT VALUES
NVA=NVA+1
VCL(NVA)=H
NVA=NVA+1
IVCL(NVA)=IZ

PEAK, BASE, AND TOP ARRAYS
NVA=NVA+1
IZP=IVCL(NVA)
IF(IZP.GT.12) GO TO 99

SET PEAK

32 NVA=16+NVVC
IVCL(NVA)=IZ
NVA=NVA+1
VCL(NVA)=H

99 IF(.NOT.PRIN2) GO TO 100
NCA=NCEC+1
NVA=9+NVVC
RETURN
END
* 
07/19/79 13:01:37
***LISTING FOR ERT1:BTRK.FCN

SUBROUTINE BTRK
IMPLICIT INTEGER*2 (I-N)
INTEGER*2 T, TA
INTEGER *4 IVR(192)
COMMON /DATA1/ ECL(224), NOFST, KOFST, ICLAD, NAM1
COMMON /DATA3/ VR(192), MXVR, NVR1
COMMON /TLIS/ T, TA, JDAY, JHR, JMIN, JSEC, IDAY, IHR, MIN, ISEC
COMMON /NVLIS/ NVARM, NCARM, NVO, ICO, IO, JVR, KTL
COMMON /KTA/ NV, NC, UCN
EQUIVALENCE(VR(1), IVR(1))
IF(NV. LE. 0. OR. NV. GT. NVARM) GO TO 10
IF(NC. LE. 0. OR. NC. GT. NCARM) GO TO 10
C DEFINE LAST ELEVATION VALUES
C
NC=NC-1*KOFST+NAM1
NVVR=(NV-1)*NVR1
NCA=3+NCEC
NVA=1+NVVR
VR(NVA)=ECL(NCA)
NCA=NCA+1
NVA=NVA+1
VR(NVA)=ECL(NCA)
NCA=NCA+2+NCEC
NVA=NVA+1
VR(NVA)=ECL(NCA)
NCA=NCA-1
NVA=NVA+1
VR(NVA)=ECL(NCA)
NVA=NVA+1
IVR(NVA)=T
NVA=NVA+1
NCA=6+NCEC
VR(NVA)=ECL(NCA)

10 RETURN
END
SUBROUTINE COMPAR
IMPLICIT INTEGER*2 (I-N)
INTEGER*2 T, TA
LOGICAL PR1, PR2, PRIN2, ARRAY
INTEGER*4 IVCL(736), IVR(192)
COMMON /DATAL/ ECL(224), NOFST, KOFST, ICLAD, NAI
COMMON /DATA2/ VCL(736), MXVC, NV1
COMMON /DATA3/ VR(192), MXVR, NVR1
COMMON /CDRAYS/ IC(32, 10), C(32, 9), ID(32, 10), D(32, 9), IM, JM
COMMON /FLGS/ ITYPE, PR1, PR2, PRIN2
COMMON /PARM/ VX, VY
COMMON /PNTR5/ NCMX, NVMIN, NVMX, IEALN, NSCAN, IEC, NVSN, NT
COMMON /CONST/ VMISW, DIV, VMAG, VMISWM, ZDIV, ADIV, A1, A2, A3, B1, B2, HDIV
COMMON /TLIS/ T, TA, JDAY, JHR, JMIN, JSEC, IDAY, IHR, MIN, ISEC
COMMON /NVLS/ NVM, NCM, NVO, ICO, IO, JVR, KTL
COMMON /COSPHI/ SINEL, COSPI2
COMMON /INTL/ MMSN, MNSN, HM, FNSN
COMMON /KTRA/ NV, NC, UCN
COMMON /RSLV/ IVU1(32), IVU2(32), IUC1(16), IUC2(16),
+ UV(32), UC(16), NCR
EQUIVALENCE(VCL(1), IVCL(1)), (VR(1), IVR(1))
IDV=1
ICV=1
IF(NCMX LE 0) RETURN
NVMXP=1
IF(NVMX GT 1) NVMXP=NVMX
DO 3 I=1, NVMXP
UV(I)=0.
IVU1(I)=0
3 IUV2(I)=0
DO 4 I=1, NCMX
UC(I)=0.
IUC1(I)=0
4 IUC2(I)=0
DO 5 I=1, IM
IC(1, I)=0
ID(1, I)=0
DO 7 J=1, JM
J1=J+1
IC(1, J1)=0
ID(1, J1)=0
C(I, J)=0.
D(I, J)=0.
7 CONTINUE
5 CONTINUE
DO 10 NC=1, NCMX
NCEC=(NC-1)*KOFST+NAI
NC1=1+NCEC
NC2=NC1+1
NC3=NC2+1
NC4=NC3+1
NC5=NC4+2
NVC=8
DO 40 NV=1, NVMXP
NVVC=(NV-1)*NV1
NVVR=(NV-1)*NV1
MLAST=0
DELNL=6.
NLNR=20*NVC
D-37
IF(IVCL(NLR) .LE. 0. AND. IVCL(NLR1) .LE. 0) GO TO 40
DTEST=(VMAG*DTTA)+(VMAG*DTTA)+VMISWM
NRA=1+NVVR
NCA=21+NVVC
DELX=ECL(NC3)-VR(NRA)-VCL(NCA)*DTTA
DELX2=DELX*DELX
IF(DELX2 .GT. ATTEST) GO TO 20
NRA=2+NVVF
NCA=21+NVVC
DELY=ECL(NC4)-VR(NRA)-VCL(NCA)*DTTA
DELY2=DELY*DELY
IF(DELY2 .GT. ATTEST) GO TO 20

