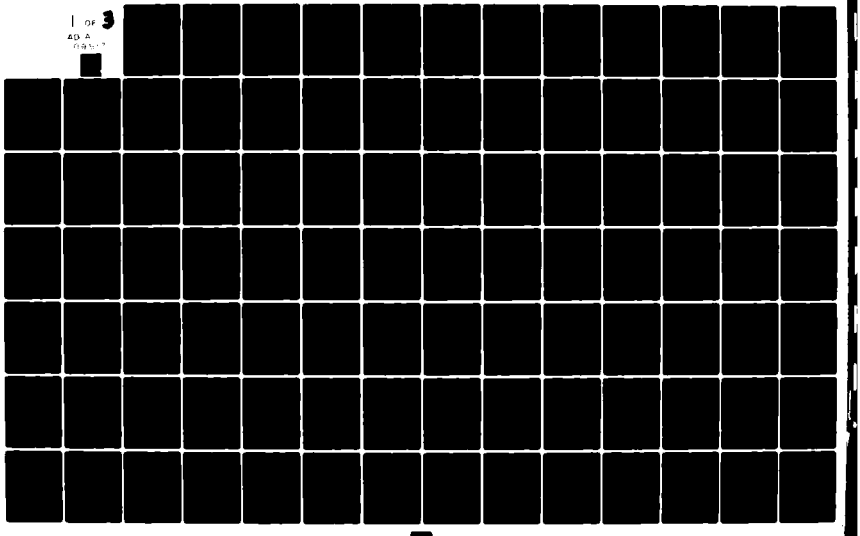
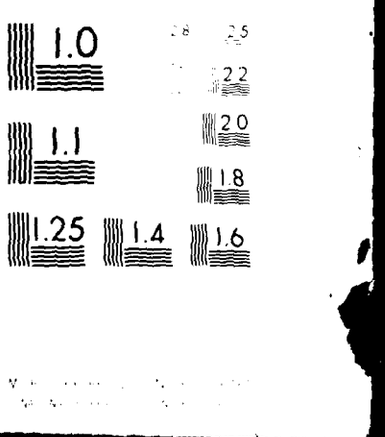
AD-A109 517 ENVIRONMENTAL RESEARCH AND TECHNOLOGY INC CONCORD MA F/G 1/2
DETECTION AND TRACKING ALGORITHM REFINEMENT.(U)
OCT 81 G B GUSTAFSON, R K CRANE DTFA01-81-Y-10521
UNCLASSIFIED ERT-P-B035 FAA/RD-81/80 NL





Within the 20 dBZ contour regions, cell detection is performed
... the reflectivity and Doppler data fields. Five Doppler

DOT/FAA/RD-81/80

Systems Research &
Development Service
Washington, D.C. 20590

Detection and Tracking Algorithm Refinement

G. B. Gustafson
R. K. Crane

October 1981

Final Report

U.S. Department of Transportation
Federal Aviation Administration

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
DOT/FAA/RD-81/80	AD-A109 517	

4. Title and Subtitle	5. Report Date
Detection and Tracking Algorithm Refinement	October 1981
	6. Performing Organization Code
7. Author's	8. Performing Organization Report No
G.B. Gustafson and R.K. Crane	P-B035 Technical Report
9. Performing Organization Name and Address	10. Work Unit No TRAIS
Environmental Research & Technology, Inc. 696 Virginia Road Concord, Massachusetts 01742	156-410-01W
	11. Contract or Grant No
	NA81RAC00072/DTFA01-81Y10521
12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
NOAA/ERL and U.S. Dept. of Transportation Boulder, Colorado FAA Washington, D.C.	Technical Report
	14. Sponsoring Agency Code
	FAA/ARD-410

15. Supplementary Notes Prepared under DOT, FAA/DOC, NOAA Interagency Agreement DTFA01-81Y10521 "Terminal Area Weather Radar Detection and Convective Prediction Development" Managed by the Aviation Weather Branch, ARD410.

16. Abstract

A previous aircraft hazard detection algorithm shown to have high detectability but with a high false alarm rate has been modified to improve reliability for aircraft warnings. The derived Doppler parameter tangential (or radial) shear is incorporated as a radar cell attribute and used in the determination of significant hazard. Further modifications to the processing structure allow for radar operation in a non-automatic mode, thereby accomodating arbitrary changes in PRF, integrator type or scan geometry. A revised output format provides a sorted hierarchical list of derived meteorological structures in a form readily adapted to a graphics display.

17. Key Words	18. Distribution Statement
Thunderstorm Turbulence Doppler Radar Cell Tracking Storm Detection Weather Radar Data Processing	Document is available to the U.S. public through the National Technical Information Service, Springfield, Virginia 22151

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	199	

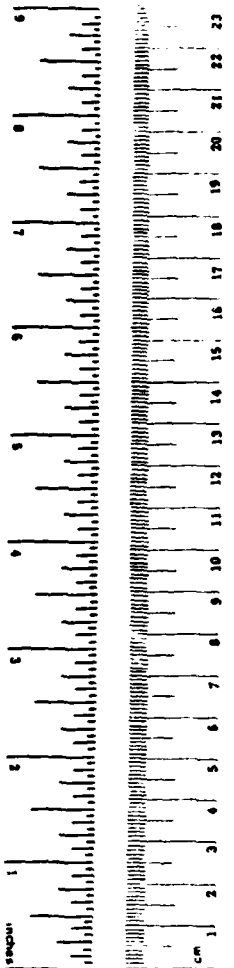
Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

METRIC CONVERSION FACTORS

Approximate Conversions to Metric Measures

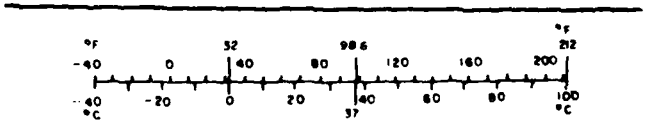
Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
in	inches	*2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
		AREA		
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
		MASS (weight)		
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
		VOLUME		
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
		TEMPERATURE (exact)		
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

*1 in = 2.54 (exactly). For other exact conversions and more detailed tables, see NBS Misc. Publ. 286, Units of Weights and Measures, Price $2.25 SD Catalog No. C13.10:286.



Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
		AREA		
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
		MASS (weight)		
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	
		VOLUME		
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
		TEMPERATURE (exact)		
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F



PREFACE

Algorithm development was performed on the MITRE Corp. Testbed for Automated Flight Services (TAFS) VAX 11/780 computer system in McLean, Virginia. The author wishes to express his appreciation for the assistance of the computer support people at TAFS, most especially Mr. Arthur McClinton.

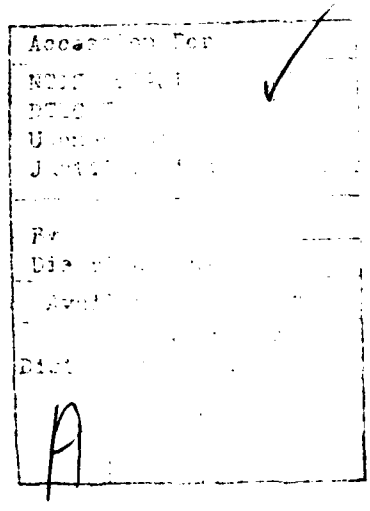


TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

LIST OF ABBREVIATIONS AND SYMBOLS

A	area
ACT VCL	number of active (updated) volume cells
AGL	above ground level
AV	average
AVG	average
AZM	azimuth
C	calibration constant
CC	complete contour
CLS CNTR	total cluster counter
cm	centimeters
CNT CNTR	total contour region counter
CN TR NO	contour track ID number
CPU	central processing unit
CS TR NO	cluster track ID number
D	Doppler parameter being processed (i.e. tangential shear)
dB	decibel
dBm	decibels relative to one milliwatt
dBZ	decibel measure of radar reflectivity
DEG	degree
DIV	mean cell divergence relative to cluster centroid
DOP	number of associated Doppler peak cells
DOP SPD	Doppler spread
DS	Doppler spread
E	east
EM/S	east-west velocity component in m/s
H	height
HGT LW	cell base height
HGT MN	cell centroid height
HGT HI	cell top height
HHMMSS	hours minutes seconds
HT	height
ID	identification number
ISO SHR	volume cell composed of only Doppler peak cells

KKM2	10^3 square kilometers
KM	kilometers
KMT/H	10^3 metric tons per hour
kw	kilowatts
LAT	latitude
L_I	integrator level
LNGTH	length
LONG	longitude
LVL	level
m	meters
MHz	megahertz
MR ID	merger ID
MT/HR	metric tons per hour
Mx	maximum
N	north
N	average receiver noise
NCL	number of clusters
NEAR DIST	nearest neighbor distance
NM/S	north-south velocity component in m/s
NN DIST	nearest neighbor distance
No	number
NO CS	number of clusters
NO FC	number of fixed contour regions
NO RD CS	number of cells in cluster used in rotation/divergence computation
NSC	number of isolated significant cells and clusters
NUM	number
NVC	number of volume cells
ORT	orientation angle from north
OVER	register overflow counter
PK	peak
P_R	received power
PRF	pulse repetition frequency
P_T	transmitter power
PWR	power
R	range

RAD VEL	radial velocity
REF	number of associated reflectivity cells
REJ	cell status code
RNG	range
ROT	cell rotation rate around cluster centroid
R_s	Doppler spread resolution
R_v	radial velocity resolution
RV	radial velocity
S	seconds
S	south
SC	isolated significant cells and clusters
SP ID	split ID
SPR	spread
TAN SHR	tangential shear
TRANS	transmitter
TRK	track
TRK NO	total volume cell counter
V	radial velocity
VBAR	average velocity
VC	volume cell
VOL	volume
V_x	east-west velocity component
V_y	north-south velocity component
W	west
WF	water flux
WTR FLUX	water flux
X	position component east of origin
XSCN	cross section
Y	position component north of origin
Z	reflectivity level

1. INTRODUCTION

A series of algorithms have been developed that provide automatic objective analysis of digital Doppler radar data. Crane (1979) described the radar cell detection and tracking routine that is employed in the current computer program. Regions of interest observed with the radar are located by contouring the reflectivity field at the 20 dBZ level. Within a contour region, relative peaks in the reflectivity or Doppler change (shear) fields are located and subjected to a multi-threshold contouring routine. Peak areas that are isolated by a contour which is 3 quantization steps (dB for reflectivity) below the peak value are declared cells. For each cell and contour region a set of attributes is accumulated which describe the characteristics of that entity. Peak cells detected at different elevation angles in a volume scan sequence are used to construct three-dimensional volume cell structures over which composite sets of attributes are accumulated. The volume cell becomes the smallest resolvable entity in the hierarchy that is developed to characterize the meteorological situation being observed.

Individual cell motion is established by associating volume cells detected on successive volume scans and tracking their centroid positions. Cell association is performed through a comparison of key attribute values and selection is made through a statistical best match function. The resultant velocity vectors must conform to a Lagrangian tracking scheme wherein a more precise track fit is required of longer lived cells.

Closely spaced volume cells that interact are declared clusters and are tracked as a single entity. Volume cells that have a high reflectivity and some degree of vertical continuity are considered to be important and are called significant cells. Isolated significant cells are classified as clusters that the radar was unable to resolve. The individual cells enclosed within clusters and contours are used to compute the motion and other attributes of the larger entities such as their physical extent, centroid location and lifetime.

At the completion of each volume scan sequence, the computed attributes compiled over the scan are output. Within the output hierarchy of contours, clusters and volume cells, each entity is assigned a unique identification number which it maintains throughout its lifetime. Each cell carries two attributes that point upward to identify the enclosing cluster and contour on.

An operational version of the cell detection and tracking routine was used to analyze non-coherent digital radar data on the Water and Power Resources Service (WPRS) Cyber-74 computer system (Gustafson, 1980). This program was initially modified to process Doppler data on the same computer (Crane, 1981). In the Cyber-74 version, a single Doppler attribute, tangential shear, was extracted and processed to form tangential shear cells. The shear cells were detected by using the magnitude of tangential shear in exactly the same way reflectivity cells had been formed using the magnitude of the reflectivity level. An identical attribute set was formed for each type (reflectivity or tangential shear) of cell which was then transferred to the track routine where the cells were associated without regard to type.

The Doppler processing routine described in this report is an expansion of the initial operational routines. It is designed to operate on a VAX 11/780 computer in near real time. The algorithm descriptions in this report primarily cover modifications to the program versions documented by Crane (1979) and Gustafson (1980).

2. THE ALGORITHMS AND THEIR USE

2.1 Calibration

Two separate subroutines are available to accept and calibrate raw integrator data from the radars in Cimarron and Norman, Oklahoma. The calibration procedure is essentially identical for the two radars, the only differences being the capability to handle the two formats of the raw data tapes (listed in Appendix D) and the integrator level/received power look-up tables. Three parameters are prepared by the calibration routines which are then passed to the cell detection and tracking program, these are: Reflectivity, Radial Velocity and the Second Moment of the Radial Velocity (Doppler Spread). A new data set containing pre-calibrated data in a form directly accessable by the cell detection and tracking program can be prepared by passing a flag to the calibration routine (refer to Appendix A). Thereafter, all runs of the program on that data set should use the routine "EXPAND" in place of the calibration routine in order to input the prepared data.

Raw radar data tapes contain received power information in the form of integrator levels set at the radar site. To convert this information into a reflectivity level in dBZ the integrator level is converted to power in dBm by a table look-up and then plugged into the equation:

$$Z = P_R + 10 \log_{10} R^2 - 10 \log_{10} P_T + C \qquad (1)$$

where Z is reflectivity in dBZ, P_R is received power in dBm which is obtained from the calibration table, R is range in km, P_T is transmitter power in Kw and C is a calibration constant. The computed reflectivity is then smoothed over a user defined number of range gates (generally two).

The magnitude of the Doppler parameters is extracted directly from the data tapes and calibrated by Equation (2) for radial velocity and Equation (3) for Doppler spread:

$$RV = (L_I - R/2)\, R_v \qquad (2)$$

where RV is radial velocity in m/s, L_I is integrator level, R is the range of integrator levels and R_v is the velocity resolution in m/s.

$$DS = R_s^2 \, (1-(1 - L_I^2) \, P_R/(P_R - N)) \qquad (3)$$

where DS is Doppler spread (m/s) , R_s is the spread resolution in m/s, L_I is the integrator level, P_R is received power in dBm and N is the average receiver noise in dBm.

The Doppler information has an inherent frequency ambiguity that the algorithm will resolve in one of two ways depending on the integrator mode in use at the time measurements were made. In the expanded integrator mode the Doppler data are collected at a PRF which is four times greater than that used to obtain the reflectivity data. The correct range interval is determined through an evaluation of the corresponding reflectivity levels at each of the folded range intervals. Normal integrator data are collected at a constant PRF and the range interval is determined by the Nth trip indicator that is operator selectable.

Independent of integrator type, the first few range gates of each fold (range interval) are eliminated from processing to minimize the effects of close in ground clutter. Further, each of the three output parameters are subjected to noise thresholding at each range gate along the radial. Calibrated data are then passed to the cell detection routine through block common.

2.2 Contouring and Cell Detection

Calibrated reflectivity data along each radial are contoured through a nested search routine at two predefined, fixed threshold levels. A base level of 20 dBZ is used to define regions in the data field within which the cell detection algorithms will be run. A second user defined level is available for display purposes only. As each radial of data is contoured a vector is defined to connect the contour endpoints on the current radial to the corresponding endpoints on the prior radial. Each new vector is assigned the identification (ID) number of any previously defined vector it connects with. Vector elements tagged with their ID and threshold level are accumulated in a temporary disk file over the entire azimuth scan. Similarly, merges and splits of contour segments are stored in a directory as they occur. After an entire scan has been processed, each vector ID is passed through the merge/split directory and reassigned a final contour ID. Vectors with a common contour ID are then linked to construct a full contour.

Within the 20 dBZ contour regions, cell detection is performed separately on the reflectivity and Doppler data fields. Five Doppler parameters are available for processing; (1) tangential shear, (2) radial shear, (3) vector shear, (4) range normalized radial velocity magnitude, and (5) range normalized Doppler spread (second moment). The magnitude of the user selected Doppler parameter is quantized into 1-dB steps in order to be in a form consistent with the reflectivity data. The cell detection algorithm is designed to locate peak regions in any quantized scalar data field that are at least 3 dB above the surrounding data and to declare them peak cells. Reflectivity and quantized Doppler data are processed by two separate passes through the cell detection routine thereby generating two peak cell data sets. Each data set is comprised of an identical cell attribute list (Table 1) describing the characteristics of the indivi diaul cells. The cell types are stored separately to allow discriminate processing by the track routine.

2.3 Cell Tracking

At the conclusion of an azimuth scan, the cell detection routine has produced both reflectivity and Doppler cell attribute lists which are stored separately. The track routine attempts to associate individual peak cells to a volume cell track established during prior scans.

In previous versions of the tracking algorithm (Gustafson, 1980), the azimuth scan sequence was assumed to have been performed in an automatic mode. For tracking purposes a volume scan was closely defined to be a series of full azimuth scans made in the same direction; these scans consisted of a fixed number of constant elevation steps requiring a fixed amount of time. In order to accomodate data collection in a non-automatic mode, the volume scan has been more loosely defined to be a series of at least two azimuth sector scans made in either direction that require greater than 150 seconds to measure. The cell-to-track association criteria have been similarly modified to minimize the impact of the data collection mode on the tracking process.

Of greatest impact is the variability in elevation steps between azimuth scans that is permitted in the new format. A volume cell is constructed from the association of individual peak cells detected on successive scans in a volume scan sequence. A large elevation gap between

TABLE 1

PEAK CELL ATTRIBUTES

1	Reflectivity - average*
2	Area
3	X } reflectivity or Doppler†
4	Y } weighted centroid position
5	Cell ID
6	Height (AGL) at centroid position
7	Range to centroid position
8	Doppler Value - average
9	Radial Velocity - average
10	Doppler Spread - average

*average values are computed over a region enclosed by a contour set 3 dB below the peak magnitude value

†position weighted by the parameter being processed, either reflectivity or Doppler

scans would normally cause the association logic to reject the new cell from inclusion in the volume cell structure. However, a relaxation of the height separation criteria of the association logic to accomodate large elevation steps could cause invalid associations such as that of an immature cell at a mid level with the cirrus overhang from a nearby mature storm. Clearly a trade-off is required; thus, the weight of the height component of the association function is defined such that a separation of between 2.5 and 3.0 km will make an association difficult (i.e. require close agreement between the other components), and a separation greater than 3 km will cause the association to be rejected.

When a cell-to-track association is successful the volume cell attribute list is updated to include the characteristics of the new cell. A list of the volume cell attributes is given in Table 2. Determination of which individual attributes are selected for update depends on the volume scan sequence and the cell type. The first seven attributes of the list are defined on the initial azimuth scan of each volume scan only and are used as a reference to the cell base. The spatial attributes such as position and vertical extent are updated on reflectivity cells only. Doppler cells are themselves considered to be an attribute of the volume cell, not a part of its physical structure. This is primarily due to the spatial distribution of turbulent shear regions around an active convective element (Crane, 1981). The reflectivity and shear peaks are not generally coincident; rather several regions of high shear may form in the vicinity of one reflectivity defined radar cell. Therefore, the Doppler cells contribute to the velocity attributes only. Captions to the right in Table 2 indicate when each attribute is updated and in which subroutine the update occurs.

Under one set of conditions the segregation of Doppler cells is overridden. An isolated Doppler peak cell that is detected outside the association range of any reflectivity cell is declared a new volume cell and its position tracked as if it were a reflectivity cell. These isolated Doppler cells often occur at a location that will, at a future time, produce a reflectivity cell. Tracking the Doppler cells in this manner yields more information about the reflectivity cell when it is eventually detected than if the isolated Doppler cells were discarded.

TABLE 2

VOLUME CELL ATTRIBUTES

1	X } reflectivity weighted	
2	Y } centroid position (km)	
3	Z: average reflectivity (dBZ)	Updated at the beginning of each volume scan
4	A: area (km^2)	
5	H: height (km)	BTRAK
6	D: Doppler value (m/s/km)	
7	R: range (km)	
8	Cell track ID	
9	Peak Cell Count	
10	Peak Cells Below Height = HM	
11	Σ Z	
12	Σ Z · X	
13	Σ Z · X^2	Updated on reflectivity and isolated Doppler cells only
14	Σ Z · Y	
15	Σ Z · Y^2	
16	Σ Z · X · Y	ATRAK
17	Σ Z · A	
18	Σ Z · H	
19	H: lowest	
20	Z: peak	
21	Z: at summit height	
22	H: summit	
23	Σ D	
24	Σ V: radial velocity (m/s)	Updated on both reflectivity and Doppler cells
25	Σ V^2	
26	Σ DS: Doppler Spread (m/s)	
27	Σ DELW: association measure	ATRAK/VTRAK
28	spare	
29	Peak Doppler Cell Count	
30	Σ D	Updated on Doppler cells only
31	Σ V	
32	Σ D · X	VTRAK
33	Σ D · Y	
34	Σ D · H	

2.4 Program Output

Data output is generated at the conclusion of each volume scan sequence. A sort routine creates a direct access binary file that contains a hierarchical list of all attributes defined or updated on the current scan. Appendix C provides a complete word description of the output format. A second formatted summary output is available as a user option (Appendix A).

The sort hierarchy consists of (1) contour regions, (2) enclosed cell clusters, (3) volume cells contained in clusters, and (4) isolated volume cells. A two record scan summary is provided as a header, and sorted contour vector information is appended as a trailer. Vector data are sorted on (1) contour track ID and (2) threshold level. Table 3 gives a schematic illustration of the sort file structure. A separate sort file is created at the end of each volume scan. At the conclusion of a program run, the individual files should be merged in the order they were created to produce a sequential record of the processed data (refer to sample command procedure Figure 4).

The optional formatted output can be configured to provide contour, cluster, or volume cell attributes, or any combination of these attribute types. Figure 1 illustrates a sample output displaying all three data types for one volume scan; the listing sequence is repeated for each volume scan. Whenever a change in PRF or integrator type is encountered in the data, the current volume scan is terminated and a radar attribute summary (Figure 2) is output.

2.5 Program Organization

The computer program consists of 22 subroutines with flow established through the main program module "DOPLR80". Figure 3 (a,b) illustrates how control cascades through the subroutines during one volume scan sequence. Note that subroutines PEAKD and COMPARE process reflectivity and Doppler data separately and each require two calls. The cell detection process is executed on each radial of data (Figure 3a) whereas the cell tracking routines are called once at the end of each azimuth scan (Figure 3b).

Each subroutine is stored in a separate file with the common generic name "FOR". The complete program is stored in an object module library referenced as "DOPLR80.OLB". This allows for editing and

TABLE 3

SORT FILE STRUCTURE

Header Records (2)	(Volume Scan 1
Contour Records (2)	(Contour 1
Cluster Record (1)	(Enclosed in Contour
Cell Record (1)	(Contained in Cluster
Cell	
Cell	
Contour	(Contour 2
Cluster	
Cell	
Cell	
•	
•	
•	
Cell	(Isolated
Vector Record (1)	
Vector	
•	
•	
•	
Header Records	(Volume Scan 2
•	
•	
•	
END OF DATA	

SCAN TIME 99 173150 - 173251 VOL SCAN 1 AZ - 2.2 TO 1.2 (DEG)
TRACK REF TIME 173049 - 173150 AZM SCAN 2/2 EL - 0.2 TO 0.4 (DEG)
AVG NOISE LEVEL = -106.6 (DBM)
AVG INTEGRATOR = 4.326
ISO SHR CLUSTERS 47

FIXED CONTOUR OUTPUT

TRK NO	CENTROID E. KM	CENTROID N. KM	AV CELL E. KM	AV CELL N. KM	Z AV DB	Z PK DB	N V C	N S C	N C L	SPR X KM	SPR L KM	O R T	WTR FLUX MT/H	AREA XSCN KKM2	VELOCITY AV EM/S	VELOCITY CELL NM/S	NEAR DIST KM	MX HI KM	MR ID NO	SP ID NO
1	7	104	7	107	34	36	1	1	0	0.0	0.0	0	0.02	0.01	0.0	0.0	0.0	1	0	0
2	4	10	6	13	34	35	1	0	0	0.0	0.0	0	0.00	0.00	0.0	0.0	0.0	0	0	0
3	160	226	162	228	33	34	1	0	0	0.0	0.0	0	0.04	0.01	0.0	0.0	0.0	6	0	0
5	147	178	148	179	35	35	1	1	0	0.0	0.0	0	0.11	0.03	0.0	0.0	0.0	5	0	0
6	166	205	167	208	34	35	1	0	0	0.0	0.0	0	0.02	0.01	0.0	0.0	0.0	6	0	0
7	151	159	151	160	43	45	2	2	0	0.0	0.0	0	0.98	0.13	0.0	0.0	0.0	4	0	0
8	185	187	179	185	42	49	4	1	0	2.2	9.4	61	2.21	0.34	0.0	0.0	0.0	6	0	0
9	-4	4	-1	3	58	68	3713			510.8	10.8	44	52.86	0.51	0.0	0.0	2.7	0	0	0

VOLUME CELL OUTPUT

TRK NO	CENTROID - E. KM	CENTROID - N. KM	- Z - AV DB	- Z - PK DB	- Z - LW DB	- Z - HI DB	HGT L W	HGT M N	HGT H I	VBAR EM/S OLD	CELL NM/S ID	SPACIAL SPRD KM	SPACIAL A KM2	(TAN) (SHR) (MSK)	DOP SPD MSK	RAD VEL M/S	RAD SPD M/S	CS TR NO	CN TR NO	D O F	R E F	R E J
1*	7	107	34	36	36	31	1	1	1	16.0	5.8	0.00	3.7	1.3	1.1	-13.4	9.5	0	1	1	2	1
2	1	15	56	56	56	56	0	0	0	16.0	5.8	0.00	0.7	8.2	14.9	-19.5	3.0	1	9	1	1	1
3*	3	15	51	54	54	54	0	0	0	16.0	5.8	0.00	0.8	4.5	13.5	-17.3	0.4	1	9	1	2	1
4*	4	16	44	45	44	45	0	0	0	16.0	5.8	0.00	0.9	5.7	5.7	-20.4	1.8	1	9	2	2	1
6*	6	18	54	56	56	51	0	0	0	16.0	5.8	0.00	0.8	8.0	5.6	-24.2	3.6	1	9	0	2	1
7	8	20	48	48	48	48	0	0	0	16.0	5.8	0.00	1.1	0.0	0.0	0.0	0.0	1	9	0	1	1
215	5	-10	47	47	47	47	0	0	0	16.0	5.8	0.00	1.0	8.5	13.0	0.8	0.0	0	9	0	1	1
217	-8	-9	58	58	58	58	0	0	0	16.0	5.8	0.00	0.9	0.0	0.0	0.0	0.0	0	9	0	1	1
218	-9	-6	64	64	64	64	0	0	0	16.0	5.8	0.00	0.5	3.5	10.1	-22.0	0.0	0	9	0	1	1
220	-14	-1	68	68	68	68	0	0	0	16.0	5.8	0.00	0.7	10.0	9.8	-11.8	0.0	0	9	0	1	1
223	-130	-256	34	34	34	34	7	7	7	16.0	5.8	0.00	21.6	3.7	0.0	-4.1	0.0	0	0	1	0	1

CLUSTER OUTPUT

TRK NO	CENTROID E. KM	CENTROID N. KM	Z AV DB	Z PK DB	N V C	SPR X KM	SPR L KM	ORT ANG DEG	CNT ID	VELOCITY AV EM/S	VELOCITY CELL NM/S	SHEAR MSKM	MX HT KM	MR ID NO	SP ID NO	CELL ROT. MSKM	CELL DIV. MSKM	NO RD CS
1	8	13	55	66	18	3.7	5.2	317	9	0.0	0.0	8.0	0	0	0	0.00	0.00	0
2	14	-10	32	32	1	0.0	0.0	0	27	0.0	0.0	10.0	0	0	0	0.00	0.00	0
3	5	-18	48	49	1	0.0	0.0	0	9	0.0	0.0	5.7	0	0	0	0.00	0.00	0
5	0	-25	37	39	1	0.0	0.0	0	28	0.0	0.0	4.6	0	0	0	0.00	0.00	0
6	-63	-155	36	36	1	0.0	0.0	0	38	0.0	0.0	2.4	3	0	0	0.00	0.00	0
8	-64	-143	41	41	1	0.0	0.0	0	38	0.0	0.0	1.3	2	0	0	0.00	0.00	0

VOL SCAN	HHMM	AREA KKM2	WFLUX KMT/H	NEAR CELL	NEIGHBOR CLST	NEIGHBOR CONT	ACT VCL	NO CS	NO FC	VELOCITY EM/S	VELOCITY NM/S	TRK NO	CLS CTR	CNT CTR	G C	OVER			
1	1730	14.8	297.39	8.7	12.9	0.0	145	8	6	16.0	5.8	228	16	45	0	0	9	0	0 0

Figure I Volume Scan Output

```
    * NORMAN *      EXPANDED INTEGRATOR

DAY  99 1980    - -    173049 CST

   PRF ---------   1084.60 (/S)
   WAVE LNGTH -      10.53 (CM)
   FREQUENCY ---   2850.00 (MHZ)
   TRANS PWR ---     28.75 (DBM)
   NOISE LVL ---   -106.60 (DBM)
   BEAM WIDTH -       0.81 (DEG)
   VEL RESOLN -       0.92 (M/S)
   MAX VEL -----     28.54 (M/S)
   SAMPLES -----     64.00 (/GATE)
   GATES -------    762
   ELEMENTS ---    380
   RNG DELAY ---  -310.00 (M)
   RNG INCR ----  1439.63 (M)
   FOLD RNG - -    1'9.20 (KM)
   FOLD GATE -       95
   NUM FOLDS -        4
   CONTOUR
    LEVEL(1) --      30     (DBZ)
    LEVEL(2) --      40     (DBZ)



COMMON ORIGIN -    NORMAN RADAR SITE
                 35.2365 N.LAT   97.4633 W.LONG

MEASUREMENTS ----- NORMAN RADAR SITE
                 35.2365 N.LAT   97.4633 W.LONG
        OFFSET    0.0000 KM N     0.0000 KM E
```

Figure 2 Radar Calibration Attributes

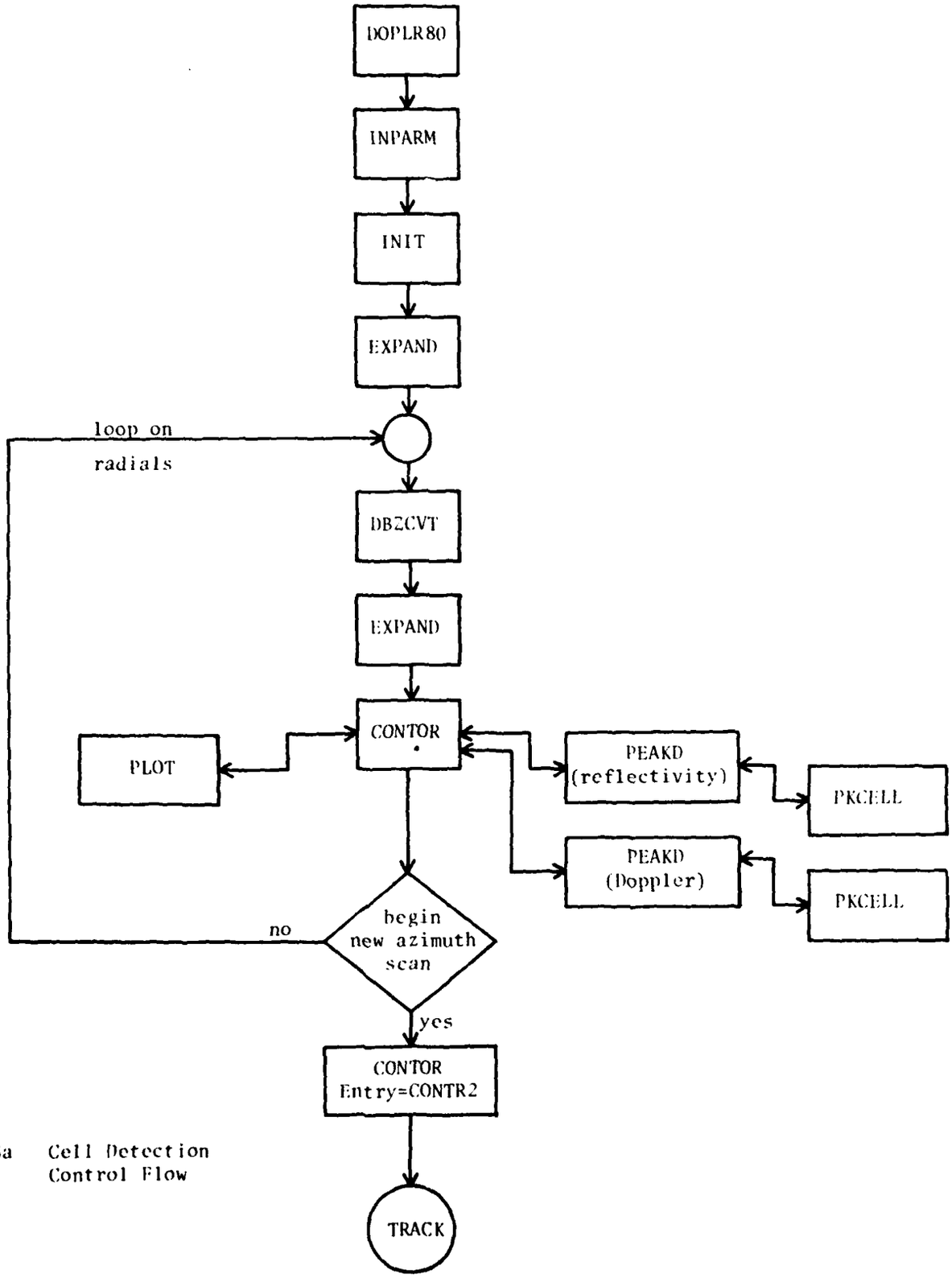


Figure 3a Cell Detection
Control Flow

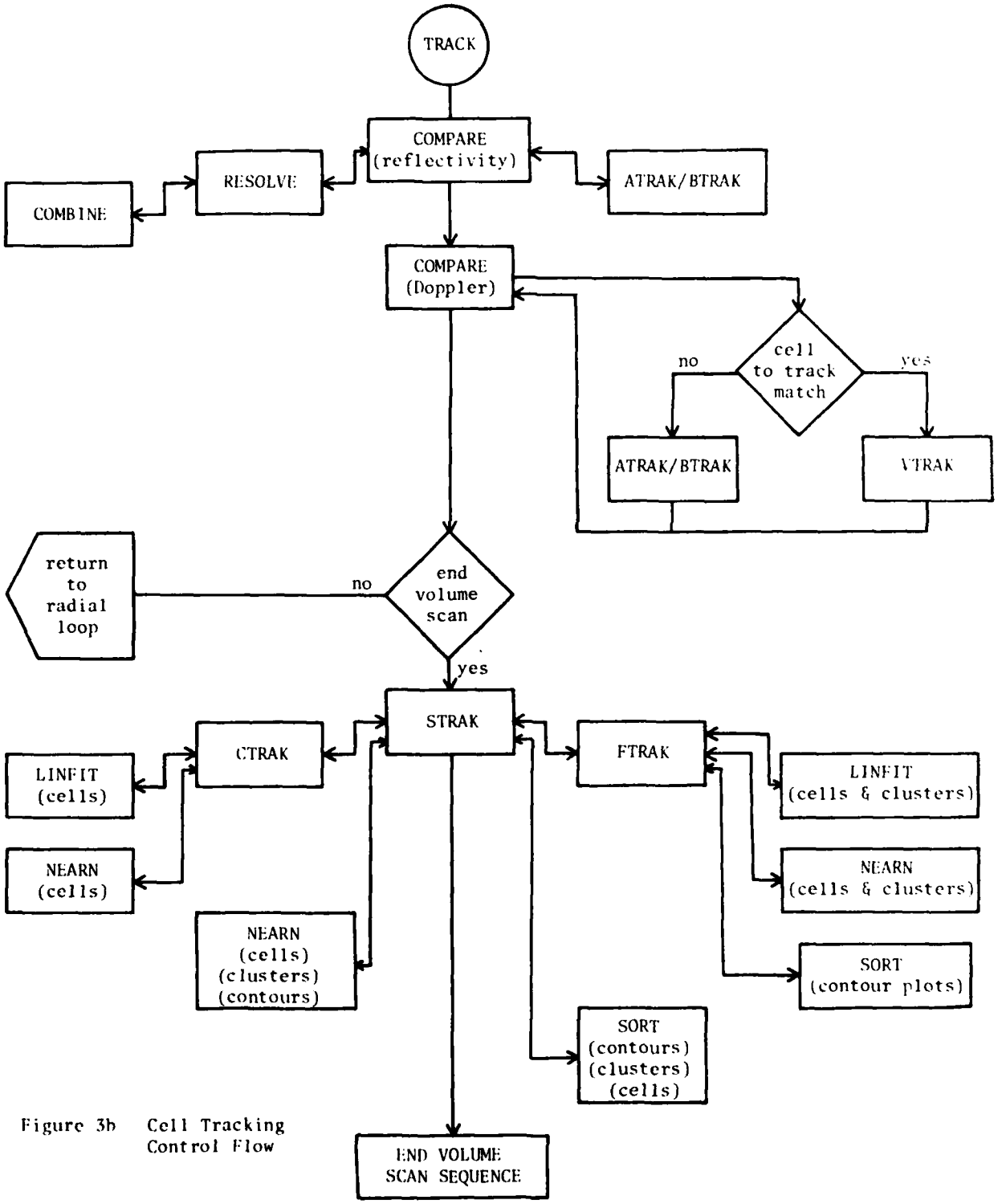


Figure 3b Cell Tracking Control Flow

recompilation of individual subroutines. The command module "LDOP.COM" creates an executable image from the object library and should be invoked just prior to program execution.

2.6 Program Execution

The computer program operates on two types of data; (1) raw integrator output and (2) calibrated reflectivity, radial velocity and Doppler spread. Operation on calibrated data requires about 20% less CPU time than raw form. There are three data input modules available. Two calibration routines input raw integrator data, perform data calibration, reformat the data, pass the calibrated data to the cell detection routine, and optionally produce a calibrated data set. Two routines are required to process data from both the Cimarron and Norman radars since each generates output in a different format (Appendix D). Each routine is stored in a file referenced by the corresponding radar name, CIMARRON.FOR and NORMAN.FOR. The third routine, in file EXPAND.FOR, inputs pre-calibrated data and passes it directly to the cell detection routine. EXPAND accepts data from either radar since the reformatting procedure is identical in the two calibration routines.

Figures 4 and 5 illustrate sample command procedures for processing raw and calibrated data respectively. DOPTAP.COM (Figure 4) can be run either interactively or in batch mode. It requires three input parameters, P1, P2 and P3. P1 is the label of the tape on which the raw data are stored, P2 is the file specification for the various output files to be produced, and P3 is the radar name. During interactive operation the user will be prompted for each input; in batch mode the input parameters must be supplied, in order, by a parameter qualifier. For example, the command:

```
SUBMIT/PARAMETER=(TAPLBL,DATFIL,RADNAM)  DOPTAP
```

will place the job DOPTAP into the batch execution queue and substitute TAPLBL for P1, DATFIL for P2 and RADNAM for P3.

After receiving the user inputs the command procedure attempts to allocate an MTA tape drive. If a drive is not currently available the procedure waits five minutes and repeats the process until successful. The operator is requested to load the data tape volume on the allocated drive and the tape is mounted.

In the LOOP section the disk directory is searched for an existing file with the same specification and generic name (SRT) as the output file

```
$!    DOPTAP COMMAND PROCEDURE
$!    OPERATES DOPLRBD PROGRAM ON UNCALIBRATED
$!    RADAR DATA LOCATED IN 4 FILES ON MAGNETIC TAPE
$!    INPUT PARAMETERS ARE P1=TAPE LABEL, P2=DATA FILE, P3=RADAR ...
$!
$ NL: DUMMY SYS$PRINT
$ NVERS=0
$!
$!    INPUT DATA TAPE, FILE AND RADAR NAME
$!
$ IF "  F$MODE() " .EQS. "BATCH" THEN -
       GOTO ALLOC_DEV
$ INQUIRE P1 TAPE LABEL
$ INQUIRE P2 FILE NAME
$ INQUIRE P3 RADAR NAME
$!
$!    ALLOCATE TAPE DRIVE
$!
$ ALLOC_DEV:
$ ALLOCATE MTA:  MT
$ IF  $STATUS  THEN GOTO MOUNT
$ WAIT 00:05
$ GOTO ALLOC_DEV
$!
$!    LOAD AND MOUNT DATA TAPE
$!
$ MOUNT:
$ REQUEST REPLY "PLEASE LOAD TAPE  P1  ON  F$LOGICAL("MT") "
$ MOUNT BLOCKSIZE=4144 MT  P1
$!
$!    FIND NEXT VERSION OF DOPT OUTPUT FILE
$!
$ LOOP:
$ NVERS=NVERS+1
$ OPEN ERROR=ASSN TEST_STAT  P2 .DAT; NVERS
$ CLOSE TEST_STAT
$ GOTO LOOP
$!
$!  ASSIGN DEVICES
$!      1> RAW INTEGRATOR RADAR DATA ON MAG TAPE
$!      2> CALIBRATED AND FORMATTED DATA OUTPUT FILE
$!      4> INTEGRATOR-TO-POWER CONVERSION TABLE
$!      5> SUMMARY OUTPUT DATA FILE
$!
$ ASSN:
$ AS MT:NOP198.DAT F1
$ AS MT:NOP199.DAT F2
$ AS MT:NOP200.DAT F3
$ AS MT:NOP201.DAT F4
$ AS  P3 .DAT FOR006
$ AS  P3 .CAL FOR004
$ AS  P2 .SUM FOR005
$ AS  P2 .ERR FOR007
$ AS  P2 .DAT; NVERS    DAT_FILE
$!
$!    CLEAN UP ANY TEMPORARY .FORMAT FILE
$!
$ ON WARNING THEN GOTO LINK_RUN
$ DEL *.TEM;*,*,*;*
$!
$!    LINK CALIBRATION ROUTINE AND EXECUTE PROGRAM
$!
$ LINK_RUN:
$ ON ERROR THEN GOTO DISMT
$ LIB DOPLRBD  P3
$ RLOOP
$ RUN DOPLRBD
$!
$!    DISMOUNT RADAR DATA TAPE
$!
$ DISMT:
$ DISM MT
$ DEAL MT
$ REQUEST "UNLOAD  P1 "
$!
$!  COPY SORTED DATA FILES ONTO OUTPUT FILE
$!
$ COPY-LOG *.TEM;* SORT_FILE
$ DEL *.TEM;*,*,*;*
$ EXIT
```

Interactive Parameter Input

Tape Drive Allocation

Tape Mount

Directory Search

Logical Device Assignment

Delete Old Files

Link Calibration Routine and Execute

Dismount Tape and Deallocate Drive

Merge Output Files

Figure 4 DOPTAP.COM Command Procedure

about to be created. If such a file exists the output file is assigned the next higher version number. Input files are assigned to a logical name consisting of "F" plus a number indicating the order in which the files are to be processed. The number sequence must begin with 1 and be continuous (F1, F2, F3...). The output file for calibrated data is given the generic name PRE and assigned to device FOR002. The integrator-to-power conversion table for the appropriate radar is assigned input device FOR004. Formatted summary data output (controlled by PRCELL, PRCLUS and PRFIXC in subroutine INPARM) is written to a generic file SUM through device FOR006. Note device FOR006, if left unassigned, defaults to the user's terminal during an interactive run. Any parity errors encountered during the tape read are listed in the ERR file assigned FOR007.

The correct calibration routine for the particular radar in use (e.g., NORMAN) is inserted into the object library (DOPLR80.OLB) by the LIB command and individual object modules are linked to form an execution file by the separate command procedure "LDOP.COM". Actual program execution is performed by invoking the command module RUN DOPLR80.

Upon completion of the program run, the data tape is dismounted and the tape drive deallocated in order to allow access by other users. Two temporary output files are created after each volume scan, one for the various contour, cluster and cell attributes and a second for the contour vectors. These files are merged in the order they were produced to form a single unformatted, binary file containing a sequential record of the full observation period. Final bookkeeping involves deleting the temporary storage files.

A second command procedure is illustrated in Figure 5 which demonstrates batch only operation on a pre-calibrated data set. In this case, two input parameters are required; these are: the job name (P1) for assignment of output file specification and the radar name (P2). The calibrated data are stored in a multi-volume tape file that is assigned to input device FOR002. Note that the integrator-to-power look-up table is not required since the conversion process was performed during the earlier run through the calibration procedure. The input routine "EXPAND" is inserted into the object library in place of the calibration routine and the program is linked and executed as before.

```
$!    DOPCAL COMMAND ROUTINE
$!    OPERATES DOPLRAN PROGRAM ON CALIBRATED RADAR DATA
$!    LOCATED IN A MULTI-VOLUME TAPE FILE " P2 ".DAT"
$!    INPUT PARAMETERS ARE P1=JOB NAME, P2=RADAR NAME
$!
$ SET NOON  ! NOPRINT
$ VERIFY
$!
$!    ALLOCATE TAPE DRIVE
$!
$ ALLOC_DEV:                                                    |
$ ALLOCATE MTA:  MT                                             |
$ IF  $STATUS  THEN GOTO MOUNT                                  |  Tape Drive Allocation
$ WAIT 00:05                                                    |
$ GOTO ALLOC_DEV                                                |
$!
$!    LOAD AND MOUNT DATA TAPE
$!
$ MOUNT:                                                        |
$ REQUEST REPLY "PLS LOAD TAPE SET ERTS ON  "F$LOGICAL("MT")  P 0"  |  Tape Mount
$ MOUNT MT: ERT8, ERT9, ERT10, ERT11                            |
$!
$!    FIND NEXT VERSION OF  ORT OUTPUT FILE
$!
$ LOOP:                                                         |
$ NVERS=NVERS+1                                                 |
$ OPEN ERROR=ACCN TEST_STAT  P1 .ERT; NVERS                     |  Directory Search
$ CLOSE TEST_STAT                                               |
$ GOTO LOOP                                                     |
$!
$!  ASSIGN DEVICES
$!      1> CALIBRATED AND FORMATTED DATA
$!      6> SUMMARY OUTPUT FILE
$!      7> ERROR OVERFLOW FILE
$!
$ ACCN:                                                         |
$ AS MT: P2 .DAT FOR006                                         |
$ AS  P1 .SUM FOR006                                            |  Logical Device Assignment
$ AS  P1 .ERR FOR007                                            |
$ AS  P1 .ERT; NVERS  .DAT_FILE                                 |
$!
$!    CLEAN UP ANY TEMPORARY STORAGE FILES
$!
$ ON WARNING THEN GOTO LINK_RUN                                 |  Delete Old Files
$ DEL *.DOP;*,*.TEM;*                                           |
$!
$!    LINK INPUT ROUTINE AND EXECUTE PROGRAM
$!
$ LINK_RUN:                                                     |
$ ON ERROR THEN GOTO DISMT                                      |  Link Data Input Routine
$ LIB DOPLRAN EXPAND                                            |  and Execute
$ @LOOP                                                         |
$ RUN DOPLRAN                                                   |
$!
$!    UNLOAD RADAR DATA TAPE
$!
$ DISMT:                                                        |
$ DISM MT                                                       |
$ DEAL MT                                                       |  Dismount Tape and
$ REQUEST "UNLOAD TAPE ON   F$LOGICAL("MT")  "                  |  Deallocate Drive
$!
$!    COPY SORTED DATA FILES ONTO OUTPUT FILE
$!
$ COPY LOG *.TEM;* SORT_FILE                                    |
$ DEL *.TEM;*,*.DOP;*                                           |  Merge Output Files
$ EXIT                                                          |
```

Figure 5 *DOPCAL.COM Command Procedure*

APPENDIX A

USER CONTROL PARAMETERS

User interaction with the computer program is accomplished through parameters set in subroutine INPARM. The user can (1) select which data are to be processed, (2) adjust the track association criteria, and (3) control the program output by defining values for the parameter list described below. Each time a parameter is changed the subroutine must be recompiled and replaced in the object library as illustrated in Appendix B.

PARAMETER	TYPE	FUNCTION
CONTRV	L	Controls whether Doppler cells are processed (T = process)
CONTRZ	L	Controls whether reflectivity cells are processed (T = process)
CALIBO	L	Controls the creation of a calibrated data set by the calibration routine (T = create data set)
PRCELL	L	Controls output of volume cell attributes through device FOR006. (T = generate output)
PRNOIS	L	Controls output of volume cells that have been rejected by the noise/ ground clutter filter. Valid only when PRCELL = T (T = generate output)
PRFIXC	L	Controls output of fixed contour attributes through device FOR006. (T = generate output)
PRCLUS	L	Controls output of cluster attributes through device FOR006. (T = generate output)
COPLOT	L	Controls generation of contour vectors at each elevation in a volume scan sequence (T = generate at all elevation angles)
CEPLOT	L	Controls generation of contour vectors at lowest elevation of volume scan sequence only (T = generate at base only)

PARAMETER	TYPE	FUNCTION
IAVGR	I	Defines the number of range gates over which the reflectivity data are to be averaged (must be ≥ 2)
BDAY	I	Julian day to begin data processing
BEGINT	I	Time to begin data processing (HHMMSS)
EDAY	I	Julian day to stop data processing
ENDT	I	Time to stop data processing (HHMMSS)
LT	I	Controls user selected reflectivity contouring (2 = generate user defined contours)
ITL(LT)	I	Reflectivity level in dBZ at which data are to be contoured
VD	R	User estimate of direction of cell motion, in degrees from north, used to initiate tracking algorithm
VW	R	User estimate of magnitude of cell motion in m/s
ISHR	I	Controls which Doppler parameter is to be processed 1) tangential shear 2) radial shear 3) vector shear 4) radial velocity 5) Doppler spread
DIV*	R	Weight of position in cell association function
ZDIV*	R	Weight of reflectivity in cell association function
HDIV*	R	Weight of height in cell association function
ADIV*	R	Weight of area in cell association function
A1†	R	Weight of current velocity in track velocity equation

*The measure of cell-to-cell association is given by
$(\Delta X^2 + \Delta Y^2)DIV + (\Delta REFL)RDIV + (\Delta HEIGHT)HDIV + (\Delta AREA)ADIV$

†The track velocity equation is
$(V_{CURRENT})\ A1 + (V_{TRACK})\ A2 + (V_{AVERAGE})\ A3$

PARAMETER	TYPE	FUNCTION
A2[†]	R	Weight of prior track velocity in track velocity equation
A3[†]	R	Weight of average velocity in track velocity equation
B1[+]	R	Weight of current average velocity in average velocity equation
B2	R	Weight of prior average velocity in average velocity equation

[+]The average cell velocity equation is
$(\Sigma V/N)$ B1 + $(V_{AVERAGE})$ B2

APPENDIX B

SAMPLE EDITING SESSION OF SUBROUTINE INPARM

```
$ EDIT INPARM.FOR                    !EDIT THE CONTENTS OF FILE INPARM
EDIT: DBC5:[CRANE]INPARM.FOR;1
*FISHR=$                             !SEARCH FOR THE STRING "ISHR="
14200          ISHR=1
*S1$2$                               !SUBSTITUTE "2" FOR "1"
14200          ISHR=2
*EB                                  !END EDITING SESSION
[DBC5:[CRANE]INPARM.FOR;1]

$ FOR INPARM                         !COMPILE THE NEW VERSION

$ LIB DOPLR81 INPARM                 !INSERT MODUAL INTO OBJECT LIBRARY

$ @LDOP                              !CREATE AN EXECUTABLE IMAGE

$ RUN DOPLR81                        !EXECUTE THE PROGRAM
```

APPENDIX C

OUTPUT FILE WORD FORMAT

HEADER RECORD (1) STRAK

Type	Word	Contents	Variable	Common
I	1	Time	KTL	NVLIS
I	2	Contouring thresholds 10^3*ITL(2)+ITL(1)	ITLS	KNCTR
I	3	Vol Scan Counter	NVSCN	PNTRS
R	4	Az Min @ NEL=1	AZLO	AZENDS
R	5	Az Max @ NEL=1	AZHI	AZENDS
R	6	Vx Avg Velocity of Cells Updated	VXS	
R	7	Vy This Scan	VYS	
R	8	Total Area	AFCS	DATA4
R	9	Total Water Flux	WFCS	DATA4
R	10	Avg NN Dist - V Cells Updated This Scan	DNN	
R	11	Avg NN Dist - SC	DCN	
R	12	Avg NN Dist - Clusters	DCA	
R	13	Reference Azimuth for Plotting	AZREF	
I	14	Radar ID Code; (13=NRO) (21=CIM)	IRADAR	AZENDS
I	15	Number Active (updated this scan) Cell Tracks	NACT	
I	16*	Number SC	KNCL=NCL+10^3	DATA5
I	17*	Number Contours	KNFL=NFL+10^3	DATA5
I	18*	Data Type = 1	IONE	

*Sort Parameters

NN DIST = Nearest Neighbor Distance

HEADER RECORD 2 FTRAK

Type	Word	Contents	Variable	Common
I	1	Time	KTL	NVLIS
R	2	E-W Offset	DLONG	DECODE
R	3	N-S Offset	DLET	
R	4	NN DIST VC Enclosed in CC	DVFN	
R	5	NN DIST SC Enclosed in CC	DFN	
R	6	NN DIST CC	DCN	
R	7	Total Area @ CC	ARCC	
R	8	Clus Index @ CC'	CI	
R	9	$\overline{WF}$ @ CC'	WFB	
R	10	WF/SC @ CC^{+}	WFS	
R	11	WF/AREA @ CC	WAB	
I	12	Number VC Enclosed in CC	NCV	
I	13	Number SC Enclosed in CC	NCS	
I	14	Number CC'	NCI	
I	15	Number CC	NCC	
I	16*	Number SC KNCL=NCL+10^3	KNCL	DATA5
I	17*	Number Contours KNFL=NFL+10^3	KNFL	DATA5
I	18*	DATA TYPE=2	ITWO	

*SORT Parameter

CC - Complete Contours

CC' - NCV $\geq$ NNMIN

CC^{+} - NCX $\geq$ 1

FIXED CONTOUR (1) FTRAK

Type	Word	Contents	Variable	Common
I	1	Time	KTL	NXLIS
I	2	Avg Reflectivity	FCL(I,2)	DATA4
R	3	X } Refl Weighted Centroid	FCL(I,3)	DATA4
R	4	Y }	FCL(I,4)	DATA4
R	5	Vx } Average Enclosed	FVX(I)	DFC
R	6	Vy } Cell Velocities	FVY(I)	DFC
R	7	Area	FCL(I,1)	DATA4
R	8	Water Flux	FCL(I,5)	DATA4
	9	Spare		
	10	Spare		
	11	Spare		
I	12	Merge Pointer	IFMGE	
I	13	Split Pointer	IFTSP	
I	14	Age ((-) CC Flag)	IFAGE(I)	
I	15	Number V Cells (Enclosed-Active)	NFV	
I	16*	Number SC (Enclosed)=NSIG+1000	KNSC	
I	17*	Contour Track ID	IFTND(I)	DFC
I	18*	Data Type = 3	ITRRE	

*SORT Parameters

FIXED CONTOUR (2) FTRAK

Type	Word	Contents		Variable	Common
I	1	Time		ITL	NVLIS
	2	Spare	ENCLOSED VOLUMES		
R	3	X } Avg Cell Centroids	ENCLOSED VOLUMES CELLS	FX(I)	UFC
R	4	Y } Avg Cell Centroids	ENCLOSED VOLUMES CELLS	FY(I)	UFC
R	5	Avg NN DIST	ENCLOSED VOLUMES CELLS	DVFN	
R	6	Spread About Avg NN DIST	ENCLOSED VOLUMES CELLS	DVFS	
R	7	Orientation Angle	ENCLOSED VOLUMES CELLS	ANGC	
R	8	Spread About Avg Cell Centroid	ENCLOSED VOLUMES CELLS	SPRDC	
R	9	X } Avg SC Centroids	ENCLOSED SC's	FXI	
R	10	Y } Avg SC Centroids	ENCLOSED SC's	FYI	
R	11	Avg NN DIST	ENCLOSED SC's	DFN	
R	12	Spread About Avg NN DIST	ENCLOSED SC's	DFS	
R	13	Orientation Angle	ENCLOSED SC's	ANGS	
R	14	Spread About Avg SC Centroid	ENCLOSED SC's	SPRDS	
I	15	Number V Cells (Enclosed-Active)		NFV	
I	16*	Number SC (Enclosed)=NSIG+1000		KNSC	
I	17*	Contour Track ID		IFTNO(I)	UFC
I	18*	Data Type = 4		IFOUR	

*SORT Parameters

CLUSTER OUTPUT CTRAK

Type	Word	Contents	Variable	Common
I	1	Time	KTL	NVTIS
I	2	Reflectivity	IE=UCE( )	UVC
R	3	X } Refl Weighted Centroid	UCX( )	UVC
R	4	Y }	UCY( )	UVC
R	5	Vx } Avg of Enclosed V Cells	UCVX( )	UVC
R	6	Vy }	UCVY( )	UVC
R	7	Spread Enclosed V Cells	SPRD	
R	8	Summit Height	UCHS( )	UVC
R	9	X } Avg Enclosed Cell Locations	XPOS( )	
R	10	Y }	YPOS( )	
R	11	Orientation Angle of Enclosed Cells	BC	
I	12	Merge Pointer	ICMGE	
I	13	Split Pointer	ICTSP( )	UVC
I	14	Age	ICAGE( )	
I	15	Number V Cells (Enclosed-Active)	IN=UCN( )	UVC
I	16*	Cluster ID	ICTNO( )	UVC
I	17*	Contour ID (Enclosing)	IFXNO	
I	18*	Data Type = 5	IFIVE	

*SORT Parameters

VOLUME CELL STRAK

Type	Word	Contents	Variable	Common
I	1	Time	KTL	NHS
I	2	Reflectivity 10 LOG(VCL(11))	IZVAL	DATA2
R	3	X } Refl Weighted Centroid	VCL(12)	DATA2
R	4	Y } Refl Weighted Centroid	VCL(14)	DATA2
R	5	Vx } Smoothed Track Velocity	VCL(47)	DATA2
R	6	Vy } Smoothed Track Velocity	VCL(48)	DATA2
R	7	Area } Refl Weighted Avg	VCL(17)	DATA2
R	8	Height } Refl Weighted Avg	VCL(18)	DATA2
R	9	Spread } Refl Weighted Avg	VCL(42)	DATA2
R	10	Doppler } Unweighted Avg	VCL(23)	DATA2
R	11	Rad Vel } Unweighted Avg	VCL(24)	DATA2
I	12	Dop Spd } Unweighted Avg	VCL(26)	DATA2
I	13	Refl & Doppler Hits IVCL(9)*10^3+IVCL(29)	NHIT	DATA2
I	14	Age	IVCL(53)	DATA2
I	15	Cell Track ID IABS(IVCL(8))	ITRKNO	DATA2
I	16*	Cluster ID	IVCL(52)	DATA2
I	17*	Contour ID IABS(IFTNO(NF))	IAFXNO	UCF
I	18*	Data Type = 6	ISIX	

*SORT Parameter

CONTOUR PLOT VECTORS (TRSL)

Type	Word	Contents		Variable	Comment
I	1	Time		KTL	NWDS
I	2*	Thresh dBZ		ITL(1	KNCTR
R	3	X_1	Segment 1	DAT(1)	
R	4	Y_1		DAT(2)	
R	5	X_2		DAT(3)	
R	6	Y_2		DAT(4)	
R	7	X_1	Segment 2	DAT(5)	
R	8	Y_1		DAT(6)	
R	9	X_2		DAT(7)	
R	10	Y_2		DAT(8)	
R	11	X_1	Segment 3	DAT(9)	
R	12	Y_1		DAT(10)	
R	13	X_2		DAT(11)	
R	14	Y_2		DAT(12)	
I	15*	ID from Contour (Segment)		ID	
I	16	ID of Low (Enclosing) Contour		IDB	
I	17*	Track ID		NF	
I	18	Data Type = 7		ISEVEN	

*SORT Parameter

APPENDIX D

RADAR DATA FORMATS

Time Series Records

Position	Contents
1-3	Julian date
4-9	Time HHMMSS
10-13	Azimuth XXX.X degrees
14-16	Elevation XX.X degrees
17-20	Range XXXX μs
21	Record type indicator 27_8
22-23	Number of gates recorded
24	Bit A 0 = 3 μs, 1 = 5 μs pulse width
24-25	Pulse width XX μs
26-29	Number of samples per gate XXXX
30	Nth trip indicator: bits 2-1 Antenna direction in horizontal: bit 4 0 = CCW, 1 = CW Antenna direction in vertical: bit 8, 0 = down, 1 = up Magnetic tape drive selector: bit A *Low gain, High gain:* bit B, 0 = low, 1 = high
31	Collection mode trigger: bits 2-1 00 - continuous, 01 = time, 10 = azimuth, 11 = external Selectable PRT bits 8-4 00 = 768 μs, 01 = 922, 10 = 1075, 11 = 1229 bit A 0 = normal integrator, 1 = expanded integrator bit B 0 = normal pulse width, 1 = wide pulse width
32	Gate spacing in μs
33-35	Collection trigger increment - time mode XX.X secs, azimuth mode XX.X degrees
36	Collection mode bits 4-2-1 000 Sector (Constant tilt) 001 RHI (Constant azimuth) 010 Time series (antenna stopped) 011 Coplanar 100 Calibration AGC Switch bit 8 AGC on = 1 Integrator range averaging switch bits B-A: 00-1, 01-2, 10-4, 11-8

Raw Doppler Format 1979, Cimarron Radar

Position	Contents
37-38	Transmitter frequency 27XX mHZ
39	Number of integrator samples when in linear mode Bits 4-2-1, k = 0-7, 2^{k+3} When in exponential mode indicates time constant Integrator mode. Bit 8: 0-linear, 1-exponential Number of samples in PPP. Bits B-A 2^{k+5}
40-41	(not used)
42	Calibration switch
43-44	Calibration value
45-48	PRT XXXX μs (fixed at 0768, see position 31 for true PRT)
49-64	Integrator values used for AGC. 6 bit binary
65	Gate 1, In-phase, most 5 significant bits
66	Gate 1, In-phase, least 5 significant bits
67	Gate 1, Quadrature, most 5 significant bits
68	Gate 1, Quadrature, least 5 significant bits
69-72	Gate 2
⋮	
125-128	Gate 16
129-132	Gate 1, second sample
⋮	
189-192	Gate 16, second sample

Length of record depends on number of samples

No. of samples	Record length
32	2112
64	4160
128	8256
256	16448
512	32832
1024	65600
2048	131136

An additional number of unused bytes is always added to each record.
This varies from 3-11 bytes.

Raw Doppler Format 1979, Cimarron Radar

Integrator/PPP records

Position	Contents
1-20	Same as time series record
21	Record type 25_8
22-64	Same
65	Integrator value for gate 1, 6 bit binary
66	Pulse pair velocity for gate 1, 6 bit binary in 2's complement
67	Pulse pair spectral width for gate 1, 6 bit binary
68	Integrator for gate 2
69	Pulse pair velocity for gate 2
70	Pulse pair width for gate 2
⋮	
2348	Integrator for gate 762
2349	Pulse pair velocity for gate 762
2350	Pulse pair width for gate 762
2351-2368	Not used

RAW DOPPLER FORMAT 1979

NORMAN DOPPLER

Time Series Records (Low PRF, Channel A)

Position	Contents
1	17$_8$ (Bits 8, 4, 2, 1 on)
2-7	Time HHMMSS
8	8-4 bits indicate delta azimuth 01 = 0.5°, 10 = 1.0°, 11 = 2.0°, 00 = None
8-11	Azimuth XXX.X degrees
12-14	Elevation XX.X degrees
15	PPI collection mode indicator Bit 8 1 = PPI mode
15-17	Range XXX μs
18	Gate spacing X μs
19	No. of samples per gate 2^{k+4}, $1 \leq k \leq 7$
20	Number of samples in PPP Bits 2-1 2^{k+5}
21-22	Antenna speed (azimuthal) X.X degrees/second (bits 8-4-2-1)
21-22	Calibration switch (Pos 22, bits B-A and Pos 21, bits B-A correspond to 8-4-2-1 value)
23-25	Julian date (Pos 23, bits 2-1, Pos 24-25, bits 8-4-2-1)
23-25	Calibration value (Pos 23, bits B-A-8-4, corresponds to 8-4-2-1 of tens digit; Pos 25, bits B-A, Pos 24, bits B-A, corresponds to 8-4-2-1 of units digit)
26	8 bit on High PRF, 4 bit on B channel, off A channel, 2 bit high gain on, 1 bit AGC on A bit 0 = normal pulse width; 1 - wide pulse width B bit if wide pulse, 0 = 3 μs, 1 - 5 μs
27	Antenna direction azimuthally. Bit 1 0 = CCW, 1 - CW Antenna direction vertically. Bit 2 0 = down, 1 up Magnetic tape drive selector. Bit 4 Expanded integrator Bit 8 0 = normal, 1 = expanded Selectable PRT Bits B-A 00 = 768 μs, 01 - 922, 10 - 1075, 11 = 1229
28	Bits 4-2-1, Number of integrator samples when in linear mode. k = 0-7, 2^{k+3} When in exponential mode indicates time constant.

Raw Doppler Format 1979, Norman Doppler

Position	Contents
28 (cont)	Integrator mode Bit 8
	0 - linear, 1 - exponential
	Multiple trip indicator Bits B-A
29	Collection mode Bits 4-2-1
	000 Sector (Constant tilt)
	001 RHI (Constant Azimuth)
	010 Time Series (antenna stopped)
	011 Coplanar
	100 Calibration
	Bits B-A-8 Step number 0-7
30	N. A. internal use
31-46	Integrator values used for AGC. 6 bit binary
47	Total number of 16 gate steps 0-7
48	Int/OCC switch 0 - off; 1 = on
49	Gate 1, In-phase, least 6 significant bits
50	Gate 1, In-phase, most 6 significant
51	Gate 1, Quadrature, least 6 significant
52	Gate 1, Quadrature, most 6 significant
53-56	Gate 2
.	
.	
.	
109-112	Gate 16
113-116	Gate 1, second sample
.	
.	
.	
173-176	Gate 16, second sample
etc.	

Length of record depends on number of samples.

Pos. 19	No. of samples	Record length
1	32	2096
2	64	4144
3	128	8240
4	256	16432
5	512	32816
6	1024	65584
7	2048	131120

Page 5

Raw Doppler Format 1979, Norman Doppler

Time Series Records (High PRF, Channel B)

Position	Contents
1-24	Same as low PRF
49-52	Gate 1, In-phase/Quadrature
⋮	
69-72	Gate 6
73-76	Gate 1, second sample
⋮	
93-96	Gate 6, second sample
etc.	

Length of record depends on number of samples.

Pos. 19	No. of samples	Record length
1	32	816
2	64	1584
3	128	3120
4	256	6192
5	512	12336
6	1024	24624
7	2048	49200

Integrator/PPP records

Position	Contents
1	15_8 (Bits 8, 4, 1 on)
2-46	Same as time series records
47-808	Integrator values for 762 gates 6 bit: 0-63
809-1570	Pulse pair velocity values for 762 gates 6 bit: 0-63; 2's complement
1571-2332	Pulse pair spectral widths for 762 gates 6 bit: 0-63

APPENDIX E

GLOSSARY

GLOSSARY

Active Cell Track	A cell track that has been updated during the current volume scan.
Age	Number of volume scans the entity has been tracked over.
Cell Track Velocity	A weighted average of (1) the scan to scan velocity of the cell, (2) the mean of scan to scan velocities of all cells updated on the prior scan and (3) the previous track velocity. Updated at the end of each volume scan.
Cluster	A narrow grouping of volume cells that interact and are tracked together.
Complete Contour	A contour that is completely within the scan bounds of the radar.
Contour Plot Segment	An X/Y pair defining one vector on the perimeter of a contour region.
Contour Threshold	A fixed level, given in dBZ, at which the reflectivity field is contoured.
Isolated Significant Cell	A significant cell that is not a part of a cluster of cells. Considered to be a cell cluster not resolved by the radar.
Merge/Split Pointers	Track ID of the entity that the contour (or cluster) merged with or split off from.
Nearest Neighbor Distance	Average closest spacing between the centroid positions of cells, clusters or contours.
Orientation	The direction, relative to north, of a least squares line fit.
Position Offset	All position information is given relative to the Norman Radar Site: 35.23651°N Lat., 97.46333°W Long. Offsets are the range from Norman to the radar making the measurements.
SC	A classification that includes both clusters of cells and significant cells not contained in clusters.

	A volume cell displaying a high degree of vertical continuity or some vertical continuity and a high reflectivity.
	Second moment of V: $\overline{V^2} - \overline{V}^2$.
	A series of at least two azimuth scans, either partial or full, lasting more than 150 seconds taken over any range of elevation angles, stepped either up or down.
Water Flux	The areal integration of rain rate measured over a contour region.