C ASSOCIATED, FIND BEST

C NRA=3+NVVR
NRA1=NRA+1
NRA2=NRA1+2
DELY=ABS(ECL(NC2)-VR(NRA))*2DIV+(DELX2+DELY2)*
1 DIV. + ABS(ECL(NC1)-VR(NRA)) * ADIV
2 + ABS(ECL(NC6)-VR(NRA)) * HDIV
IF(DELYL .GT. VMISW) MLAST=1
GO TO 20
30 NVC=NVC+1
IF(NVC .GT. 1) GO TO 31
IUC1(NC)=NV
UC(NC)=DELY
GO TO 33
31 IF(IUC1(NC) .EQ. NV) GO TO 33
NVT=NV
IF(DELY .LT. UC(NC)) GO TO 32
35 IF(IUC2(NC) .LE. 0) GO TO 36
I=IUC2(NC)
IF(I .GT. IM) GO TO 361
39 J=ID(I,1)+1
ID(I,1)=J
IF(J .LE. JM) GO TO 37
C PRINT 101, JM, NSCAN, NVT, NC, J
JO=JO+1
J=JM
37 ID(I, J+1)=NVT
D(I,J)=DELY
GO TO 33
36 I=IDV
IDV=I+1
IF(I .LE. IM) GO TO 38
C PRINT 102, IM, NSCAN, NVT, NC, I
361 IQ=IQ+1
I=IM
38 IUC2(NC)=I
GO TO 39
32 DX=UC(NC)
UL(NC)=DELY
DELY=DX
NVT=IUC1(NC)
IUC1(NC)=NV
GO TO 35
33 IF(IUV1(NV) .NE. 0) GO TO 21
IUV1(NV)=NC
UV(NV)=DELY
GO TO 40

C CLUSTER
21 IF(IUV1(NV).EQ. NC) GO TO 40
   NCT=NC
   IF(DELW.LT. UV(NV)) GO TO 22
25 IF(IUV2(NV).EQ. 0) GO TO 26
   I=IUV2(NV)
   IF(I GT IM) GO TO 261
29 J=IC(1,1)+1
   IC(1,1)=J
   IF(J LE JM) GO TO 27
   C PRINT 101, JM, NSCAN, NV, NCT, J
C 101 FORMAT(' NO. OF CELLS IN CLUSTER EXCEEDS JM = ', 5110)
   JO=J0+1
   J=JM
27 J1=J+1
   IC(I,J1)=NCT
   C(I,J)=DELW
   GO TO 40
26 I=ICV
   ICV=I+1
   IF(I LE IM) GO TO 28
   C PRINT 102, IM, NSCAN, NV, NCT, J
C 102 FORMAT(' NO. OF ENTRIES IN CLUSTER ARRAY EXCEEDS IM = ', 5110)
261 IO=IO+1
   I=IM
28 IUV2(NV)=I
   GO TO 29
22 DX=UV(NV)-DELW
   UV(NV)=DELW
   DELW=DX
   NCT=IUV1(NV)
   IUV1(NV)=NC
   GO TO 25
C C
C \text{NO COMPARV, TRY VCL}
C
20 NCA=5+NVVC
   DELT=T-IVCL(NCA)
   ATEST=VMISHW*(VMAG*DELT)**(VMAG+DELT)
   NCA=1+NVVC
   NCA1=21+NVVC
   DELX=ECL(NC3)-VCL(NCA)-VCL(NCA1)*DELT
   DELX2=DELX**DELX
   IF(DELX2 .GT. ATEST) GO TO 40
   NCA=2+NVVC
   NCA1=22+NVVC
   DELY=ECL(NC4)-VCL(NCA)-VCL(NCA1)*DELT
   DELY2=DELY**DELY
   IF(DELY2 .GT. ATEST) GO TO 40
   NCA=3+NVVC
   NCA1=NCAC+1
   NCA2=NCAC+2
   DELN=ABS((ECL(NC2)-FLOAT(IVCL(NCA)))*ZDIV)**(DELX2
   1)**(DELX2)*DIV+1 + ABS(ECL(NC1)-VCL(NCA1))*DIV
   2 +ABS(ECL(NC6)-VCL(NCA2))*DIV
   IF(MLAST. NE 0 AND DELWL.LT. DELWL) DELN=DELWL
   IF(DELY .LE. VMISHW) GO TO 30
40 CONTINUE
   IF(NVC .GT. 0) GO TO 10
C C
C \text{ISOLATED CELL, NO COMPAR}
C
C IF(NVMX. LT. NVMIN) GO TO 501
DO 50 NV=NVMIN, NVMX
   NVVC=(NV-1)*NVC1
50 CONTINUE
IF(IVC(L(NCA)) .EQ. 0 .AND. IVCL(NCA1) .EQ. 0) GO TO 55

50 CONTINUE

IF(NV .LE. NVARM) GO TO 51

C PRINT 103. NVARM, NV
C 103 FORMAT(C) NO. OF ACTIVE TRACKS EXCEEDS ARRAY MAXIMUM NVMX ='

C 1 2110

NV=NV+1
NV=NVARM

51 NVMX=NV

55 NVMIN=NV

IF(NVNX .LE. 0) NVMX=NV
UCN=UC(N)
CALL ATRAK
CALL BTRAK
IUV1(NV)=NC
IUC1(NC)=NV
UV(NV)=0 0
UC(NC)=0 0

10 CONTINUE

C HAVE LIST OF COMPARISONS, NOW RESOLVE CONFLICTS

DO 60 NV=1,NVMX
IF(IUV1(NV) .LT. 0) GO TO 60
IF(IUV1(NV) .EQ. 0) GO TO 61
NC=IUV1(NV)
IF(NC GT NCMX) GO TO 61
IF(IUC1(NC) .LE. 0) GO TO 61
IF(IUV2(NV) .EQ. 0 .AND. IUC2(NC) .EQ. 0) GO TO 70
NCR=NC
CALL RESOLV
NC=NCR
GO TO 60