APPENDIX F

LISTING OF COMPUTER PROGRAM CODE

```
C     ***********************************************************************
C
C     NAME:       DOPLOT
C     ABSTRACT:   TEST DOS SCAN STRATEGY
C
C     PURPOSE:    MONITORS SWEEP OF RADAR DATA AND DIRECTS
C                 CONTROL FLOW THROUGH THE SUBROUTINES.
C                 TESTS FOR CHANGE IN ANTENNA ROTATION DIRECTION
C                 OR ELEVATION ANGLE TO END AN AZIMUTH SCAN.
C                 TESTS FOR DROP BELOW FIXED ELEVATION ANGLE TO
C                 END VOLUME SCAN.
C
C     INTERFACES:
C        CALLING MODS.  NONE
C        CALLED MODS.   INFARM,INIT,EXPAND,DRZCVT,CONTOR,
C                       AIRAK,BIRAK,COMPARE,STRAK
C        COMMON BLOCKS
C          AZENDS,AZM,AZ2,CNT,COUNT,DATAS,DATAI,FLGS,IBAK,
C          INSUB,INTL,ISOS,NVLTS,NVLTT,PARM,PNTRS,PWORK,
C          RADCOM,REFL,TILTD,TLTS,UNUSE,VEL
C
C     COMMENTS:  DEFAULT REQUIRES MORE THAN 1 AZ SCAN IN ORDER TO
C                FORM A VOLUME SCAN.  ALGORITHM AUTOMATICALLY
C                OVERIDES DEFAULT AND BEGINS A NEW VOLUME SCAN
C                ON CHANGE OF PRF OR INTEGRATOR TYPE.
C
C     VERSION:   1.0  DEC/VAX 11-780
C     DATE:      4/30/81
C     DESIGN:    RKCRANE & GBGUSTAFSON
C     PROGMR:    GBGUSTAFSON
C
C     ***********************************************************************
C
      LOGICAL PRINT1,COPLOT,CONTRZ,CEPLOT,CONTRV,CALTBO,PROVER,
     +        PRCELL,PRSIG,PREIXC,PRCLUS,PRSCAN,PRHEAD,PRNOIS
      INTEGER SEC,TSEC
      INTEGER BDAY,EDAY,BEGINT,ENDT
      INTEGER W,WI,TS,TI,TO
      INTEGER HR,HV,B,C
      INTEGER YEAR,DAY,TIME
C
      DIMENSION IECL(10,128),IESCL(10,128)
C
C     ------------------------------------------------------------------
      COMMON /DATA1/   ECL(10,128),NCO,NCMX,NRJC
      COMMON /DATAS/   ESCL(10,128),NSCO,NSCMX,NSRJC
      COMMON /COUNT/   IXR,IXS
      COMMON /INTL/    MHSN,MNSN,HM,FNSN,FNSRN,NCLN,NFLN,MZSN,NNMIN,FCAZ
      COMMON /NVLIS/   NVARM,NCARM,NVO,NFO,ICO,IO,JO,JYR,IBL,KTL
      COMMON /NVLIT/   KTLL,NKNID,NKDO,IZTH,NKIMX,ITHR,IFXC(1024),HIST
      COMMON /TLIS/    TSEC,JDAY,JHR,JMIN,JSEC,IDAY,THR,TMIN,ISEC
      COMMON /PNTRS/   NVMIN,NVMX,IELSN,NSCAN,IESNL,NVSCN,NT
      COMMON /CNT/     COSPHI,SINEL,COSPI2,ZMIN,ELAST,SPRM,IFXMX
      COMMON /VEL/     TS(382),TI(382),HV(382),RV(382),RS(382)
      COMMON /REFL/    W(382),WI(382),HR(382)
      COMMON /PWORK/   KMAX,TO(100),JMXDB,JMAX,IAMAX,IR,JR,
     +                 IMXJMX,NCL,NID,NIDF,IMX,IMN
      COMMON /AZM/     AZMUTH,DAZF,AZLAST,NA,ELEVAT,B,C
```

```
      COMMON /PARM/    PRINT,COPLOT,CPPLOT,CONTRZ,CONTRV,CALIBO,
     +                 NUME,NUMR
      COMMON /FLGS/    PRCELL,PRSTG,PRLEXC,PRCLUS,PRSCAN,PRHLAD,
     +                 PRNOTS,PROVER
      COMMON /INSUB/   BEGINT,LNDT,DLTTR,SCON,ICOMP,
     +                 DAZM,BDAY,EDAY
      COMMON /AZZ/     SINA,COSA,DELTAZ,ISCANF,NEL
      COMMON /RADCOM/  YEAR,DAY,TIME,ITILT,IAZI,RDRM,RTRM,
     +                 BEAWI,PWRBAR,RAWBAR
      COMMON /TILTD/   DELT,ELTEST
      COMMON /ISOS/    ISOCTR
      COMMON /IBAR/    SUMN,NUM
      COMMON /UNUSE/   NPWC,NSER
      COMMON /AZENDS/  AZLO,AZHI,AZREF,ELOW,ELAVE,IRADAR
      COMMON /ICON/    SETFI,RNGDLY,SETEL,RANG(382),IHGT(382)
C     ------------------------------------------------------------------
C
      EQUIVALENCE (IECL(1,1),IECI(1,1))
      EQUIVALENCE (ESCL(1,1),IESCL(1,1))
C
      PARAMETER(RPD=.017453,DAZMIN=0.1)
      PARAMETER(ITMIN=150,NELMN=2)
      PARAMETER(IEOF=0,NORMAL=1,NEWPRF=2,INITAL=10)
C
      REWIND 1
C
C     INITIALISE CALIBRATION CONSTANTS AND TRACK COUNTERS
C
      CALL INPARM
      CALL INIT
      ISOCTR=0
      NVSCN=0
      NVNXT=NVSCN+1
      FNSN=1.009
      ICODE=INITAL
      KTL=0
      KILL=0
      NSCAN=1
      IFIRST=1
C
C     INPUT INITIAL RADIAL
C
      CALL EXPAND(ICODE)
C
C
      DAZT=2.5*BEAWI
      DELT=.5*BEAWI
      AZNXT=FLOAT(IAZI)/10.
      ELNXT=FLOAT(ITILT)/10.
    4 NEL=0
      NAZSC=0
      AZLO=AZNXT
      ITIME=TIME
      NVNXT=NVSCN+1
      OPEN(UNIT=8, STATUS='SCRATCH', FORM='UNFORMATTED')
C
C     PREPARE FOR NEW SCAN
C
    5 ISCANF=0
      ELSUM=0.0
```

```
      ENDIF
      IF(NEL.EQ.1)
C
      DSEL=1./SIN(ELNXT*RPD)
      II=1
      DO 11 N=1,14
      I=(N*DSEL-RDKM)/CLKM+1
      IF(I.GT.382) I=382
      DO 12 J=II,I
   12 IHGT(J)=N
      IF(I.GE.380) GO TO 13
   11 II=I
   13 CONTINUE
C
      NA=0
      AZSTAR=AZNXT
      ELSTAR=ELNXT
C
C     LOOP THROUGH EACH RADIAL IN AZIMUTH SCAN
C
   21 NA=NA+1
      AZLAST=AZMUTH
      AZMUTH=AZNXT
      ELEVAT=ELNXT
      ELSUM=ELSUM+ELEVAT
      IF(NA.EQ.10 .AND. NEL.EQ.1) AZRET=AZMUTH
C
C     PREPARE REFLECTIVITY AND VELOCITY DATA FOR PROCESSING
C
      CALL DBZCVT
C
C     INPUT NEXT RADIAL AND COMPUTE CHANGE IN AZ AND EL ANGLES
C
   81 CALL EXPAND(ICODE)
C
      IF(IDAY.GT.EDAY.OR.(IDAY.EQ.EDAY.AND.ITME.GE.ENDT)) ICODE=EOI
      AZNXT=FLOAT(IAZI)/10.
      ELNXT=FLOAT(ITILT)/10.
      DELTEL=ABS(ELSTAR-ELNXT)
C
C     TEST FOR END OF SCAN
C       (EL ANGLE CHANGE GT DELT)
```

```
      IF(DELTET.GT.DETT.OR.ICODE.NE.NORMAL) GO TO 121
C
C     AZIMUTH INCREMENT WITHIN LIMITS (DAZMIN TO DAZT)
C
      AZNXT=AZMUTH
      IF(ABS(DAZ).GE.DAZMIN) GO TO 82
      IF(NA.EQ.1) GO TO 20
      GO TO 81
   82 DELTAZ=DAZ
      IF (ABS(DAZ).GT.DAZT) GO TO 101
C
C    NEW ROTATION DIRECTION, END SCAN
C
      DAZS=SIGN(1.,DAZ)
      IF(NA.EQ.1)DAZF=DAZS
      IF(DAZF.EQ.DAZS) GO TO 141
      IF(DAZF.NE.DAZS)GO TO 121
C
C    CHECK FOR 360 - 0 DEG CROSSOVER
C
  101 DAZ=DAZ+360.*DAZF
      IF(ABS(DAZ).GE.DAZMIN) GO TO 102
      IF(NA.EQ.1) GO TO 20
      GO TO 81
   82 DELTAZ=DAZ
      IF (ABS(DAZ).GT.DAZT) GO TO 100
      DAZS=SIGN(1.,DAZ)
      IF (DAZF.NE.DAZS) GO TO 121
      GO TO 141
C
  100 DAZ=AZMUTH-AZLAST
      IF(ABS(DAZ).LT.DAZMIN) GO TO 81
C
C    LARGE DELTA AZ, CHECK FOR ABNORMAL ANTENNA ROTATION
C
      DELTAZ=AZNXT-AZMUTH
      IF(DELTAZ.LT.(-180.)) DELTAZ=DELTAZ+360.
      IF(DELTAZ.GT.180.) DELTAZ=DELTAZ-360.
      DAZS=SIGN(1.,DELTAZ)
      IF(DAZF.EQ.DAZS) GO TO 95
      AZST=AZSTAR
      DAST=AZMUTH-AZSTAR
      DASS=SIGN(1.,DAST)
      IF(DASS.NE.DAZF) AZST=AZSTAR-DAZF*360.
      IF(AZNXT.GT.AZST.AND.AZNXT.LT.AZMUTH.AND.DAZF.GT.0) GO TO 85
      IF(AZNXT.LT.AZST.AND.AZNXT.GT.AZMUTH.AND.DAZF.LT.0) GO TO 85
      GO TO 95
   85 IX1=IAZI/1000
      IX2=(IAZI-1000*IX1)/100
      IX3=(IAZI-1000*IX1-100*IX2)/10
      IF(PROVER) WRITE(7,2222)TIME,JHR,JSEC,AZMUTH,IX1,IX2,IX3
C
C    AZMUTH SHIFT TOO LARGE, CLOSE OFF CELLS
C
   25 IF(ABS(DAZ).GT.DAZT)DAZ=DAZU
      DELTAZ=ABS(DAZ*RPD)
      CALL CONTOR
      NA=NA+1
      ELSUM=ELSUM+ELEVAT
```

```
C
C    PROCESS FINAL RADIAL IN AZ SCAN
C
  131 ISCANT=ISCANF
      ISCANF=0
      DAZ=ABS(AZMUTH-AZLAST)
      DELTAZ=DAZ*REPD
C
C    PROCESS FINAL RADIAL IN SCAN
C
      IF(ABS(DAZ).GE.DAZMIN) CALL CONTOR
      ISCANF=ISCANT
      ELAVE=ELSUM/NA
      ELEVAT=ELAVE
C
      IF(NEL.GT.1) GO TO 130
      ELOW=ELAVE
      AZHI=AZMUTH
      AZNORM=-DAZF*RELAWT/2.
      AZLO=AZLO+AZNORM
      IF(AZLO.LT.0.) AZLO=AZLO+360.
      IF(AZLO.GT.360.) AZLO=AZLO-360.
      AZHI=AZHI+AZNORM
      IF(AZHI.LT.0.) AZHI=AZHI+360.
      IF(AZHI.GT.360.) AZHI=AZHI-360.
C
C    ASSOCIATE END AROUND CELLS
C
  130 CALL CONTR2
C
C    INCREMENT SCAN COUNTER
C
```

```
      NSCAN=NSCAN+1
      ENSCN=ENSCN+1
      IDAY=JDAY
      ITIM=ITIM
      IF(ICODE.EQ.IEOF) GO TO 200
C
C     SUBROUTINE REFL = BEAM CENTERS SCAN TO SCAN
C
      CALL COMPARE(ECL,NCMX,IECL,1)
      CALL COMPARE(ESCL,NSCMX,IESCL,2)
C
  155 CONTINUE
      IF(ELEV.LT.ELTEST) NAZSC=NAZSC+1
      ITIMO=TIME-ITIME
      IF((ITIMO.LT.ITMIN .OR. NEL.LT.NELMN .OR. ELNXT.GT.ELTEST)
     &    .AND. ICODE.EQ.NORMAL) GO TO 5
C
C     DUMP OUT VOLUME SCAN SUMMARY TO DISPLAY
C     CLEAN UP COUNTERS FOR NEXT VOL SCAN
C
      NVSCN=NVSCN+1
      IF(NVSCN.EQ.1) KTLT=KTL
      AVGINT=SUMN/NUM
      IF(IREREAD) WRITE(...,140) IDAY,IHR,IMIN,ISEC,TIME,
     &                           NVSCN,AZLO,AZHI,
     &                           JHR,JMIN,JSEC,IHR,IMIN,ISEC,
     &                           NEL,NAZSC,ELOW,ELAVE,
     &                           PWRBAK,AVGINT,ISOCTR
      ISOCTR=0
      CALL STRAK
      IF(ICODE.EQ.IEOF) STOP
      CLOSE(8)
      GO TO 3
C
C     STORE INITIAL SCAN REFLECTIVITY CELL ATTRIBUTES
C
  200 IFIRST=2
      IDAY=IDAY
      IHR=IHR
      IMIN=IMIN
      ISEC=ISEC
      IZMIN=ZMIN
      ITHR=IZHR*IZMIN*100+MZSN*1000
      IF(NCMX.LE.0) GO TO 40
      DO 10 NC=1,NCMX
      CALL ATRAK(NT,NC,NC,0.,ECL,NCMX,IECL,1)
      CALL RTRAK(NC,NC,ECL)
   10 CONTINUE
   40 NVMX=NCMX
      NVMIN=NVMX
      IF(NVMIN.LE.0) NVMIN=1
C
C     INITIALIZE CLUSTER ASSN.  - REFL TO REFL
C
      CALL COMPARE(ECL,NCMX,IECL,1)
      GO TO 135
C
C     THROW OUT SCANS OF LESS THAN 10 RADIALS
C
  210 NEL=NEL-1
```

```
[illegible]
[illegible]
      CLOSE(3)
      GO TO 4
[illegible] 1X,5H [illegible] 6,3H  HOUR [illegible]
     1          1X,8H AZMUTH [illegible] 1X [illegible]
     1          1X,5H EX [illegible] 1X,5H [illegible]
[illegible] FORMAT( 1 [illegible] 9X, SCAN TIME [illegible] I4 [illegible]
     1           3X,'VOL SCAN [illegible] I4 [illegible]
     1           9X,'TRACE REF TIME [illegible]
     1           3X, AZM SCAN [illegible] I3 [illegible] I1 [illegible]
     1          42X,'AVG NOISE LEVEL [illegible]
     1          42X,'AVG INTEGRATOR [illegible] I2,3 [illegible]
     1          42X,'ISO SHR CLUSTERS [illegible] I5 [illegible]
      END
```

```
      BLOCK DATA
C
C     ****************************************************************
C
C     NAME:       BLKDAT
C     DESCRIPT:   DEF (P=35, S=0) (AA)
C
C     PURPOSE:    INITIALIZE CONSTANTS AND ARRAYS
C
C     COMMON BLOCKS:
C        ARYM4,A2M,CDRAYS,CLIST,CNI,DATA4,DATA5,DVAL,
C        EDIT,FILTER,FIXED,FLGS,HEADC,INSUB,INTL,KNCTR,KNTRL,
C        MORED,MULT,NVLIS,NVLIT,OLFS,ORCOM,PARM,PRSTORE,PVSTORE,
C        PWORK,QUANTX,REFL,TITID,TLIS,TMAX,VEL,WTND,ZLOOK
C
C     COMMENTS:   STORAGE SYSTEM REQUIRES
C                  JR=KMAX
C                  IP1,2,3(1:JMX*JR,NEC)
C                  NIEMAX=NPA*IEMAX
C                  IB,ICC(NIEMAX*NEC)
C                  ITA,IPNT(KMAX,IEMAX,NEC)
C                  CTR,C1,2,3,DL(IEMAX*NEC)
C                  IDC,IDV(IEMAX)
C                  T(JMX1B)
C                  ICVNT,TRVNT(NEC)
C                  NIDAT=NIDP*IAT
C                  ATR(NIDAT*NEC)
C                  IDSLOT,DSI(NIDP*NEC)
C                  REFL_UP(NUP);  TSHR_UP(NUV)
C                  TATR,VATR(NIDP*NUMAX)
C                  KDD(NEC)
C                  IACT,IACV(NIDP),  IPRNG(JMAX)
C                  IPTA,IDC(IEMAX,NEC)
C                  IMN=2, IMX=NCL-1
C                  W,WT,TS,TT(NCL)
C
C     VERSION:    1.0  DEC/VAX 11-780
C     DATE:       4/30/81
C     DESIGN:     RKCRANE
C     PROGMR:     GROGSTAFSON
C
C     ****************************************************************
      LOGICAL PRINT1,COPLOT,CONTRZ,CEPLOT,CONTRV,PROVER,NTEST,CALIBO
      LOGICAL PRCELL,PRSIG,PRFIXC,PRCLUS,PRSCAN,PRHEAD,PRNOIS
      INTEGER WT,W,HR,TT,TS,HV,R,C
      INTEGER TL,BEGINT,ENDT,BDAY,EDAY
      INTEGER TSEC,TM,TML,TMX
      REAL*8 SVA,SVB,SVC,SA2,SB2,SC2,SAB,SAC,SBC,SV2,SB,SC
C
      COMMON /MULT/     VIDPT(4),NACTT,NTEST
      COMMON /TLIS/     TSEC,JDAY,JHR,JMIN,JSEC,IDAY,IHR,IMIN,ISEC
      COMMON /UNUSE/    NPWC,NSER
      COMMON /ORCOM/    COMLAT,COMLONG,COMHGT,COMRAD(2)
      COMMON /NVLIS/    NVARM,NCARM,NVO,NFO,ICO,IO,JO,JYR,LBL,KTI
      COMMON /NVLIT/    KTLL,NKNTD,NKDO,IZTH,NKDMX,ITHR,IFXC(1024),IITST
      COMMON /CLST/     ICLN,ICLIST(256),ICLMX
      COMMON /ZLOOK/    JZOFF,ZZARY(91),KRATE(91)
      COMMON /INTL/     MHSN,MNSN,HM,FNSN,FNSRN,NCLN,NFLN,
     1                  MZSN,NNMLN,FCAZ
      COMMON /CDRAYS/   IC(128,10),CD(128,9),ID(128,10),D(128,9),
```

```
     +                    IM,JM,MCIDX
      COMMON /IVAL/       [illegible]
      COMMON /IMAX/       IM,IMI,IMX
      COMMON /DATA4/      ECL(9,256),AFCS,WFCS,NIMX,GEAUM,
     +                    KNIDC(1024),NFTA
      COMMON /CNT/        COSPHI,SINEL,COSP2,ZMIN,FLAST,CPHM,IFXMX
      COMMON /FIDIR/      IFCDIR(512)
      COMMON /DATA5/      NCCL,NFL,JCL,JCTNO(128),WCX(256),WCY(256),WCZ(
     1                    JFTNO(256),WFX(256),WFY(256),WFZ(256),
     2                    IFMG(256),ICMG(128),ICZC(128),NUMM
      COMMON /PARM/       PRINT1,COPLOT,CEPLOT,CONTRZ,CONTRV,CALCTG1,
     +                    NUMF,NUMR
      COMMON /FLGS/       PRCELL,PRSIG,PRFIXC,PRCLUS,PRSCAN,PRHEAD,
     +                    PRNOIS,PROVER
      COMMON /OFFS/       IZOFF,FZOFF
      COMMON /INSUB/      BEGINT,ENDT,DELTR,SCON,TCOMP,
     +                    DAZM,BDAY,EDAY
      COMMON /AZM/        AZMUTH,DAZES,AZLAST,NA,ELEVAT,B,C
      COMMON /MORED/      INPRF,SCALE
      COMMON /FILTER/     TATRMN,AREAMN,CELMN(2),SVMX
      COMMON /QUANTX/     VQUANT
      COMMON /PWORK/      KMAX,T(100),JMXDB,JMAX,IAMAX,IR,JR,
     +                    IMXJMX,NCL,NID,NIDF,IMX,IMN
      COMMON /FIXED/      NFC,TL(2),ICC(256),IB(256),NFA,IEMAX,ICVNT(2),
     +                    IBVNT(2),ATR(2560),IAT,NIDF,KDD(2),
     +                    IDSLOT(512)
      COMMON /PRSTORE/    NUF,TATR(1856),NUMAX,IACT(64),IDC(32),IPRNG(64),
     +                    IPTAR(32,2),ITAR(30,32,2),IPNTR(30,32,2),
     +                    IP1R(1920,2),IP2R(1920,2),IP3R(1920,2)
      COMMON /PVSTORE/    NUV,VATR(1856),NVMAX,IACV(64),IDV(32),IPVRNG(64),
     1                    IPTAV(32,2),ITAV(30,32,2),IPNTV(30,32,2),
     +                    IP1V(1920,2),IP2V(1920,2),IP3V(1920,2)
      COMMON /VEL/        TS(382),TI(382),HV(382),RV(382),RS(382)
      COMMON /REFL/       W(382),WI(382),HR(382)
      COMMON /TILTD/      DELT,ELTEST
      COMMON /KNTBL/      KNID(2),KNIDA(2,1024)
      COMMON /KNCTR/      LT,ITL(2),KLVL,JNID,JNIDA(2,1024),
     +                    KNIDM(2),KNIDL(1,1024)
      COMMON /ARYMX/      NIDF2,NIEMX,NIDAT,NIDAT2
      COMMON /HEADC/      H1(3,5),H2(3,5),H3(3,5)
      COMMON /WIND/     SVA(14,8),SVB(14,8),SVC(14,8),
     +                  SA2(14,8),SB2(14,8),SC2(14,8),
     +                  SAB(14,8),SAC(14,8),SBC(14,8),
     +                  SV2(14,8),SB(14,8),SC(14,8),NUM(14,8)
C
      DATA COMLAT/35.23651/,COMLONG/97.46333/,
     +     COMHGT/0.3697/,COMRAD/'  NO','RMAN'/
C     DATA COMLAT/35.47533/,COMLONG/97.81314/,
C    +     COMHGT/0.39939/,COMRAD/'CIMA','RRON'/
      DATA KNID/2*0/,KNIDA/2048*0/,JNIDA/2048*0/,KNIDL/1024*0/
      DATA DELT/.5/,ELTEST/.6/,SCALE/1.0/
      DATA VQUANT/2.0/,B/2/,C/1/
      DATA PRINT1/.FALSE./,COPLOT/.FALSE./,CONTRZ/.FALSE./,
     1     CEPLOT/.FALSE./,CONTRV/.FALSE./,PROVER/.FALSE./
     2     PRCELL/.FALSE./,PRSIG/.FALSE./,PRFIXC/.FALSE./,
     3     PRCLUS/.FALSE./,PRSCAN/.FALSE./,PRHEAD/.FALSE./
      DATA NUMF/0/,NUMR/999/
C
C    ZMIN IS THRESHOLD LEVEL ON PEAK CELLS
C
```

```
      DATA ZMIN/20.0/,TSEC/0/
      DATA IZOFF/30/
      DATA SVA/112*0./,SVB/112*0./,SVC/112*0./,SA2/112*0./,SB2/11.*0.
      DATA SC2/112*0./,SV2/112*0./,SAB/112*0./,SAC/112*0./,SBC/11 *0.
      DATA SB/112*0./,SC/112*0./,NUM/112*0/
      DATA BEGINT/0/,ENDT/250000/,BDAY/0/,EDAY/366/
      DATA DELTR/1.0/,ICOMP/2/
      DATA AREAMN/0.5/,DAZM/0.017/
      DATA NPWC/0/,NSER/0/
      DATA IPTAR/64*0/,ITAR/1920*0/,IPNTR/1920*0/,
     +     IP1R/3840*0/,IP2R/3840*0/,IP3R/3840*0/
      DATA IPTAV/64*0/,ITAV/1920*0/,IPNTV/1920*0/,
     +     IP1V/3840*0/,IP2V/3840*0/,IP3R/3840*0/
      DATA NIDF/256/,IAT/5/,NPA/4/,IEMAX/32/,NFC/2/
      DATA NIDF2/512/,NIEMX/128/,NIDAT/1280/,NIDAT2/2560/
C
      DATA KMAX/30/,JMXDR/100/,JMAX/64/,IAMAX/315/,IR/7/,IE/30 /
     +     IMXJMX/64/,NCL/382/
      DATA NID/128/,NUF/9/,NTDF/64/,NUMAX/29/
      DATA NUV/9/,NVMAX/20/
      DATA IMX/381/,IMN/2/
C
C     TRACK VALUES
C
      DATA VDPT/.001,.00066,.0005,.00033/,NTEST/.TRUE./,NACTI/0
      DATA DELA/0.098/,ICLMX/256/,IFXC/1024*0/
      DATA NKNID/1/,NNIDC/1024*0/,NKDMX/1024/
      DATA TMX/300/,IM/128/,JM/9/,MCDX/1/,ID/1280*0/,IC/1280*0/
      DATA IFCDIR/512*0/,JCL/0/,JFL/0/,CD/1152*0./,D/1152*0./
      DATA NVARM/460/,MNSN/5/,MHSN/1/,HM/6./,FNSRN/.1/,IFXMX/1/
      DATA TM/300/,NCARM/128/,NFARM/256/,JCTN0/128*0/,JCTN0/256*0/
      DATA H1/'TAN','DOP','RAD','RAD','DOP','RAD','TOT','DOP','RAD',
     +        'RAD','DOP','TAN','DOP','TAN','RAD'/
      DATA H2/'SHR','SPD','VEL','SHR','SPD','VEL','SHR','SPD','VEL',
     +        'VEL','SPD','SHR','SPD','SHR','VEL'/
      DATA H3/'MSK','MSK','M/S','MSK','MSK','M/S','MSK','MSK','M/S',
     +        'M/S','MSK','MSK','MSK','MSK','M/S'/
C
      END
```

```
C
C     ********** ***************************************************
C
C     NAME:     [illegible]
C     [illegible]
C
C     [illegible] INPUT PARAMETERS BEFORE PROGRAM [illegible]
C
C       [illegible]
C       CALLING MODS:    DOPLR80
C       CALLED MODS:     NONE
C       COMMON BLOCKS:
C         DOPLR,CONST,FLGS,GATE,INSUB,KNCTR,MORED,OFFS,PARM,VPARM
C
C     COMMENTS:   MUST BE RECOMPILED AFTER CHANGES
C
C     VERSION:    1.0  DEC/VAX 11/780
C     DATE:       4/30/81
C     DESIGN:     RKCRANE
C     PROGMR:     GBGUSTAFSON
C
C     *****************************************************************
C
      LOGICAL PRINT1,COPLOT,CONTRZ,CEPLOT,CONTRV,PROVER,PRNOIS,
     +        PRCELL,PRSIG,PRFIXC,PRCLUS,PRSCAN,PRHEAD,CALIBO,[illegible]
      INTEGER BDAY,EDAY,BEGINT,ENDT
      INTEGER YEAR,DAY,TIME,RNGDEL,RI
C
      COMMON /GATE/    IAVGR
      COMMON /PARM/    PRINT1,COPLOT,CEPLOT,CONTRZ,CONTRV,CALIBO,
     +                 NUMF,NUMR
      COMMON /FLGS/    PRCELL,PRSIG,PRFIXC,PRCLUS,PRSCAN,PRHEAD,
     +                 PRNOIS,PROVER
      COMMON /INSUB/   BEGINT,ENDT,DELTR,SCON,ICOMP,
     +                 DAZM,BDAY,EDAY
      COMMON /DOPLR/   A(5),D(5),NSM,ISQ(20),ISHR
      COMMON /MORED/   INPRF,SCALE
      COMMON /OFFS/    IZOFF,FZOFF
      COMMON /VPARM/   VX,VY,VXI,VYI,TMKTL,TMKTLL
      COMMON /KNCTR/   LT,ITL(2),KLVL,JNID,JNIDA(2,1024),
     +                 KNIDM(2),KNIDL(1,1024)
      COMMON /CONST/   VMISW(2),DIV,VMAG,VMISWM,ZDIV,ADIV,
     +                 A1,A2,A3,B1,B2,HDIV
C
C      ------------------------------------------------------------
      PARAMETER(T=.TRUE.,F=.FALSE.)
C      ------------------------------------------------------------
C
C     OUTPUT AND PROCESSING SWITCHES  (DEFAULT=FALSE)
C
C      *DEVICE 6
C       PRINT1 ---- CONTOR   CLUSTER OUTPUT, LOWEST EL
C       PRCELL ---- STRAK     NORMAL VOLUME CELL SUMMARY OUTPUT
C       PRSIG ----- STRAK     SUMMARY OUTPUT ON SIG CELLS ONLY
C       PRFIXC ---- FTRAK     NORMAL FIXED CONTOUR SUMMARY OUTPUT
C       PRCLUS ---- CTRAK     NORMAL CLUSTER SUMMARY OUTPUT
C       PRHEAD ---- DOPLR80
C                   STRAK
C                   FTRAK
```

```
C                    CTRAK     HEADING FOR ANY SUMMARY OUTPUT
C      PRNOIS ------ STRAK     INCLUDE CELLS FLAGED AS NOISE IN OUTPUT
C      PROVER ------ CONTOR
C                    PEAKD     ARRAY OVERFLOW MESSAGES
C      PRSCAN ------ ATRAK     SCAN BY SCAN V. CELL UPDATE, UNLABLED
C                    RESOLVE   SCAN BY SCAN CLUSTER UPDATE, UNLABLED
C
C     *CONTOUR SEGMENTS
C      COPLOT ------ CONTOR    FIXED CONTOR PLOT OUTPUT
C                    PEAKD     PEAK CELL PLOT OUTPUT
C      CEPLOT ------ CONTOR    FIXED CONTOR PLOT OUTPUT ON LOW EL ONLY
C                              NOTE -- IF CEPLOT=T, COPLOT=F
C
C     *PEAKD PROCESSING
C      CONTRZ ------ CONTOR    CALL PEAKD ON REFL DATA
C      CONTRV ------ CONTOR    CALL PEAKD ON SHEAR DATA
C
C     *CALIBRATED DATA
C      CALIBO ------ EXPAND    OUTPUT CALIBRATED DATA TO DEVICE 2
C
      CONTRV=T
      CONTRZ=T
      CALIBO=F
      PRINT1=F
      PRCELL=T
      PRNOIS=T
      PREIXC=T
      PRCLUS=T
      COPLOT=F
      CEPLOT=T
C
      IF(PRINT1.OR.PRCELL.OR.PRSIG.OR.PREIXC.OR.PRCLUS) PRHEAD=T
      IF(CEPLOT) COPLOT=F
C
C    INPUT SMOOTHING CONSTANT ON REFLECTIVITY (IAVGR)
C
      IAVGR=2
C
C    SELECT TIME INTERVAL TO BE PROCESSED
C
C          BDAY         BEGIN DAY DESIRED
C          BEGINT       BEGIN TIME DESIRED (HHMMSS)
C          EDAY         END DAY DESIRED
C          ENDT         END TIME DESIRED (HHMMSS)
C
      BDAY=0
      BEGINT=102700
      EDAY=169
      ENDT=103300
C
C
C    FIXED THRESHOLD CONTOURING LEVELS
C
C           LT     NUMBER OF CONTOUR LEVELS
C           ITL    THRESHOLD LEVELS IN DBZ (ASCENDING ORDER)
C           KLVL   THRESHOLD LEVEL FOR CELL DETECTION
C
      LT=2
      ITL(1)=20
      ITL(2)=40
```

```
      LEVEL 1
C
C     STEERING LEVEL WINDS - MAGNITUDE AND DIR.
C           VD  DIRECTION FROM (DEG)
C           VM  SPEED (M/S)
C
      VD=250.0
      VW=17.0
      VW=.001*VW
      VD=.0174533*VD
      VX=-VW*SIN(VD)
      VY=-VW*COS(VD)
      VXI=VX
      VYI=VY
C
C     SET RADIAL VELOCITY SMOOTHING CONSTANT
C
      NSM=5
C
C     SET WHICH VELOCITY MEASURE TO PROCESS
C        ISHR = 1 - TANGENTIAL SHEAR
C               2 - RADIAL SHEAR
C               3 - VECTOR SHEAR
C               4 - RADIAL VELOCITY (ABS)
C               5 - DOPPLER SPREAD
C
      ISHR=1
C
C     DEFINE VOLUME CELL ASSOCIATION WEIGHTS
C
      DIV= 0.20
      ZDIV=0.10
      HDIV=0.08
      ADIV=0.04
      VMISW(1)=6.0
      VMISW(2)=11.0
C
C     DEFINE VOLUME CELL TRACKING WEIGHTS
C
      A1=.6
      A2=.4
      A3=.0
      B1=.7
      B2=.3
C
      RETURN
      END
```

```
      SUBROUTINE  INIT
C
C     ****************************************************************
C
C     NAME :     INIT
C     PROJECT: FRT B035 629 (FAA)
C
C     PURPOSE: TO INITIALISE VARIOUS PARAMETERS, WEIGHTS
C              AND COUNTERS BEFORE DATA PROCESSING BEGINS
C
C     INTERFACES:
C        CALLING MOD.   DOPLR80
C        CALLED MODS.   NONE
C        COMMON BLOCKS
C          CLST,CNT,DATA5,DATA1,DATA2,DATA3,DOPLR,ECONST,
C          FIXED,INSUB,INTL,KNCTR,NVLIS,NVLIT,OFFS,ORCOM,
C          PNTRS,QUANTX,RADAR,VPARM,ZLOOK
C
C     COMMENTS:  REQUIRES OUTPUT FROM INPARM
C
C     VERSION:   1.0  DEC/VAX 11-780
C     DATE:      4/30/81
C     DESIGN:    RKCRANE
C     PROGMR:    GBGUSTAFSON
C
C     ****************************************************************
C
      INTEGER TL,BEGINT,ENDT,BDAY,EDAY
      DIMENSION IVCL(53,460)
C
      COMMON /QUANTX/ VQUANT
      COMMON /FIXED/  NFC,TL(2),IC(256),IB(256),NPA,IEMAX,ICVNT(2),
     +                IBVNT(2),ATR(2560),IAT,NIDF,KDD(2),IDSLOT(512)
      COMMON /INSUB/  BEGINT,ENDT,DELR,SCON,ICOMP,
     +                DAZM,BDAY,EDAY
      COMMON /KNCTR/  LT,ITL(2),KLVL,JNID,JNIDA(2,1024),
     +                KNIDM(2),KNIDL(1,1024)
      COMMON /ORCOM/  COMLAT,COMLONG,COMHGT,COMRAD(2)
      COMMON /RADAR/  HMF
      COMMON /CLST/   ICLN,ICLIST(256),ICLMX
      COMMON /DATA1/  ECL(10,128),NCO,NCMX,NRJC
      COMMON /DATA5/  ESCL(10,128),NSCO,NSCMX,NSRJC
      COMMON /DATA2/  VCL(53,460)
      COMMON /DATA3/  IVR(6,460)
      COMMON /ECONST/ EARTH,TSDIV,ZNDRS
      COMMON /CNT/    COSPHI,SINEL,COSPI2,ZMIN,ELAST,SPRM,IFXMX
      COMMON /PNTRS/  NVMIN,NVMAX,IELSN,NSCAN,IESNL,NVSCN,NT
      COMMON /NVLIS/  NVARM,NCARM,NVO,NFO,ICO,IO,JO,JYR,LBL,KTL
      COMMON /NVLIT/  KTLL,NKNID,NKDO,IZTH,NKDMX,ITHR,IFXC(1024),HTST
      COMMON /INTL/   MHSN,MNSN,HM,FNSN,FNSRN,NCLN,NFLN,MZSN,NNMIN,FCAZ
      COMMON /VPARM/  VX,VY,VXI,VYI,TMKTL,TMKTLL
      COMMON /ZLOOK/  JZOFF,ZARY(91),RRATE(91)
      COMMON /OFFS/   IZOFF,FZOFF
      COMMON /DOPLR/  A(5),D(5),NSM,TSQ(20),ISHR
C
      PARAMETER(SUM=3.73206)
      PARAMETER(ACON=.6/SUM,DCON=.4/SUM)
C
      EQUIVALENCE(VCL(1,1),IVCL(1,1))
C
```

```
      JYR=80
C
C    INITIALISE COUNTERS
C
      NT=0
      NRJC=0
      NSRJC=0
      NCLN=0
      NFLN=0
      ICLN=0
      NVO=0
      ICO=0
      IO=0
      JO=0
      ELAST=0.
      IESNL=0
      IELSN=0
      EARTH=1.21*2.*6371.3
      ICLN=1
C
C    SET TEST LIMITS
C
      TSDIV=0.05
      VMAG=.01
      SPRM=2.
      HTST=0.4
      MZSN=40
      NNMIN=5
C
C    COMPUTE MIN WEIGHT (AREA*REFL) FOR A SIG CONTOUR
C
      FCA=125.
      IFCZ=20
      FCAZ=FCA*IFCZ*.001
C
C    SET HEIGHT LEVEL FOR % DETECTED
C
      HMP=COMHGT+HM
      RSN=30.
      ZNOISE=20.
      ZNDRS=ZNOISE-20*ALOG10(RSN)
C
C    INITIALISE ATTRIBUTE STORAGE ARRAYS
C
      DO 40 J=1,ICLMX
   40 ICLIST(J)=0
      DO 20 JX=1,NVARM
      DO 10 MX=1,53
   10 IVCL(MX,JX)=0
      VCL(47,JX)=VX
      VCL(48,JX)=VY
      DO 11 KX=1,6
   11 IVR(KX,JX)=0
   20 CONTINUE
C
C    DEFINE REFLECTIVITY OFFSETS
C
      NFC=LT
      ICOMP=2
C
```

```
      IZOFF=IZOFF
      JZOFF=ITL(I)-1
      DO 25 I=1,NEC
   25 ITL(I)=ITL(I)+IZOFF
C
C   CONSTRUCT LINEAR Z AND RAIN RATE TABLES
C   DEL LAWS & PARSONS Z/R RELATIONSHIP     Z = AA * R**BB
C
      AA=400.
      BB=1.4
      AR=ALOG10(AA)/BB
      BR=0.1/BB
      DO 30 IX=1,91
      ZN=IX+JZOFF
      ZARY(IX)=10.**(ZN/10.)
      RRATE(IX)=10.**(BR*ZN-AR)
   30 CONTINUE
C
C   CONSTRUCT SHEAR QUANTIZATION TABLE IN 1/VQUANT DB STEPS
C   QUANTISE SQRT OF TSQ FOR DOPPLER SPREAD(ISHR=5)
C
      DVQ=1./VQUANT
      IF(ISHR.EQ.5) DVQ=DVQ*2.
      DO 50 N=1,20
   50 TSQ(N)=10.**(FLOAT(N)*.1*DVQ)
C
C   DEFINE RADIAL VELOCITY SMOOTHING WEIGHTS
C
      A(1)=.5*ACON
      A(2)=.86603*ACON
      A(3)=ACON
      A(4)=A(2)
      A(5)=A(1)
      D(1)=.5*DCON
      D(2)=.86603*DCON
      D(3)=DCON
      D(4)=D(2)
      D(5)=D(1)
C
C
      RETURN
      END
```

```
      SUBROUTINE EXPAND(ICODE)
C
C*****************************************************************
C
C   NAME:       EXPAND
C   PROJECT:    ERT 8020-820 (FAA)
C   PROJECT:    ERT 4577 600 (FAA)
C
C   PURPOSE:    READ UNPACKED RADAR DATA
C
C   INTERFACES:
C      CALLING MOD.   DOPLR80
C      CALLED MODS.   NONE
C      INPUT PARM.
C        1) ICODE - DIRECTS PROGRAM CONTROL IN EXPAND
C              1   NORMAL PROCESSING
C             10   FIRST PASS, INITIALISE CONSTANTS
C      OUTPUT PARM.
C        1) ICODE - DIRECTS PROGRAM CONTROL IN DOPLR80
C              0   END OF DATA FILE, END PROCESSING
C              2   NEW PRF, END CURRENT VOLUME SCAN
C      COMMON BLOCKS
C        AZENDS,AZM,DECODE,DOPLR,FILTER,FLGS,GATE,IBAR,INSUB,
C        KNCTR,NEWCO,ORCOM,OUTPAR,PWORK,RADCOM,TCON,UNUSE
C
C   COMMENTS:   READS DATA THAT IS OUTPUT BY NORMAL
C               UNPACKING AND CALIBRATING ROUTINES
C               (CIMARRON OR NORMAN)
C
C   VERSION:    1.0   DEC/VAX 11-780
C   DATE:       5/6/81
C   DESIGN:     RKCRANE
C   PROGMR:     GBGUSTAFSON
C
C*****************************************************************
C
      LOGICAL PRCELL,PRSIG,PRFIXC,PRCLUS,PRSCAN,PRHEAD,PRNOIS
      INTEGER YEAR,DAY,TIME,ITILT,IAZI
      INTEGER TRPOW,T,TS,TI,HV
      INTEGER B,C,BEGINT,ENDT,BDAY,EDAY
      CHARACTER*8 RADAR(2)
      DIMENSION HINTG(2,2)
C
      COMMON /INSUB/   BEGINT,ENDT,DELR,SCON,ICOMP,
     +                 DAZM,BDAY,EDAY
      COMMON /KNCTR/   LT,ITL(2),KLVL,JNTD,JNIDA(2,1024),
     +                 KNIDM(2),KNIDL(1,1024)
      COMMON /RADCOM/  YEAR,DAY,TIME,ITILT,IAZI,RDKM,RIKM,
     +                 BEAWI,PWRBAR,RAWBAR
      COMMON /AZM/     AZMUTH,DAZES,AZLAST,NA,ELEVAT,B,C
      COMMON /NEWCO/   ZI(380),VI(380),SI(380),
     +                 VMX,VMD,NRG,NREC,NRGR
      COMMON /GATE/    IAVGR
      COMMON /DOPLR/   A(5),D(5),NSM,TSQ(20),ISHR
      COMMON /PWORK/   KMAX,T(100),JMXDB,JMAX,IAMAX,IR,JR,
     +                 IMXJMX,NCL,NID,NIDP,IMX,IMN
      COMMON /TCON/    SETRI,RNGDLY,SETEL,RANG(380),IHGT(380)
      COMMON /ORCOM/   COMLAT,COMLONG,COMHGT,COMRAD(2)
      COMMON /DECODE/  UF(9),HEIGHT,DLONG,DLAT
      COMMON /FILTER/  TATRMN,AREAMN,CELMN(2),SVMX
```

```
      COMMON /FLGS/    PRCELL,PRSIG,PREIXC,PRCLUS,PRSCAN,PRHEAD,PRNOIS
      COMMON /OUTPAR/  MOUT,NOUT,NOCTR,NOTCH(30)
      COMMON /IBAR/    SUMN,NUM
      COMMON /AZENDS/  AZLO,AZHI,AZREF,ELOW,ELAVE,IRADAR
C
      DATA IFLAG/1/
      DATA HINTG/'  NO','RMAL','EXPA','NDED'/
      DATA RADAR/' NORMAN ','CIMARRON'/
      PARAMETER(IEOF=0,NORMAL=1,NEWPRF=2,INITAL=10)
C
      IF(ICODE.EQ.NORMAL .OR. IFLAG.EQ.1) GO TO 6
C
C   OUTPUT HEADER INFORMATION ON CHANGE OF PRF OR INTEGRATOR
C
      IF(.NOT.PRHEAD) GO TO 6
      WRITE(6,170) RADAR(IRC),HINTG(1,INT),HINTG(2,INT),DAY,YEAR,TIME,
     +             PRF,WAVECM,FRQ,TRSPWR,CZER1,BEAWI,SV,VMX,PULSE
      WRITE(6,171) NRG,NRGR,RNGDLY,SETR1,FLDR,IGFOLD,
     +             IFOLD
C
C   INPUT PREPARED HEADER RECORD
C
    6 READ(2,END=10) TIME,DAY,YEAR,IAZI,ITILT,RLAT,RLONG,HEIGHT,DLAT,
     +          DLONG,IRADAR,BEAWI,IPRT,INT,FRQ,WAVEL,PRF,
     +          PWRBAR,RAWBAR,PULSE,AVGPWR,WAVECM,TRSPWR,
     +          CZERO,NRG,NCL,IMX,NRGR,NOCTR,FLDR,NOTCH,
     +          IGFOLD,IFOLD,RDKM,RIKM,SETRI,RNGDLY,
     +          VMX,VMD,SV,SVMX,SUMN,NUM,NPEC,ICODE,
     +          ZI,VI,SI
C
C   IF CHANGE IN PRF OR INTEGRATOR, INPUT RANGE GATE POSN
C
      IF(IFLAG.EQ.1) GO TO 65
      IF(ICODE.EQ.NEWPRF) GO TO 101
      RETURN
C
C    *END OF RECORD
C
   10 CONTINUE
      ICODE=IEOF
      RETURN
C
C   OUTPUT HEADER INFORMATION
C
   65 CONTINUE
      IF(.NOT.PRHEAD) RETURN
C
      IF(IABS(IRADAR).EQ.13) IRC=1
      IF(IABS(IRADAR).EQ.21) IRC=2
      WRITE(6,170) RADAR(IRC),HINTG(1,INT),HINTG(2,INT),DAY,YEAR,TIME,
     +             PRF,WAVECM,FRQ,TRSPWR,CZERO,BEAWI,SV,VMX,PULSE
  170 FORMAT('1'///7X,'* ',A8,' *',
     +       5X,2A4,' INTEGRATOR'//3X,'DAY',I4,' 19',I2,
     +       3X,'- -',3X,I6.6,' CST'//5X,'PRF ---------- ',F8.2,
     +       ' (/S)'/5X,'WAVE LNGTH - ',F8.2,' (CM)'/5X,
     +       'FREQUENCY -- ',F8.2,' (MHZ)'/5X,'TRANS PWR --- ',
     +       F8.2,' (DBM)'/5X,'NOISE LVL -- ',F8.2,' (DBM)'/5X,
     +       'BEAM WIDTH - ',F8.2,' (DEG)'/5X,'VEL RESOLN -- ',F8.2,
     +       ' (M/S)'/5X,'MAX VEL ----- ',F8.2,' (M/S)'/5X,'SAMPLES'
     +       ' ------ ',F8.2,' (/GATE)')
```

```
      WRITE(6,1'1) NRG,NRGR,RNGDLY,SETRL,FLDR,IOFOLD,
     +                 IFOLD
  1'1 FORMAT(5X,'GATES ---- = ',I5/5X,
     +        'ELEMENTS = ',I5/5X,'RNG DELAY --- ',F8.2,' (M)'/5X,
     +        'RNG INCR  = ',F8.2,' (M)'/5X,'FOLD RNG --- ',F8.2,
     +        ' (KM)'/5X,'FOLD GATE -- ',I5/5X,'NUM FOLDS = ',I5)
C
      IF(BDAY.GT.0) WRITE(6,172) BDAY,BEGINT,EDAY,ENDT
  172 FORMAT(5X,'BEGIN DAY -- ',I5/5X,'BEGIN TIME = ',I8.6,' (CST)'/
     +       5X,'END DAY --- ',I5/5X,'END TIME --- ',I8.6,' (CST)')
      WRITE(6,174)
  174 FORMAT(5X,'CONTOUR')
      DO 175 K=1,LT
  175 WRITE(6,173) K,LIL(K)
  173 FORMAT(6X,'LEVEL(',I1,') -- ',I5,4X,'(DBZ)')
      WRITE(6,169) COMRAD(1),COMRAD(2),COMLAT,COMLONG,
     +             RADAR(IRC),RLAT,RLONG,DLAT,DLONG
  169 FORMAT(////2X,'COMMON ORIGIN = ',2A4,' RADAR SITE'/
     +       18X,F7.4,' N.LAT',F9.4,' W.LONG'//
     +        2X,'MEASUREMENTS ---- ',A8,' RADAR SITE'/
     +       18X,F7.4,' N.LAT',F9.4,' W.LONG'/
     +        9X,'OFFSET',F10.4,' KM N',F10.4,' KM E')
      IFLAG=0
C
C   COMPUTE GATE RANGES ON INITIAL SCAN AND CHANGE IN PRF
C
  101 CONTINUE
      DO 111 I=1,NRGR
  111 RANG(I)=RDKM+RIKM*(I-1)
      RETURN
      END
```

```
      SUBROUTINE EXPAND(ICODE)
C
C  ***************************************************************************
C
C   NAME:      EXPAND
C   PROJECT:   ERT 8035-620 (FAA)
C
C   PURPOSE:   READ CIMARRON INTEGRATOR DATA TAPE
C              AND CALIBRATE REFLECTIVITY AND DOPPLER
C              PARAMETERS
C
C   INTERFACES:
C      CALLING MOD.   DOPLR80
C      CALLED MODS.   NONE
C      INPUT PARM.
C        1) ICODE - DIRECTS PROGRAM CONTROL IN EXPAND
C              1  NORMAL PROCESSING
C             10  FIRST PASS, INITIALISE CONSTANTS
C      OUTPUT PARM.
C        1) ICODE - DIRECTS PROGRAM CONTROL IN DOPLR80
C              0  END OF DATA FILE, END PROCESSING
C              2  NEW PRF, END CURRENT VOLUME SCAN
C      COMMON BLOCKS
C        AZENDS,AZM,DECODE,DOPLR,FILTER,FLGS,GATE,IBAR,INSUB,
C        KNCTR,NEWCO,ORCOM,OUTPAR,PWORK,RADCOM,TCON,UNUSE
C
C   COMMENTS:  READS CIMARRON DATA, EXPANDED INTEGRATOR,
C              AVERAGES REFL DATA BY IAVGR,
C              AND UNFOLDS DOPPLER DATA IN RANGE.
C
C   VERSION:   1.0  DEC/VAX 11-780
C   DATE:      5/6/81
C   DESIGN:    RKCRANE
C   PROGMR:    GBGUSTAFSON
C
C  ***************************************************************************
C
      LOGICAL PRCELL,PRSIG,PRFIXC,PRCLUS,PRSCAN,PRHEAD,PRNOIS
      REAL*8 SUM
      CHARACTER*2 INPUT(4)
      INTEGER YEAR,DAY,TIME,ITILT,IAZI
      INTEGER TRPOW,NBUF(64),NDAT(762,3),T,TS,TI,HV
      INTEGER B,C,BEGINT,ENDT,BDAY,EDAY
      DIMENSION CAL(64,2),RVAL(380),PWRI(380),PRT(4),IGFOLD(4),
     +          RDEL(4,2),HINTG(2,2),ISN(2),INORM(2)
C
      COMMON /RADCOM/ YEAR,DAY,TIME,ITILT,IAZI,RDKM,RIKM,
     +                BEAWI,PWRBAR,RAWBAR
      COMMON /AZM/    AZMUTH,DAZES,AZLAST,NA,ELEVAT,B,C
      COMMON /NEWCO/  ZI(380),VI(380),SI(380),
     +                VMX,VMD,NRG,NREC,NRGR
      COMMON /GATE/   IAVGR
      COMMON /INSUB/  BEGINT,ENDT,DELR,SCON,ICOMP,
     +                DAZM,BDAY,EDAY
      COMMON /DOPLR/  A(5),D(5),NSM,TSQ(20),ISHR
      COMMON /PWORK/  KMAX,T(100),JMXDB,JMAX,IAMAX,IR,JR,
     +                IMXJMX,NCL,NID,NIDP,IMX,IMN
      COMMON /TCON/   SETRI,RNGDLY,SETEL,RANG(380),IHGT(380)
      COMMON /ORCOM/  COMLAT,COMLONG,COMHGT,COMRAD(2)
      COMMON /DECODE/ UP(9),HEIGHT,DLONG,DLAT
```

```
      COMMON /[illegible]  TAIRMN,AREAMN,CELMN(2),SVMX
      COMMON /[illegible]  PRCELL,PRSTG,PRTEXC,PRCLUS,PRSCAN,PRHEAD,PRNOIS
      COMMON /IBUF/    SUMN,NUM
      COMMON /UNUSE/   NPWC,NSLR
      COMMON /FUNCTN/  IT,ITL(2),KLVL,JNID,JNIDA(2,1024),
     +                 KNIDM(2),KNIDL(1,1024)
      COMMON /OUTPAR/ MOUT,NOUT,NOCTR,NOTCH(30)
      COMMON /AZENDS/ AZLO,AZHI,AZREF,ELOW,ELAVE,IRADAR
C
      BYTE  MBUF(2368)
C
      PARAMETER(IEGR=21,ITIMS=23,IVOFF=2,ISOFF=3)
      PARAMETER(CV2=0.14989625,CV=299.9725)
      PARAMETER(REARTH=111.2,RPD=.017453,PI2=9.8696,AMPK=1000.)
      PARAMETER(NRGM=380,INREF=64)
      PARAMETER(IEOF=0,NORMAL=1,NEWPRF=2,INITAL=10)
      DATA INOT/3/,IPRTO/-1/,INTO/-1/,PRT/768.,922.,1075.,1229./
      DATA RDEL/-1.010,-1.130,-1.100,-1.250,-1.300,-1.480,-1.730,-2.450/
      DATA HINTG/'  NO','RMAL','EXPA','NDED'/,NREC/0/
      DATA ISN/1,-1/,INORM/1,4/
      DATA INPUT/'F1','F2','F3','F4'/,IDEV/0/,IERR/0/
C
      IFLAG=0
      IF(ICODE.EQ.INITAL) GO TO 11
      IF(ICODE.EQ.NORMAL) GO TO 6
C
C    OUTPUT HEADER INFORMATION ON CHANGE OF PRF OR INTEGRATOR
C
      IF(.NOT.PRHEAD) GO TO 7
      WRITE(6,170) HINTG(1,INT),HINTG(2,INT),DAY,YEAR,TIME,PRF,WAVECM,
     +             FRQ,TRSPWR,CZER1,BEAWI,SV,VMX,PULSE
      WRITE(6,171) NRG,NRGR,RNGDLY,SETRI,FLDR,IGFOLD(1),
     +             IFOLD
    7 ICODE=NORMAL
    6 CONTINUE
C
C    AVERAGE LAST 20 UNNOTCHED GATES TO COMPUTE AVG PWR
C
  105 DO 106 I=NGB20,NGBL
      N=NDAT(I,1)
      IF(CAL(N,1).GE.OFF) GO TO 106
      SUMN=SUMN+N
      NUM=NUM+1
  106 CONTINUE
C
C     *INPUT TAPE RECORD
C
    5 READ(1,END=10,ERR=20) MBUF
      DO 25 I=1,INREF
   25 NBUF(I)=MBUF(I)
      IF(NBUF(31).GE.32) GO TO 55
      DO 30 I=1,762
      IREF=INREF+(I-1)*3
      DO 30 J=1,3
      IJ=J+IREF
   30 NDAT(I,J)=MBUF(IJ)
C    1 FORMAT(23(100A1),68A1)
      NREC=NREC+1
      GO TO 60
C
```

```
C     *END OF RECORD
C
   10 CONTINUE
      CLOSE(1)
   11 IDEV=IDEV+1
      OPEN(UNIT=1, FILE=INPUT(IDEV), ERR=12,
     +     FORM='UNFORMATTED', STATUS='OLD')
      GO TO 5
C
   12 ICODE=IEOF
      RETURN
C
C     *PARITY ERROR
C
   20 CONTINUE
      WRITE(6,10000)NREC
10000 FORMAT(1H0,' PARITY ERROR, RECORD ',I10)
      IERR=IERR+1
      IF(IERR.GT.100) GO TO 12
      GO TO 5
C
C     *WIDE PULSE WIDTH
C
   55 NPWC=NPWC+1
      GO TO 5
C
C    TEST FOR INTEGRATOR (NBUF(21)=21)
C          OR TIME SERIES (NBUF(21)=23)
C
   60 CONTINUE
      IF(NBUF(21).EQ.ITEGR) GO TO 90
      IF(NBUF(21).EQ.ITMS) NSER=NSER+1
      GO TO 5
C
C     *NORMAL PROCESSING
C
   90 CONTINUE
C
C    DEFINE TIME AND TEST FOR CORRECT DATA TIME INTERVAL
C
      TIME=((((NBUF(4)*10+NBUF(5))*10+NBUF(6))*10+NBUF(7))
     +                  *10+NBUF(8))*10+NBUF(9)
      DAY=(10*NBUF(1)+NBUF(2))*10+NBUF(3)
      IF(DAY.LT.BDAY.OR.(DAY.EQ.BDAY.AND.TIME.LT.BEGINT)) GO TO 5
C
C
C    INITIALISE CALIBRATION CONSTANTS  ****CIMARRON****
C
      IF(ICODE.LT.INITAL) GO TO 100
      RLAT=35.47533
      RLONG=97.81314
      HEIGHT=0.39939
      DLAT=(RLAT-COMLAT)*REARTH
      DLONG=(COMLONG-RLONG)*REARTH*COS(COMLAT*RPD)
C
      YEAR=80
      NRG=762
      INTO=JMOD(NBUF(31),32)
      INTO=INTO/16+1
C
```

```
C    DEFINE PRT MULTIPLIER FOR RANGE INCREMENT CALC.
C
       SPDPRT=CV2/PRT(1)
C
       BEAWI=0.81
       BEAWI2=BEAWI*RPD
       BEAWI2=1./(BEAWI2*BEAWI2)
       CZERO=102.76
       TRSPWR=10.*ALOG10(750.)
       OFFSET=CZERO-TRSPWR
C
C    INPUT RECEIVED POWER CALIBRATION TABLE
C
       REWIND 4
       READ(4,4) (CAL(I,1),I=1,64)
     4 FORMAT(8F6.1)
       DO 50 I=1,64
    50 CAL(I,1)=CAL(I,1)+OFFSET
C
C
   100 CONTINUE
C
C    DEFINE BEAM ORIENTATION (TILT AND AZIMUTH)
C
       IAZI=((10*NBUF(10)+NBUF(11))*10+NBUF(12))*10+NBUF(13)
       ITILT=10*NBUF(15)+NBUF(16)
       IF(NBUF(14).EQ.45) ITILT=-ITILT
       IF(NBUF(14).NE.45) ITILT=100*NBUF(14)+ITILT
C
C    IF NEW PRF OR NEW INTEGRATOR, FORCE NEW VOLUME SCAN (ICODE=2)
C
       IPRT=JMOD(NBUF(31),16)
       IPRT=IPRT/4+1
C
C    INT=1, NORMAL INTEGRATOR
C    INT=2, EXPANDED INTEGRATOR
C
       INT=JMOD(NBUF(31),32)
       INT=INT/16+1
       IRADAR=NBUF(21)*ISN(INTO)
       IF(IPRT.EQ.IPRTO.AND.INT.EQ.INTO) GO TO 101
       IPRTO=IPRT
       INTO=INT
       IF(ICODE.EQ.NORMAL) ICODE=NEWPRF
       FRQ=2700+10*NBUF(37)+NBUF(38)
       WAVEL=CV/FRQ
       PRF=1.E6/PRT(IPRT)
C
C    DEFINE SAMPLING REGION
C
       IFOLD=INORM(INT)
       IAVGD=IAVGR*IFOLD
       DAVGR=1./IAVGR
       DIV=1./IAVGD
       DIVS=1./(IAVGD*32.*32.)
       NRGR=NRG*DAVGR
       IF(NRGR.GT.NRGM) NRGR=NRGM
       NGBL=NRGR-1
       NGB20=NGBL-20
       NCL=NRGR+2
```

```
      IMX=NRGR+1
C
C    RANGE DELAY, FN OF PRT AND INTEGRATOR TYPE
C
      RDKM=RDEL(IPRT,INT)
      RNGDLY=RDKM*AMPK
C
C    RANGE INCREMENT =(IAVGD SEC E-6) * (PRT/BASEPRT) * (2*C)
C
      RIKM=IAVGD*PRT(IPRT)*STDPRT
      SETRI=RIKM*AMPK
      DRIKM=1./RIKM
C
C    COMPUTE RANGE INTERVAL
C
      DMIN=RIKM*.5
      NMIN=1
      DO 111 I=1,NRGR
      RANG(I)=RDKM+RIKM*(I-1)
      IF(RANG(I).GT.DMIN) GO TO 110
      NMIN=I+1
      GO TO 111
  110 RVAL(I)=20.*ALOG10(RANG(I))
  111 CONTINUE
C
C    EXPANDED INTEGRATOR, DETERMINE POSITION OF FOLDING OFFSETS
C
      NOCTR=NMIN+1
      DO 18 I=1,NOCTR
   18 NOTCH(I)=I
      IF(IFOLD.EQ.1) GO TO 17
      INOT=NMIN+2
      FLDR=PRT(IPRT)*CV2
      NRGD=FLDR*DRIKM+.5
      DO 15 N=1,IFOLD
      IR=NRGD*N-1
C
C    ELIMINATE GATES IMMEDIATELY AROUND FOLDING OFFSET FROM PROCESSING
C
      DO 14 I=1,INOT
      NOCTR=NOCTR+1
   14 NOTCH(NOCTR)=IR+I
   15 IGFOLD(N)=IR
      GO TO 16
C
C    NORMAL INTEGRATOR, NO FOLDING
C
   17 IGFOLD(1)=NRGR
      NOCTR=NOCTR+1
      NOTCH(NOCTR)=NRGR
      NRGD=NRGR
C
   16 CONTINUE
      IF(NOTCH(NOCTR).LE.NRGR) GO TO 13
      NOCTR=NOCTR-1
      GO TO 16
C
C    VELOCITY RESOLUTION (SV) = WAVEL/4. * PRF * (1./31.)
C
   13 VMX=WAVEL/4. * PRF
```

```
      VMID=.*VMX
      SV=VMX/3.
C
C   SPREAD RESOLUTION (SV2) = VMX*VMX*2/(PI*PI)
C
      SV2=VMX*VMX*2./PI2
      SVMX=SQRT(SV2)
      SVMX=.9*SVMX
C
C   PULSES PER RESOLUTION ELEMENT
C
      PULSE=((10*NBUF(26)+NBUF(27))*10+NBUF(28))*10+NBUF(29)
      ROOTP=1./SQRT(PULSE)
C
C   INITIALISE THRESHOLD LEVELS
C
      AVGPWR=OFFSET-CZERO
      PWRBAR=AVGPWR-OFFSET
      OFF=18.*ROOTP+AVGPWR
      RTHRS=AVGPWR+5.
      VTHRS=AVGPWR+10.
      STHRS=AVGPWR+15.
      VNOISE=VTHRS+5.
      SNOISE=STHRS+5.
C
C   OUTPUT HEADER INFORMATION
C
      IF(ICODE.LT.INITAL) GO TO 101
      IF(.NOT.PRHEAD) GO TO 102
      WAVECM=WAVEL*100.
      CZER1=-CZERO
      WRITE(6,170) HINTG(1,INT),HINTG(2,INT),,DAY,YEAR,TIME,PRF,WAVECM,
     +             FRQ,TRSPWR,CZER1,BEAWI,SV,VMX,PULSE
  170 FORMAT('1'///7X,'* CIMARRON *',
     +       5X,2A4,' INTEGRATOR'//3X,'DAY',I4,' 19',I2,
     +       3X,'- -',3X,I6.6,' CST'//5X,'PRF ------- ',F8.2,
     +       ' (/S)'/5X,'WAVE LNGTH - ',F8.2,' (CM)'/5X,
     +       'FREQUENCY -- ',F8.2,' (MHZ)'/5X,'TRANS PWR -- ',
     +       F8.2,' (DBM)'/5X,'NOISE LVL -- ',F8.2,' (DBM)'/5X,
     +       'BEAM WIDTH - ',F8.2,' (DEG)'/5X,'VEL RESOLN - ',F8.2,
     +       ' (M/S)'/5X,'MAX VEL ---- ',F8.2,' (M/S)'/5X,'SAMPLES'
     +       ' ---- ',F8.2,' (/GATE)')
      WRITE(6,171) NRG,NRGR,RNGDLY,SETRI,FLDR,IGFOLD(1),
     +             IFOLD
  171 FORMAT(5X,'GATES ------ ',I5/5X,
     +       'ELEMENTS --- ',I5/5X,'RNG DELAY -- ',F8.2,' (M)'/5X,
     +       'RNG INCR --- ',F8.2,' (M)'/5X,'FOLD RNG -- ',F8.2,
     +       ' (KM)'/5X,'FOLD GATE -- ',I5/5X,'NUM FOLDS -- ',I5)
      IF(BDAY.GT.0) WRITE(6,172) BDAY,BEGINT,EDAY,ENDT
  172 FORMAT(5X,'BEGIN DAY -- ',I5/5X,'BEGIN TIME - ',I8.6,' (CST)'/
     +       5X,'END DAY ---- ',I5/5X,'END TIME --- ',I8.6,' (CST)')
      WRITE(6,169) COMRAD(1),COMRAD(2),COMLAT,COMLONG,
     +             RLAT,RLONG,DLAT,DLONG
  169 FORMAT(////2X,'COMMON ORIGIN - ',2A4,' RADAR SITE'/
     +      18X,F7.4,' N.LAT',F9.4,' W.LONG'//
     +       2X,'MEASUREMENTS -- CIMARRON RADAR SITE'/
     +      18X,F7.4,' N.LAT',F9.4,' W.LONG'/
     +       9X,'OFFSET',F10.4,' KM N',F10.4,' KM E')
C
      IFLAG=1
```

```
  102 ICODE=NORMAL
  101 CONTINUE
C
C   CALIBRATE
C
C   AVERAGE BY IAVGR, REFLECTIVITY DATA (M=1)
C
C      REFL(DBZ) = RCVD PWR(DBM) + 10LOG(R**2) - TRNS PWR(DBM) + NOISE
C
  120 M=1
      N=1
      NOT=NOTCH(N)
C
C   NOTCH GATES IMMEDIATELY AROUND FOLD RING
C
      DO 130 I=1,NRGR
      PWRI(I)=-999.
      VI(I)=-999.
      ZI(I)=-999.
      SI(I)=0.
      IF(I.EQ.NOT) GO TO 129
C
      IR=(I-1)*IAVGR
      PWR=0.
      DO 135 J=1,IAVGR
      K=J+IR
  135 PWR=PWR+CAL(NDAT(K,M),1)
      PWR=PWR*DAVGR
      IF(PWR.GT.RTHRS) ZI(I)=PWR+RVAL(I)
      PWRI(I)=PWR
      GO TO 130
C
  129 N=N+1
      IF(N.GT.NOCTR) N=NOCTR
      NOT=NOTCH(N)
  130 CONTINUE
C
C   UNFOLD POSITION OF VELOCITY DATA OVER IFOLD RANGE BLOCKS
C
  131 N=2
      M=3
      DO 140 I=NMIN,NRGD
      PWRMX=PWRI(I)
      IMAX=I
      IF(IFOLD.LT.2) GO TO 146
      DO 145 J=1,IFOLD
      K=I+IGFOLD(J)
      IF(K.GT.NRGR) GO TO 146
      PWR=PWRI(K)
      IF(PWR.LE.PWRMX) GO TO 145
      PWRMX=PWR
      IMAX=K
  145 CONTINUE
C
C   SUBJECT VELOCITY DATA TO RAW POWER THRESHOLDING
C
  146 CONTINUE
      IF(PWRMX.LE.VTHRS) GO TO 140
C
C   CALIBRATE & AVERAGE VELOCITY (N=2) AND SPREAD (M=3) DATA
```

```
C
C       RAD VEL(M/S)  = (INT. VALUE - RANGE/2 + .5) * VEL RESOLN
C       DOP SPD(M/S)  = SPD RESOLN**2 (1-(1-VAL**2) * PREC/(PREC-NOISE))
C
        IR=(I-1)*IAVGD
C
C    RAD VEL IS OFFSET FROM INTEGRATOR IN RANGE BY IVOFF
C    DOP SPRD IS OFFSET FROM INTEGRATOR IN RANGE BY ISOFF
C
        IV=IR+IVOFF
        IS=IR+ISOFF
        IF(IV+IAVGD.GT.NRG.OR.IS+IAVGD.GT.NRG) GO TO 140
        V=0.
        S=0.
        DO 150 J=1,IAVGD
        KV=J+IV
        KS=J+IS
        V=V+(NDAT(KV,N)-31.5)*SV
        SPD=NDAT(KS,M)
  150   S=S+SPD*SPD
        SI2=S*DIVS
        ANOISE=10.**((AVGPWR+OFFSET)/10.)
        RECPWR=10.**((PWRMX+OFFSET)/10.)
        PWRCOEF=RECPWR/(RECPWR-ANOISE)
C
C    SUBJECT 1ST 3 GATES IN EACH BLOCK
C       TO NOISE THRESHOLDING
C
        SIMAX=0.
        IF(I.GT.3) GO TO 160
        IF(PWRMX.GT.SNOISE) GO TO 161
        IF(PWRMX.GT.VNOISE) GO TO 162
        GO TO 140
C
C    COMPUTE DOPPLER SPREAD AND NORMALISE BY 1/2 BEAWI*RANG(KM)
C
  160   CONTINUE
        IF(PWRMX.LE.STHRS) GO TO 162
  161   SIMAX= SV2 * (1.-(1.-SI2)*PWRCOEF) * BEAWI2/(RANG(I)*RANG(I))
        IF(SIMAX.GT.0.) SI(IMAX)=SIMAX
  162   VI(IMAX)=V*DIV
  140   CONTINUE
C
C    OUTPUT PREPARED DATA TO DISK
C
        WRITE(2) TIME,DAY,YEAR,IAZI,ITILT,RLAT,RLONG,HEIGHT,DLAT,
       +         DLONG,IRADAR,BEAWI,IPRT,INT,FRQ,WAVEL,PRF,
       +         PWRBAR,RAWBAR,PULSE,AVGPWR,WAVECM,TRSPWR,
       +         CZER1,NRG,NCL,IMX,NRGR,NOCTR,FLDR,NOTCH,
       +         IGFOLD(1),IFOLD,RDKM,RIKM,SETRI,RNGDLY,
       +         VMX,VMD,SV,SVMX,SUMN,NUM,NREC,ICODE
C
        IF(NEWPRF .OR. IFLAG.EQ.1) WRITE(2) RANG
        WRITE(2) ZI
        WRITE(2) VI
        WRITE(2) SI
C
        RETURN
        END
```

```
      SUBROUTINE EXPAND(ICODE)
C
C  ***************************************************************************
C
C   NAME:      EXPAND
C   PROJECT:   ERT 8035-620 (FAA)
C
C   PURPOSE:   READ NORMAN INTEGRATOR DATA TAPE
C              AND CALIBRATE REFLECTIVITY AND DOPPLER
C              PARAMETERS
C
C   INTERFACES:
C      CALLING MOD.   DOPLR80
C      CALLED MODS.   NONE
C      INPUT PARM.
C        1) ICODE - DIRECTS PROGRAM CONTROL IN EXPAND
C              1   NORMAL PROCESSING
C             10   FIRST PASS, INITIALISE CONSTANTS
C      OUTPUT PARM.
C        1) ICODE - DIRECTS PROGRAM CONTROL IN DOPLR80
C              0   END OF DATA FILE, END PROCESSING
C              2   NEW PRF, END CURRENT VOLUME SCAN
C      COMMON BLOCKS
C        AZENDS,AZM,DECODE,DOPLR,FILTER,FLGS,GATE,IBAR,INSUB,
C        KNCTR,NEWCO,ORCOM,OUTPAR,PWORK,RADCOM,TCON,UNUSE
C
C   COMMENTS:  READS NORMAN DATA, EXPANDED INTEGRATOR,
C              AVERAGES REFL DATA BY IAVGR,
C              AND UNFOLDS DOPPLER DATA IN RANGE.
C
C   VERSION:   1.0  DEC/VAX 11-780
C   DATE:      5/6/81
C   DESIGN:    RKCRANE
C   PROGMR:    GBGUSTAFSON
C
C  ***************************************************************************
C
      LOGICAL PRCELL,PRSIG,PREIXC,PRCLUS,PRSCAN,PRHEAD,PRNOIS,
     +        PRINT1,COPLOT,CEPLOT,CONTRZ,CONTRV,CALIBD,PROVER
      REAL*8 SUM
      CHARACTER*2 INPUT(4)
      INTEGER YEAR,DAY,TIME,ITILT,IAZI
      INTEGER TRPOW,NBUF(46),NDAT(762,3),T,TS,TI,HV
      INTEGER B,C,BEGINT,ENDT,BDAY,EDAY
      DIMENSION CAL(64,2),RVAL(380),PWRI(380),PRT(4),IGFOLD(4),
     +          FLDR(4),RDEL(4,2),HINTG(2,2),TRSPWR(2),
     +          CZERO(2),INORM(2),ISN(2)
C
      COMMON /INSUB/  BEGINT,ENDT,DELR,SCON,ICOMP,
     +                DAZM,BDAY,EDAY
      COMMON /KNCTR/  LT,ITL(2),KLVL,JNID,JNIDA(2,1024),
     +                KNIDM(2),KNIDL(1,1024)
      COMMON /RADCOM/ YEAR,DAY,TIME,ITILT,IAZI,RDKM,RIKM,
     +                BEAWI,PWRBAR,RAWBAR
      COMMON /AZM/    AZMUTH,DAZES,AZLAST,NA,ELEVAT,B,C
      COMMON /NEWCO/  ZI(380),VI(380),SI(380),
     +                VMX,VMD,NRG,NREC,NRGR
      COMMON /GATE/   IAVGR
      COMMON /DOPLR/  A(5),D(5),NSM,TSQ(20),ISHR
      COMMON /PWORK/  KMAX,T(100),JMXDB,JMAX,IAMAX,IR,JR,
```

```
     +                  IMX,JMX,NCL,NTD,NTDP,IMX,IMN
      COMMON /CCOM/     SETRI,RNGDLY,SETEL,RANG(380),IHGT(382)
      COMMON /ORCOM/    COMLAT,COMLONG,COMHGT,COMRAD(2)
      COMMON /DECODE/   UF(9),HEIGHT,DLONG,DLAT
      COMMON /FILTER/   IATEMN,AREAMN,CELMN(2),SVMX
      COMMON /FLGS/     PRCELL,PRSIG,PRFIXC,PRCLUS,PRSCAN,PRHEAD,
     +                  PRNOIS,PROVER
      COMMON /PARM/     PRINT1,COPLOT,CEPLOT,CONTRZ,CONTRV,CALIBO,
     +                  NUMF,NUMR
      COMMON /OUTPAR/   MOUT,NOUT,NOCTR,NOTCH(30)
      COMMON /IRAR/     SUMN,NUM
      COMMON /AZLNDS/   AZLO,AZHI,AZREF,ELOW,ELAVE,IRADAR
      COMMON /UNUSE/    NPWC,NSER
C
      BYTE MBUF(2332)
C
      PARAMETER(LTLGR=13,ITIMS=15,IVOFF=2,ISOFF=2)
      PARAMETER(CV2=0.14989625,CV=299.9725)
      PARAMETER(REARTH=111.2,RPD=.017453,PI2=9.8696,AMPK=1000.)
      PARAMETER(NRGM=380,INREF=46)
      PARAMETER(IEOF=0,NORMAL=1,NEWPRF=2,INITAL=10)
      DATA INOT/3/,IPRTO/-1/,INTO/-1/,PRT/768.,922.,1075.,1229./
      DATA RDEL/-.310,-.190,-.130,-.130,-.310,-.310,-.180,-.160/
      DATA HINTG/'  NO','RMAL','EXPA','NDED'/,NREC/0/
      DATA INORM/1,4/,ISN/1,-1/,NTRIPO/-1/
      DATA INPUT/'F1','F2','F3','F4'/,IDEV/0/,IERR/0/
C
      IFLAG=0
      IF(ICODE.EQ.INITAL) GO TO 11
      IF(ICODE.EQ.NORMAL) GO TO 6
C
C    OUTPUT HEADER INFORMATION ON CHANGE OF PRF OR INTEGRATOR
C
      IF(.NOT.PRHEAD) GO TO 7
      WRITE(6,170) HINTG(1,INT),HINTG(2,INT),DAY,YEAR,TIME,PRF,WAVECM,
     +             FRQ,TRSPWR(INT),CZER1,BEAWI,SV,VMX,PULSE
      WRITE(6,171) NRG,NRGR,RNGDLY,SETRI,FLDR(IPRT),IGFOLD(1),
     +             IFOLD
    7 ICODE=NORMAL
    6 CONTINUE
C
C    AVERAGE LAST 20 UNNOTCHED GATES TO COMPUTE AVG PWR
C
  105 DO 106 I=NGB20,NGBL
      N=NDAT(I,1)
      IF(CAL(N,INT).GE.OFF) GO TO 106
      SUMN=SUMN+N
      NUM=NUM+1
  106 CONTINUE
C
C     *INPUT TAPE RECORD
C
    5 READ(1,END=10,ERR=20) MBUF
      DO 25 I=1,INREF
   25 NBUF(I)=MBUF(I)
      IF(JMOD(NBUF(26),32).GT.16) GO TO 55
      DO 30 J=1,3
      JREF=INREF+(J-1)*762
      DO 30 I=1,762
      IJ=I+JREF
```

```
   30  NDAT(I,J)=MBUF(IJ)
C     1 FORMAT(23(100A1),68A1)
      NREC=NREC+1
      GO TO 60
C
C     *END OF RECORD
C
   10 CONTINUE
      CLOSE(1)
   11 IDEV=IDEV+1
      OPEN(UNIT=1, FILE=INPUT(IDEV), ERR=12,
     +     FORM='UNFORMATTED', STATUS='OLD')
      GO TO 5
C
   12 ICODE=IEOF
      RETURN
C
C     *PARITY ERROR
C
   20 CONTINUE
      WRITE(7,10000)NREC
10000 FORMAT(1HO,' PARITY ERROR, RECORD ',I10)
      IERR=IERR+1
      IF(IERR.GT.100) GO TO 12
      GO TO 5
C
C     *WIDE PULSE WIDTH
C
   55 NPWC=NPWC+1
      GO TO 5
C
C    TEST FOR INTEGRATOR (NBUF(1)=13)
C          OR TIME SERIES (NBUF(1)=15)
C
   60 CONTINUE
      IF(NBUF(1).EQ.ITEGR) GO TO 90
      IF(NBUF(1).EQ.ITIMS) NSER=NSER+1
      GO TO 5
C
C     *NORMAL PROCESSING
C
   90 CONTINUE
C
C    DEFINE TIME AND TEST FOR CORRECT DATA TIME INTERVAL
C
      TIME=(((((NBUF(2)*10+NBUF(3))*10+NBUF(4))*10+NBUF(5))
     +                  *10+NBUF(6))*10+NBUF(7)
      DAY=(10*JMOD(NBUF(23),4)+JMOD(NBUF(24),16))*10+JMOD(NBUF(25),16)
      IF(DAY.LT.BDAY.OR.(DAY.EQ.BDAY.AND.TIME.LT.BEGINT)) GO TO 5
C
C
C    INITIALISE CALIBRATION CONSTANTS  ****NORMAN****
C
      IF(ICODE.LT.INITAL) GO TO 100
      DO 21 I=1,4
   21 FLDR(I)=CV2*PRT(I)
      RLAT=35.23651
      RLONG=97.46333
      HEIGHT=0.3697
      DLAT=(RLAT-COMLAT)*REARTH
```

```
      WRONG=(COMLONG-RLONG)*BEAMTH*COS(COMLAT*RPD)
C
      YEAR=80
      NRG='62'
      ENCODE(3,101,NRG)(2**16)
      INTO IPTO/511
C
C   DEFINE PRT MULTIPLIER FOR RANGE INCREMENT CALC.
C
      STDPRT=CV2-PRT(1)
C
      BEAWI=0.81
      BEAWI2=BEAWI*RPD
      BEAWI2=1./(BEAWI2*BEAWI2)
C
C   INPUT RECEIVED POWER CALIBRATION TABLE
C
      REWIND 4
      DO 40 J=1,2
      READ(4,3) CZERO(J),TPWR
    3 FORMAT(35X,F6.1,F6.0/)
      READ(4,4) (CAL(I,J),I=1,64)
    4 FORMAT(23X,8F6.1)
      TRSPWR(J)=10.*ALOG10(TPWR)
      OFFSET=CZERO(J)+TRSPWR(J)
      DO 50 I=1,64
   50 CAL(I,J)=CAL(I,J)-OFFSET
   40 CONTINUE
C
  100 CONTINUE
C
C   DEFINE BEAM ORIENTATION (TILT AND AZIMUTH)
C
      IAZI=((10*JMOD(NBUF(8),4)+NBUF(9))*10+NBUF(10))*10+NBUF(11)
      ITILT=10*NBUF(13)+NBUF(14)
      IF(NBUF(12).EQ.45) ITILT=-ITILT
      IF(NBUF(12).NE.45) ITILT=100*NBUF(12)+ITILT
C
C   IF NEW PRF OR NEW INTEGRATOR, FORCE NEW VOLUME SCAN (ICODE=2)
C
      IPRT=JMOD(NBUF(27),64)
      IPRT=IPRT/16+1
C
C   INT=1, NORMAL INTEGRATOR
C   INT=2, EXPANDED INTEGRATOR
C
      INT=JMOD(NBUF(27),16)
      INT=INT/8+1
      NTRIP=NBUF(29)/16
      IRADAR=NBUF(1)*ISN(INTO)
      IF(IPRT.NE.IPRTO .OR. INT.NE.INTO) GO TO 75
      IF(NTRIP.NE.NTRIPO .AND. INT.EQ.1) GO TO 70
      GO TO 101
C
C   COMPUTE CALIBRATION CONSTANTS
C
   75 IPRTO=IPRT
      INTO=INT
      IF(ICODE.EQ.NORMAL) ICODE=NEWPRF
      FRQ=2850.
```

```
      WAVEL  CV/FRQ
      PRF  1.E6/PRT(IPRT)
C
C     DEFINE  SAMPLING REGION
C
      IFOLD=INORM(INT)
      IAVGD=IAVGR*IFOLD
      DAVGR=1./IAVGR
      DIV=1./IAVGD
      DIVS=1./(IAVGD*32.*32.)
      NRGR=NRG*DAVGR
      IF(NRGR.GT.NRGM) NRGR=NRGM
      NGBL=NRGR-1
      NGB20=NGBL-20
      NCI=NRGR+2
      IMX=NRGR+1
C
C     RANGE  INCREMENT =(IAVGD SEC E-6) * (PRT/BASEPRT) * (C/2)
C
      RIKM=IAVGD*PRT(IPRT)*STDPRT
      SETRI=RIKM*AMPK
C
      DRIKM=1./RIKM
      DMIN=RIKM*.5
      NMIN=(DMIN - RDEL(IPRT,INT))*DRIKM + .5
      NOCTR=NMIN+1
      DO 18 I=1,NOCTR
   18 NOTCH(I)=1
      IF(INT.EQ.1) GO TO 17
C
C     EXPANDED INTEGRATOR, DETERMINE POSITION OF FOLDING OFFSETS
C     ELIMINATE GATES IMMEDIATELY AROUND FOLDS FROM PROCESSING
C
      NTRIP=0
      INOT=NMIN+2
      NRGD=FLDR(IPRT)*DRIKM+.5
      DO 15 N=1,IFOLD
      IR=NRGD*N-1
      DO 14 I=1,INOT
      NOCTR=NOCTR+1
   14 NOTCH(NOCTR)=IR+I
   15 IGFOLD(N)=IR
      GO TO 16
C
C     NORMAL  INTEGRATOR, NO FOLDING
C
   17 IGFOLD(1)=NRGR
      NOCTR=NOCTR+1
      NOTCH(NOCTR)=NRGR
      NRGD=NRGR
C
   16 CONTINUE
      IF(NOTCH(NOCTR).LE.NRGR) GO TO 13
      NOCTR=NOCTR-1
      GO TO 16
C
C     VELOCITY RESOLUTION (SV) = WAVEL/4. * PRF * (1./31.)
C
   13 VMX=WAVEL/4. * PRF
      VMD=2.*VMX
```

```
      SV=VMX/3.
C
C   SPECTRUM RESOLUTION (SVR)     (VMX*VMX*2)/(PI*PI)
C
      SVR=(VMX*VMX*2.)/(PI*PI)
      SVMX=SQRT(SVR)
      SVMX=.9*SVMX
C
C   PULSES PER RESOLUTION ELEMENT
C
      PULSE=2.0**(NBUF+12)/4.)
      ROOTP=1./SQRT(PULSE)
C
C   INITIALISE THRESHOLD LEVELS
C
      AVGPWR=TRSPWR(INT)
      OFF=10. * ROOTP+AVGPWR
      OFFSET=CZERO(INT)+TRSPWR(INT)
      PWRBAK=AVGPWR+OFFSET
      RTHRS=AVGPWR+5.
      VTHRS=AVGPWR+10.
      STHRS=AVGPWR+15.
      VNOISE=VTHRS+5.
      SNOISE=STHRS+5.
C
C   RANGE DELAY, FN OF PRT AND INTEGRATOR TYPE
C
   70 NTRIPO=NTRIP
      RDKM=RDEL(IPRT,INT)+NTRIP*FLDR(IPRT)
      RNGDLY=RDKM*AMPK
C
C   COMPUTE RANGE INTERVAL
C
      DO 111 I=1,NRGR
      IP=I+1
      RANG(IP)=RDKM+RIKM*(I-1)
      IF(RANG(IP).LE.DMIN) GO TO 111
      RVAL(I)=20.*ALOG10(RANG(IP))
  111 CONTINUE
      RANG(382)=RANG(381)+RIKM
C
C   OUTPUT HEADER INFORMATION
C
      IF(ICODE.LT.INITAL) GO TO 101
      IF(.NOT.PRHEAD) GO TO 102
      WAVECM=WAVEL*100.
      WRITE(6,170) HINTG(1,INT),HINTG(2,INT),DAY,YEAR,TIME,PRF,WAVECM,
     +             FRQ,TRSPWR(INT),CZERO(INT),BEAWI,SV,VMX,PULSE
  170 FORMAT('1'///7X,'* NORMAN *',
     +       5X,2A4,' INTEGRATOR'//3X,'DAY',I4,' 19',I2,
     +       3X,'- -',3X,I6.6,' CST'//5X,'PRF ---------- ',F8.2,
     +       ' (/S)'/5X,'WAVE LNGTH = ',F8.2,' (CM)'/5X,
     +       'FREQUENCY -- ',F8.2,' (MHZ)'/5X,'TRANS PWR -- ',
     +       F8.2,' (DBM)'/5X,'NOISE LVL -- ',F8.2,' (DBM)'/5X,
     +       'BEAM WIDTH = ',F8.2,' (DEG)'/5X,'VEL RESOLN = ',F8.2,
     +       ' (M/S)'/5X,'MAX VEL ---- ',F8.2,' (M/S)'/5X,'SAMPLES'
     +       ' ---- ',F8.2,' (/GATE)')
      WRITE(6,171) NRG,NRGR,RNGDLY,SETTI,FLDR(IPRT),IGFOLD(1),
     +             IFOLD
  171 FORMAT(5X,'GATES ------ ',I5/5X,
```

```
     +        'ELEMENTS = ',I5/5X,'RNG DELAY = ',F8.2,' (M)'/5X,
     +        'RNG INCR = ',F8.2,' (M)'/5X,'FOLD RNG = ',F8.2,
     +        ' (KM)'/5X,'FOLD GATE = ',I5/5X,'NUM FOLDS = ',I5)
      IF(BDAY.GT.0) WRITE(6,172) BDAY,BCGINT,EDAY,LNDT
  172 FORMAT(5X,'BEGIN DAY = ',I5/5X,'BEGIN TIME = ',I8.6,' (CST)'/
     +       5X,'END DAY = ',I5/5X,'END TIME = ',I8.6,' (CST)')
      WRITE(6,174)
  174 FORMAT(5X,'CONTOUR')
      DO 175 K=1,LT
  175 WRITE(6,173) K,ITL(K)
  173 FORMAT(6X,'LEVEL(',I1,') = ',I5,4X,'(DBZ)')
      WRITE(6,169) COMRAD(1),COMRAD(2),COMLAT,COMLONG,
     +             RLAT,RLONG,DLAT,DLONG
  169 FORMAT(////2X,'COMMON ORIGIN = ',2A4,' RADAR SITE'/
     +       18X,F7.4,' N.LAT',F9.4,' W.LONG'//
     +       2X,'MEASUREMENTS = NORMAN RADAR SITE'/
     +       18X,F7.4,' N.LAT',F9.4,' W.LONG'/
     +       9X,'OFFSET',F10.4,' KM N',F10.4,' KM E')
C
      IFLAG=1
  102 ICODE=NORMAL
  101 CONTINUE
C
C   CALIBRATE
C
C   AVERAGE BY IAVGR, REFLECTIVITY DATA (M=1)
C
C      REFL(DBZ) = RCVD PWR(DBM) + 10LOG(R**2) - TRNS PWR(DBM) + NOISE
C
  120 M=1
      N=1
      NOT=NOTCH(N)
C
C   NOTCH GATES IMMEDIATELY AROUND FOLD RING
C
      DO 130 I=1,NRGR
      PWRI(I)=-999.
      VI(I)=-999.
      ZI(I)=-999.
      SI(I)=0.
      IF(I.EQ.NOT) GO TO 129
C
      IR=(I-1)*IAVGR
      PWR=0.
      DO 135 J=1,IAVGR
      K=J+IR
  135 PWR=PWR+CAL(NDAT(K,M),INT)
      PWR=PWR*DAVGR
      IF(PWR.GT.RTHRS) ZI(I)=PWR+RVAL(I)
      PWRI(I)=PWR
      GO TO 130
C
  129 N=N+1
      IF(N.GT.NOCTR) N=NOCTR
      NOT=NOTCH(N)
  130 CONTINUE
C
C   UNFOLD POSITION OF VELOCITY DATA OVER IFOLD RANGE BLOCKS
C
  131 N=2
```

```
      M=3
      DO 140 I=NMIN,NRGD
      IP=I+1
      PWRMX=PWRI(I)
      IMAX=I
      IF(IFOLD.LT.2) GO TO 146
      DO 145 J=1,IFOLD
      K=I+IGFOLD(J)
      IF(K.GT.NRGR) GO TO 146
      PWR=PWRI(K)
      IF(PWR.LE.PWRMX) GO TO 145
      PWRMX=PWR
      IMAX=K
  145 CONTINUE
C
C   SUBJECT VELOCITY DATA TO RAW POWER THRESHOLDING
C
  146 CONTINUE
      IF(PWRMX.LE.VTHRS) GO TO 140
C
C   CALIBRATE & AVERAGE VELOCITY (N=2) AND SPREAD (M=3) DATA
C
C       RAD VEL(M/S) = (INT. VALUE - RANGE/2 + .5) * VEL RESOLN
C       DOP SPD(M/S) = SPD RESOLN**2 (1-(1-VAL**2) * PREC/(PREC-NOISE))
C
      IR=(I-1)*IAVGD
C
C   RAD VEL IS OFFSET FROM INTEGRATOR IN RANGE BY IVOFF
C   DOP SPRD IS OFFSET FROM INTEGRATOR IN RANGE BY ISOFF
C
      IV=IR+IVOFF
      IS=IR+ISOFF
      IF(IV+IAVGD.GT.NRG.OR.IS+IAVGD.GT.NRG) GO TO 140
      V=0.
      S=0.
      DO 150 J=1,IAVGD
      KV=J+IV
      KS=J+IS
      V=V+(NDAT(KV,N)-31.5)*SV
      SPD=NDAT(KS,M)
  150 S=S+SPD*SPD
      SI2=S*DIVS
      ANOISE=10.**((AVGPWR+OFFSET)/10.)
      RECPWR=10.**((PWRMX+OFFSET)/10.)
      PWRCOEF=RECPWR/(RECPWR-ANOISE)
C
C   SUBJECT 1ST 3 GATES IN EACH BLOCK
C     TO NOISE THRESHOLDING
C
      SIMAX=0.
      IF(I.GT.3) GO TO 160
      IF(PWRMX.GT.SNOISE) GO TO 161
      IF(PWRMX.GT.VNOISE) GO TO 162
      GO TO 140
C
C   COMPUTE DOPPLER SPREAD AND NORMALISE BY 1/2 BEAWI*RANG(KM)
C
  160 CONTINUE
      IF(PWRMX.LE.STHRS) GO TO 162
  161 SIMAX= SV2 * (1.-(1.-SI2)*PWRCOEF) * BEAWI2/(RANG(IP)*RANG(IP))
```

```
      IF(SIMAX.GT.0.) SI(IMAX)=SIMAX
  162 VI(IMAX)=V*DIV
  140 CONTINUE
C
C   OUTPUT PREPARED DATA TO DISK
C
      IF(CALIBO) WRITE(2) TIME,DAY,YEAR,IAZI,ITILT,RLAT,RLONG,
     +          HEIGHT,DLAT,DLONG,IRADAR,BEAWI,IPRT,INT,FRQ,WAVEL,
     +          PRF,PWRBAR,RAWBAR,PULSE,AVGPWR,WAVECM,TRSPWR(INT),
     +          CZERO(INT),NRG,NCL,IMX,NRGR,NOCTR,FLDR(IPRT),NOTCH,
     +          IGFOLD(1),IFOLD,RDKM,RIKM,SETRI,RNGDLY,
     +          VMX,VMD,SV,SVMX,SUMN,NUM,NREC,ICODE,
     +          ZI,VI,SI
      RETURN
      END
```

```
      SUBROUTINE DBZCVT
C
C     ****************************************************************
C
C     NAME:     DBZCVT
C     PROJECT: ERT 8035-620 (FAA)
C
C     PURPOSE:  TO CALIBRATE AVERAGED OUTPUT FROM EXPAND,
C               UNFOLD RADIAL VELOCITY IN RANGE AND SET
C               UP REFL, SHEAR, RAD VEL & DOP SPREAD
C               ARRAYS FOR PROCESSING
C
C     INTERFACES:
C        CALLING MOD.   DOPLR80
C        CALLED MODS.   NONE
C        INPUT PARM.    NONE
C        OUTPUT PARM.   NONE
C        COMMON BLOCKS
C          AZM,DOPLR,FILTER,FIXED,INSUB,NEWCO,OFFS,OUTPAR,
C          PWORK,RADCOM,REFL,TCON,VEL
C
C     COMMENTS:  ARRAYS THAT PASS DATA TO THE PROCESSING ROUTINES,
C                COMMON VEL & REFL, HAVE NO DATA IN THE FIRST AND
C                LAST STORAGE LOCATIONS TO INDICATE BEGINNING AND
C                END OF RADIAL
C
C     VERSION:  1.0  DEC/VAX 11-780
C     DATE:     5/6/81
C     DESIGN:   RKCRANE, JHO & GBGUSTAFSON
C     PROGMR:   GBGUSTAFSON
C
C     ****************************************************************
C
      INTEGER YEAR,DAY,TIME,ITILT,IAZI
      INTEGER B,C,OLD
      INTEGER W,WI,HR,TS,TI,HV,WOLD
      INTEGER TL,BEGINT,ENDT,BDAY,EDAY
C
      DIMENSION VEL(380,2),WOLD(380),SHR(5)
C
      COMMON /DOPLR/  A(5),D(5),NSM,TSQ(20),ISHR
      COMMON /RADCOM/ YEAR,DAY,TIME,ITILT,IAZI,RDKM,RIKM,
     +                BEAWI,PWRBAR,RAWBAR
      COMMON /INSUB/  BEGINT,ENDT,DELR,SCON,ICOMP,
     +                DAZM,BDAY,EDAY
      COMMON /FIXED/  NFC,TL(2),IC(256),IB(256),NPA,IEMAX,ICVNT(2),
     +                IBVNT(2),ATR(2560),IAT,NIDF,KDD(2),IDSLOT(512)
      COMMON /FILTER/ TATRMN,AREAMN,CELMN(2),SVMX
      COMMON /OFFS/   IZOFF,FZOFF
      COMMON /TCON/   SETRI,RNGDLY,SETEL,RANG(382),IHGT(382)
      COMMON /VEL/    TS(382),TI(382),HV(382),RV(382),RS(382)
      COMMON /REFL/   W(382),WI(382),HR(382)
      COMMON /PWORK/  KMAX,T(100),JMXDB,JMAX,IAMAX,IR,JR,
     +                IMX,JMX,NCL,NID,NIDP,IMX,IMN
      COMMON /AZM/    AZMUTH,DAZES,AZLAST,NA,ELEVAT,B,C
      COMMON /NEWCO/  ZI(380),VI(380),SI(380),
     +                VMX,VMD,NRG,NREC,NRGR
      COMMON /OUTPAR/ MOUT,NOUT,NOCTR,NOTCH(30)
C
C
```

```
      PARAMETER(RPD=.01745329,RP10D=1.7453E-3,AKPM=.001,AKPKK=1000.)
      DATA TAZ10/0/,NEW/1/,WOLD/380*0/
C
      OLD=NEW
      NEW=NEW+1
      IF(NEW.GT.2) NEW=1
C
C     SET MIN AREA IN KILO-KM
C
      IATRMN=AREAMN*AKPKK/RIKM
      CFI=BEAWI*RPD*RIKM
      CELMN(1)=1.5*CEL
      CELMN(2)=3.5*CEL
      SETEL=ELEVAT
      W(1)=0
      W(NCL)=0
      TS(1)=-999
      TS(NCL)=-999
      RV(1)=0.
      RV(NCL)=0.
      RS(1)=0.
      RS(NCL)=0.
C
C     UNFOLD VELOCITY VALUES TO SAME UNAMBIGUOUS INTERVAL
C
      I=1
    5 IF(VI(I).GT.-999.) GO TO 10
      I=I+1
      IF(I.GT.NRGR) GO TO 25
      GO TO 5
C
   10 VL=VI(I)
C
    8 I=I+1
      IF(I.GT.NRGR) GO TO 25
      IF(VI(I).LT.-990.) GO TO 8
C
C     TEST OUT IN RANGE
C
    4 V=VI(I)
      IF(V-VL.GT.VMX) V=V-VMD
      IF(VL-V.GT.VMX) V=V+VMD
C
C     TEST RELATIVE TO PRIOR RADIAL
C
      IF(VEL(I,OLD).LT.-990..OR.NA.EQ.1) GO TO 7
      VB=VEL(I,OLD)
      IF(V-VB.GT.VMX) V=V-VMD
      IF(VB-V.GT.VMX) V=V+VMD
C
C     TEST AGAINST MEAN
C
    7 J=I
    9 J=J+1
      IF(J.GT.NRGR) GO TO 15
      IF(VI(J).LT.-990.) GO TO 9
      VMEAN=(VL+VI(J))/2.
      IF(V-VMEAN.GT.VMX) V=V-VMD
      IF(VMEAN-V.GT.VMX) V=V+VMD
      VL=V
```

```
      VI(I)=V
      I=J
      GO TO 6
C
   15 VI(I)=V
   25 CONTINUE
C
C    DEFINE:  REFLECTIVITY
C             TANGENTIAL VELOCITY SHEAR
C             RADIAL VELOCITY
C    OVER NRGR RANGE INTERVALS
C
      DO 21 I=1,NRGR
      IP=I+1
      TS(IP)=-999
      VEL(I,NEW)=-999.
      DO 22 N=1,ISHR
   22 SHR(N)=-999.
C
C    OFFSET REFLECTIVITY DATA
C
      W(IP)=ZI(I)+FZOFF
      IF(W(IP).LE.TL(1)) W(IP)=0
C
C    SMOOTH VELOCITY DATA OVER -NSM- VALUES
C
      INM=I-NLOW
      VINEW=0.0
      VOLD=0.0
      J=0
      SUM=0
      DO 40 IN=1,NSM
      N=INM+IN
      IF(N.LT.1.OR.N.GT.NRGR) GO TO 40
      IF(VI(N).LT.-990.) GO TO 40
      J=J+1
      VINEW=VINEW+A(IN)*VI(N)
      SUM=SUM+A(IN)
      IF(VEL(N,OLD).LT.-990.) GO TO 40
      VOLD=VOLD+D(IN)*VEL(N,OLD)
      SUM=SUM+D(IN)
   40 CONTINUE
      IF(J.GT.2) VEL(I,NEW)=(VINEW+VIOLD)/SUM
C
C    TEST FOR BAD DOPPLER VALUES
C
      VNEW=VEL(I,NEW)
      IF(W(IP).LE.TL(1) .OR. VNEW.LT.-990.) GO TO 60
C
C    DEFINE TANGENTIAL SHEAR (M/S/KM)
C
      IF(ISHR.EQ.2) GO TO 45
      IF(NA.EQ.1 .OR. IAZI.EQ.IAZIO .OR.
     +    WOLD(I).LE.TL(1) .OR. VEL(I,OLD).LT.-990.)
     +    GO TO 44
      VSCALE=RANG(I)*(IAZI-IAZIO)*RP1OD
      SHR(1)=ABS((VNEW-VEL(I,OLD))/VSCALE)
C
   44 CONTINUE
      IF(ISHR.GT.3) GO TO 50
```

```
      IF(ISHR.EQ.1) GO TO 60
C
C    DEFINE RADIAL SHEAR (M/S/KM)
C
   45 CONTINUE
      IF(I.GT.1) SHR(2)=ABS(VNEW-VEL(I-1,NEW))/RIKM
C
      IF(ISHR.NE.3) GO TO 60
C
C    DEFINE TOTAL (VECTOR) SHEAR (M/S/KM)
C
      IF(SHR(1).GT.-990..AND.SHR(2).GT.-990.)
     +   SHR(3)=SQRT(SHR(1)*SHR(1)+SHR(2)*SHR(2))
      GO TO 60
C
C    DEFINE RADIAL VELOCITY AS SHEAR VECTOR
C
   50 CONTINUE
      IF(ISHR.NE.4) GO TO 55
      SHR(4)=ABS(VNEW)
      GO TO 60
C
C    DEFINE DOPPLER SPREAD AS SHEAR VECOTR (M/S/KM) - NORMALISED
C
   55 CONTINUE
      IF(ISHR.NE.5) GO TO 59
      SII=SI(I)
      IF(SII.GT.0.) SHR(5)=SII
      SII=0.
      IF(SHR(1).GT.-990.) SII=SHR(1)
      SI(I)=SII
      GO TO 60
C
C    BAD ISHR CODE, DEFAULT TO TANG SHEAR (1)
C
   59 ISHR=1
C
C    QUANTIZE DOPPLER VALUE TO BE PROCESSED INTO (1/VQUANT) DB STEPS
C    IF DOPPLER SPREAD, QUANTISE SQUARE ROOT
C
   60 S=SHR(ISHR)
      IF(S.LT.-990.) GO TO 20
      DO 30 N=1,20
      IF(S.LE.TSQ(N)) GO TO 35
   30 CONTINUE
C
C    PREPARE ALL PROCESSING ARRAYS
C
   35 TS(IP)=N-1
   20 WOLD(I)=W(IP)
      RS(IP)=SI(I)
   21 RV(IP)=VNEW
C
C    FILL IN NOTCHED GATES FOR CONTOUR DETECTION
C
      DO 26 N=1,NOCTR
      I=NOTCH(N)
      IP=I+1
   26 W(IP)=W(I)
      IAZIO=IAZI
```

```
      SUBROUTINE CONTOR
C
C     ****************************************************************
C
C     NAME:     CONTOR
C     PROJECT: ERT 8035-620 (FAA)
C
C     PURPOSE:  TO DETECT REGIONS OF THE REFLECTIVITY AND SHEAR
C               FIELDS THAT ARE ABOVE PRESET FIXED THRESHOLD
C               LEVELS, TO ASSOCIATE THESE REGIONS FROM RADIAL TO
C               RADIAL AND TO ACCUMULATE ATTRIBUTES OVER EACH
C               CONTOUR REGION.  CONTOUR BOUNDARYS ARE OUTPUT TO AN
C               INTERNAL PLOT FILE THROUGH SUBR PLOT.  EACH
C               SUPER THRESHOLD EVENT IS FURTHER PROCESSED BY
C               SUBR. PEAKD TO DETECT PEAK VALUES.  CONTOUR SEGMENTS
C               AND THEIR ATTRIBUTES ARE JOINED TO FORM COMPLETE
C               CONTOURS IN CONTR2.  A DIRECTORY OF THE CONTOUR
C               SEGMENT ID'S AND THEIR BASE CONTOUR ID IS MAINTAINED
C               AS THE JNIDA(ARRAY).
C
C     INTERFACES:
C        CALLING MOD.   DOPLR81
C        CALLED MODS.   PLOT,PEAKD
C        INPUT PARM.    NONE
C        OUTPUT PARM.   NONE
C        COMMON BLOCKS
C          ARYMX,AZM,AZ2,CNT,COUNT,DATA5,DATA1,DATA4,DECODE,
C          FILTER,FIXED,FLGS,INSUB,KNCTR,KNTBL,MORED,NVLIS,
C          NVLIT,OFFS,OUTPAR,PARM,PNTRS,PRSTORE,PVSTORE,PWORK,
C          RADCOM,REFL,SECOND,TANGENT,TCON,TLIS,TMAX,VEL,
C          VPARM,ZLOOK
C
C     COMMENTS:  PEAKD IS CALLED TWICE, FIRST FOR REFLECTIVITY
C                EVENTS AND SECOND FOR SHEAR EVENTS.
C
C     VERSION:  1.0  DEC/VAX 11-780
C     DATE:     5/6/81
C     DESIGN:   RKCRANE & GBGUSTAFSON
C     PROGMR:   GBGUSTAFSON
C
C     ****************************************************************
C
      LOGICAL PRINT1,COPLOT,CONTR2,CEPLOT,CPLOT,CONTRV,CALIBO,PROVER
C
      INTEGER YEAR,DAY,TIME
      INTEGER SEC,STARTT,BEGINT,ENDT,BDAY,EDAY
      INTEGER HV,B,C,HR,TL,TSEC,TM,TML,TMX
      INTEGER W,WI,WC,WB,DELTW,TS,TI,DELTV
C
      REAL DI(64),CTR(64),CI1(64),CI2(64),CI3(64),DSI(512)
C
      DIMENSION IFCI(9,256),IECL(10,128),IESCL(10,128),
     +          IA(256),IAVNT(2)
C
      COMMON /KNTBL/   KNID(2),KNIDA(2,1024)
      COMMON /KNCTR/   LT,ITL(2),KLVL,JNID,JNIDA(2,1024),
     +                 KNIDM(2),KNIDL(1,1024)
      COMMON /OFFS/    IZOFF,FZOFF
      COMMON /RADCOM/  YEAR,DAY,TIME,ITILT,IAZI,RDKM,R1KM,
     +                 BEAWI,PWRBAR,RAWBAR
```