70 UCN=UC(NC)
CALL ATRAK
CALL BTRAK
IUV1(NV)=IUV1(NV)
IUC1(NC)=IUC1(NC)
UV(NV)=0 0
UC(NC)=0 0
GO TO 60

C NO NC COMPARE, FIX HEIGHT STATISTICS

61 IF(FNSN .LT. 1.1) GO TO 60
NVCA=7*(NV-1)+NVC1
CA=VCL(NVCA)
HTC=CA*SINEL+CA*CA*COSPI2

60 CONTINUE
RETURN
END
SUBROUTINE RESOLV
LOGICAL PRIN2, PR1, PR2
IMPLICIT INTEGER*2 (I-N)
INTEGER*4 IVCL(736), NVT, IVS, IVT, KV, J, IDIJ, JJ, NCT, ICS, ICT,
+KC, ICIJ

DIMENSION V(384)
COMMON /NVLIS/ NVARM, NCARM, NVO, ICO, IO, JYR, KTL
COMMON /COMB/ IV(32,7), IVMX
COMMON /PNTSR/ NCMX, NVPN, IELS, NSCAN, IESNL, NVSCN, NT
COMMON /CDRAYS/ IC(32,10), C(32,9), ID(32,10), D(32,9), IM, JM
COMMON /FLGS/ ITYPE, PR1, PR2, PRIN2
COMMON /DATA2/ VCL(736), MMVC, NVC1
COMMON /CONST/ VMISW, DIV, VMAG, VMISWM, ZDIV, ADIV, A1, A2, A3, B1, B2, HDIV
COMMON /INTL/ MHSN, MNSN, HM, FNSN
COMMON /KTR/ N, NC, UCN
COMMON /RSLV/ UV1(32), UV2(32), IUC1(16), IUC2(16),
+EQUIVALENCE(VCL(1), IVCL(1))

IVMX=NVARM

C HAVE CLUSTER, ORDER LISTS

DO 4 I=1,6
4 IV(1,I)=0
IVT=4
IVS=5
ICT=1
ICS=2
JV=1
JC=1
LV=0
LC=0
KV=1
KC=1
NCT=NCA
KOF2=128
KOF3=256
IF(NCT.LE.0. OR. NCT.GT. NCMX) GO TO 100

C PROCESS NCT

65 IF(IUC1(NCT).LE.0. OR. IUC1(NCT).GT. NVMX) GO TO 66
IF(UC(NCT).LE.0.) GO TO 66
NVT=IUC1(NCT)
IF(UC(NCT).GT.0.) UC(NCT)=-UC(NCT)
CALL COMBIN(NVT, IVS, IVT, KV, J)
IF(IUC2(NCT).LE.0. OR. IUC2(NCT).GT. IM) GO TO 62
I=IUC2(NCT)
JX=ID(I,1)
IF(JX.LE.0) GO TO 62
IF(JX.GT. JM) JX=JM
ID(I,1)=-ID(I,1)
JX1=JX+1

DO 611 J=2, JX1
611 IDIJ=ID(I,J)
CALL COMBIN(IDIJ, IVS, IVT, KV, JJ)

611 CONTINUE

C PROCESS NVT

62 IF(UV1(NVT).LE.0. OR. UV1(NVT).GT. NCMX) GO TO 63
IF(\/NVT) .LE. 0. ) GO TO 63
NCT=IUV1(NVT)
IF(\/NVT) .GT. 0. ) U\/NVT)=-\/NVT)
CALL COMBIN\(NCT, ICS, ICT, KC, J\)
IF(IUV2(NVT) . LE. 0. OR. IUV2(NVT). GT. IM) GO TO 63
I=IUV2(NVT)
JX=IC(I,1)
IF(JX . LE. 0. ) GO TO 63
IF(JX. GT. JM) JX=JM
IC(I,1)=-IC(I,1)
JX1=JX+1
DO 621 J=JX1, JX-1
ICIJ=IC(I, J)
CALL COMBIN\(ICIJ, ICS, ICT, KC, JJ\)
621 CONTINUE

RUN COMPAR LIST TO FLUSH OUT FULL SET

63 DO 631 K=JV, KV
NVT=IV(K, IVS)
IF(NVT. LE. 0. OR. NVT. GT. NVARM) GO TO 631
IF(\/NVT) . LT. 0. ) NCT=IUV1(NVT). LE. NCMX) GO TO 64
631 CONTINUE
GO TO 66

64 JV=K
LC=LC+1
GO TO 62

66 DO 661 K=JC, KC
NCT=IV(K, ICS)
IF(NCT. LE. 0. OR. NCT. GT. NCMX) GO TO 661
IF(UC\(NCT) . LT. 0. ) GO TO 661
IF(IUC1(NCT) . GT. 0. AND. IUC1(NCT). LE. NVARM) GO TO 67
661 CONTINUE
GO TO 68

67 JC=K
LV=LV+1
GO TO 65

68 IF(LC . EQ. 0) GO TO 69
LC=0
JV=1
JC=1
LV=0
GO TO 63

69 IF(LV . EQ. 0) GO TO 70
LV=0
JV=1
LC=0
JC=1
GO TO 66

HAVE ORDERED LIST, NOW FIND BEST MATCH

70 IF(KC. LE. 1 . OR. KV. LE. 1) GO TO 100
KV=KV-1
IF(KV. GT. IVMX) GO TO 100
KC=KC-1
IF(KC. GT. IVMX) GO TO 100
IMSM=0
DO 701 K=1, KV
NV=IV(K, IVS)
IF(NV. LE. 0. OR. NV. GT. NVMX) GO TO 701
IF(\/NV) . LT. 0. ) U\/NV)=-\/NV)
701 CONTINUE

DO 71 K=1, KC
KA2=K+KOF2
VCIC)=0.
VCIC2)=0.
VICKA3)=0.
IVCK, ICT)=0
NC-IVCK,ICS)

VCIC)=0.

IVCK,ICT)=6
IVCK, 3)=0
IVTCNX,ICN).