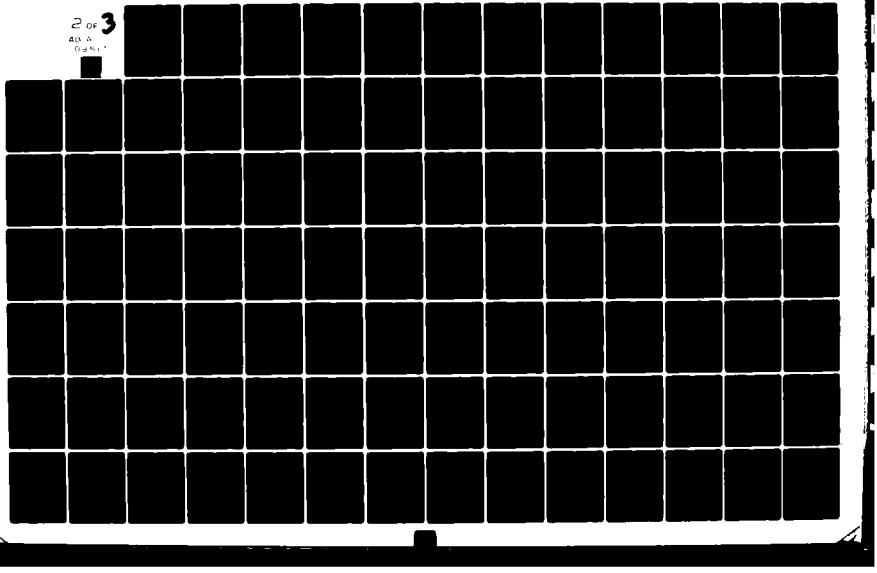
```
      COMMON /ICOM/     SETRI,RNGTBL,SITEL,RANG(382),THGT(382)
      COMMON /INSCAN/   BEGINT,ENDT,DELTR,CGL,ICOM,
     +                  DAZM,BDAY,EDAY
      COMMON /AZM/      AZMUTH,DAZF,AZLAST,NA,ELEVAT,R,C
      COMMON /PARM/     PRINTI,COPLOT,CEPLOT,CONTEC,CONTRV,CALIBO,
     +                  NUMF,NUMR
      COMMON /FLGS/     FRCELL,FRSTG,FRTEXC,FRCLUS,FRCLAD,FRHEAD,
     +                  FRNOIS,FROVER
      COMMON /MORED/    INPRF,SCALE
      COMMON /AZ2/      SINA,COSA,DELTAZ,ISCANF,NEL
      COMMON /ZLOOK/    JZOFF,ZARY(91),RRATE(91)
      COMMON /FILTER/   IATRMN,AREAMN,CELMN(2),SVMX
      COMMON /FIXED/    NFC,TL(2),TC(256),TB(256),NFA,IEMAX,IEVNT(2),
     +                  IRVNT(2),ATR(2560),IAT,NIDF,KDD(2),
     +                  IDSLOT(512)
      COMMON /PRSTORE/  NUF,TATR(1856),NUMAX,IACT(64),IDC(32),IPTNC(64),
     +                  IPTAR(32,2),ITAR(30,32,2),IPNTR(30,32,2),
     +                  IP1R(1920,2),IP2R(1920,2),IP3R(1920,2)
      COMMON /PVSTORE/  NUV,VATR(1856),NVMAX,IACV(64),IDV(32),IPVRNG(64),
     +                  IPTAV(32,2),ITAV(30,32,2),IPNTV(30,32,2),
     +                  IP1V(1920,2),IP2V(1920,2),IP3V(1920,2)
      COMMON /VEL/      TS(382),TL(382),HV(382),RV(382),RS(382)
      COMMON /REFL/     W(382),WI(382),HR(382)
      COMMON /PWORK/    KMAX,T(100),JMXDB,JMAX,IAMAX,IR,JR,
     +                  IMXJMX,NCL,NID,NIDP,IMX,IMN
      COMMON /SECOND/   SEC
      COMMON /OUTPAR/   MOUT,NOUT,NOCTR,NOTCH(30)
      COMMON /ARYMX/    NIDF2,NIEMX,NIDAT,NIDAT2
      COMMON /DECODE/   UP(9),HEIGHT,DLONG,DLAT
      COMMON /NVLIS/    NVARM,NCARM,NVO,NFO,ICO,IO,JO,JYR,LBL,KTL
      COMMON /NVLIT/    KTLL,NKNID,NKDO,IZTH,NKDMX,ITHR,IFXC(1024),HTSI
      COMMON /PNTRS/    NVMIN,NVMX,IELSN,NSCAN,IESNL,NVSCN,NT
      COMMON /TLIS/     TSEC,JDAY,JHR,JMIN,JSEC,IDAY,IHR,IMIN,ISEC
      COMMON /TMAX/     TM,TML,TMX
      COMMON /CNT/      COSPHI,SINEL,COSPHI2,ZMIN,ELAST,SPRM,IFXMX
      COMMON /VPARM/    VX,VY,VXI,VYI,TMKTL,TMKTLL
      COMMON /DATA1/    ECL(10,128),NCO,NCMX,NRJC
      COMMON /DATA5/    ESCL(10,128),NSCO,NSCMX,NSRJC
      COMMON /DATA4/    FCL(9,256),AFCS,WFCS,NFMX,NFARM,KNIDC(1024),NFIA
      COMMON /COUNT/    IXR,IXS
      COMMON /TANGENT/  SINAZ(2),COSAZ(2)
C
      EQUIVALENCE(FCL(1,1),IFCL(1,1))
      EQUIVALENCE(ECL(1,1),IECL(1,1))
      EQUIVALENCE(ESCL(1,1),IESCL(1,1))
C     ------------------------------------------------------------------
C
C     IRMX=IEMAX*NFC
C
      PARAMETER(IRMX=64,IUPMX=9,AMPK=1000.)
      PARAMETER(RPD=.017453,DPR=57.29578,MPAD=2)
      PARAMETER(IPUP=3000,IPDN=2000)
      DATA LDV/3/,LTV/3/
C     ------------------------------------------------------------------
C
C     NORMALISE AZMUTH OF EACH RADIAL BY 1/2 BEAM WIDTHS
C
      AZRAD=AZMUTH*RPD
      AZNORM=AZMUTH-DAZF*BEAWI/2.
      IF(AZNORM.LT.0.) AZNORM=AZNORM+360.
```

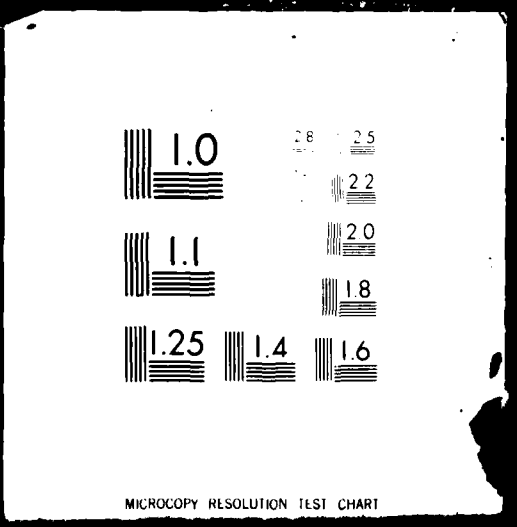
```
      IF(AZNORM.GT.360.) AZNORM=AZNORM-360.
      AZNORM=AZNORM*RPD
      AZNOW=AZNORM
      IF (NA.GT.1) GO TO 61
C
C     INITIALIZE ONCE EACH SCAN (NA=1)
C
      TEMP=0.0
      AZSTAR=AZNORM
      STARTT=(TL(1)-IZOFF)*100+(TL(2)-IZOFF)
      IF(LT.EQ.1)STARTT=STARTT-(TL(2)-IZOFF)
C
C     CPLOT=TRUE, PLOT OUTPUT ON ALL SCANS
C     CEPLOT=TRUE, PLOT OUTPUT ON LOW SCAN ONLY
C     COPLOT IS INTERNAL PLOT TAPE SWITCH
C
      COPLOT=CPLOT
      IF(CEPLOT.AND.NEL.EQ.1)COPLOT=CEPLOT
C
      ELRAD=ELEVAT*RPD
      COSPHI=COS(ELRAD)
      SINEL=SIN(ELRAD)
      COSPHI2=COSPHI*COSPHI
      IZTH=STARTT/100.
C
C     SET TRACK REFERENCE TIME FOR THIS SCAN
C
      IDAY=DAY
      IMIN=TIME/100
      ISEC=TIME-IMIN*100
      SEC=ISEC
      IHR=IMIN/100
      IMIN=IMIN-IHR*100
C
C     CONVERT THIS TIME TO SECONDS FROM START OF YEAR
C
      TSEC=((IDAY*24+IHR)*60+IMIN)*60+ISEC
C
C     SET KTL AT BEGINNING OF VOLUME SCAN
C
      IF(NEL.GT.1) GO TO 30
      KTL=TSEC
   30 TMKTL=TSEC-KTL
      TMKTLL=TSEC-KTLL
C
      TELSN=NEL
      MOUT=0
      NMR=1
      NMV=1
      NVCEL=1
      NCEL=1
      DO 31 K=1,IUPMX
   31 UP(K)=0.0
      DO 3111 J=1,NCL
      TI(J)=TS(J)
 3111 WI(J)=W(J)
      DO 51 L=1,NIDAT2
   51 AIR(L)=0.0
      DO 53 L=1,NIDF2
      DSI(L)=0.0
```

```
    53 IDSLOT(L)=0
       DO 54 L=1,NIDF
       IR(L)=0
    54 IC(L)=0
       DO 55 L=1,IRMX
    55 CIR(L)=0.0
       DO 56 J=1,NFC
       IRVNT(J)=0
       KDD(J)=0
       LIMIT=KNID(J)
       IF(LIMIT.LE.0) GO TO 56
       DO 57 K=1,LIMIT
    57 KNIDA(J,K)=0
    56 KNID(J)=0
C
C    INITIALZE CONTOUR ARRAYS FOR EACH RADIAL
C
    61 TEMP=AZLAST-DAZF*BEAWI/2.
       IF(TEMP.LT.0.) TEMP=TEMP+360.
       IF(TEMP.GT.360.) TEMP=TEMP-360.
       TEMP=TEMP*RPD
C
       KIBC=0
       KATR=0
       KCTR=0
       KSLOT=0
       DO 91 K=1,IRMX
       CI1(K)=0.
       CI2(K)=0.
       CI3(K)=0.
    91 DI(K)=0.
       DO 101 K=1,IEMAX
       IDC(K)=0
       IDV(K)=0
   101 CONTINUE
       DO 102 J=1,JMAX
       IPRNG(J)=0
       IPVRNG(J)=0
   102 CONTINUE
       DO 111 K=1,NFC
       ICVNT(K)=0
   111 CONTINUE
       IP=0
       IPV=0
       IPVB=0
       IPB=0
C
C    FIND REFLECTIVITY EVENTS ABOVE FIXED THRESHOLD LEVEL (TL)
C
       DO 281 I=2,NCL
       DO 231 K=1,NFC
       JK=K-1
       KIBC=NIEMX*JK
       KIC=IEMAX*JK
       WC=W(I)
       WB=W(I-1)
       IF(WC.LE.TL(K).OR.I.EQ.NCL) GO TO 241
       IF (WB.GT.TL(K)) GO TO 151
C
C    COUNT EVENTS
```

```
C
      IE=ICVNT(K)+1
      IF(IE.LE.IEMAX) GO TO 1411
      IF(PROVER) WRITE(7,1412)IEMAX,K
 1412 FORMAT(1X,'EVENT COUNTER EXCEEDED MAX VALUE, IMAX=',I6,5X,
     *'K=',I4)
      IE=IEMAX
 1411 ICVNT(K)=IE
      IF (K.EQ.KLVL) IE0=IE
      IE=IE-1
      IK=IE*NPA+KIBC
      IC(1+IK)=I-1
      IC(3+IK)=IE0
C
C    TALLY EVENT ATTRIBUTES
C
  151 R=RANG(I-1)*AMPK
      INDX=WC-TL(KLVL)+1
      IF(INDX.GT.91)INDX=91
      IF(INDX.LE.0)INDX=1
      WR=R*ZARY(INDX)
      KICE=ICVNT(K)+KIC
      CI1(KICE)=CI1(KICE)+R
      CI2(KICE)=CI2(KICE)+WR
      CI3(KICE)=CI3(KICE)+R*WR
C
C    COMPUTE WATER CONTENT ON LOWEST EL
C
      IF(NEL.EQ.1) DI(KICE)=DI(KICE)+RRATE(INDX)*R
C
  231 CONTINUE
      GO TO 281
C
C    LOCATE END OF EVENT
C
  241 DO 271 KL=K,NFC
      KIBC=NIEMX*(KL-1)
      IF (WR.LE.IL(KL)) GO TO 281
      IE=ICVNT(KL)-1
      IK=IE*NPA+KIBC
  271 IC(2+IK)=I-1
  281 CONTINUE
C
C    END EVENT DETECTION LOOPS
C
      COSAZ(1)=COS(TEMP)
      SINAZ(1)=SIN(TEMP)
      COSA=COS(AZRAD)
      SINA=SIN(AZRAD)
      COSAZ(2)=COS(AZNOW)
      SINAZ(2)=SIN(AZNOW)
C*
C*   EVENT ASSOCIATION, RADIAL TO RADIAL
C*
      DO 611 K=1,NFC
      JK=K-1
      KIC=IEMAX*JK
      KIBC=NIEMX*JK
      KATR=NIDAT*JK
```