IFCKV. LE. CKC-IMSM)

KNC=O
DO 72 K=1, KC
IFCIVCFGS)

NC=IYCK, ICS)
IF(UC(NC). LE. 0. )UC(NC)=--UC(NC)
NV=IUC1(NC)

IF(NV. LE. 0. OR. NV. GT. NVMX) GO TO 71
IF(IVUV1(NV). LT. 0)GO TO 71
IF(IVUV1(NV). NE. NC)GO TO 71
I=UC(NC)
UV(NV)=-UV(NV)
UC(NC)=-UC(NC)
GO TO 71

711 IMSM=IMSM+1
71 CONTINUE

IFKV. LE. (KC-IMSM). OR. IMSM.EQ. 0) GO TO 75

C FIRST ROUND MIN WEIGHT SELECTION

KNV=0
KNC=0
DO 72 K=1, KC
IF(IYCK, ICS)

NC=IVK, ICS)
IF(NC. LE. 0. OR. NC. GT. NCMX) GO TO 72
IF(UC(NC). LE. 0. 1)GO TO 72
NV=IUC1(NC)

IF(NV. LE. 0. OR. NV. GT. NVMX)GO TO 724
IF(UC(NV). GT. 0. 1)GO TO 725
IVCK, 7)=NV
KNV=KNV+1

724 I=IUC2(NC)

IF(I. LE. 0. OR. I. GT. IM) GO TO 721
JX=ID(I, 1)

IF(JX. LT. 0)JX=-JX
IF(JX LE. 0. OR. JX. GT. JM) GO TO 721
NV=0

DWT=999.
DO 723 J=1, JX
J1=J+1

NVT=ID(I, J1)

IF(NVT. LE. 0. OR. NVT. GT. NVMX) GO TO 723
IF(UC(NVT). LE. 0. 1) GO TO 723

DELW=DCI, 3)

IF(DELW. LE. a)

IFCDELW. LT.DTW-EL
IFCDWT.EQ. DELW) NV=NVT

723 CONTINUE

IF(NV . LE. 0 . OR. NV. GT. NVMX. OR. DWT. GT. VMISW. OR. DWT. LT. 1)
1 GO TO 721
GO TO 726

725 DWT=UC(NC)
726 KA2=K+KOF2

VCKA2)=DWT

IF(VU(NV). GT. 0 .)UV(NV)=-UV(NV)
IF(UHC(NC). GT. 0 .)UC(NC)=-UC(NC)

IVK, ICT)=NV

GO TO 72

721 KNC=KNC+1
C
IV(KNC, IVT) = K

72 CONTINUE
IF(KNV .LE. 0. AND. KNC .LE. 0) GO TO 75
IF(KNC .EQ. 0. OR. KNC .GT. KC) GO TO 80

C CASCAD REORDER OF COMPAR LIST
C
J = 0
731 J = J + 1
IF(J .GT. KNC) GO TO 80
K = IV(J, IVT)
IF(K .LE. 0. OR. K .GT. KC) GO TO 731
NC = IV(K, ICS)
IF(NC .LE. 0. OR. NC .GT. NCMX) GO TO 739
NV = IUC1(NC)
IF(NV .LE. 0. OR. NV .GT. NVMX) GO TO 739
DO 738 L = 1, KC
IF(IV(L, 3) .EQ. NV) GO TO 7381
IF(IV(L, IXT) .EQ. NV) GO TO 7382
738 CONTINUE
GO TO 739
7381 NCT = IV(L, ICS)
DELT = VMISW + V(L)
GO TO 7383
7382 NCT = IV(L, ICS)
KA2 = L + KOF2
DELT = VMISW + V(KA2)
7383 KT = L
IF(NCT .LE. 0. OR. NCT .GT. NCMX) GO TO 739
IF(IUC1(NCT) .LE. 0. OR. IUC2(NCT) .LE. 0) GO TO 739
DELT = UC(NC)
IF(DELW LT 0) DELW = -DELW
I = IUC2(NCT)
IF(I .LT. 0) I = -I
IF(I .GT. 1) GO TO 739
JX = ID(I, 1)
IF(JX LT 0) JX = -JX
IF(JX .LE. 0. OR. JX .GT. JM) GO TO 739
DWT = 999.
NVB = 0
DWT1 = 999
NV1 = 0

DO 732 L = 1, JX
IF(D(I, L) .LE. 0.1) GO TO 732
IF(D(I, L) LT DWT) DWT = D(I, L)
L1 = L + 1
NV1 = IABS(ID(I, L1))
IF(DWT .EQ. D(I, L)) NVB = NV1
IF(NV1 .LE. 0. OR. NV1 .GT. NVMX) GO TO 732
IF(UV(NV1) .GT. 0.1 AND. D(I, L) LT DWT) DWT1 = D(I, L)
IF(DWT1 .EQ. D(I, L)) NV1 = NV1
732 CONTINUE
IF(NVB .LE. 0. OR. NVB .GT. NVMX) GO TO 739
1 GO TO 739
734 IF(NV1 .LE. 0. OR. NV1 .GT. NVMX) GO TO 731
1 GO TO 735
DEEW1 = DELW + DWT1
DELL2 = DELW + DWT
IF(DEEL1 .GT. DELT) GO TO 735
IV(K, 6) = NV
KA3 = K + KOF:
V(KA3) = DELW
IV(KT, 6) = NV1
KA3 = KT + KA3
V(KA3) = DWT1
IF(DELW2 .GE. DELW1) GO TO 739
IV(KT,7)=NVB
GO TO 739

735 DO 736 I=1, KC
IF(IV(I,3) .EQ. NVB) GO TO 739
IF(IV(I, ICT) .EQ. NVB) GO TO 737
736 CONTINUE
GO TO 739

737 KA2=I+KOF2
DELT=DELT+V(KA2)
DELW2=DELW+DWT+YHISW
IF(DELW2 .GT. DELT) GO TO 739
IV(K,6)=NV
KA2=K+KOF3
V(KA3)=DELW
IV(KT,6)=NVB
KA3=K+KOF3
KA2=I+KOF2
V(KA3)=DWT
IV(I, ICT)=0
V(KA2)=0.
739 IV(K,7)=0
KNV=KNV-1
GO TO 731

EXCHANGE PAIRS FOR MIN MEASURE

80 IF(KNV .LE. 0 OR KNV .GT. KC) GO TO 75
DO 801 K=1, KC
NVB=IV(K,7)
IF(NVB .LE. 0 OR NVB .GT. NVMX) GO TO 801
IF(IV(UV2(NVB)). LE. 0) GO TO 801
NC=IV(K, ICS)
IF(NC .LE. 0 OR NC .GT. NCMX) GO TO 801
NV=IV(K, 2)
DO 802 L=1, KC
IF(NVB .EQ. IV(L, ICT) OR NVB .EQ. IV(L, 3)) GO TO 803
802 CONTINUE
GO TO 801
803 NCB=IV(L, ICS)
I=IV(UV2(NVB)
IF(I .LT. 0) I=-I
JX=IC(I, 1)
IF(JX .LT. 0 OR JX .GT. JM) GO TO 801
DO 807 J=1, JX
J1=J+1
IF(IC(I, J1), EQ. NCB) GO TO 808
807 CONTINUE
GO TO 801
808 DSET=C(I, J)
DUC=UC(NC)
IF(DUC .LT. 0) DUC=-DUC
DELWB=DSET+DUC
KA2=K+KOF2
KA3=K+KOF3
DELI1=V(KA2)
KA=KA-KOFST
IF(DELI1 .LE. 1) DELW1=V(KA2)
IF(DELI1 .LE. 1) DELW1=V(K)
KA2=L+KOF2
KA3=L+KOF3
DELI2=V(KA3)
IF(DELI2 .LE. 1) DELW2=V(KA2)
IF(DELI2 .LE. 1) DELW2=V(L)
DELW=DELW1+DELW2
IF(DELW .EQ. DELW) GO TO 804
UPDATE ATTRIBUTES

DO 78 K=1, KC
KA3=K+KOF3
KA2=K+KOF2
IF(PRIN2)
  1 WRITE(6, 788) IV(K, ICS), IV(K, 6), IV(K, ICT),
  2 IV(K, 3), V(KA3), V(KA2), V(K)
788 FORMAT(1X, 415, 3F8.2)
NC=IV(K, ICS)
IF(NC LE. 0. OR. NC. GT. NCMX) GO TO 78
IF(IUC1(NC). LE. 0) GO TO 78
NV=IV(K, 6)
IF(NV. LE. 0. OR. NV. GT. NVMX) GO TO 810
DWT=V(KA3)
GO TO 820
810 NV=IV(K, ICT)
IF(NV. LE. 0. OR. NV. GT. NVMX) GO TO 811
DWT=V(KA2)
GO TO 820
811 NV=IV(K, 3)
IF(NV. LE. 0. OR. NV. GT. NVMX) GO TO 79
DWT=V(K)
820 IF(DWT. LE. 0.1. OR. DWT. GT. VMISW) GO TO 79
IF(IUV1(NV). EQ. 0) GO TO 79
UCN=DWT
CALL ATRA;
CALL BRA:
GO TO 77
79 IF(NVMX. LT. NVMIN) GO TO 7911
DO 791 I=NVMIN, NVMX
NVVC=(I-1)*NVV1
IVA1=IVA+11
IF(IVCL(IVA1). EQ. 0 . AND. IVCL(IVA). EQ. 0) GO TO 792
791 CONTINUE
7911 I=NVMX+1
IF(I. LE. NVARM) GO TO 7921
PRINT 103, NVARM, I
C 103 FORMAT(1X, 10, A) NO. OF ACTIVE TRACKS EXCEEDS ARRAY MAXIMUM NVMX =
C 1  2110)
NVO=NVO+1
I=NVARM
GO TO 77
7921 NVMX=I
792 NV=1
NVMIN=1
UCN=0
CALL ATRA;
CALL BTRAK
77 IUV1(NV)=NC
IUC1(NC)=-NV
UV(NV)=0.
UC(NC)=0.
78 CONTINUE
DO 99 K=1, NV
NV=IV(K,IV5)
IF(NV. LE. 0. OR. NV. GT. NVMX) GO TO 99
IF(IUV1(NV). LE. 0) GO TO 99
IUV1(NV)=-IUV1(NV)
IF(FNSN. LT. 1.1) GO TO 99
NVA=7+(NV-1)*NVC1
VA=VCL(NVA)
HTC=VA*SINEL+VA*VA*COSPI2
99 CONTINUE
100 RETURN
END
SUBROUTINE COMBIN(N, IS, IT, K, J)
IMPLICIT INTEGER*2 (I-N)
INTEGER*4 N, IS, IT, K, J
COMMON /COMB/ IV(32, 7), IVMX

I=IS
IS=IT
IT=I
L=0
DO 10 J=1, K
IVJ=IV(J, IT)
IF(IVJ.LT.0) IVJ=-IVJ
IF(IVJ.NE.0) GO TO 40
20 IF(IVJ.EQ.0) GO TO 40
10 IV(J, IS)=IV(J, IT)
J=K
40 L=L+1
IV(J, IS)=N
30 DO 50 I=J, K
IL=I+L
50 IV(IL, IS)=IV(I, IT)
K=K+L
IF(K.GE.IVMX) GO TO 70
IV(K, IS)=0
GO TO 80
70 WRITE(6, 100) K, IVMX
100 FORMAT(' ERROR IN COMBIN, 13, K, IVMX', 2X, 2I10)
K=IVMX-1
80 RETURN
END