AD-A109 517 ENVIRONMENTAL RESEARCH AND TECHNOLOGY INC CONCORD MA F/G 1/2
DETECTION AND TRACKING ALGORITHM REFINEMENT.(U)
OCT 81 G B GUSTAFSON, R K CRANE DTFA01-81-Y-10521
UNCLASSIFIED ERT-P-B035 FAA/RD-81/80 NL





```
      KCTR=IEMAX*JR
      KSLOT=NIDF*JR
C
C    IBVNT/ICVNT ARE NUMBER OF EVENTS ON PRIOR/CURRENT RADIAL
C
      JEM=IBVNT(K)
      IEM=ICVNT(K)
      IPU=IPUP+K
      IPD=IPDN+K
C
C    KEVENT = CURRENT RADIAL
C    IEVENT = PRIOR RADIAL
C
      KEVENT=1
      IEVENT=1
  331 KK=(KEVENT-1)*NPA+KIBC
      IK=(IEVENT-1)*NPA+KIBC
      IK1=IK+NPA
      KK1=KK+NPA
  332 IF (IB(2+IK).EQ.0.AND.IC(2+KK).EQ.0) GO TO 601
C
C    TEST FOR OVERLAPPING (ASSOCIATED) EVENTS
C
      IF (IB(1+IK).GT.IC(2+KK)) GO TO 471
      IF (IB(2+IK).LT.IC(1+KK)) GO TO 471
C
C    ASSOCIATED - UPDATE CURRENT EVENT WITH ASSOCIATION ID
C
      IID=IB(IK1)
      MIID=(IID-1)*IAT+KATR
      MIID1=MIID+IAT
      IF(IID.LE.0.OR.IID.GT.KDD(K)) GO TO 471
      IC(KK1)=IID
      IF(.NOT.COPLOT) GO TO 341
C
C    GENERATE FIXED CONTOUR PLOT TAPE (DEVICE 4)
C        LEFT SIDE
C
C    ......0....0....
C         /
C    ....0.......0...
C
      IDSL=IID+KSLOT
      CALL PLOT(IB(1+IK),1,IC(1+KK),2,IDSLOT(IDSL),K)
C
C    SUM ASSOCIATED EVENT ATTRIBUTES INTO CONTOUR
C
  341 IF(ISCANF.NE.0) GO TO 3811
      KICE=KEVENT+KIC
      ATR(1+MIID)=ATR(1+MIID)+DELTAZ*CI1(KICE)
      ATR(2+MIID)=ATR(2+MIID)+DELTAZ*CI2(KICE)
      ATR(3+MIID)=ATR(3+MIID)+SINA*DELTAZ*CI3(KICE)
      ATR(4+MIID)=ATR(4+MIID)+COSA*DELTAZ*CI3(KICE)
      IE1=IC(3+KK)
      IID1=IC(IE1*NPA)
C
C    OUTPUT ASSOCIATED EVENTS
C
C
C    UPDATE EVENT TO CONTOUR POINTER
```

```
C
      IF(ATR(MIID1).EQ.0.)ATR(MIID1)=IDSLOT(IID1)
C
C   FLAG EDGES (-)
C
      IF (IC(1+KK).EQ.1.OR.IC(2+KK).EQ.IMX) ATR(MIID1)=
     1   -ABS(ATR(MIID1))
C
C   SUM RAIN RATE OVER CONTOUR AREA, LOWEST EL, ALL THRESH
C
      IF (NEL.GT.1) GO TO 381
      KDSLT=IID+KSLOT
      DSI(KDSLT)=DSI(KDSLT)+DI(KICE)*DELTAZ
C
C   TEST FOR MERGE OR SPLIT OF EVENTS, RADIAL TO RADIAL
C
  381 IF(IEVENT.GE.JEM)GO TO 441
      IF (IB(1+IK1).GT.IC(2+KK)) GO TO 441
C
C   DRAW DOWN TO PRESENT AZMUTH.
C
C   ...0....0...0....0..
C            !
C   ....0...!......0....
C
      IF(.NOT.COPLOT) GO TO 6003
      IDSL=IB(NPA+IK1)+KSLOT
      CALL PLOT(IB(2+IK),1,IB(2+IK),2,IDSLOT(IDSL),K)
C
C   DRAW OVER TO IEVENT+1
C
C   ...0....0...0... 0..
C            !
C   ....0...!_ _....0....
C
      CALL PLOT2(IB(1+IK1),2,IDSLOT(IDSL),K)
C
C   DRAW UP TO PREVIOUS AZIMUTH
C
C   ...0....0...0....0..
C            !   !
C   ....0...!_ _!..0....
C
      CALL PLOT2(IB(1+IK1),1,IDSLOT(IDSL),K)
C
C   MERGE - SLIDE EVENT COUNTER TO OUTTER EVENT
C
 6003 IEVENT=IEVENT+1
      IK=IK1
      IK1=IK+NPA
      KID=IB(IK1)
 3815 MKID=(KID-1)*IAT+KATR
      MKID1=MKID+IAT
C
C   IF MERGED EVENTS HAVE COMMON ID (PREVIOUSLY ASSOCIATED)
C     GO ON TO NEXT EVENT
C
      IF(KID.EQ.IID)GO TO 381
C
C   TEST THAT EVENTS ARE REAL
```

```
C
      IF(KID.GT.0.AND.KID.LE.KDD(K))GO TO 401
      GO TO 381
401   IF(ATR(MKID1).EQ.0.0.OR.ATR(MIID1).EQ.0.0)GO TO 381
C
C   SUM MERGED CONTOUR ATTRIBUTES AND ZERO UNUSED SLOTS
C
      IATT=IAT-1
      DO 411 J=1,IATT
      JI=J+MIID
      JK=J+MKID
      ATR(JI)=ATR(JI)+ATR(JK)
  411 ATR(JK)=0.0
C
      IDSLT=IID+KSLOT
      KDSLT=KID+KSLOT
      IF(NEL.GT.1) GO TO 432
      DSI(IDSLT)=DSI(IDSLT)+DSI(KDSLT)
      DSI(KDSLT)=0.0
C
C   IF ANY MERGED EVENT IS EDGE FLAGGED(-), FLAG ALL
C
  432 IF(ATR(MKID1).LT.0..AND.ATR(MIID1).GT.0.) ATR(MIID1)=-ATR(MIID1)
      ATR(MKID1)=0.0
C
C   RESET ASSN ID OF ALL ASSOCIATED EVENTS TO 1ST EVENT
C
      DO 430 J=1,JEM
      JE=J*NPA+KIBC
      IF(IB(JE).EQ.KID) IB(JE)=IID
  430 CONTINUE
C
      DO 431 J=1,KEVENT
      JE=J*NPA+KIBC
      IF(IC(JE).EQ.KID) IC(JE)=IID
  431 CONTINUE
C
C   RESET KNID OF ALL ASSOCIATED EVENTS TO 1ST EVENT
C
      KNIDI=IDSLOT(IDSLT)
      KNIDK=IDSLOT(KDSLT)
      IDSLOT(KDSLT)=0
      IF(KNIDK.LE.0.OR.KNIDK.GT.KNID(K)) GO TO 381
      KNIDA(K,KNIDK)=KNIDI
      GO TO 381
C
C   CLOSE END AROUND EVENTS
C
 3811 ICTR=KEVENT+KCTR
      KNIDU=0
      KNIDT=ABS(CTR(ICTR))
   48 IF(KNIDT.LE.0.OR.KNIDT.GT.KNID(K)) GO TO 381
      KNIDU=KNIDA(K,KNIDT)
      IF(KNIDU.EQ.KNIDT) GO TO 49
      KNIDT=KNIDU
      GO TO 48
   49 NIDEK=KDD(K)
      DO 3812 JK=1,NIDEK
      JDSLT=JK+KSLOT
      IF(IDSLOT(JDSLT).EQ.KNIDU) GO TO 3813
```

```
 3812 CONTINUE
C
      GO TO 381
 3813 KID=JK
      GO TO 3815
  441 IF(KEVENT+1.GT.IEM) GO TO 451
      IF(IC(1+KK1).GT.IB(2+IK)) GO TO 451
C
C   DRAW LINE CONNECTING IC(N) TO IC(N+1)
C
C   ....................
C
C   ..0...0_____0.....0.
C
      IF(.NOT.COPLOT) GO TO 6001
      KDSL=IC(KK1)+KSLOT
      CALL PLOT(IC(2+KK),2,IC(1+KK1),2,IDSLOT(KDSL) K)
      KID=IC(NPA+KK1)
      KDSL=KID+KSLOT
C
C   INCREMENT EVENT COUNTER, CURRENT RADIAL
C
6001  KEVENT=KEVENT+1
C
C   SPLIT - UPDATE EVENT WITH ASSN ID OF 1ST EVENT
C
      KK=KK1
      KK1=KK+NPA
      IF(IID.LE.0.OR.IID.GT.KDD(K))GO TO 471
      IC(KK1)=IID
      GO TO 341
C
C              RIGHT SIDE.
C
C   ...0........0.......
C               |
C   ......0.....0.......
C
451   IF(.NOT.COPLOT) GO TO 6002
      IDSL=IB(IK1)+KSLOT
      CALL PLOT(IB(2+IK),1,IC(2+KK),2,IDSLOT(IDSL),K)
      KDSL=IC(KK1)+KSLOT
C
C   INCREMENT COUNTERS, BOTH RADIALS
C
6002  IEVENT=IEVENT+1
      KEVENT=KEVENT+1
      IK=(IEVENT-1)*NPA+KIBC
      KK=(KEVENT-1)*NPA+KIBC
      IK1=IK+NPA
      KK1=KK+NPA
C
C   LOOP BACK THROUGH ASSN PROCESS AGAIN FOR NEW EVENTS
C
      IF(KEVENT.GT.IEM.AND.IEVENT.GT.JEM)GO TO 601
      IF(KEVENT.GT.IEM)GO TO 481
      IF(IEVENT.GT.JEM)GO TO 521
      GO TO 332
C
C   EVENTS ARE NOT ASSOCIATED
```

```
C
471    IF (IB(1+IK).EQ.0) GO TO 521
       IF (IC(1+KK).EQ.0) GO TO 481
       IF (IC(2+KK).LT.IB(1+IK)) GO TO 511
C
C    NO EVENT ON CURRENT RADIAL, CLOSE CONTOUR
C
  481 IID=IB(IK1)
      MIID=(IID-1)*IAT+KATR
      MIID1=MIID+IAT
      IDSL=IID+KSLOT
      IF(IID.LE.0) GO TO 802
      IF(ATR(1+MIID).GE.TATRMN) GO TO 802
C
C    ELIMINATE CONTOUR REGION IF TOO SMALL
C
      IDSL=IID+KSLOT
      IDSLOT(IDSL)=0
      DO 8005 I=1,IAT
 8005 ATR(I+MIID)=0
  802 CONTINUE
C
C    CLOSE OUT CONTOUR ON IB
C
      IF(.NOT.COPLOT) GO TO 6004
      IF(IDSLOT(IDSL).EQ.0) GO TO 6004
C
C    ....O..O...
C        |
C    ....|......
C
      CALL PLOT(IB(1+IK),1,IB(1+IK),2,IDSLOT(IDSL),K)
C
C    ....O..O...
C        |
C    ....|__....
C
      CALL PLOT2(IB(2+IK),2,IDSLOT(IDSL),K)
C
C    ....O..O...
C        |  |
C    ....|__|...
C
      CALL PLOT2(IB(2+IK),1,IDSLOT(IDSL),K)
C
C    INCREMENT EVENT COUNTER, PREVIOUS RADIAL
C
 6004 IEVENT=IEVENT+1
      IK=(IEVENT-1)*NPA+KIBC
      IK1=IK+NPA
      IF(KEVENT.GT.IEM.AND.IEVENT.GT.JEM)GO TO 601
      IF(IEVENT.GT.JEM)GO TO 521
      IF(KEVENT.GT.IEM)GO TO 481
      IF (IC(1+KK).LE.IB(2+IK)) GO TO 332
      IF (IC(2+KK).NE.0) GO TO 501
      GO TO 332
501   IF (IB(1+IK).EQ.0) GO TO 521
511   IF (IC(1+KK).GT.IB(2+IK)) GO TO 331
C
C    UNASSOICATED, START NEW CONTOUR
```

```
C
521    IF (IC(1+KK).EQ.0) GO TO 562
C
C    LOCATE EMPTY ASSN ID SLOT
C
       DO 5522 IID=1,NIDF
       IDSL=IID+KSLOT
       IF(IDSLOT(IDSL).NE.0)GO TO 5522
       IDD=IID
       IC(KK1)=IID
       IF(ISCANF.NE.0) GO TO 522
       KNID(K)=KNID(K)+1
       IF(KNID(K).LE.NKDMX) GO TO 523
       KNID(K)=NKDMX
       WRITE(7,540) K,NA,NEL
  540 FORMAT(1X,'TOO MANY SEGMENTS ON TL',I2,' RAD',I3,' EL',I3)
  523 IDSLOT(IDSL)=KNID(K)
       KNIDA(K,KNID(K))=KNID(K)
       GO TO 5523
C
  522 IDSLOT(IDSL)=KEVENT
       GO TO 5523
C
C    NOTE
C        WHEN ALL ID,S ARE USED,
C        ID(NIDF,K) WILL BURDEN ALL OTHER CELLS
C
 5522 CONTINUE
       IC(KK1)=NIDF
       IF((MOUT.EQ.0).AND.PROVER) WRITE(7,6007)
       MOUT=MOUT+1
 6007 FORMAT(' HELP    TOO MANY FIXED CONTOURS, NIDF EXCEEDED')
       KNID(K)=KNID(K)+1
       IDSLOT(IDSL)=KNID(K)
       IID=NIDF
       IDD=IID
 5523 KDD(K)=MAX0(KDD(K),IDD)
       IID=IDD
C
C    PLOT INITIAL  EDGE
C
       IF(.NOT.COPLOT.OR.NA.EQ.1) GO TO 527
       KID=IC(NPA+KK)
       KDSL=KID+KSLOT
C
C    ..................
C
C    ...O_____O.......
C
       CALL PLOT(IC(1+KK),2,IC(2+KK),2,IDSLOT(IDSL),K)
  527 IF(ISCANF.NE.0) GO TO 561
C
C    SUM NEW CONTOUR ATTRIBUTES
C
       MIID=(IID-1)*IAT+KATR
       MIID1=IAT+MIID
       MIDD=(IDD-1)*IAT+KATR
       IMIDD=IAT+MIDD
       KICE=KEVENT+KIC
       ATR(1+MIDD)=DELTAZ*CI1(KICE)+ATR(1+MIDD)
```

```
      ATR(2+MIDD)=DELTAZ*CI2(KICE)+ATR(2+MIDD)
      ATR(3+MIDD)=SINA*DELTAZ*CI3(KICE)+ATR(3+MIDD)
      ATR(4+MIDD)=COSA*DELTAZ*CI3(KICE)+ATR(4+MIDD)
      IE1=IC(3+KK)
      IID1=IC(NPA*IE1)
      ATR(IMIDD)=IDSLOT(IID1)
C
C   FLAG EDGES (-)
C
      IF (IC(1+KK).EQ.1.OR.IC(2+KK).EQ.IMX) ATR(IMIDD)=-
     1ABS(ATR(IMIDD))
C
C   SUM RAIN RATE OVER AREA, LOWEST EL
C
      IF (NEL.GT.1)   GO TO 561
      DSI(IDD+KSLOT)=DI(KICE)*DELTAZ
  561 CONTINUE
C
C   INCREMENT EVENT COUNTER AND PASS THROUGH ASSOCIATION AGAIN
C
  562 KEVENT=KEVENT+1
      KK=(KEVENT-1)*NPA+KIBC
      KK1=KK+NPA
      IF(KEVENT.GT.IEM.AND.IEVENT.GT.JEM)GO TO 601
      IF(KEVENT.GT.IEM)GO TO 481
      IF(IEVENT.GT.JEM)GO TO 521
      GO TO 332
  601 CONTINUE
C
C   END CONTOUR ASOCIATION
C
  611 CONTINUE
C*
C*  IDENTIFY TL(1) SEGMENTS WHICH ENCLOSE HIGHER THRESH SEGMENTS
C*
      IF(NEL.NE.1) GO TO 650
      K=NFC
      IEMX=ICVNT(1)
      IF(IEMX.LE.0) GO TO 650
  615 J=K-1
      IF(ICVNT(K).LE.0) GO TO 635
      KIBC=NIEMX*J
      KSLOT=NIDF*J
      JE=1
      JK=KIBC
C
C   LOOP THROUGH TL(1) SEGMENTS
C
      DO 620 IE=1,IEMX
      IK=(IE-1)*NPA
C
C   LOOP THROUGH TL(K) (HIGHER) SEGMENTS
C
  625 CONTINUE
      IF(IC(1+JK).GT.IC(2+IK)) GO TO 620
      IF(IC(1+JK).GE.IC(1+IK).AND.IC(2+JK).LE.IC(2+IK)) GO TO 645
      KNIDI=0
      GO TO 619
  645 KNIDI=IDSLOT(IC(4+IK))
  619 KNIDJ=IDSLOT(IC(4+JK)+KSLOT)
```

```
      KNIDL(J,KNIDJ)=KNIDI
      JE=JE+1
      IF(JE.GT.ICVNT(K)) GO TO 635
      JK=(JE-1)*NPA+KIBC
      GO TO 625
C
  620 CONTINUE
C
C   DECREMENT COUNTER TO NEXT LOWER THRESHOLD LEVEL
C
  635 K=J
      IF(K.GT.1) GO TO 615
  650 CONTINUE
C*
C*  LOCATE REFL AND SHEAR PEAKS WITHIN CONTOUR SEGMENTS
C*
      IEMX=ICVNT(KLVL)
      KIBC=NIEMX*(KLVL-1)
C
C   NOTCH 3 GATES AROUND FOLDING OFFSETS FOR CELL DETECTION
C         NOTE: ZI(I)=W(I+1)
C
      DO 270 N=1,NOCTR
      I=NOTCH(N)+1
  270 W(I)=0
C
C   LOOP THROUGH EACH SEGMENT ON THE KLVL THRESHOLD
C
      DO 280 IE=1,IEMX
      IAD=(IE-1)*NPA+KIBC
      IL=IC(1+IAD)
      IH=IC(2+IAD)
C
C   REFL PEAK DETECTION
C
      DO 285 I=IL,IH
      DELTW=W(I)-W(I-1)
      IF(DELTW.LT.0) GO TO 171
      IF(DELTW.GT.0) IPB=I-1
      GO TO 181
C
  171 CONTINUE
      IF (IPB.EQ.0) GO TO 181
      IP=IP+1
      IF(IP.LE.JMAX) GO TO 1711
      IF(PROVER) WRITE(6,1913)IP,IEVENT
 1913 FORMAT(1X,17HN PEAKS EXCEEDED,2I6)
      IP=JMAX
      GO TO 181
 1711 IPRNG(IP)=(I+IPB)/2
      IPB=0
  181 CONTINUE
C
C   SHEAR PEAK DETECTION
C
      IF(TS(I).EQ.-999) GO TO 191
      IF(TS(I-1).EQ.-999) GO TO 201
      DELTV=IABS(TS(I))-IABS(TS(I-1))
      IF(DELTV.LT.0) GO TO 191
      IF(DELTV.EQ.0) GO TO 285
```

```
  201 IPVB=I-1
      GO TO 285
C
  191 IF(IPVB.EQ.0) GO TO 285
      IPV=IPV+1
      IF(IPV.LE.JMAX) GO TO 1912
      IF(PROVER) WRITE(6,1913)IPV,IEVENT
      GO TO 285
 1912 IPVRNG(IPV)=(I+IPVB)/2
      IPVB=0
  285 CONTINUE
C
C   END OF SEGMENT, CLOSE OFF ANY REMAINING PEAKS
C
C   REFL
      IF (IPB.EQ.0) GO TO 251
      IP=IP+1
      IF(IP.LE.JMAX)GO TO 242
      IF(PROVER) WRITE(6,1913)IP,IEVENT
      IP=JMAX
      GO TO 243
  242 IPRNG(IP)=(I+IPB)/2
  243 IPB=0
  251 IDC(IE)=IP
C   SHEAR
      IF(IPVB.EQ.0) GO TO 261
      IPV=IPV+1
      IF(IPV.LE.JMAX) GO TO 252
      IPV=JMAX
      GO TO 253
  252 IPVRNG(IPV)=(I+IPVB)/2
  253 IPVB=0
  261 IDV(IE)=IPV
  280 CONTINUE
C
C   REFLECTIVITY CELL DETECTION
C
      B=C
      C=C+1
      IF(C.GT.MXAD) C=1
      IF(.NOT.CONTRZ)GO TO 800
      CALL PEAKD(W,LDV,TL(KLVL),3,NCEL,TATR,NUMAX,IACT,IDC,
     +           IPRNG,HR,NMR,TS,IPTAR,ITAR,IPNTR,
     +           IP1R,IP2R,IP3R)
C
C   TANGENTIAL SHEAR CELL DETECTION
C
  800 IF(.NOT.CONTRV) GO TO 8000
      CALL PEAKD(TS,LTV,0,4,NVCEL,VATR,NVMAX,IACV,IDV,
     +           IPVRNG,HV,NMV,W,IPTAV,ITAV,IPNTV,
     +           IP1V,IP2V,IP3V)
 8000 CONTINUE
C
C   ISCANF = (+/-)1 INDICATES END OF SCAN
C
      IF(ISCANF.NE.0) GO TO 871
C
C   PREPARE FOR NEXT RADIAL
C
      DO 810 K=1,NFC
```

```
      JK=K-1
      NTBC=NTEMX*JK
      IBVNT(K)=ICVNT(K)
      IEMX=MAX0(1,IBVNT(K))
      NMAX=NPA*IEMX+NTBC
      NMIN=1+NTBC
C
C     ON FIRST RADIAL SAVE SEGMENT ID'S FOR END AROUND ASSOCIATION
C
      IF(NA.NE.1) GO TO 815
      KSLOT=NTIDC*JK
      ICTR=IEMAX*JK
      DO 805 J=1,IEMX
      IID=IC(J*NPA+NTBC)
      IF(IID.LE.0) GO TO 805
      ICTR=IID+KCTR
      CTR(ICTR)=IDSLOT(IID+KSLOT)
  805 CONTINUE
C
C     SET CURRENT RADIAL SEGMENTS INTO PRIOR RADIAL ARRAYS
C     ON FIRST RADIAL, LOW SCAN SAVE CONTOUR END POINTS FOR PLOT
C
      IF(NEL.NE.1) GO TO 815
      IAVNT(K)=ICVNT(K)
      DO 811 NIE=NMIN,NMAX
      IA(NIE)=IC(NIE)
      IB(NIE)=IC(NIE)
  811 IC(NIE)=0
      GO TO 810
C
  815 CONTINUE
      DO 816 NIE=NMIN,NMAX
      IB(NIE)=IC(NIE)
  816 IC(NIE)=0
  810 ICVNT(K)=0
C
      RETURN
```

```
      ENTRY CONTR2
C
C    ************************
C    FINISH SCAN, FINAL RADIAL
C    ************************
C
      IF (ISCANF,EQ,0) GO TO 871
C*
C*   INITIAL AND FINAL AZMUTHS MATCH
C*
      TEMP=AZNORM
      DELTAZ=AZSTAR-TEMP
      AZNOW=AZSTAR
C
C    RECOVER INITIAL RADIAL DATA AND ASSOCIATE INITIAL TO FINAL RADIALS
C
      DO 8611 IX=1,NCL
      TS(IX)=TI(IX)
 8611 W(IX)=WI(IX)
      GO TO 61
C*
C*   SORT KNIDA TABLE TO ESTABLISH BASE CONTOUR ID VALUE
```

```
C*
  821 CONTINUE
      DO 700 K=1,NFC
      IEMX=KNID(K)
      IF(IEMX.LT.1) GO TO 700
      DO 701 IE=1,IEMX
      J=IE
  702 I=KNIDA(K,J)
      IF(I.EQ.0) GO TO 701
      J=J-1
      GO TO 702
  701 KNIDA(K,IE)=I
  700 CONTINUE
      IF(ISCANF.GT.0) GO TO 872
C*
C*   INITIAL AND FINAL AZMUTHS DO NOT MATCH
C*
      DO 821 K=1,NFC
      JK=K-1
      KIBC=NIEMX*JK
      KATR=NTDAT*JK
      KSLOT=NIDF*JK
      KCTR=IEMAX*JK
      IEMX=IBVNT(K)
      IF(IEMX.LT.1) GO TO 825
C
C    FLAG FINAL AZMUTH BOUNDARY CONTOUR ID VALUES NEGATIVE
C
      DO 822 IE=1,IEMX
      IK1=IE*NPA+KIBC
      IK=IK1-NPA
      IDD=IB(IK1)
      MIDD=IAT*IDD+KATR
      ATR(MIDD)=-ABS(ATR(MIDD))
C
C    CLOSE OFF CONTOUR PLOT ON FINAL RADIAL
C
      IF(.NOT.COPLOT) GO TO 822
      IDSL=IDD+KSLOT
      CALL PLOT(IB(1+IK),2,IB(2+IK),2,IDSLOT(IDSL),K)
  822 CONTINUE
C
C    FLAG INITIAL AZMUTH BOUNDARY CONTOUR ID VALUES NEGATIVE
C
  825 IEMX=IAVNT(K)
      IF(IEMX.LT.1) GO TO 821
      DO 823 IE=1,IEMX
      ICTR=IE+KCTR
      KNIDU=0
      KNIDT=ABS(CTR(ICTR))
      IF(KNIDT.GT.KNID(K).OR.KNIDT.EQ.0) GO TO 823
      KNIDU=KNIDA(K,KNIDT)
      DO 829 J=1,NIDF
      IDY=IDSLOT(J+KSLOT)
      IF(KNIDU.EQ.IDY) GO TO 8291
  829 CONTINUE
      GO TO 823
 8291 IATA=J*IAT+KATR
      ATR(IATA)=-ABS(ATR(IATA))
C
```

```
C    PLOT CONTOUR EDGES ON INITIAL RADIAL
C
       IF(.NOT.COPLOT) GO TO 823
       IF (CII-1)*NPATHIRC
       AZNOW AZSTAE
       COSAZ(I) COS(AZNOW)
       SINAZ(I) SIN(AZNOW)
       CALL PLOT(IA(IFIR)*I,IA(2*IFIR)*I,KNIDU,K)
  823 CONTINUE
  821 CONTINUE
C*
C*   ESTABLISH A CONTOUR ID TABLE ON LOWEST ELEVATION
C*
  822 CONTINUE
       IF(NEI.NE.1) GO TO 937
C
       JNTID=0
       DO 704 K=1,NEC
       JEMX=KNTID(K)
       IF(JEMX.LT.1) GO TO 704
       NENTID=KNTIDM(K)
       KNTIDM(K)=JEMX
C
C    COUNT CONTOURS AT EACH THRESHOLD LEVEL
C
       DO 705 JE=1,JEMX
       IF(JE.NE.KNIDA(K,JE)) GO TO 705
       JNTID=JNTID+1
       JNTIDA(K,JE)=JNTID
  705 CONTINUE
C
C    REDUCE KNIDA DIRECTORY TO BASE CONTOUR ID VALUES
C
       DO 706 JE=1,JEMX
       J=KNIDA(K,JE)
  706 JNTIDA(K,JE)=JNTIDA(K,J)
C
C    DIRECT ENCLOSED CONTOUR POINTERS TO BASE CONTOUR ID
C
       IF(K.EQ.1) GO TO 704
       L=K-1
       DO 708 JE=1,JEMX
       J=KNTID(L,JE)
  708 KNTID(L,JE)=KNIDA(1,J)
C
C    ESTABLISH DIRECTORY ON NEXT HIGHER THRESH LEVEL
C
  704 CONTINUE
C
C      ******************************
C      PREPARE FIXED CONTOR ATRIBUTES
C      ******************************
C
       IF(PRINT1) WRITE(6,712)
  712 FORMAT(17X,       'AVE   /-------LOCATION-------/    AREA  AVE  FIX'/
     +  5X,'TLD  AREA  REFL   EAST NORTH RANGE    AZM  RESLN PRECP CTR'/
     +'   ID DBZ  KKM2   DBZ    KM    KM    KM     DEG  ELMNT MT/HR REF')
C
       DACEL=SETRI*DAZM*1.E-03
       IF(KNTID(KLVL).LT.1) GO TO 1413
```

```
      K=NLVL
      JK=K-1
      KEMX=KNTID(K)
      KSLOT=NTOP*JK
      KATR=NTDAT*JK
C
C     ZERO CONTOUR ACCUMULATORS AT START OF SCAN
C
      IXCT=0
      NEO=0
      NKTID=0
      NK=0
      AECS=0
      WECS=0
      NEMX=0
      IF(NKNTID.EQ.0) GO TO 9321
      DO 9322 ID=1,NKNTID
 9322 KNTOC(ID)=0
 9321 CONTINUE
C
C     CALCULATE NUMBER OF CONTOURS FOR LOWEST LEVEL
C
      NIDK=KID(KLVL)
C
C     LOOP THROUGH EACH SEGMENT
C
      DO 9332 J=1,NIDK
      JAT=(J-1)*IAT+KATR
      JAK=J+KSLOT
C
C     TEST STORAGE SLOT FOR DATA
C
      IF(IDSLOT(JAK).EQ.0) GO TO 9332
      ID1=J
C
C     CHECK QUALITY OF DATA
C
      IF(ATR(1+JAT).LT.TATRMN.OR.ATR(2+JAT).LE.0) GO TO 9332
C
C     DECODE CONTOUR ATTRIBS
C
      ABAR=ATR(1+JAT)
      ZBAR=ATR(2+JAT)/ABAR
      ZABAR=1./(ABAR*ZBAR)
      ABAR=SETRI*ABAR
      XBAR=ATR(3+JAT)*ZABAR
      YBAR=ATR(4+JAT)*ZABAR
C
C     ADJUST UNITS AND COMPUTE FINAL ATTRIBS
C
      ABAR=ABAR*1.E-9*COSPHI
      XBAC=XBAR*1.E-3
      YBAC=YBAR*1.E-3
      RBAR=SQRT(XBAC*XBAC+YBAC*YBAC)
      RCELLS=ABAR/(RBAR*DACEL)
      AZBAR=ATAN2(XBAC,YBAC)*DPR
      IF(AZBAR.LT.0.) AZBAR=AZBAR+360.
      ZBAR=10.*ALOG10(ZBAR)
      TPREC=DST(J)*SETRI*1.E-9*COSPHI
C
```

```
C     TRACE ASSOCIATED CELLS TO A COMMON ID
C
        DATA JATTIG/
        KNTDR=0
        KNIDS=ATR(IATA)
        KNIDR=IABS(KNIDS)
        IF(KNIDR.LE.0.OR.KNIDR.GT.KEMX) GO TO 6023
C
C     FLAG EDGES NEGATIVE
C
        KNIDQ=SIGN(KNTDA(KIVL,KNIDR),KNIDS)
 6023 CONTINUE
C
C     INCREMENT CONTOUR COUNTER
C
        IXCT=IXCT+1
        IF(IXCT.GT.NFARM) GO TO 9323
C
C     STORE FIXED CONTOUR ATTRIBS IN FCL(ARRAY)
C
C           (1) AREA
        FCL(1,IXCT)=ABAR
C           (2) AVG. REFLECTIVITY
        FCL(2,IXCT)=ZBAR
C           (3) LAST CENTROID POSITION
        FCL(3,IXCT)=XBAC*COSPHI+DLONG
C           (4) NORTH CENTROID POSITION
        FCL(4,IXCT)=YBAC*COSPHI+DLAT
C           (5) TOTAL WATER FLUX
        FCL(5,IXCT)=TPREC
C           (6) CLUSTER CELL ID
        IFCL(6,IXCT)=KNIDQ
C
        IFCL(7,IXCT)=0
        IFCL(8,IXCT)=0
        IFCL(9,IXCT)=0
        IF(PRINT1)
       +WRITE(6,716) ID1,ITL(K),ABAR,ZBAR,FCL(3,IXCT),FCL(4,IXCT),
       +RBAR,AZBAR,RCELLS,TPREC,KNIDQ
  716 FORMAT(1X,I3,I4,F7.3,F5.1,3F6.1,F7.1,2F6.2,I4)
C
C     TEST THAT ID IS VALID
C
        KKNID=IABS(KNIDQ)
        IF(KKNID.LE.0.OR.KKNID.GT.KEMX) GO TO 9322
C
C     SET CELL/CONTOUR POINTER
C
        KNTDC(KKNID)=IXCT
        IF(KKNID.GT.NK) NK=KKNID
        GO TO 9325
C
C     INCREMENT ID OVERFLOW COUNTER
C
 9322 NKDO=NKDO+1
        GO TO 9325
C
C     INCREMENT CONTOUR OVERFLOW COUNTER
C
 9323 NFO=NFO+1
```

```
C
C    SUM AREA AND WATER FLUX TOTALS FOR ALL CONTOUR REGIONS
C
 9325 AFCS=AFCS+ABAR
      WFCS=WFCS+TPREC
 9332 CONTINUE
C
C    RUN THROUGH THE KNIDA TABLE FOR THE CELL/CONTOUR POINTER
C
      DO 931 IE=1,KNID(KLVL)
      ID=KNIDA(KLVL,IE)
      IF(ID.LE.0.OR.IE.GT.NEMX.OR.IE.EQ.ID) GO TO 931
      KNIDC(IE)=KNIDC(ID)
  931 CONTINUE
  932 CONTINUE
C
C    CLEAN UP COUNTERS AT END OF SCAN
C
      NEMX=IXCT-NFO
      NCMX=IXR-NCO
      NSCMX=IXS-NSCO
      IESNL=NEL
      ILAST=ELEVAT
      IF(NK.GT.0) NKNID=NK
C
C    TRACE ALL ASSOCIATED REFLECTIVITY CELLS TO BASE CONTOUR ID
C
      DO 9475 J=1,NCMX
      IUP=0
      IUR=IABS(IECL(5,J))
      IF(IUR.LE.0.OR.IUR.GT.KNID(KLVL)) GO TO 9425
      IUP=KNIDA(KLVL,IUR)
 9425 CONTINUE
      IECL(5,J)=IUP
      IF(IUP.GT.NKDMX) IECL(5,J)=NKDMX
 9475 CONTINUE
C
C    TRACE ALL ASSOCIATED SHEAR CELLS TO BASE CONTOUR ID
C
      DO 9578 J=1,NSCMX
      IUV=0
      IUR=IABS(IESCL(5,J))
      IF(IUR.LE.0.OR.IUR.GT.KNID(KLVL)) GO TO 9525
      IUV=KNIDA(KLVL,IUR)
 9525 CONTINUE
      IESCL(5,J)=IUV
      IF(IUV.GT.NKDMX) IESCL(5,J)=NKDMX
 9578 CONTINUE
C
C     *
C
 1413 CONTINUE
      ISCANF=0
      RETURN
      END
```

```
      SUBROUTINE PLOT(IR,I,JR,J,KNID,K)
C
C     ***************************************************************
C
C     NAME:       PLOT
C     PROJECT:    EET A1/?? 600 (TAA)
C
C     PURPOSE:    WRITE CONTOUR BOUNDARY LOCATIONS TO AN
C                 INTERNAL SCRATCH FILE.  INCLUDE KNID
C                 NUMBER ON WHICH TO ASSOCIATE SEGMENTS.
C
C     INTERFACES:
C        CALLING MOD: CONTOR
C        CALLED MODS: NONE
C        INPUT PARMS: IR=RANGE GATE OF X1,Y1
C                     I=RADIAL CODE OF X1,Y1
C                     JR=RANGE GATE OF X2,Y2
C                     J=RADIAL CODE OF X2,Y2
C                     KNID=CONTOUR SEGMENT ID
C                     K=THRESHOLD LEVEL
C        OUTPUT PARM: NONE
C        COMMON BLOCKS
C          UPDATED:   NONE
C          READ:      TCON,TANGENT,DECODE
C
C     COMMENTS:  CONTOURS ARE OUTPUT ON 2 THRESHOLD LEVELS
C
C     VERSION:   1.0  DEC/VAX-11
C     DATE:      1/26/81
C     DESIGN:    GRGUSTAFSON
C     PROGMR:    GRGUSTAFSON
C
C     ***************************************************************
C
      COMMON /TCON/     SETRT,RNGDLY,SETEL,RANG(382),IHGT(382)
      COMMON /TANGENT/ SINA(2),COSA(2)
      COMMON /DECODE/  UF(9),HEIGHT,DLONG,DLAT
C
      R1=RANG(IR-1)
      X1=SINA(I)*R1+DLONG
      Y1=COSA(I)*R1+DLAT
      GO TO 20
C
C     USE LAST X2,Y2 PAIR AS CURRENT X1,Y1 PAIR
C
      ENTRY PLOT2(JR,J,KNID,K)
      X1=X2
      Y1=Y2
C
   20 R2=RANG(JR-1)
      X2=SINA(J)*R2+DLONG
      Y2=COSA(J)*R2+DLAT
C
C     OUTPUT KNID, THRESH LEVEL AND VECTOR END POINTS TO INTERNAL FILE
C
      IF(KNID.LE.0) RETURN
      WRITE(8) KNID,K,X1,Y1,X2,Y2
C
      RETURN
      END
```

```
      SUBROUTINE PEAK(U,LDB,TM,ITY,NCELL,TATR,NUMAX,IACT,IDC,
     +                IPCRNG,HB,NMX,V,IPTA,ITA,IPNT,
     +                IP1,IP2,IP3)
C
C    ***************************************************************
C
C    NAME:      PEAK
C    PROJECT:   ERT A579-600 (FAA)
C
C    PURPOSE:   ASSOCIATE REGIONS OF PEAK REFLECTIVITY OR SHEAR
C               VALUES WITHIN CONTOUR REGIONS TO FORM PEAK CELLS,
C               ACCUMULATE CELL ATTRIBUTES INCLUDING:
C                    1)CENTROID LOCATION
C                    2) AREA
C                    3) HEIGHT
C                    4) REFL
C                    5) SHEAR
C                    6) CONTOUR ID (POINTER)
C               AND TEST FOR VALID ATTRIBUTE VALUES.
C
C    INTERFACES:
C       CALLING MOD.   CONTOR
C       CALLED MODS.   PKCELL
C       INPUT PARM.    U,LDB,TM,ITY,IACT,IDC,IPCRNG,IPTA,
C                      ITA,IPNT
C       OUTPUT PARM.   NCELL,TATR,HB,NMX,V,IP1,IP2,IP3
C       COMMON BLOCKS
C         AZM,AZ2,COUNT,DATA5,DATA1,DECODE,FILTER,FIXED,FLGS,
C         INSUB,KNCTR,KNTBL,OFFS,OUTPAR,PARM,PWORK,QUANTX,TCON,
C         VEL,WIND
C
C    COMMENTS:  SHEAR PARAMETER MAY BE EITHER TANGENTIAL SHEAR,
C               RADIAL SHEAR OR TOTAL SHEAR.  CELL BOUNDARYS
C               ARE DEFINED BY CONTOUR 3DB BELOW PEAK VALUE.
C
C    VERSION:   1.3  DEC/VAX-11
C    DATE:      1/6/81
C    DESIGN:    RKCRANE
C    PROGMR:    GBGUSTAFSON
C
C*****************************************************************
C
      LOGICAL PRINT1,COPLOT,CEPLOT,CONTRZ,CONTRV,CALIBO,PROVER
      LOGICAL NAONE
      REAL TATR(1856)
      REAL*8 SVA,SVB,SVC,SA2,SB2,SC2,SAB,SAC,SBC,SV2,SB,SC
C
      INTEGER U(382),HB(382),IACT(64),IDC(32),
     +        IPCRNG(64),IECL(10,128),IESCL(10,128)
      INTEGER W,WI,HR,TS,TI,HV,V(382)
      INTEGER BEGINT,ENDT,BDAY,EDAY
      INTEGER TL,T,TM,B,C
C
      DIMENSION IPTA(32,2),ITA(30,32,2),IPNT(30,32,2)
      DIMENSION IP1(1920,2),IP2(1920,2),IP3(1920,2)
      DIMENSION GOUT(20)
C
      COMMON /QUANTX/ VQUANT
      COMMON /KNTBL/  KNID(2),KNIDA(2,1024)
      COMMON /KNCTR/  LT,ITL(2),KLVL,JNID,JNIDA(2,1024),
```

```
     +                    KNIDM(2),KNTDL(1,1024)
      COMMON /PARM/       PRINT1,COPLOT,CEPLOT,CONTRZ,CONTRV,CALIBO,
     +                    NUME,NUMK
      COMMON /FLGS/       PRCELL,PRSIG,PREIXC,PRCLUS,PRSCAN,PRHEAD,
     +                    PRNOIS,PROVER
      COMMON /PWORK/      KMAX,IC(100),JMXDB,JMAX,IAMAX,IR,JK,
     +                    IMX,JMX,NCL,NID,NIDP,IMX,IMN
      COMMON /OFFS/       IZOFF,FZOFF
      COMMON /FIXED/      NEC,IL(2),IC(256),IB(256),NPA,IEMAX,ICVNT(2),
     +                    IBVNT(2),ATR(2560),IAT,NIDF,KDD(2),
     +                    IDSLOT(512)
      COMMON /ICON/       SETET,RNGDLY,SETEI,RANG(382),IHGT(382)
      COMMON /INSUB/      BEGINT,ENDT,DELTR,SCON,ICOMP,
     +                    DAYM,BDAY,EDAY
      COMMON /AZM/        AZMUTH,DAZES,AZLAST,NA,ELEVAT,B,C
      COMMON /AZ2/        SAZ,CAZ,DAZ,ISCANF,NEL
      COMMON /FILTER/     IATRMN,AREAMN,CELMN(2),SVMX
      COMMON /OUTPAR/     MOUT,NOUT,NOCTR,NOTCH(30)
      COMMON /DECODE/     UP(9),HEIGHT,DLONG,DLAT
      COMMON /DATA1/      ECL(10,128),NCO,NCMX,NRJC
      COMMON /DATA5/      ESCL(10,128),NSCO,NSCMX,NSRJC
      COMMON /COUNT/      IXF,IXS
      COMMON /VEL/        IS(382),IT(382),HV(382),RV(382),RS(382)
      COMMON /CNT/         CEL,SEL,CEL2,ZMIN,ELAST,SPRM,IFXMX
      COMMON /WIND/       SVA(14,8),SVB(14,8),SVC(14,8),
     +                    SA2(14,8),SB2(14,8),SC2(14,8),
     +                    SAB(14,8),SAC(14,8),SBC(14,8),
     +                    SV2(14,8),SB(14,8),SC(14,8),NUM(14,8)
C
      EQUIVALENCE(ECL(1,1),IECL(1,1)),(ESCL(1,1),IESCL(1,1))

      PARAMETER(IZERO=0,JATMX=1856,IPMX=1920,RPD=.017453,DPO=1./45.)
C     ------------------------------------------------------------------
C
C
C
C    SET IFLAG=1 FOR INTERMEDIATE PRINT OUT TO DEVICE 6
C        GIVES IATR(ARRAY) & UP(ARRAY) VALUES
C
      PARAMETER(IFLAG=0)
C
C    SET IFLAG2=1 FOR PRINT OUT OF PEAK CELL UPDATES ON DEVICE 6
C
      PARAMETER(IFLAG2=0)
C
C    LM   NUMBER OF ATTRIBUTES TO BE COMPUTED
C
      PARAMETER(LM=9)
      PARAMETER(LMM=LM-1,IDX=LM+1)
C
C    IFM IS NUMBER OF EVENTS ABOVE THRESHOLD(TL) ON CURRENT RADIAL
C
C    ITY = 3   REFLECTIVITY EVENTS
C    ITY = 4   DOPPLER VELOCITY SHEAR EVENTS
C
      IFM=ICVNT(KLVL)
C
      NAX=NA
      NCLM=NCL-1
      LMDF=LM*NIDP
      IUMX=(NUMAX-2)/LM
```

```
      IF (LDB.GT.LDMX) LDB=LDMX
      LDBM1=LDB-1
      NUMP=2+LM*LDB
      NUMP1=NUMP+1
      IDX=NUMP-1
      IDXOF=IDX*NIDP
      IB2=1+(LDB-1)*LM
C
C    PREPARE FOR BACKGROUND WIND SUMMATION PROCEDURE
C
      IF(IIY.NE.4) GO TO 2106
      CELCAZ=CEL*CAZ
      CELSAZ=CEL*SAZ
      SEL2=SEL*SEL
      CELCAZ2=CELCAZ*CELCAZ
      CELSAZ2=CELSAZ*CELSAZ
      CELSCZ=CELCAZ*CELSAZ
      SCLCAZ=SEL*CELCAZ
      SCLSAZ=SEL*CELSAZ
      IOCT=AZMUTH*DPO+1
 2106 CONTINUE
C
C
C    NA = RADIAL COUNTER
C    ISCANF = +/-1   END OF SCAN
C
C    ZERO CELL ATTRIB ARRAYS AND CELL COUNTERS ONCE EACH SCAN
C
      IF(NA.NE.1 .OR. ISCANF.NE.0) GO TO 2109
      NOUT=0
      NMX=1
      DO 2107 I=1,NIDP
 2107 IACT(I)=0
      DO 2108 J=1,JATMX
 2108 TATR(J)=0.
C
C    ZERO CURRENT RADIAL ARRAYS
C
 2109 NGM=0
      DO 23 K=1,KMAX
   23 IPNT(K,1,C)=0
C
C                OUTER C EVENT LOOP
C
 1044 NADNF=NA.EQ.1
      JEM=IBVNT(KLVL)
      IF(IEM.LE.0)GO TO 952
      IF(IEM.GT.IEMAX)IEM=IEMAX
C
C    BEGIN NORMAL PROCESSING.
C
C    LOOP ON EACH EVENT ON CURRENT RADIAL
C    LOCATE PAKS AND SET THRESH VALUE -LDB- DOWN
C    ASSOCIATE PEAK EVENTS RADIAL  TO RADIAL AT EACH THRESH LEVEL
C
      DO 951 IE=1,IEM
      IE1=IE-1
      IECA=(IE1)*NPA
      ICEST=IC(1+IECA)
      ICESP=IC(2+IECA)
```

```
      LID1=IC(NPA+IECA)
      IF(LID1.EQ.0)GO TO 951
      IATA=LID1*IAT
      IF(.NOT.NAONE .OR. ISCANF .NE. 1) GO TO 938
      IPT=IPTA(IE,C)
      GO TO 940
C
  938 IPTA(IE,C)=0
      IPL=0
      IF(IE.EQ.1)GO TO 232
      DO 233 K=1,KMAX
  233 IPNT(K,IE,C)=IPNT(K,IE1,C)
      IPL=IDC(IE1)
  232 IP=IDC(IE)
      IF(IP.LE.IPL)GO TO 951
      IPL=IPL+1
      JE1=0
      JE2=0
C
C               FIND B EVENTS ASSOCIATED WITH C EVENTS.
C               JEM IS NO. OF EVENTS IN PREVIOUS RADIAL.
C
      IF(JEM .EQ. 0) GO TO 41
      IF(JEM.GT.IEMAX)JEM=IEMAX
C
C   LOCATE ALL JE EVENTS ASSOCIATED WITH IE EVENT
C
      DO 31 JE=1,JEM
      JEA=(JE-1)*NPA
      IF(IB(2+JEA) .LT. ICEST) GO TO 31
      IF(IB(1+JEA) .GT. ICESP) GO TO 41
      JE2=JE
      IF (JE1.EQ.0) JE1=JE
   31 CONTINUE
C
C    FIND THRESHOLDS FOR IE EVENT
C
   41 DO 51 J=1,JMXDB
   51 T(J)=0
      NTHRES=1
C
C    LOOP THROUGH ALL CONTOUR PEAKS ON CURRENT RADIAL
C
      DO 71 L=IPL,IP
      IF(L .GT.JMAX)GO TO 71
      IR1=IPCRNG(L)
      IF(IR1.LT.ICEST)GO TO 71
      IF (IR1.GT.ICESP)GO TO 712
      IU=U(IR1)
      MTT=IABS(IU)-TM+1
C
C   FIND ALL UNIQUE THRESH VALUES FOR CURRENT PEAK EVENT
C
      DO 711 K=1,LDB
      IT=MTT-K
      IF(IT.LE.0)GO TO 711
      IF(IT.GT.JMXDB)IT=JMXDB
      IF(T(IT).EQ.0)NTHRES=NTHRES+1
      T(IT)=1
  711 CONTINUE
```

```
   71 CONTINUE
C
C    ARRAY OVERFLOW
C
   72 IPSRT=0
      IF(NTHRES.GT.KMAX)IPSRT=NTHRES-KMAX
      IPT=1
C
C    STORE THE -KMAX- HIGHEST THRESHOLD LEVELS
C
      DO 91 L=1,JMXUR
      IF(I(L),LE,0,) GO TO 91
      ITA(IPT,IE,C)=L*ITM-1
      IPSRT=IPSRT-1
      IF(IPSRT.GT.0)GO TO 91
      IPT=IPT+1
91    CONTINUE
      IPT=IPT-1
      IF(IPT.GE.JR)IPT=JR
      IFTA(IE,C)=IPT
      IF(IPT.LE.0)GO TO 951
C
C    FIND ALL SEGMENTS WITHIN IE EVENT THAT ARE ENCLOSED BY CONTOUR
C
      IBGN=ICEST+1
      IND=ICESP+1
C
C    LOOP ON RANGE
C
      DO 161 I=IBGN,IND
      I1=I-1
C
C    LOOP ON THRESHOLD
C
      DO 131 K=1,IPT
      IF (U(I).EQ.-999) GO TO 141
      IF(IABS(U(I)).LE.ITA(K,IE,C)) GO TO 141
      IF (U(I1).EQ.-999) GO TO 121
      IF(IABS(U(I1)).GT.ITA(K,IE,C)) GO TO 131
C
C    START RANGE FOR SEGMENT (CONTOUR)
C
  121 IPNT(K,IE,C)=IPNT(K,IE,C)+1
      IF(IPNT(K,IE,C).LE.IMXJMX) GO TO 1211
C
      IF(PROVER) WRITE(7,1212)ITY,K,IE
 1212 FORMAT(2X,'NUMBER OF SEGMENTS EXCEEDS IMX',5I10,/5I10)
C
      IPNT(K,IE,C)=IMXJMX
1211  IPE=IPNT(K,IE,C)
      IREG=I1
      IPEK=IPE+(K-1)*JMAX
      IP1(IPEK,C)=IREG
      IP3(IPEK,C)=0
  131 CONTINUE
      GO TO 161
C
C    SUM BACKGROUND WIND BELOW LOWEST INCLUSION THRESHOLD
C
  141 CONTINUE
```

```
      IF(K.GT.2 .OR. ITY.NE.4 .OR. RV(I).LT.-990.) GO TO 142
      IF(RANG(I).LT.25. .OR. RANG(I).GT.150.) GO TO 142
      RADV=RV(I)
      JF IHGT(I)
      SVA(J,IOCT)=SVA(J,IOCT)+SEL*RADV
      SVB(J,IOCT)=SVB(J,IOCT)+CELCAZ*RADV
      SVC(J,IOCT)=SVC(J,IOCT)+CELSAZ*RADV
      SA2(J,IOCT)=SA2(J,IOCT)+SEL2
      SB2(J,IOCT)=SB2(J,IOCT)+CELCAZ2
      SC2(J,IOCT)=SC2(J,IOCT)+CELSAZ2
      SV2(J,IOCT)=SV2(J,IOCT)+RADV*RADV
      SAB(J,IOCT)=SAB(J,IOCT)+SCLCAZ
      SAC(J,IOCT)=SAC(J,IOCT)+SCLSAZ
      SBC(J,IOCT)=SBC(J,IOCT)+CELSCZ
      SB(J,IOCT)=SB(J,IOCT)+CELCAZ
      SC(J,IOCT)=SC(J,IOCT)+CELSAZ
      NUM(J,IOCT)=NUM(J,IOCT)+1
  142 CONTINUE
C
C   END RANGE FOR SEGMENT
C
      DO 151 KL=K,IPT
      IF (U(I1).EQ.-999) GO TO 161
      IF(IABS(U(I1)).LE.ITA(KL,IE,C)) GO TO 161
      IPE=IPNT(KL,IE,C)
      IREG=I1
      IPEK=IPE+(KL-1)*JMAX
      IP2(IPEK,C)=IREG
  151 CONTINUE
  161 CONTINUE
C
C   ASSOCIATE PEAK EVENTS AT EACH THRESHOLD LEVEL -IPT-
C
C   LOOP ON THRESHOLD -KC- HIGHEST TO LOWEST
C
940   DO 941 LC=1,IPT
      KC=IPT-LC+1
      IF(KC.LE.0)GO TO 941
      ITHRESH=ITA(KC,IE,C)
      KCC=(KC-1)*JMAX
      NPC=IPNT(KC,IE,C)
      NPL=0
      IF(IE.GT.1)NPL=IPNT(KC,IE1,C)
      IF(NPC.LE.NPL)GO TO 941
      NPL=NPL+1
C
C   LOOP ON IE EVENT SEGMENTS ENCLOSED BY KC THRESHOLD CONTOUR
C
      DO 931 IPE=NPL,NPC
      IPEKC=IPE+KCC
      IHRM=IP1(IPEKC,C)
      IHR=IHRM+1
      IHD=IP2(IPEKC,C)
      K=KC+1
      KJMAX=KC*JMAX
      NPK=0
      IATM=0.
      LPL=0
      LPE=IPNT(K,IE,C)
```

```
      IF(IE.GT.1)LPL=IPNT(K,IE1,C)
      LPL=LPL+1
      IF(LPE.LT.LPL.OR.K.GT.IPT)GO TO 193
      DO 191 L=LPL,LPE
      LK=L+KJMAX
      IF(IP2(LK,C).LT.IHBM) GO TO 191
      IF(IP1(LK,C).GT.IHD) GO TO 193
C
C   NPCEL IS FOR NEXT HIGHER (ENCLOSED) THRESHOLD ON C RADIAL
C
      NPCEL=IP3(LK,C)
      IF(NPCEL.LE.0)GO TO 1911
      TATM=AMAX1(TATM,TATR(NPCEL))
      IF(TATM.EQ.TATR(NPCEL))NPK=NPCEL
      IF(ABS(TATR(NPCEL)).GT.(ITHRESH+LDB))GO TO 932
191   CONTINUE
      GO TO 193
932   NPK=-NPCEL
      GO TO 193
 1911 NPK=-(NIDP+1)
C
C   ASSOCIATE CELLS ON PRIOR RADIAL, TOP DOWN
C
  193 NPK=0
      IF(NAONE) GO TO 361
      TATM=0.
      IF (JE2.EQ.0) GO TO 371
C
C   LOOP THROUGH EVENTS ON PRIOR RADIAL
C
      DO 261 JE=JE1,JE2
      JEA=(JE-1)*NPA
      IF(IB(2+JEA) .LT. IHBM)  GO TO 261
      IF(IB(1+JEA) .GT. IHD)  GO TO 3661
C
C   JE EVENT ON PRIOR RADIAL IS ASSOCIATED
C
      IPB=IPTA(JE,B)
      IF(IPB.LE.0)GO TO 261
C
C   PRIOR RADIAL, LOOP ON THRESH -KB- HIGHEST TO LOWEST
C
      DO 291 LB=1,IPB
      KB=IPB-LB+1
      KBB=(KB-1)*JMAX
      NP1=0
      IF(JE.GT.1)NP1=IPNT(KB,JE-1,B)
      NP2=IPNT(KB,JE,B)
      IF(NP2.LE.NP1)GO TO 291
      NP1=NP1+1
C
C   LOOP ON JE EVENT SEGMENTS ENCLOSED BY KB CONTOUR
C   COMPARE WITH KC CONTOUR
C
      DO 281 JPE=NP1,NP2
      JPEKB=JPE+KBB
      IF(IP2(JPEKB,B).LT.IHBM) GO TO 281
      IF(IP1(JPEKB,B).GT.IHD) GO TO 291
C
C   LPCEL IS CONTOUR THRESHOLD ON B RADIAL
```

```
C
      LPCEL=IPS(JPEKB,B)
      IF(LPCEL.LE.0)GO TO 281
      IF(ITHRESH.LE.ITA(KB,JE,B))GO TO 282
      IF(ITA(KB,JE,B)+1.LT.TATR(LPCEL))GO TO 281
C
C   FIND PEAK THRESHOLD LEVEL
C
282   TATM=AMAX1(TATM,TATR(LPCEL))
      IF(TATM.NE.TATR(LPCEL))GO TO 281
      MPK=LPCEL
      KPM=KB
      JBM=JE
281   CONTINUE
291   CONTINUE
251   CONTINUE
C
C   END OF JE COMPARE LOOP
C
3661  IF(MPK.EQ.0)GO TO 371
      IF(ABS(TATR(MPK)).GT.ITHRESH+LDB)MPK=-MPK
      GO TO 421
371   DO 194 I=IHB,IHD
      IF(HB(I).EQ.-999)GO TO 194
      IF(IABS(HB(I)).LE.ITHRESH)GO TO 194
      IF(NPK.EQ.0)GO TO 931
      IF(NPK.GT.0)GO TO 366
      GO TO 3662
  194 CONTINUE
C
C                HAVE B COMPARE WITHIN RANGE
C
  3661 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
      IF(NPK.LE.0.OR.NPK.GT.NMX)GO TO 3662
C
C                NO PRIOR RADIAL FOR COMPARISON, INCREMENT NPCEL
C
      NPCEL=NPK
359   INDX=TATR(NPCEL)-ITHRESH-1
      IF(INDX.GE.LDB.OR.INDX.LE.0)GOTO 366
      IN=1+INDX*LM
      INX=IDX+INDX*LM
      NPIN=NPCEL+(IN-1)*NIDP
      MPIN=NPCEL+(IN-LMM-1)*NIDP
      IF(TATR(NPIN).NE.0. .OR. NAONE) GO TO 3921
      IF(TATR(MPIN).LE.0.)GO TO 366
      MPC=NPCEL
      NPCEL=TATR(MPIN)
      IF(MPC.EQ.NPCEL.OR.NPCEL.GT.NMX)GO TO 366
      GO TO359
 3921 IPS(IPEKC,C)=NPCEL
      IF(NAONE .AND. ISCANF.EQ.1) GO TO 366
      IN2=NPCEL+IN*NIDP
```

```
      IF(IATR(IN2).EQ.-999.) GO TO 419
      IN3=IN2+NIDP
      IN4=IN3+NIDP
      IN5=IN4+NIDP
      IN6=IN5+NIDP
      IN7=IN6+NIDP
      IN8=IN7+NIDP
      IN9=IN8+NIDP
      IST=IHB
      ISP=IHD
      DO 411 I=IST,ISP
      R=RANDU(ISEIRI*(FLOAT(I+1).5)
      RD=R*DAZ
      RU=RD*ABS(FLOAT(U(I)))
      RU2=R*RU
      IATR(IN2)=IATR(IN2)+RD
      IATR(IN3)=IATR(IN3)+RU
      IATR(IN4)=IATR(IN4)+SAZ*RU2
      IATR(IN5)=IATR(IN5)+CAZ*RU2
      IF(V(I).EQ.-999) GO TO 411
      IATR(IN6)=IATR(IN6)+RD*IABS(V(I))
      IATR(IN7)=IATR(IN7)+RV(I)*RD
      IATR(IN8)=IATR(IN8)+RS(I)*RD
      IATR(IN9)=IATR(IN9)+RD
  411 CONTINUE
  419 NIX=NPCEL+(INX-1)*NIDP
      IATR(NIX)=SIGN(FLOAT(NA),IATR(NIX))
      IF(NAONE) IATR(NIX)=SIGN(IATR(NIX),-1.0)
      IF(IST.LE.2..OR.ISP.GE.IMX)IATR(IN2)=-999.
      GO TO 366
3662  NPCEL=-NPK
366   IF(NPCEL.GT.NMX.OR.NPCEL.LE.0)GO TO 931
      IMDX=IATR(NPCEL)-ITHRESH-1
C
C    COMBINE PRIOR(LPCEL) WITH CURRENT(NPCEL) EVENT AT THIS LEVEL
C          COMBINE BY SETTING AREA AS POINTER AND IDX TO NA = 0
C
      IF(LPE.LT.LPL.OR.K.GT.IPT)GO TO 931
      DO 365 L=LPL,LPE
      LK=L+KJMAX
      IF(IP2(LK,C).LT.IHBM) GO TO 365
      IF(IP1(LK,C).GT.IHD) GO TO 931
      LPCEL=IP3(LK,C)
      IF(LPCEL.LE.0.OR.LPCEL.GT.NMX)GO TO 365
      LPX=LPCEL+LMDP
      IF(IATR(LPX).EQ.0.)GO TO 365
      IF(NPCEL.EQ.LPCEL)GO TO 365
      INDX=IATR(LPCEL)-ITHRESH-1
      IF(INDX.GE.LDB)GO TO 365
      IF(INDX.LE.0)INDX=0
      IND=IDX+INDX*LM
      LPND=LPCEL+(IND-1)*NIDP
      IF(IATR(LPND).EQ.0.)GO TO 365
      IND=INDX+1
      IPG=0
      DO 3663 J=IND,LDB
      IN=(J-1)*LM+1
      LPIN=LPCEL+(IN+LM-1)*NIDP
      IF(IATR(LPIN).EQ.NA)IPG=IPG+1
      DO 3663 I=1,LM
```

```
      LPLIN=LPCEL+(IN+1)*NIDP
3663  IATR(LPLIN)=0.
      IF(IPG.EQ.0.OR.IE.LE.1)GO TO 3664
      DO 3665 I=1,IE
      IPTI=IPTA(I,C)
      IF(IPTI.LE.0)GO TO 3665
      DO 3666 KT=1,IPTI
      NPCL=0
      IF(I.GT.1)NPCL=IPNT(KT,I-1,C)
      NPCT=IPNT(KT,I,C)
      IF(NPCT.LE.NPCL)GO TO 3666
      NPCL=NPCL+1
      KTT=(KT-1)*JMAX
      DO 3667 LP=NPCL,NPCT
      LPKT=LP+KTT
      IF(LPCEL.NE.IP3(LPKT,C))GO TO 3667
      INDXT=IATR(NPCEL)-ITA(KT,I,C)-1
      IF(INDXT.LT.LDB)GO TO 3668
 3669 IP3(LPKT,C)=IZERO
      GO TO 3667
3668  IF(IMDX.GE.LDB)GO TO 3669
      IP3(LPKT,C)=NPCEL
3667  CONTINUE
3666  CONTINUE
3665  CONTINUE
      IPG=0
3664  IF(IMDX.GE.LDB)GO TO 365
      IACT(LPCEL)=-NPCEL
      LPLX=LPCEL+(1+INDX*LM)*NIDP
      IATR(LPLX)=NPCEL
      IF(INDX.NE.0)GO TO 365
      IACT(LPCEL)=-NIDP-1
365   CONTINUE
      GO TO 931
C
C               COMBINE NPCEL AND LPCEL, PEAK VALUES EQUAL
C
C
C                COMBINE WITH B RADIAL CELLS
C
421   IF(MPK.LE.0)GOTO 422
      IF(NPK.LT.0)GO TO 3662
      NGM=0
      LPCEL=MPK
      LPX=LPCEL+LMDP
      IF(ABS(IATR(LPX)).EQ.NA.AND.NPK.EQ.0.AND.ITA(KC,IE,C).GT.
     *ITA(KBM,JBM,B))GO TO 485
      INDX=IATR(LPCEL)-ITHRESH-1
      IMDX=INDX
      IF(NPK.GT.0)IMDX=IATR(NPK)-ITHRESH-1
      IF(IMDX.LE.INDX)GO TO 4212
      NGM=1
      NPCEL=NPK
      IND=INDX
      INDX=IMDX
      IMDX=IND
      GO TO 4213
4212  IF(INDX.LT.0)GO TO 481
      NPCEL=LPCEL
C
```

```
C     CONDENSE WITH B = RADIAL, C=LEVEL LOWER
C
 4221 IF(INDX.GE.IDO)GO TO 4221
      IN=INDX*LM
 5312 NPIX=NPCEL+(IN+LM)*NIDP
      NPIN=NPCEL+(IN+1)*NIDP
      IF(TATR(NPIX).NE.0.)GO TO 5311
      IF((TATR(NPIN).LE.0.).AND.(NGM.EQ.0))GO TO 4221
      IF(NGM.NE.1)GO TO 5312
 5313 IM=IMDX*LM
      IF(IM.LT.0)GO TO 5311
      IF(LPCEL.LE.0.OR.LPCEL.GT.NMX)GO TO 422
      LPIX=LPCEL+(IM+LM)*NIDP
      LPIM=LPCEL+(IM+1)*NIDP
      IF(TATR(LPIX).NE.0.)GO TO 5311
      IF(TATR(LPIM).GT.0.)GO TO 5313
      LPCEL=NPCEL
      GO TO 4221
5313  LPCEL=TATR(LPIM)
      IF(LPCEL.EQ.NPCEL.OR.LPCEL.GT.NMX)GO TO 4221
      IMDX=TATR(LPCEL)-ITHRESH-1
      GO TO 5314
5312  NPCEL=TATR(NPIN)
      IF(NPCEL.LE.0.OR.NPCEL.GT.NMX)GO TO 4221
      INDX=TATR(NPCEL)-ITHRESH-1
      GO TO 4213
 5311 IF3(IPERC,C)=NPCEL
      NPIN=NPCEL+(IN+1)*NIDP
      IF(TATR(NPIN).EQ.-999.)GO TO 8012
      IN3=NPIN+NIDP
      IN4=IN3+NIDP
      IN5=IN4+NIDP
      IN6=IN5+NIDP
      IN7=IN6+NIDP
      IN8=IN7+NIDP
      IN9=IN8+NIDP
      IST=IHB
      ISP=IHD
      DO 531 I=IST,ISP
      R=RANDLY+SETRI*(FLOAT(I-1)-.5)
      RD=R*DAZ
      RU=RD*ABS(FLOAT(U(I)))
      RU2=R*RU
      TATR(NPIN)=TATR(NPIN)+RD
      TATR(IN3)=TATR(IN3)+RU
      TATR(IN4)=TATR(IN4)+SAZ*RU2
      TATR(IN5)=TATR(IN5)+CAZ*RU2
      IF(V(I).EQ.-999) GO TO 531
      TATR(IN6)=TATR(IN6)+RD*IABS(V(I))
      TATR(IN7)=TATR(IN7)+RV(I)*RD
      TATR(IN8)=TATR(IN8)+RS(I)*RD
      TATR(IN9)=TATR(IN9)+RD
  531 CONTINUE
 8012 NPIX=NPCEL+(IN+LM)*NIDP
      TATR(NPIX)=SIGN(FLOAT(NA),TATR(NPIX))
      IF(NAONE) TATR(NPIX)=SIGN(TATR(NPIX),-1.0)
      IF(IST.LE.2.OR.ISP.GE.IMX)TATR(NPIN)=-999.
      LPCEL=NPCEL
      GO TO 4221
C
```

```
C                COMBINE WITH R-RADIAL, C-LEVEL HIGHER
C
C                IF FIRST COMBINE, AREA=0, IF SECOND OR HIGHER, AREA=-1.
C                TEST AREA TO ESTABLISH NEW NUMBERS
C
  422  INDX=INDX
       IND=NUMP-1
       INDP=IDB
       INS=2
       IPG=0
       IATR(LPCEL)=ITHRESH+1
       LPMP=LPCEL+LDXNP
       IATR(LPMP)=ABS(ATR(IATA))
       IF(INDX.GE.LDB)GO TO 482
       IND=IDB-INDX
       DO 4832 I=INDX,LDBM
       LPXM=LPCEL+(I+1)*LMDP
       IF(IATR(LPXM).EQ.NA)IPG=IPG+1
4832   CONTINUE
       DO 483 I=1,IND
       DO 483 J=1,LM
       IN=1+J+(IDB-I)*LM
       IM=1+J+(IND-I)*LM
       LPIN=LPCEL+(IN-1)*NIDP
       LPIM=LPCEL+(IM-1)*NIDP
483    IATR(LPIN)=IATR(LPIM)
       IND=INDX*LM+1
       INDP=INDX
482    DO 4835 I=1,LDB
       LPXM=LPCEL+I*LMDP
       IF(ABS(IATR(LPXM)).EQ.NA)IPG=IPG+1
4835   CONTINUE
       DO 484 I=INS,IND
       LPI=LPCEL+(I-1)*NIDP
484    IATR(LPI)=0.
       DO 4841 I=1,INDP
       LPXM=LPCEL+I*LMDP
4841   IATR(LPXM)=NA
       IF(IPG.EQ.0.OR.IE.LE.1)GO TO 488
       DO 4831 I=1,IE
       IPTT=IPTA(I,C)
       IF(IPTT.LE.0) GO TO 4831
       DO 4833 KT=1,IPTT
       NPCL=0
       IF(I.GT.1) NPCL=IPNT(KT,I-1,C)
       NPCT=IPNT(KT,I,C)
       IF(NPCT.LE.NPCL)GO TO 4833
       NPCL=NPCL+1
       KTT=(KT-1)*JMAX
       DO 4834 LP=NPCL,NPCT
       LPKT=LP+KTT
       IF(LPCEL.NE.IP3(LPKT,C))GO TO 4834
       INDXT=IATR(LPCEL)-IIA(KT,I,C)-1
       IF(INDXT.LT.LDB)GO TO 4834
       IP3(LPKT,C)=IZERO
4834   CONTINUE
4833   CONTINUE
4831   CONTINUE
       IPG=0
  488  IN=0
```

```
      IF(LPCEL.LE.0.OR.LPCEL.GT.NMX)GO TO 931
      LPDX=LPCEL-NIDP
      IATR(LPDX)=NA
      IP3(IPENC,C)=LPCEL
      NPCEL=LPCEL
      NGM=0
      GO TO 512
485   DO 486 I=1,NIDP
      IF(IACT(I).EQ.0)GO TO 487
  486 CONTINUE
C
C     NO EMPTY SLOT, OVERWRITE LAST SLOT
C
      IF(PROVER) WRITE(7,644)
      I=NIDP
  487 LPCEL=I
      IACT(I)=1
      NMX=MAXO(NMX,I+1)
      IF(NMX.GT.NIDP)NMX=NIDP
      IATR(LPCEL)=ITHRESH+1
      LPMP=LPCEL+LDXNP
      IATR(LPMP)=ABS(ATR(IATA))
      GO TO 488
  427 LPCEL=IABS(MPK)
C
C     TEST LPCEL AND ESTABLISH ADDRESS FOR AREA POINTER AND NA
C
 4221 IF(LPCEL.GT.NMX.OR.LPCEL.LE.0)GO TO 3662
C     THRESH LEVEL
      IMDX=IATR(LPCEL)-ITHRESH-1
      IF(IMDX.LT.0)GO TO 632
      LD=1+IMDX*LM
      LDLM=LD+LM
      IF(LDLM.LE.0.OR.LDLM.GT.NUMAX) GO TO 632
      LPLD=LPCEL+LD*NIDP
      LDNP=LPCEL+(LDLM-1)*NIDP
      LPCELL=IATR(LPLD)
      IF(LPCELL.GT.NMX) GO TO 632
C
C     FLUSH NPCEL SET
C
      DO 441 JE=JE1,JE2
      JEA=(JE-1)*NPA
      IF(IB(2+JEA) .LT. IHBM)  GO TO 441
      IF(IB(1+JEA) .GT. IHD)  GO TO 632
      IPB=IPTA(JE,B)
      IF(IPB.LE.0)GO TO 441
      DO 471 LB=1,IPB
      KB=IPB-LB+1
      KBB=(KB-1)*JMAX
      MPB=IPNT(KB,JE,B)
      MPL=0
      IF(JE.GT.1)MPL=IPNT(KB,JE-1,B)
      IF(MPB.LE.MPL)GO TO 471
      MPL=MPL+1
      DO 461 JPE=MPL,MPB
      JPEKB=JPE+KBB
      IF(IP2(JPEKB,B).LT.IHBM) GO TO 461
      IF(IP1(JPEKB,B).GT.IHD) GO TO 471
      NPCEL=IP3(JPEKB,B)
```

```
      IF(NPCEL.LE.0.OR.NPCEL.GT.NMX)GO TO 461
      IF(LPCEL.EQ.NPCEL)GO TO 461
      IF(ITA(KB,JE,B).NE.ITA(KC,IE,C))GO TO 461
C
C        COMBINE AT TB=TC LEVEL
C
C.2   INDX=TATR(NPCEL)+THRESH+1
      IF(INDX.GE.LDD)GO TO 461
      IF(INDX.LT.0)GO TO 8511
      IF(INDX.LT.LDD)GO TO 861
8511  DO 512 I=1,LM
      NDI=ND+I
      NPND=NPCEL+(NDI-1)*NIDP
8512  TATR(NPND)=0.
      IF3(JPEKB,B)=IZERO
      GO TO 461
 8511 IF3(JPEKB,B)=IZERO
      DO 8512 J=2,NUMP
      NPJ=NPCEL+(J-1)*NIDP
8512  TATR(NPJ)=0.
      IACT(NPCEL)=-(NIDP+1)
      GO TO 461
  861 ND=1+INDX*LM
      IF(NPCEL.LE.0 .OR. NPCEL.GT.NMX) GO TO 8612
      NDLM=ND+LM
      IF((NDLM).LE.0 .OR. (NDLM).GT.NUMAX) GO TO 8612
      GO TO 8611
 8612 WRITE(6,8210)LPCEL,NPCEL,LD,LM,ND,NA,JPE,KB,IPE,KC
 8210 FORMAT(10I10)
      GO TO 461
 8611 NDNP=NPCEL+(NDLM-1)*NIDP
      IF(TATR(LDNP).NE.0..AND.TATR(NDNP).NE.0.)GO TO 8911
      IF(TATR(LDNP).EQ.0. .AND.LPCELL.LE.0)GO TO 851
      IF(TATR(LDNP).GT.0.)GO TO 8912
      LPCEL=LPCELL
      GO TO 4221
 8912 NPND=NPCEL+ND*NIDP
      IF(TATR(NDNP).EQ.0..AND.TATR(NPND).LE.0.)GO TO 8913
      NPCEL=TATR(NPND)
      IF(NPCEL.LE.0.OR.NPCEL.GT.NMX.OR.NPCEL.EQ.LPCEL)GO TO 461
      IF3(JPEKB,B)=NPCEL
      GO TO 502
 8913 LD1=LD+1
      DO 8914 I=1,LM
      LPLDI=LPCEL+(LD1+I)*NIDP
8914  TATR(LPLDI)=0.
      IF3(JPEKB,B)=IZERO
      GO TO 4221
 8911 IBNDRY=0
      LPLD=LPCEL+LD*NIDP
      NPND=NPCEL+ND*NIDP
      IF(TATR(LPLD).EQ.-999..OR.TATR(NPND).EQ.-999.)
     X IBNDRY=1
      DO 891 I=1,LMM
      LDI=LPCEL+(LD+I-1)*NIDP
      NDI=NPCEL+(ND+I-1)*NIDP
      IF(IBNDRY.EQ.0)TATR(LDI)=TATR(NDI)+TATR(LDI)
      TATR(NDI)=0.
891   CONTINUE
```

```
      IF(IBNDRY(J,C,1)ATR(IPID)=-999.
      NDNP=NPCEL+(NDHLM-1)*NIDP
      ATR(NDNP)=0.
      IATR(NPNB)=IPCEL
      IACT(NPCEL)=IPCEL
      IP3(IPEKC,C)=IPCEL
441   CONTINUE
471   CONTINUE
481   CONTINUE
      IF(NPK.LE.0)GO TO 3662
      NPCEL=IPCEL
      GO TO 366
C
C              UNASSOCIATED
C
  631 IF(NAONE .AND. ISCANF .NE.0) GO TO 630
      DO 642 J=1,NIDP
      IF(IACT(J).EQ.0)GO TO 643
642   CONTINUE
C
C   NO EMPTY SLOT, OVERWRITE LAST SLOT
C
      IF(PROVER) WRITE(7,644)
 644  FORMAT(5X,' TOO MANY CELLS')
      J=NIDP
      GO TO 643
C
630   NCELT=NCELT+1
      IP3(IPEKC,C)=NCELT
      GO TO 931
643   NPCEL=J
      IACT(J)=1
      NMX=MAX0(NMX,J+1)
      IF(NMX.GT.NIDP)NMX=NIDP
      IP3(IPEKC,C)=NPCEL
      DO 671 I=1,NUMP
      NPI=NPCEL+(I-1)*NIDP
      ATR(NPT)=0.0
671   CONTINUE
  591 ATR(NPCEL)=ITHRESH+1
      NPMP=NPCEL+LDXNP
      ATR(NPMP)=ABS(ATR(IATA))
      IST=IHB
      ISP=IHD
      NP2=NPCEL+NIDP
      NP3=NP2+NIDP
      NP4=NP3+NIDP
      NP5=NP4+NIDP
      NP6=NP5+NIDP
      NP7=NP6+NIDP
      NP8=NP7+NIDP
      NP9=NP8+NIDP
      DO 621 I=IST,ISP
      R=RANDLY+SETRI*(FLOAT(I-1)-.5)
      RD=R*DAZ
      RU=RD*ABS(FLOAT(U(I)))
      RU2=R*RU
      ATR(NP2)=RD+ATR(NP2)
      ATR(NP3)= RU+ATR(NP3)
      ATR(NP4)= SAZ*RU2+ATR(NP4)
```

```
      TATR(NP5)=TATR(NP5)+ CAZ*RU2
      IF(V(I).EQ.-999) GO TO 621
      TATR(NP6)=TATR(NP6)+RD*TABS(V(I))
      TATR(NP7)=TATR(NP7)+RV(I)*RD
      TATR(NP8)=TATR(NP8)+RS(I)*RD
      TATR(NP9)=TATR(NP9)+RD
  621 CONTINUE
      NPIDX=NPCELHLMDP
      TATR(NPIDX)=NA
      IF(NAONE) TATR(NPIDX)=-TATR(NPIDX)
      IF(TST.LE.2.OR.ISP.EQ.IMX)TATR(NP2)=-999.
C
C   END -IPE- CONTOUR SEGMENT LOOP
C
  931 CONTINUE
C
C   END -KC- THRESHOLD LOOP
C
  941 CONTINUE
C
C   END -IE- EVENT ASSOCIATION LOOP
C
  951 CONTINUE
```

```
C
C     CLEAN UP IATR AND IC ARRAYS
C     TEST EACH SEGMENT, ZERO ARRAYS OF COMBINED OR CLOSED SEGMENTS
C
C     LOOP ON EACH CONTOUR LEVEL
C
        DO 9512 I=1,NMX
        IF(IACT(I).EQ.0.OR.IATR(I).LE.0.)GO TO 9512
        IF(IACT(I).GE.0)GO TO 9611
C
C     LOOP ON CURRENT RADIAL EVENTS  IE
C
        DO 9613 IE=1,IEM
        IPT=IPTA(IE,C)
        IF(IPT.LE.0)GO TO 9613
C
C     LOOP ON THRESHOLDS ON IE EVENT
C
        DO 9618 KC=1,IPT
        NPC=IPNT(KC,IE,C)
        NPL=0
        IF(IE.GT.1)NPL=IPNT(KC,IE-1,C)
        IF(NPC.LE.NPL)GO TO 9618
        NPL=NPL+1
C
C     MATCH IE SEGMENT WITH LEVEL BEING TESTED
C
        KCC=(KC-1)*JMAX
        DO 9619 IPE=NPL,NPC
        IPEKC=IPE+KCC
        IF(I.NE.IP3(IPEKC,C))GO TO 9619
        IF(IACT(I).LT.-NIDP)GO TO 9614
        INDX=IATR(I)-ITA(KC,IE,C)-1
        IF(INDX.GE.LDB.OR.INDX.LT.0) GO TO 9619
        IXDX=I+(INDX+1)*LMDP
        IF(IATR(IXDX).NE.0.)GO TO 9619
```

```
      IIXM=II+(I-1)*LMDF
      IF(IATR(IIXM).NE.-IACT(I))GO TO 9614
      IF(IFLAG.EQ.1) IACT(I)
      GO TO 9619
 9614 IF(IFLAG.EQ.1)IZERO
9611  CONTINUE
9613  CONTINUE
9617  CONTINUE
      IF(IACT(I).GE.-NIDF)GO TO 9517
      IACT(I)=0
      IATR(I)=0.
      GO TO 9512
9517  DO 9513 J=1,LDB
      ILM=I+(IF(J-1)*LM)*NIDF
      IXLM=ILM+LMDF
      IF(IATR(ILM).EQ.-IACT(I).AND.IATR(IXLM).EQ.0.)
     +   GO TO 9514
9513  CONTINUE
      GO TO 9611
9514  IATR(ILM)=0.
9511  DO 9512 K=2,LDB
      IZS=0
      IIXM=I+K*LMDF
      IIXL=IIXM-LMDF
      ILM=IIXL+NIDF
      IF(IATR(IIXM).NE.0..AND.IATR(ILM).EQ.0.) IZS=1
      IF(IFIX(IATR(IIXM)).LE.(NAX-1).AND.IATR(IIXM).GT.0.
     +   .AND.IFIX(IATR(IIXL)).EQ.NAX) IZS=1
      IF(IZS.NE.1) GO TO 9612
      DO 9509 IKK=ILM,IIXM,NIDF
      IATR(IKK)=0.
 9509 CONTINUE
9612  CONTINUE
C
      IF(IFLAG.NE.1) GO TO 4568
      DO 4567 J=1,NUMF
      GOUT(J)=IATR(I+(J-1)*NIDF)
      IF(ABS(GOUT(J)).GT.999999999.) GOUT(J)=GOUT(J)/1000.
 4567 CONTINUE
      WRITE(6,998)I,IACT(I),(GOUT(J),J=1,NUMF)
C
 4568 IACT(I)=1
9512  CONTINUE
      IF(NAONE .AND. ISCANF.EQ.0) GO TO 1030
      IF (NAONE .AND. ISCANF .EQ. 1) GO TO 959
      GO TO 952
C
C   COMBINE LAST RADIAL IN SCAN TO FIRST RADIAL IN SCAN
C
959   DO 9590 IE=1,IEM
      IPT=IPIA(IE,C)
      IF(IPT.LE.0) GO TO 9590
      DO 9591 IC=1,IPT
      KC=IPT-IC+1
      KCC=(KC-1)*JMAX
      NPC=IPNT(KC,IE,C)
      NPL=0
      IF(IE.GT.1)NPL=IPNT(KC,IE-1,C)
      NPL=NPL+1
      DO 9591 IPE=NPL,NPC
```

```
      IPEAC=IPEAKC
      NPCEL=IP3(IPEAC,C)
      IF(TATR(NPCEL+LMDP).GE.0.) GO TO 9591
      IF(NPCEL .LE. 0) GO TO 9591
      LPCEL=IP3(IPEAC,B)
      IF(LPCEL .LT. 0) GO TO 9591
      INDX=IATR(LPCEL)-IATR(NPCEL)
      IF(INDX .LT. 0) GO TO 958
C
C    PEAK LPCEL .GE. PEAK NPCEL
C
      IF(INDX .GE. LDB) GO TO 955
      IND=LDB-INDX
      DO 9592 I=1,IND
      IM=1+LM+(IND-I)*LM
      NPIM=NPCEL+(IM-1)*NIDP
      IF(TATR(NPIM).GE.0.)GO TO 9592
      IN=1+LM+(LDB-I)*LM
      LPIN=LPCEL+(IN-1)*NIDP
      TATR(LPIN)=NAX-1
      NPINM=NPCEL+(1+(IND-I)*LM)*NIDP
      LPLDM=LPCEL+(1+(LDB-I)*LM)*NIDP
      IF(TATR(NPINM).EQ.-999.) TATR(LPLDM)=-999.
      IF(TATR(LPLDM).EQ.-999.)GO TO 9592
      DO 9593 J=1,LMM
      IN=1+J+(LDB-I)*LM
      IM=1+J+(IND-I)*LM
      LPIN=LPCEL+(IN-1)*NIDP
      NPIM=NPCEL+(IM-1)*NIDP
9593  TATR(LPIN)=TATR(NPIM)+TATR(LPIN)
9592  CONTINUE
9693  DO 9694 I=1,NUMP
      NPI=NPCEL+(I-1)*NIDP
9694  TATR(NPI)=0.
      IACT(NPCEL)=0
      GO TO 9591
  955 LPLDX=LPCEL+(LDX-1)*NIDP
      TATR(LPLDX)=NAX-1
      GO TO 9591
958   INDX=-INDX
      IF(INDX .GE. LDB) GO TO 9591
C
C PEAK NPCEL .GT. LPCEL
C
      IND=LDB-INDX
      DO 9691 I=1,IND
      IN=1+LM+(LDB-I)*LM
      NPIN=NPCEL+(IN-1)*NIDP
      IF(TATR(NPIN).GE.0.)GO TO 9691
      TATR(NPIN)=NAX-1
      LPIND=LPCEL+(1+(IND-I)*LM)*NIDP
      NPLDB=NPCEL+(1+(LDB-I)*LM)*NIDP
      IF(TATR(LPIND).EQ.-999.) TATR(NPLDB)=-999.
      IF(TATR(NPLDB).EQ.-999.)GO TO 8019
      DO 9692 J=1,LMM
      IN=1+J+(LDB-I)*LM
      IM=1+J+(IND-I)*LM
      NPIN=NPCEL+(IN-1)*NIDP
      LPIM=LPCEL+(IM-1)*NIDP
9692  TATR(NPIN)=TATR(LPIM)+TATR(NPIN)
```

```
9[illegible]    ON END
                [illegible]
9915    IATR(NPCEL+IL0XNP)=IATR(IPCEL+IL0XNP)
        NPCEL=LPCEL
        GO TO 9[illegible]
9921    CONTINUE
 9[illegible]  CONTINUE
C
C     END OF CLEAR UP AND END AROUND ASSOCIATION

   [illegible]  DO 971 I=1,NMX
        IL0=(I-1)*NTDP
        IF(IGR(I).EQ.0)GO TO 991
        IF(IATR(IL0X).EQ.0.)GO TO 9912
        IF(ABS(IATR(IL0X)).EQ.MAX-1)GO TO 971
        GO TO 991
C
C       CHECK BACKGROUND COMING DOWN
C
   971  INBR=0
        ITERM=1
        DO 9716 J=1,LOBM
        IJLM=1+(I+(J-1)*LM)*NTDP
        IJLM3=IJLM+NTDP
        IF(IATR(IJLM).LE.0..OR.IATR(IJLM3).EQ.0.)GO TO 9902
 9716   CONTINUE
        JEMM=MAXO(1,JEM)
        DO 9711 J=1,JEMM
        NPAJ=J*NPA
        IID1=IB(NPAJ)
        IATA=IID1*IAT
        KNIDB=0
        KNIDT=ABS(ATR(IATA))
 9901   IF(KNIDT.LE.0.OR.KNIDT.GT.KNID(KLVL)) GO TO 9800
        KNIDB=KNIDA(KLVL,KNIDT)
        IF(KNIDT.EQ.KNIDB) GO TO 9800
        KNIDT=KNIDB
        GO TO 9801
 9800   INUMP=I+LOXNP
        KNIDY=0
        KNIDX=ABS(IATR(INUMP))
 9803   IF(KNIDX.LE.0.OR.KNIDX.GT.KNID(KLVL)) GO TO 9802
        KNIDY=KNIDA(KLVL,KNIDX)
        IF(KNIDX.EQ.KNIDY) GO TO 9802
        KNIDX=KNIDY
        GO TO 9803
 9802   IF(KNIDB.NE.KNIDY) GO TO 9711
        IPB=IPTA(J,B)
        DO 9717 K=1,IPB
        ITHRESH=ITA(K,J,B)
        IF((IATR(I)-ITHRESH).NE.LDB)GO TO 9717
C
        NP=IPNT(K,J,B)
        NL=0
        IF(J.GT.1)NL=IPNT(K,J-1,B)
        NL=NL+1
        KJMAX=(K-1)*JMAX
        DO 9713 N=NL,NP
        NK=N+KJMAX
C
```

```
C     REJECT ON PEAK THRESHOLD
C
        ITERM=2
        IF(I.NE.IF3(NK,B))GO TO 9713
        INBR=INBR+1
        IST=IF1(NK,B)
        ISP=IF2(NK,B)+1
        DO 9715 I=IST,ISP
        IF(U(I).EQ.-999)GO TO 9715
        IF(IABS(U(I)).GT.ITHRESH) GO TO 9982
 9715 CONTINUE
9713  CONTINUE
 9717 CONTINUE
9711  CONTINUE
        ITERM=3
        IF(INBR.EQ.0)GO TO 9982
        IF(IATR(IIDX).LT.0.)GO TO 991
        ITERM=4
        IID2=IIID2+NTIDF
C
C     REJECT IF AREA TOO SMALL
C
        IF(IATR(IID2).LE.IATRMN)GO TO 9982
        JNCL=(NCELL-1)*LM
C
C     DECLARE A CELL
C
        DO 981 J=1,LMM
        JN=(J-1)*NTIDF+IID2
        UF(J)=IATR(JN)
981     CONTINUE
C
C     TEST PEAK PARAMETER AND AREA FOR A VALID CELL
C
        IF(UP(2).EQ.0..OR.UP(1).LE.0.) GO TO 727
        DIVV=0.
        IF(UP(8).GT.0) DIVV=1./(UP(8)*VQUANT)
C
C     REFLECTIVITY CELL (ITY=3) OR SHEAR CELL (ITY=4)
C