*
SUBROUTINE STRAK
IMPLICIT INTEGER*2 (I-N)
INTEGER*2 T, TM, TA
INTEGER*4 IVCL(736), ITCL(21), IVR(192)
DIMENSION ATST(62), NUM(62)
LOGICAL PRI, PR2, PRIN2
DIMENSION Tcl(21), DUM(6)
COMMON /TMAX/ TM
COMMON /FLGS/ I TYPE, PRI, PR2, PRIN2
COMMON /PNTRS/ NCMX, NVMIN, NVMX, IELS, NSCAN, IESNL, NVCNL, NT
COMMON /TLIS/ T, TA, JDAY, JHR, JMIN, JSEC, IDAY, IHR, MIN, ISEC
COMMON /NVLIS/ NVARM, NARM, NVO, ICO, IO, JO, JYR, KTL
COMMON /CONST/ VMISW, DIV, VMAG, VMISWM, ZDIV, A1, A2, A3, B1, B2, HDIV
COMMON /INTL/ MHSN, MNSN, HM, FNSN
COMMON /VPARM/ VX, VY
COMMON /DATA1/ ECL(224), NOFST, KOFST, ICLAD, NAM
COMMON /DATA2/ VCL(736), MXVC, NVC1
COMMON /DATA3/ VR(192), MXVR, NVR1
EQUIVALENCE(TCL(1), ITCL(1))
EQUIVALENCE(VCL(1), IVCL(1)), (VR(1), IVR(1))
DATA IZERO/0/, IOUT/31/, LAN/62/
MAN=0
NVOUT=NVMX-1
DO 19 N=1, LAN
ATST(N)=0
19 NUM(N)=0
VXC=0
VYC=0
NSN=0
NACT=0
NFIN=FNSN-1.
IF(NFIN.LE.0) GO TO 55
WRITE(2) KTL, NVMSC, JDAY, JHR, JMIN, JSEC
1 ,NFIN, NSCAN, NT, NVMX, JYR, NVO
2 ,10, JO, IZERO, VX, VY, DUM
IF(PRI. OR. PR2)
WRITE(6, 1000) JDAY, JHR, JMIN, JSEC, NVCNL
1000 FORMAT(1HL, 3X, 'PROGRAM TRACK4 (78207) ', 5X, 'TRACK TIME', 414,
15X, 'VOL SCAN', 16/,'1X, 'TRK LOCATION DBZ CELL PEAK ',
2 'SURFACE SUMMIT VELOCITY A'/, 1X
3 'ID EAST NORTH AV VOL DBZ HT DBZ HT ',
4 'EAST NORTH ', 1X
5 'KM KM KM 'KM 'KM ',
6 'M/S M/S L'
DO 20 NV=1, NVOUT
NVR=(NV-1)*NVC1
NV23=23+NVR
A=VCL(NV23)
DO 21 N=1, IOUT
NN=NN+MAN
IF(A.GT.ATST(NN)) GO TO 22
21 CONTINUE
NN=NN+1
22 LAN=MAN
MAN=MAN+IOUT
IF(MAN.GT. IOUT) MAN=0
IFMN=0
DO 20 N=1, IOUT
MM=M+MAN
ML=M+LAN
IF(QLNE. NN OR. IFML. EQ. 1) GO TO 24
ATST(MM)=A
NUM(MM)=NV
LAN=LAN-1
IFML=1
GO TO 23
24 ATST(MM)=ATST(ML)
NUM(MM)=NUM(ML)
23 CONTINUE
20 CONTINUE
50 DO 100 N=1, IOUT
NN=N+MAN
NV=NUM(NN)
IF(NV. EQ. 0) GO TO 100
NVA=(NV-1)*NVC1
NV1=NV1+1
NV2=NV2+1
NV3=NV3+1
NV4=NV4+1
NV5=NV5+1
NV6=NV6+1
NV7=NV7+1
NV8=NV8+1
NV9=NV9+1
NV10=NV10+1
NV11=NV11+1
NV12=NV12+1
NV13=NV13+1
NV14=NV14+1
NV15=NV15+1
NV16=NV16+1
NV17=NV17+1
NV18=NV18+1
NV19=NV19+1
NV20=NV20+1
NV21=NV21+1
NV22=NV22+1
NV23=NV23+1
NR1=1+(NV-1)*NVR1
NR2=NR1+1
NR3=NR2+1
NR4=NR3+1
NR5=NR4+1
NR6=NR5+1
IF(NVCL(NV9). LE. 0) GO TO 200
Z2DIV=1.0/VCL(NV10)
ITCL(1)=IVCL(NV5)
TCL(2)=VCL(NV11)*Z2DIV
TCL(3)=VCL(NV12)*Z2DIV
TCL(4)=10.0*ALOG10(VCL(NV10)/FLOAT(IVCL(NV9)))
TCL(5)=VCL(NV13)*Z2DIV
ITCL(7)=IVCL(NV16)
TCL(8)=VCL(NV17)
ITCL(10)=IVCL(NV3)
TCL(11)=VCL(NV4)
TCL(12)=VCL(NV6)
IM3=IVCL(NV16)-3
30 ITCL(15)=IVCL(NV15)
TCL(16)=VCL(NV14)
IVL8=IVCL(NV8)
IF(IVL8. LT. 0) IVL8=-IVL8
ITCL(17)=IVL8
ITCL(18)=0
ITCL(20)=0
ITCL(21)=0
VXT=VCL(NV21)
VYT=VCL(NV22)
ITIM=IVCL(NV20)
JTIM=ITCL(1)
IF ITIM.EQ.0 OR ITIM.EQ.JTIM GO TO 40
DELM=JTIM-ITIM
DELM=1.0/DELM
TCL(20)=(TCL(2)-VCL(NV18))*DELM
TCL(21)=(TCL(3)-VCL(NV19))*DELM
VXT=TCL(20)
VYT=TCL(21)
40 IVCL(NV20)=ITCL(1)
VCL(NV19)=TCL(3)
VCL(NV18)=TCL(2)
VCL(NV21)=A1*VXT+A2*VCL(NV21)+A3*VX
VCL(NV22)=A1*VYT+A2*VCL(NV22)+A3*VY
VR(NR1)=TCL(2)
VR(NR2)=TCL(3)
VR(NR3)=TCL(4)
VR(NR4)=VCL(NV4)
IVR(NR5)=ITCL(1)
VR(NR6)=TCL(8)
IZVAL=VR(NR3)
IDTC=FLOAT(IVCL(NV9))/(FNSN-1.)*10.+.5
NSN=NSN+1
VXC=VXC+TCL(20)
VVC=VVC+TCL(21)
VXP=VCL(NV21)*1000.
VYP=VCL(NV22)*1000.
NACT=NACT+1
WRITE(2) TCL, VCL(NV39), VCL(NV40), IDTC
IF(PR1) GO TO 59
IF(NOT. PR2) GO TO 59
IF(IDTC.GT. MNSN OR IZVAL .GT. 35) AND IVCL(NV8) .GT. 0
1 IVCL(NV8)=-IVCL(NV8)
IF(IVCL(NV8).GE.0) GO TO 59
59 WRITE(6,1005) ITCL(17), VR(NR1), VR(NR2), IZVAL, TCL(5), ITCL(7),
ITCL(8), ITCL(10), TCL(12), ITCL(15), TCL(16),
2VXP, VYP, IDTC
1005 FORMAT(4X, I4, 2F5.0, I3, F6.1, I3, F5.1, I5, F5.1, I5,
+ 3F5.1, 2X, 12)
58 DO 41 I=9,14
IA=I+NVA
41 IVCL(IA)=0
IVCL(NV23)=0
GO TO 100
200 IF(IVCL(NV20).LE.0) GO TO 102
IF((T-IVCL(NV20)).LE.TM) GO TO 100
DO 101 I=1,20
IA=I+NVA
101 IVCL(IA)=0
IA=23+NVA
VCL(IA)=0
IF(NV.LT.NVMIN) NVMIN=NV
102 IVCL(NV23)=0
100 CONTINUE
IF(NSN.EQ.0) GO TO 56
VN=NSN
VX=B1*VXC/VN+B2*VX
VY=B1*VVC/VN+B2*VY
DO 43 JI=1, IOUT
II=IJ+MAN
I=NUM(IJ)
IF (I.EQ.0) GO TO 43
IF(IVCL(NVA).GT.0) GO TO 43
NA=21+(I-1)*NVCL
VCL(NA)=VX
NA=NA+1
VCL(NA)=VY
43 CONTINUE
VXP=VX*1000.
VYP=VYY*1000.
56 IF(NVSCN.GT.1) GO TO 55
WRITE(6,1003)
1003 FORMAT(///IX. VOL START TIME NO EL LAST TRACK',
1 ' OVERFLOW AV VELOCITY',/IX,
2 'SCAN: DAY HHMM SS SCANS SCAN TOTAL ACT.',
3 ' NV IC I J EAST NORTH')
55 WRITE(6,1004) NVSCN, JDAY, JHR, JMIN, JSEC, NFN, NSCAN, NT, NACT
1004 FORMAT(1X, 15, 14, 13, 12, 13, 15, 216, 15, 17, 314, 1X, 2F6 1)
NVSCN=NVSCN+1
NVO=0
ICO=0
IO=0
JO=0
JDAY=IDAY
JHR=JHR
JMIN=JMIN
JSEC=JSEC
FNSN=1.009
KTL=T
RETURN
END
Two types of output are produced following each volume scan:

1. A list of attributes describing each of the 31* most significant cells detected and tracked within the volume scan:
   a. ITCL (17) - the ID number of this cell track
   b. VR (1) - cell location to east [(-)west] of radar (km)
   c. VR (2) - cell location to north [(-)south] of radar (km)
   d. IZVAL = VR (3) - average reflectivity of cell (dBZ)
   e. TCL (5) - cell volume (km$^3$)
   f. ITCL (7) - peak reflectivity of cell (dBZ)
   g. TCL (8) - height of peak reflectivity (km)
   h. ITCL (10) - reflectivity at base of cell (dBZ)
   i. TCL (12) - height of cell base (km)
   j. ITCL (15) - reflectivity at summit of cell (dBZ)
   k. TCL (16) - height of cell summit (km)
   l. VXP = VCL (39)*10$^5$ - cell velocity toward east [(-)west] (m/s)
   m. VYL = VCL (40)*10$^3$ - cell velocity toward north [(-)south] (m/s)
   n. ID'C = IVCL (9)/(FNSN-l)*10
   number of scans cell detected/number of scans per volume scan
   percent of elevation scans cell detected at

2. A summary list of statistics for the entire volume scan:
   a. NV:CN - volume scan number
   b. JD:Y, JHR, JMIN, JSEC - start time of volume scan
   c. NFN = FNSN-1 - number of elevation scans processed in this volume scan
   d. NSCAN - number of last elevation scan in the volume scan
   e. NT - number of cell tracks updated this scan
   f. NACT - number of possible cell tracks stored from current and previous scans
   g. NSO - number of significant cells detected but not outputted
   h. ICO - number of internally paired and clustered cells over the dimensions of CLUST array in subroutine RESOLV
   i. IO - number of detected cells over array dimensions of cluster array
      in subroutine COMPAR
   j. JO - number of detected cells over array dimensions of IC array in subroutine COMPAR
   k. VXP - an estimate of the average velocity east [(-)west] any new cells will have (m/s), set to 0 on first scan
   l. VYP - an estimate of the average velocity north [(-)south] any new cells will have (m/s), set to 0 on first scan

*see Appendix F - Option to Increase Number of Significant Cells
### SAMPLE OUTPUT OF FIRST VOLUME SCAN

**PROGRAM TRACK 4 (781207)***

<table>
<thead>
<tr>
<th>TRK</th>
<th>LOCATION</th>
<th>DBZ</th>
<th>CELL PEAK</th>
<th>SURFACE</th>
<th>SUMMIT</th>
<th>VELOCITY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>KM</td>
<td>KM</td>
<td>KM</td>
<td>KM</td>
<td>M/S</td>
</tr>
<tr>
<td>17</td>
<td>-90.</td>
<td>40</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
</tr>
<tr>
<td></td>
<td>-69.</td>
<td>34</td>
<td>3</td>
<td>34</td>
<td>34</td>
<td>34</td>
</tr>
<tr>
<td>2</td>
<td>-96.</td>
<td>37</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
</tr>
<tr>
<td>18</td>
<td>-89.</td>
<td>32</td>
<td>47</td>
<td>47</td>
<td>47</td>
<td>47</td>
</tr>
<tr>
<td>3</td>
<td>-56.</td>
<td>67</td>
<td>42</td>
<td>42</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>19</td>
<td>-88.</td>
<td>38</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
</tr>
<tr>
<td>20</td>
<td>-86.</td>
<td>38</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
</tr>
<tr>
<td>5</td>
<td>-94.</td>
<td>34</td>
<td>47</td>
<td>47</td>
<td>47</td>
<td>47</td>
</tr>
<tr>
<td>21</td>
<td>-97.</td>
<td>53</td>
<td>39</td>
<td>39</td>
<td>39</td>
<td>39</td>
</tr>
<tr>
<td>4</td>
<td>-50.</td>
<td>-66</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
</tr>
<tr>
<td>22</td>
<td>-91.</td>
<td>50</td>
<td>42</td>
<td>42</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>30</td>
<td>-93.</td>
<td>37</td>
<td>46</td>
<td>46</td>
<td>46</td>
<td>46</td>
</tr>
<tr>
<td>6</td>
<td>-43.</td>
<td>63</td>
<td>37</td>
<td>37</td>
<td>37</td>
<td>37</td>
</tr>
<tr>
<td>23</td>
<td>-37.</td>
<td>54</td>
<td>52</td>
<td>52</td>
<td>52</td>
<td>52</td>
</tr>
<tr>
<td>7</td>
<td>-4.</td>
<td>25</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>14</td>
<td>-72.</td>
<td>53</td>
<td>44</td>
<td>44</td>
<td>44</td>
<td>44</td>
</tr>
<tr>
<td>8</td>
<td>-50.</td>
<td>60</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
</tr>
<tr>
<td>24</td>
<td>-75.</td>
<td>83</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
</tr>
<tr>
<td>25</td>
<td>-73.</td>
<td>56</td>
<td>39</td>
<td>39</td>
<td>39</td>
<td>39</td>
</tr>
<tr>
<td>9</td>
<td>-27.</td>
<td>2</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>43</td>
</tr>
<tr>
<td>10</td>
<td>-14.</td>
<td>25</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>11</td>
<td>-6.</td>
<td>15</td>
<td>42</td>
<td>42</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>12</td>
<td>-80.</td>
<td>61</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>13</td>
<td>-17.</td>
<td>18</td>
<td>39</td>
<td>39</td>
<td>39</td>
<td>39</td>
</tr>
<tr>
<td>15</td>
<td>-4.</td>
<td>53</td>
<td>49</td>
<td>49</td>
<td>49</td>
<td>49</td>
</tr>
<tr>
<td>26</td>
<td>-57.</td>
<td>82</td>
<td>43</td>
<td>43</td>
<td>43</td>
<td>43</td>
</tr>
<tr>
<td>16</td>
<td>-6.</td>
<td>1</td>
<td>41</td>
<td>41</td>
<td>41</td>
<td>41</td>
</tr>
<tr>
<td>27</td>
<td>-54.</td>
<td>80</td>
<td>42</td>
<td>42</td>
<td>42</td>
<td>42</td>
</tr>
<tr>
<td>28</td>
<td>-62.</td>
<td>55</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>36</td>
</tr>
<tr>
<td>29</td>
<td>-52.</td>
<td>-78</td>
<td>45</td>
<td>45</td>
<td>45</td>
<td>45</td>
</tr>
</tbody>
</table>

---

**VOL SCAN START TIME NO EL LAST TRACK OVERFLOW AV VELOCITY**

<table>
<thead>
<tr>
<th>SCAN</th>
<th>DAY</th>
<th>HHMM</th>
<th>SS</th>
<th>SCANS</th>
<th>TOTAL ACT.</th>
<th>NV</th>
<th>IC</th>
<th>J</th>
<th>EAST</th>
<th>NORTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>120</td>
<td>1646</td>
<td>59</td>
<td>3</td>
<td>4</td>
<td>32</td>
<td>31</td>
<td>25</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

---

E-3
APPENDIX F

Options to Increase:
Number of Significant Cells
Number of Active Tracks
Option: Increase Number of Significant Cells Processed Each Scan

Both the real-time program and the post-mission program are set up to process 16 significant cells each scan. To increase the number of cells the dimensions of several arrays and address offsets must be increased.

In block common DATA1 - for each additional cell
  increase dimension of ECL by 14
  increase NOPST* by 1
  increase ICLAD* by 7

In block common RSLV - for each additional cell
  increase dimension of: IUC1, IUC2&UC by 1

In block common NVLIS - for each additional cell
  increase NCARM* by 1

In subroutine RESOLV - for each additional cell
  increase dimension of V by 24
  increase KOE2&KOF3 by 8

Note: to process more than 32 cells each scan the number of active tracks must be increased to at least that number (see option to increase active tracks).

*variable set in block DATA
Option: Increase Number of Active Tracks Updated Each Scan and Outputted Each Volume Scan

Both the real-time program and the post-mission program are set up to update and output 31 active tracks at any one scan. To increase the number of tracks, the dimensions of several arrays and addresses offsets must be increased.

In block common DATA2 - for each additional track
  increase dimension of VCL and IVCL by 23
  increase MXVC* by 23

In block common DATA3 - for each additional track
  increase dimension of VR and IVR by 6
  increase MXVR* by 7

In block common RSLV - for each additional track
  increase dimension of: HUV1, HUV2 & IV by 1

In block common CDRAYS - for each additional track
  increase the first dimension of: IC, IC, ID & ID by 1
  increase IM* by 1

In block common NVLTS - for each additional track
  increase NVARM* by 1

In block common COMB - for each additional track
  increase the first dimension of IV by 1

In subroutine STRAK - for each additional track
  increase dimension of ATST & NUM by 2
  increase IOUT by 1
  increase LAN by 2

*variable set in block DATA