        IF(ITY.EQ.3) GO TO 721
        IF(ITY.EQ.4) GO TO 722
        GO TO 727
C
C     STORE REFL CELL ATTRIBS IN ECL(ARRAY)
C
  721 REFL=UP(2)/UP(1)-FZOFF
        SHEAR=UP(5)*DIVV
        CALL PKCELL(ECL,IECL,REFL,SHEAR,IXR,KNIDY,NRJC,NCO)
C
        IF(IFLAG2.NE.1) GO TO 726
        WRITE(6,7720)
 7720 FORMAT(1X,'REFLECTIVITY')
        WRITE(6,7721) IXR,ECL(1,IXR),ECL(2,IXR),ECL(3,IXR),ECL(4,IXR),
     +ECL(5,IXR),ECL(6,IXR),ECL(7,IXR),ECL(8,IXR)
 7721 FORMAT(14X,I3,4F6.1,I5,3F6.1)
        GO TO 726
C
C     STORE SHEAR CELL ATTRIBS IN ESCL(ARRAY)
C
```

```
C     ***** FIND CELL CENTERS FOR EZOUT
C     ***** FOR CONTOUR PLOTS *****
      CALL CNCEL (NCELL,ITESCL,KEEL,SHEAR,IXS,KNTUY,NSEGB,NSEGO)
C
      IF(IFLAG.NE.1) GO TO 726
      WRITE(6,7225)
 7225 FORMAT(1X,'TAN= SHEAR')
      WRITE(6,7226) IXS,ESCL(1,IXS),ESCL(2,IXS),ESCL(3,IXS),ESCL(4,IXS),
     *ESCL(5,IXS),ESCL(6,IXS),ESCL(7,IXS),ESCL(8,IXS)
C
C    OUTPUT CELL POSITION ON CONTOUR PLOT
C
  726 CONTINUE


  727 CONTINUE
C
      IF(IFLAG.EQ.1)WRITE(6,9910)NCELL,(UP(J),J=1,LM)
9910  FORMAT(1X,'HCC',I4,4X,9F13.2)
C
      ITERM=5
      NCELL=NCELL+1
      IF(NCELL.LE.NID) GO TO 9982
C
C    NO EMPTY SLOT, OVERWRITE LAST SLOT
C
      IF(NOUT.EQ.0.AND.PROVER) WRITE(7,963)
  963 FORMAT(' HELP -- TOO MANY PEAK CELLS, NID EXCEEDED')
      ITERM=6
      NOUT=NOUT+1
      NCELL=NID
      GO TO 9982
C
9912  DO 9913 J=1,LDB
      INDP1=1+J*LM
      IINDP=I+(INDP1-1)*NIDP
      IF(ABS(TATR(IINDP)).EQ.NA)GO TO 991
      IF(TATR(IINDP).LT.0..AND.TATR(IINDP).NE.-999.)GO TO 991
9913  CONTINUE
      ITERM=7
C
9982  CONTINUE
      IF(IFLAG.NE.1) GO TO 5679
      DO 5678 J=1,NUMP
      GOUT(J)=TATR(I+(J-1)*NIDP)
      IF(ABS(GOUT(J)).GT.999999999.) GOUT(J)=GOUT(J)/1000.
 5678 CONTINUE
      WRITE(6,999)ITERM,I,IACT(I),(GOUT(J),J=1,NUMP)
999   FORMAT(1X,2HXX,I1,I4,I6,F7.2,8F13.2,/,(20X,8F13.2))
998   FORMAT(3X,I4,I6,F7.2,8F13.2,/,(20X,8F13.2))
C
C    CLEAR TEMPORARY CELL STORAGE ARRAYS
C
 5679 DO 982 J=1,NUMP
      IJA=I+(J-1)*NIDP
982   TATR(IJA)=0.
      IACT(I)=0
991   CONTINUE
1030  CONTINUE
C
```

```
C     RESET AND CLEAR FOR END AROUND TESTING
C
      IF(ISCANE.NE.1 .OR. NAONE) GO TO 1040
      NA=1
      NOEL=1
      DO 400 I=1,IEMX
      IE3(I,B)=IE3(I,C)
  400 IE3(I,C)=0
      GO TO 1044
C
 1040 DO 1 I=2,NCLM
      MH=-999
      IF (U(I-1).NE.-999) MH=IABS(U(I-1))
      IF (U(I).NE.-999) MH=MAX0(MH,IABS(U(I)))
      IF (U(I+1).NE.-999) MH=MAX0(MH,IABS(U(I+1)))
    1 UE(I)=MH
      RETURN
      END
```

```
      SUBROUTINE TT (XX,CELL,ICELL,REFL,SHEAR,IX,KNID,NF,NCO)
C
C*****************************************************************
C
C     NAME:       TT
C     PURPOSE:    TEST [illegible]
C
C     ENTRY:      DECODE CODE, LOOP OVER CELL ATTRIBUTES OR BOTH
C                 REFLECTIVITY AND SHEAR FIELDS.  TRANSFORMS CENTROID
C                 TO A CODE AND RETURN TO A COMMON ORIGIN.
C
C     CALLED BY:
C        CALLING MODE:    PLACED
C        CALLED MODE:     NONE
C        ENTRY DATA:      REFL,SHEAR,KNID
C        RETURN DATA:     CELL,ICELL,IX,NF,NCO
C        COMMON BLOCKS:
C           DECODE:       NONE
C           READ:         DECODE,ICON,CNT,FCONST,NVELS
C
C     COMMENTS:   TESTS THAT REFLECTIVITY AND ALTITUDE
C                 VALUES ARE AT REASONABLE LEVELS.
C
C     VERSION:    1.0  DEC/VAX 11
C     DATE:       12/6/81
C     DESIGN:     GREGORSTAFSON
C     PROGMR:     GREGORSTAFSON
C
C*****************************************************************
C
      DIMENSION CELL(10,128),ICELL(10,128)
     1          ,REFL(128),SHEAR(128),KNID(128)
      COMMON /DECODE/ UF(9),HEIGHT,DLONG,DLAT
      COMMON /ICON/   SETRI,RNGDLY,SETEL,RANG(382),IHGT(382)
      COMMON /CNT/    COSPHI,SINEL,COSPHI2,ZMIN,ELAST,SPRM,TFXMX
      COMMON /FCONST/ EARTH,TSDIV,ZNORS
      COMMON /NVELS/  NVARM,NCARM,NVO,NFO,ICO,IO,JO,JYR,LBL,KTL
C
      PARAMETER(ZMAXC=70.,HMAXC=17.)
C
C     TEST QUALITY OF DATA
C
      IF(REFL.LT.ZMIN.OR.REFL.GT.ZMAXC.OR.KNID.EQ.0) GO TO 20
C
C     DECODE FUNDAMENTAL CELL ATTRIBUTES
C
      DIVR=.001/UF(2)
      DIVV=0.
      IF(UF(8).GT.0.) DIVV=1./UF(8)
      XPOSN=UF(3)*DIVR
      YPOSN=UF(4)*DIVR
      AREA=UF(1)*SETRI*1.E-6*COSPHI
      R2=XPOSN*XPOSN + YPOSN*YPOSN
      RANGE=SQRT(R2)
      ALTUDE=RANGE*SINEL + R2*COSPHI2/EARTH
      RADVEL=UF(6)*DIVV
      SPDVEL=UF(7)*DIVV
C
C     ADJUST POSITION TO COMMON ORIGIN
C
```

```
      XPOSN=XPOSN*COSPHI+DLONG
      YPOSN=YPOSN*COSPHI+DLAT
      ALTUDE=ALTUDE+HEIGHT
C
C   ALTITUDE IN RANGE
C
      IF(ALTUDE.GT.HMAXC) GO TO 20
C
C   INCREMENT CELL COUNTER
C
      IX=IX+1
      IF(IX.GT.NCARM) GO TO 30
C
C       (1) REFLECTIVITY AVERAGED OVER CONTOUR 3DB BELOW PEAK
      CELL(1,IX)=REFL
C       (2) AREA
      CELL(2,IX)=AREA
C       (3) EAST CENTROID POSITION
      CELL(3,IX)=XPOSN
C       (4) NORTH CENTROID POSITION
      CELL(4,IX)=YPOSN
C       (5) CELL ID
      ICELL(5,IX)=KNID
C       (6) HEIGHT AGL
      CELL(6,IX)=ALTUDE
C       (7) RANGE TO CENTROID
      CELL(7,IX)=RANGE
C       (8) TANGENTIAL SHEAR ACROSS CELL
      CELL(8,IX)=SHEAR
C       (9) RADIAL VELOCITY
      CELL(9,IX)=RADVEL
C       (10) RADIAL VELOCITY SPREAD
      CELL(10,IX)=SPDVEL
C
      RETURN
C
C   COUNT REJECTED CELLS
C
   20 NRJC=NRJC+1
      RETURN
C
C   COUNT ARRAY OVERFLOW (TOO MANY CELLS)
C
   30 NCO=NCO+1
      RETURN
      END
```

```
      SUBROUTINE UPDATE(NT,NV,NC,DELW,ECL,NCMX,TECL,IZTS)

C     ***************************************************************
C
C     NAME:       ATLAS
C     PROJECT:    TEST WSR-88D ANCF (TAA)
C
C     PURPOSE:    TO ACCUMULATE ASSOCIATED PEAK CELL ATTRIBUTES
C                 IN ORDER TO FORM 3 DIMENSIONAL VOLUME CELLS
C
C     INTERFACES:
C        CALLING MODS:   DOPLER80,COMPARE,RESOLVE
C        CALLED MODS:    VTRAP(ENTRY=ISOV)
C        INPUT PARM:
C          1) NV  - INDEX OF PRIOR CELL TRACK
C          2) NC  - INDEX OF CURRENT PEAK CELL
C          3) DELW - MEASURE OF ASSOCIATION FROM COMPARE (NC TO NV)
C          4) ECL  - ARRAY CONTAINING REAL PEAK CELL ATTRIBUTES
C          5) NCMX - NUMBER OF PEAK CELLS DETECTED ON CURRENT SCAN
C          6) TECL - ARRAY CONTAINING INTEGER PEAK CELL ATTRIBUTES
C          7) IZTS - FLAG INDICATING REFL(1) OR SHEAR(2) TYPE CELL
C        OUTPUT PARM:
C          1) NT - VOLUME CELL TRACK ID
C        COMMON BLOCKS
C          UPDATTED:  NVLIT,DATA2,CNT
C          READ:      VPARM,ZLOOK,RADAR,NVLIS,NVLIT,PNTRS,INTL,
C                     FLGS,DATA2,DATA4,CNT,CLST
C
C     COMMENTS:  NORMALLY UPDATES ONLY ON REFL CELLS,
C                SINGLE EXCEPTION IS AN ISOLATED SHEAR CELL
C                WHTCH IS UPDATTED AS A REFL CELL ON ITS
C                FIRST OCCURENCE ONLY
C
C     VERSION:  2.0  DEC/VAX-11
C     DATE:     12/15/80
C     DESIGN:   RKCRANE
C     PROGMR:   GBGUSTAFSON
C
C     ***************************************************************
C
      LOGICAL PRCELL,PRSIG,PRFIXC,PRCLUS,PRSCAN,PRHEAD,PRNOIS,PROVER
      DIMENSION TECL(10,128),ECL(10,128),IVCL(53,460),IZTST(2)
C
      COMMON /VPARM/ VX,VY,VXI,VYI,TMKTL,TMKTLL
      COMMON /ZLOOK/ JZOFF,ZARY(91),RRATE(91)
      COMMON /RADAR/ HMP
      COMMON /NVLIS/ NVARM,NCARM,NVO,NFO,ICO,IO,JO,JYR,LBL,KTL
      COMMON /NVLIT/ KTLL,NKNID,NKDO,IZTH,NKDMX,ITHR,IFXC(1024),HTST
      COMMON /PNTRS/ NVMIN,NVMX,IELSN,NSCAN,IESNL,NVSCN,NTT
      COMMON /INTL/  MHSN,MNSN,HM,FNSN,FNSRN,NCLN,NFLN,MZSN,NNMIN,FCAZ
      COMMON /FLGS/  PRCELL,PRSIG,PRFIXC,PRCLUS,PRSCAN,PRHEAD,
     +               PRNOIS,PROVER
      COMMON /DATA2/ VCL(53,460)
      COMMON /DATA4/ ECL(9,256),AFCS,WFCS,NFMX,NFARM,KNIDC(1024),NETA
      COMMON /CNT/   COSPHI,SINEL,COSPI2,ZMIN,ELAST,SPRM,IFXMX
      COMMON /CLST/  ICLN,ICLIST(256),ICLMX
C
      EQUIVALENCE(VCL(1,1),IVCL(1,1))
C
      DATA IZTST/0,1000/
```

```
      IF(IE.LE.0.OR.NC.GT.NCMX) RETURN
      IF(IV.LE.0.OR.NV.GT.NVARM) RETURN
      IF(IVCL(37,NV).EQ.0) THEN
        IVCL(37,NV)=IE
        XCL(47,NV)=XCL(47,NV)*TMELT
        XCL(48,NV)=XCL(48,NV)*TMELT
      ELSEIF(IVCL(37,NV).NE.IE) THEN
        IVCL(37,NV)=IE
      ENDIF
      IF(NEQ.LE.NSCLAN) GO TO 50
      IF(NV.LE.GT.2) WRITE(6,1000) TELSN,NV,NC,IZ,H,X,Y,A,DELW
     *  ,IVCL(37,NV)
 1000 FORMAT(2X,4I6,5E8.2,I8)
   50 IF(IVCL(9,NV) .GT. 0)  GO TO 10
      IVCL(39,NV)=IVCL(29,NV)*ELMSGT(IZTS)

C     DEFINE LOWEST ELEVATION VALUES

      IVCL(1,NV)=Y
      IVCL(2,NV)=Y
      IVCL(3,NV)=IZ
      IVCL(4,NV)=A
      IVCL(6,NV)=ICL(8,NC)
      IVCL(5,NV)=H
      IVCL(12,NV)=H
      IVCL(7,NV)=ICL(7,NC)
      IVCL(37,NV)=IEXD
      IF(TELSN.GT.1) IVCL(37,NV)=-IABS(IVCL(37,NV))
      IF(IVCL(8,NV) .NE. 0) GO TO 10
      IVCL(8,NV)=NT

C     INCREMENT CONTOURABLE ARRAYS

      IVCL(9,NV)=ICOLL
      IF(I.EQ.IZI) IZI=I
      IF(I.GE.IZI+1) IZI=IZI+1
      IVCL(9,NV)=I
      IF(IVCL(9,NV).GT.NKUMX) GO TO 101

C     DETERMINE CONTOUR ID (KNIDLO) AT CELL BASE

      KNIDLO=IVCL(37,NV)
      IF(KNIDLO.LT.0.OR.KNIDLO.GT.NKNID) GO TO 101
      NE=KNIDC(KNIDLO)
      IF(NE.LE.0.OR.NE.GT.NEMX) GO TO 101

C     CHECK IF CELL HAS BEEN ASSIGNED A PRIOR CONTOUR ID (KNID)

      KNID=IABS(IEXD)
      IF(KNID.LE.0.OR.KNID.GT.NKNID) GO TO 103
      NE=KNIDC(KNID)
      IF(NE.LE.0.OR.NE.GT.NEMX) GO TO 103

C     CORRECT: ASSIGN TO CONTOUR WITH LARGEST AREA

      IF(NE.EQ.NE1) GO TO 101
      IF(ECL(1,NE).GE.ECL(1,NE1)) GO TO 101
```

```
C
C     C A L C U L A T E  P I X E L  A T T E N U A T I O N
C
      IF(IZTS.EQ.1) GO TO 110
      IF(IZ.GT.IVCL(20,NV)) GO TO 110
      IF(IZ.LT.IVCL(21,NV)) GO TO 110
C
C     D E F I N E  C E L L  A T T R I B U T E S
C
C     IF(IZ.GT.IVCL(20,NV).AND.VCL(10,NV).GT.VCL(10,NV)+1.
C
C     S U M  C E L L  I N D I C E S
C
      VCL(11,NV)=VCL(11,NV) + Z
      VCL(12,NV)=VCL(12,NV) + Z**2
      VCL(13,NV)=VCL(13,NV) + Z**3
      VCL(14,NV)=VCL(14,NV) + Z*Y
      VCL(15,NV)=VCL(15,NV) + Z*Y**2
      VCL(16,NV)=VCL(16,NV) + Z**2*Y
      VCL(17,NV)=VCL(17,NV) + Z*A
      VCL(18,NV)=VCL(18,NV) + Z*H
C
C     U P D A T E  C U M U L A T I V E  D O F  &  P E A K  A T T R I B S
C
      VCL(23,NV)=VCL(23,NV) + ICL(8,NC)
      VCL(24,NV)=VCL(24,NV) + U
      VCL(25,NV)=VCL(25,NV) + U*U
      VCL(26,NV)=VCL(26,NV) + ICL(10,NC)
      VCL(27,NV)=VCL(27,NV) + DELW
C
C     S U M M I T  V A L U E S
C
      IF(H.LT.VCL(19,NV)) VCL(19,NV)=H
      IF(H.LT.VCL(22,NV)) GO TO 111
      VCL(22,NV)=H
      IVCL(21,NV)=IZ
  111 CONTINUE
C
C     I S  C E L L  U P D A T E  T S  A T T R I B U T E S  A N D  R E T U R N
C
      IF(IZTS.EQ.1) GO TO 112
      CALL TSOV(NV,NC)
      IF(IVCL(20,NV).EQ.0) IVCL(20,NV)=IZ
      RETURN
C
C     D E F I N E  P E A K  R E F L
C
  112 CONTINUE
      IF(IZ.GT.IVCL(20,NV)) IVCL(20,NV)=IZ
      RETURN
      END
```

```
      SUBROUTINE BTRAK(NV,NC,ECL)

C*****************************************************************

C NAME:     BTRAK
C MODULE: RRT A579 600 (AAA)

C PURPOSE: TO STORE PRIOR SCAN ATTRIBUTES OF VOLUME CELL TRACKS

C INTERFACES:
C    CALLING MOD.   DOPER80,COMPARE,RESOLVE
C    CALLED MODS.   NONE
C    INPUT PARM.
C       1) NV    INDEX OF CELL TRACK
C       2) NC    INDEX OF PRIOR SCAN PEAK CELL
C       3) ECL    ARRAY CONTAINING PEAK CELL ATTRIBUTES
C    OUTPUT PARM.   NONE
C    COMMON BLOCKS
C      UPDATED:  DATA3
C      READ:       DATA2,NVELS,VPARM

C COMMENTS:  UPDATES REFL CELLS ONLY

C VERSION:   1.0  DEC/VAX 11
C DATE:      FY 16/80
C PROGMR:    KRUEGER
C PROGMR:    GUSTAFSON

C*****************************************************************

      INTEGER I
      DIMENSION ECL(10,128)

      COMMON /DATA3/ VR(6,460)
      COMMON /DATA2/ VCL(53,460)
      COMMON /NVELS/ NVARM,NCARM,NVO,NFO,ICO,IO,JO,JYR,LBL,KTL
      COMMON /VPARM/ VX,VY,VXT,VYT,TMKTL,TMKTLL

      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

      VR(1,NV)=ECL(3,NC)-VCL(47,NV)*TMKTL
      VR(2,NV)=ECL(4,NC)-VCL(48,NV)*TMKTL
      VR(3,NV)=ECL(1,NC)
      VR(4,NV)=ECL(2,NC)
      VR(5,NV)=ECL(6,NC)
      VR(6,NV)=ECL(8,NC)
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ATRAN(NV,NC,DELW)
C
C     ******************************************************************
C
C     NAME:     ATRAN
C     PURPOSE: TO STORE ATTRIBUTES OF VOLUME CELL 1 ARE
C              THAT ARE UPDATTED BY A SHEAR CELL
C
C     CALLING SEQUENCES:
C        CALLING MODS:  COMPARE,ATRAN(ENTRY ISOV)
C        CALLED MODS:   NONE
C
C        INPUT PARM:
C          1) NV - INDEX OF CELL TRACK
C          2) NC - INDEX OF PEAK CELL
C          3) DELW - MEASURE OF ASSOCIATION FROM COMPARE (NC TO NV)
C        OUTPUT PARM:  NONE
C        COMMON BLOCKS
C          UPDATTED:  DATA2
C          READ:      DATA5
C
C     COMMENTS:  UPDATES ON ASSOCIATED SHEAR CELLS ONLY
C
C     VERSION:  1.0  DEC/VAX-11
C     DATE:     12/16/80
C     DESIGN:   GBGUSTAFSON
C     PROGMR:   GBGUSTAFSON
C
C     ******************************************************************
C
C
      DIMENSION IVCL(53,460)
C
      COMMON /DATA2/ VCL(53,460)
      COMMON/ DATA5/ ESCL(10,128),NSCO,NSCMX,NSRJC
C
      EQUIVALENCE(VCL(1,1),IVCL(1,1))
C
      TS=ESCL(8,NC)
      V=ESCL(9,NC)
C
      IVCL(29,NV)=IVCL(29,NV)+1
C
C     STORE ATTRIB ON ASSOCIATED TS CELLS
C
      VCL(23,NV)=VCL(23,NV)+TS
      VCL(24,NV)=VCL(24,NV)+V
      VCL(25,NV)=VCL(25,NV)+V*V
      VCL(26,NV)=VCL(26,NV)+ESCL(10,NC)
      VCL(27,NV)=VCL(27,NV)+DELW
C
C     STORE ATTRIB ON TS CELLS ONLY
C
      ENTRY ISOV(NV,NC)
C
      VCL(30,NV)=VCL(30,NV)+TS
      VCL(31,NV)=VCL(31,NV)+V
```

```
      VCL(32,NV)=VCL(32,NV)+TS*ESCL(3,NC)
      VCL(33,NV)=VCL(33,NV)+TS*ESCL(4,NC)
      VCL(34,NV)=VCL(34,NV)+TS*ESCL(6,NC)

      RETURN
      END
```

```
      SUBROUTINE CC2 (PEC,NEMX,IECL,IETS)
C
C     ********************************************************************
C
C                 CC2POST
C     PURPOSE:    RELOCATE PEAK CELLS
C
C     ENTRIES:    TO ASSOCIATE PEAK CELLS TO EXISTING
C                 CELL TRACKS BY EXTRAPOLATING BACK ALONG THE TRACK
C                 VELOCITY VECTOR, TO COMPUTE A MEASURE OF
C                 THE GOODNESS OF ASSOCIATION, AND TO RELEASE ALL
C                 OF CELLS IN CLUSTERS.
C
C     CALLED BY:
C        CALLING MODS.  MODPER80
C        CALLED MODS.   ATRAK,BTRAK,VTRAK,RESOLV
C        INPUT PARM.
C          1) PEC  - ARRAY CONTAINING REAL PEAK CELL ATTRIBUTES
C          2) NCMX - NUMBER OF PEAK CELLS DETECTED ON CURRENT SCAN
C          3) IECL - ARRAY CONTAINING INTEGER PEAK CELL ATTRIBUTES
C          4) IETS - FLAG INDICATING REEL(1) OR SHEAR(2) TYPE CELL
C        OUTPUT PARM.  NONE
C        COMMON BLOCKS
C          UPDATED:  CDRAYS,CLST,PNTRS,NVLTS,NVLTT,CNT,UVC
C          READ:     DATA2,DATA3,CDRAYS,VPARM,PNTRS,CONST,
C                    NVLTS,NVLTT,ECONST
C
C     COMMENTS:  ISOLATED SHEAR CELLS ARE  TREATED AS
C                REEL CELLS IF:
C                     1) IT IS THE ONLY CELL ASSOC. WITH A TRACK AND
C                     2) ALL OTHER CELLS ASSOC. TO SAME TRACK ARE
C                        ISOLATED SHEAR CELLS
C
C     VERSION: 1.1  DEC/VAX-11
C     DATE:    12/05/80
C     DESIGN:  KKCRANE
C     PROGMR:  GBGUSTAFSON
C
C     ********************************************************************
C
      DIMENSION TVCL(53,460),IECL(10,128)
      DIMENSION ECL(10,128)
C
      COMMON /DATA2/  VCL(53,460)
      COMMON /DATA3/  VR(6,460)
      COMMON /CDRAYS/ IC(128,10),C(128,9),ID(128,10),D(128,9),
     +                IM,JM,MCDX
      COMMON /VPARM/  VX,VY,VXI,VYI,TMKTL,TMKTLL
      COMMON /PNTRS/  NVMIN,NVMX,IELSN,NSCAN,IESNI,NVSCN,NT
      COMMON /CONST/  VMISW(2),DIV,VMAG,VMISWM,ZDIV,ADIV,
     +                A1,A2,A3,B1,B2,HDIV
      COMMON /NVLTS/  NVARM,NCARM,NVO,NFO,ICO,IO,JO,JYR,IBL,KTL
      COMMON /NVLTT/  KTLL,NKNTD,NKDO,IZTH,NKDMX,ITHR,IFXC(1024),IEL,
      COMMON /CNT/    COSPHI,SINEL,COSP12,ZMIN,ELAST,SPRM,IFXMX
      COMMON /ECONST/ EARTH,TSDIV,ZNDRS
      COMMON /UVC/    UV(512),UC(256),THV(512,2),IHC(256,2),IV(128,
      COMMON /TSOS/   TSOCTR
C
      DATA INITAL/0/
C
```

```
      EQUIVALENCE (VCL(1,1),IVCL(1,1))
C
      VMISWM=(VMISW(IZIS)-1.)/DIV
      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.
      IUV(I,1)=0
    3 IUV(I,2)=0
      DO 4 I=1,NCMX
      UC(I)=0.
      IUC(I,1)=0
    4 IUC(I,2)=0
      DO 5 I=1,MCIDX
      IC(I,1)=0
      ID(I,1)=0
      DO 7 J=1,JM
      IC(I,J+1)=0
      ID(I,J+1)=0
      C(I,J)=0.
      D(I,J)=0.
    7 CONTINUE
    5 CONTINUE
C                                                                       1
C   BEGIN NC COMPARE LOOP
C
      DO 10 NC=1,NCMX
      NVC=0
C
C   COMPARE ALL NC CELLS TO EACH VCL TRACK
C
      DO 40 NV=1,NVMXP
      MLAST=0
      DELW=0.
      DELWL=0.
C
C   COMPARE CURRENT CELL TO LAST CELL
C
      IF(IVCL(53,NV).LE.0 .AND. IVCL(9,NV).LE.0) GO TO 40
      DTTA=TMKTL
      IF(IVCL(9,NV).LE.0) DTTA=TMKTLL
      ATEST=(VMAG*DTTA)*(VMAG*DTTA)+VMISWM
      DELX=ECL(3,NC)-VR(1,NV)-VCL(47,NV)*DTTA
      DELX2=DELX*DELX
      IF(DELX2 .GT. ATEST) GO TO 20
      DELY=ECL(4,NC)-VR(2,NV)-VCL(48,NV)*DTTA
      DELY2=DELY*DELY
      IF(DELY2 .GT. ATEST) GO TO 20
      DHT = (ECL(6,NC) - VR(5,NV))
C
C   CURRENT CELL CAN EXTRAPOLATE BACK TO LAST CELL
C   COMPUTE MEASURE OF ASSOCIATION TO NV CELL
C
      DELWI = ABS(ECL(1,NC) - VR(3,NV)) * ZDIV
     1      +     (  DELX2   +  DELY2  ) *  DIV + 1.
     2      + ABS(ECL(2,NC) - VR(4,NV)) * ADIV
     3      +         DHT     *   DHT     * HDIV
```

```
C
      IF(DELWL.LT.....W.IZT.)) MLAST =
C
C   COMPARE CURRENT CELL TO VCL TRACK
C
   30 DELY DELA
      ATEST=VMTSW(MAG*DELT)*(VMAG*DELT)
      DELX=VCL(3,NC)-VCL(1,NV)-VCL(47,NV)*DELT
      DELX2=DELX*DELX
      IF(DELX2.GT.ATEST) GO TO 40
      DELY=VCL(4,NC)-VCL(2,NV)-VCL(48,NV)*DELT
      DELY2=DELY*DELY
      IF(DELY2.GT.ATEST) GO TO 40
      DHT=VCL(6,NC)-VCL(5,NV)
C     IF(IVCL(2,NV).EQ.0.AND.DHT.GT.HTSIG) GO TO 401
      ZVAL=IVCL(3,NV)
C
C   CURRENT CELL CAN EXTRAPOLATE BACK TO CELL BASE
C   COMPUTE MEASURE OF ASSOCIATION TO VCL TRACK
C
   40 DELW = ABS(FCL(1,NC) -    ZVAL    ) * ZWTV
     1      +    (  DELX2   +   DELY2   ) *  HTV +1.
     2      + ABS(ECL(2,NC) - VCL(4,NV)) * AWTV
     3      +         DHT     *    DHT     * HWTV
C
      IF(MLAST.NE.0.AND.DELWL.LT.DELW) DELW=DELWL
      IF(DELW.LE..9) GO TO 40
      IF(DELW.GT.VMTSW(12TS)) GO TO 40
C
C   DELW IS VMTSW, BOTH WAYS
C
      NVC=NVC+1
      IFVN=0
      IFVO=0
      IF(NVC .EQ. 1) GO TO 41
      IF(IUC(NC,1) .EQ. NV) GO TO 33
      NVT=IUC(NC,1)
      IF(IVCL(9,NV).EQ.0 .OR. IVCL(9,NVT).EQ.0) GO TO 34
C
C   OVERIDE SHEAR TRACK ASSOCIATIONS (IFV()=1) WITH REEL TRACKS
C
      IF(IVCL(9,NV).EQ.1.AND.IVCL(29,NV).GE.1000) IFVN=1
      IF(IVCL(9,NVT).EQ.1.AND.IVCL(29,NVT).GE.1000) IFVO=1
C
C   IF PRIOR AND CURRENT TRACK(NV) TYPES EQUAL, FIND BEST
C
      IF(IFVN.EQ.IFVO) GO TO 34
C
C   DO NOT USE PRIOR SHR TRACK IF CURRENT ASSN TO REEL TRACK
C
      IF(IFVN.EQ.0.AND.IFVO.EQ.1) GO TO 42
C
C   DO NOT USE CURENT SHR TRACK IF PRIOR ASSN TO REEL TRACK
C
      NVC=NVC-1
      GO TO 40
C
C   NORMAL PROCESSING
C
   34 NVT=NV
```

```
      IF(IEVN.EQ.1) ISOCTR=ISOCTR+1
      IF(DELW .GE. UC(NC)) GO TO 35
C
C     HAVE NEW BEST FIT, REORDER LIST
C
      DX=UC(NC)
      UC(NC)=DELW
      DELW=DX
      NVT=IUC(NC,1)
      IUC(NC,1)=NV
C
C     KEEP TRACK OF ALL ASSOCITATIONS TO NC
C
   34 IF(IZTS.EQ.2) GO TO 40
      IF(IUC(NC,2) .LE. 0) GO TO 36
      I=IUC(NC,2)
      IF(I.GT.IM) GO TO 361
      GO TO 39
   36 I=IDV
      IDV=I+1
      IF(I .LE. IM) GO TO 38
  361 IO=IO+1
      I=IM
   38 IUC(NC,2)=I
   39 I=ID(I,1)+1
      ID(I,1)=J
      IF(J .LE. JM) GO TO 37
      JO=JO+1
      J=JM
   37 ID(I,J+1)=NVT
      D(I,J)=DELW
      GO TO 33
C
C     RESTART LIST ON CURRENT REFL TRACK
C
   42 NVC=1
   41 IUC(NC,1)=NV
      UC(NC)=DELW
C
C     SET BEST CELL TO TRACK MATCH
C
   33 IF(IZTS.EQ.2) GO TO 40
      IF(IUV(NV,1) .NE. 0) GO TO 21
      IUV(NV,1)=NC
      UV(NV)=DELW
      GO TO 40
C
C           CLUSTER
C
   21 IF(IUV(NV,1) .EQ. NC) GO TO 40
      NCT=NC
      IF(DELW .GE. UV(NV)) GO TO 25
   22 DX=UV(NV)
      UV(NV)=DELW
      DELW=DX
      NCT=IUV(NV,1)
      IUV(NV,1)=NC
   24 IF(IUV(NV,2) .EQ. 0) GO TO 26
      I=IUV(NV,2)
      IF(I.GT.IM) GO TO 261
```

```
      GO TO 35
   35 I=IC
      IC(I)=I
      IF(I.LT.IM) GO TO 38
   37 IC=IC+1
      I=IM
   38 IUV(NV,1)=I
   39 I=IC(I,1)+1
      IC(I,1)=I
      IF(I.LT.IM) GO TO 37
      IC=IC+1
      I=IM
   40 IC(I,IC)=NCT
      CC(I,1)=DELW
      GO TO 40
C
C     TEST TO ALLOW CELL BASE TO RISE ON SECOND OBSERV. ONLY
C
C  401 IF(IVCL(53,NV).NE.1) GO TO 30
      IF(CHIT.LE.2.) GO TO 402
C
   40 CONTINUE
      IF(NVC.EQ.0) GO TO 45
C
C    FOUND MATCH ON REEL CELL, RESOLVE CONFLICTS
C
      IF(IZTS.NE.2) GO TO 10
      NV=IUC(NC,1)
      IF(IVCL(9,NV).EQ.0) GO TO 56
C
C    FOUND MATCH ON TS CELL, SUM ATTRIBUTES
C
      CALL VTRAK(NV,NC,UC(NC))
      GO TO 10
C
C    NO COMPARE, FIND EMPTY NV SLOT AND BEGIN NEW CELL
C
   45 IF(NVMX.LT.NVMIN) GO TO 501
      DO 50 NV=NVMIN,NVMX
      IF(IVCL(53,NV).EQ.0 .AND. IVCL(9,NV).EQ.0) GO TO 55
   50 CONTINUE
  501 NV=NVMX+1
      IF(NV .LT. NVARM) GO TO 51
      NVO=NVO+1
      NV=NVARM
   51 NVMX=NV
   55 NVMIN=NV
      IF(NVMX.LT.0) NVMX=NV
      IF(INITIAL.EQ.0) GO TO 57
   56 CALL ATRAK(NT,NV,NC,UC(NC),ECL,NCMX,IECL,IZTS)
      CALL BTRAK(NV,NC,ECL)
   57 IUV(NV,1)=NC
      IUC(NC,1)=NV
      UV(NV)=0.0
      UC(NC)=0.0
   10 CONTINUE
C
C    RUN COMPARE LIST ON NV, MINIMISE DELW ON NC    NV ASSOCIATION
C
      IF(IZTS.EQ.2) GO TO 100
```

```
      DO 50 NV=1,NVMXP
      NC=IUV(NV,1)
      IF(NC.LT.0 .OR. NC.GT.NCMX) GO TO 60
      IF(IUC(NC,1) .LE. 0) GO TO 60
      IF(IUV(NV,2).NE.0 .OR. IUC(NC,2).NE.0) GO TO 70
      IF(INITAL.EQ.0) GO TO 71
      IF(IUC(NC,1).NE.NV) GO TO 60
C
      CALL ATRAK(NT,NV,NC,UC(NC),ECL,NCMX,IECL,IZTS)
      CALL BTRAK(NV,NC,ECL)
C
   71 IUV(NV,1)=-IUV(NV,1)
      IUC(NC,1)=-IUC(NC,1)
      UV(NV)=0.0
      UC(NC)=0.0
      GO TO 60
C
C   RESOLVE CONFLICTS
C
   70 CALL RESOLVE(NC,ECL,NCMX,IECL,IZTS,INITAL)
   60 CONTINUE
C
C   ESTABLISH ASSN COUNT THIS SCAN
C
  100 MCDX=MAXO(IDV,ICV)
      IF(MCDX.LE.0) MCDX=1
      IF(MCDX.GT.IM) MCDX=IM
C
C    ASSOCIATE KNID VALUES
C
      DO 789 I=1,NCMX
      NV=IABS(IUC(I,1))
      IF(NV.LE.0.OR.NV.GT.NVMX) GO TO 789
      IFXD=IVCL(37,NV)
      IF(IFXD.GT.0) GO TO 789
      IXD=IECL(5,1)
      IF(IXD.LE.0.OR.IXD.GT.NKIDMX) GO TO 789
      IVCL(37,NV)=IFXC(IXD)
  789 CONTINUE
C
C     ZERO KNID DIRECTORY
C
      DO 788 I=1,IFYMX
  788 IFXC(I)=0
      IFXMX=1
      INITAL=1
      RETURN
      END
```

```
C     SUBROUTINE          COMCANE  ...  ...  ...  ...  ...  ...
C
C     ***********************************************************************
C
C     PURPOSE:      RESORTS
C     CALLED BY:    ...  ...  ...  ...  ...  ...  ...  ...
C
C     METHOD:       USES A CLUSTER OF CELLS TO FIND ONE  ...  ...
C                   ASSOCIATION MATRIX THAT MINIMIZES  ...  ...
C                   ASSOCIATION WEIGHTS.
C
C     SUBROUTINES:
C        CALLING MODE:   COMCANE
C        CALLED MODE:    NEARN,COMRENE
C        INPUT PARMS:    ...,...,...,...,...,...,...
C        OUTPUT PARMS:   NONE
C        COMMON BLOCKS:
C          UPDATED:      ...,...,...,...,...,...,...
C          READ:         NVARS,PNTRS,FLGS,DATA2,CONST,CNT,FCONST,INTL
C
C     COMMENTS:     CALLED ONLY WHEN MORE THAN ONE ...  ...  ...
C                   TO THE SAME CELL  TRACK IN COMCANE.
C
C     VERSION:      1.0  DEC/VAX-11
C     DATE:         1/6/81
C     DESIGN:       RKOPANE
C     PROGRAM:      JHWILLAND
C
C     ***********************************************************************
C
      LOGICAL PRCELL,PRSIG,PRFIXC,PRCLUS,PRSCAN,PRHEAD,PRNOIS,PROVER
      DIMENSION IVCL(53,460),V(128,3),ICL(10,128),ICL(10,128)
C
      COMMON /NVARS/  NVARM,NCARM,NVO,NFO,ICO,IO,JO,JYR,IBL,KTI
      COMMON /PNTRS/  NVMIN,NVMX,ICLSN,NSCAN,ICSNI,NVSCN,NI
      COMMON /CLST/   ICLN,ICLIST(256),ICLMX
      COMMON /CDRAYS/ IC(128,10),C(128,9),ID(128,10),D(128,9),IM,IM,...
      COMMON /FLGS/   PRCELL,PRSIG,PRFIXC,PRCLUS,PRSCAN,PRHEAD,
     1                PRNOIS,PROVER
      COMMON /DATA2/  VCL(53,460)
      COMMON /CONST/  VMISW(2),DIV,VMAG,VMISWM,ZDIV,ADIV,
     1                A1,A2,A3,B1,B2,HDIV
      COMMON /CNT/    COSPHI,SINEL,COSPI2,ZMIN,ELAST,SFRM,IFXMX
      COMMON /FCONST/ EARTH,TSDIV,ZNDRS
      COMMON /INTL/   MHSN,MNSN,HM,FNSN,FNSRN,NCLN,NELN,MZSN,NNMIN,...
      COMMON /UVC/    UV(512),UC(256),IUV(512,2),IUC(256,2),IV(128,...
C
      EQUIVALENCE(VCL(1,1),IVCL(1,1))
      DATA IVMX/128/
C
C
C           HAVE CLUSTER, ORDER LISTS
C
      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=IUC(1)
      IF(NCT.LE.0.OR.NCT.GT.NCMX) GO TO 100
C     PROCESS NCT
   62 IF(IUC(NCT,1) .LE. 0.OR.IUC(NCT,1).GT.NVMX) GO TO 66
      IF(UC(NCT) .LE. 0.)GO TO 66
      NVT=IUC(NCT,1)
      UC(NCT)=-ABS(UC(NCT))
      CALL COMBINE(NVT,IVS,IVT,KV,J,IVMX)
      IF(IUC(NCT,2) .LE. 0.OR.IUC(NCT,2).GT.IM) GO TO 62
      I=IUC(NCT,2)
      IX=ID(I,1)
      IF(IX .LE. 0) GO TO 62
      IF(IX.GT.JM) IX=JM
      ID(I,1)=ID(I,1)
      DO 611 J=I,IX
      CALL COMBINE(ID(I,J+1),IVS,IVT,KV,JJ,IVMX)
  611 CONTINUE
C     PROCESS NVT
   63 IF(IUV(NVT,1) .LE. 0.OR.IUV(NVT,1).GT.NCMX) GO TO 63
      IF(UV(NVT) .LE. 0.)GO TO 63
      NCT=IUV(NVT,1)
      UV(NVT)=-ABS(UV(NVT))
      CALL COMBINE(NCT,ICS,ICT,KC,J,IVMX)
      IF(IUV(NVT,2) .LE. 0.OR.IUV(NVT,2).GT.IM) GO TO 63
      I=IUV(NVT,2)
      JX=IC(I,1)
      IF(JX .LE. 0) GO TO 63
      IF(JX.GT.JM) JX=JM
      IC(I,1)=IC(I,1)
      DO 621 J=I,JX
      CALL COMBINE(IC(I,J+1),ICS,ICT,KC,JJ,IVMX)
  621 CONTINUE
C
C           RUN COMPARE LIST TO FLUSH OUT FULL SET
C
   63 DO 631 K=JV,KV
      NVT=IV(K,IVS)
      IF(NVT.LE.0.OR.NVT.GT.NVARM) GO TO 631
      IF(UV(NVT).LE.0.) GO TO 631
      IF(IUV(NVT,1).GT.0.AND.IUV(NVT,1).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).LE.0.) GO TO 661
      IF(IUC(NCT,1).GT.0.AND.IUC(NCT,1).LE.NVARM) GO TO 67
  661 CONTINUE
      GO TO 68
   67 JC=K
      LV=LV+1
      GO TO 65
```

```
   65 IF(IC .EQ.      ) GO TO 70
      IC
      JV=1
      V=
      I=
      GO TO 65
   70 IF(IV .EQ.  ) GO TO 70
      IV=0
      JV=1
      I=
      K=1
      GO TO 65
C
C           HAVE ORDERED LIST, NOW FIND BEST MATCH
C
   75 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
      UV(NV)=ABS(UV(NV))
  701 CONTINUE
      DO 71 K=1,KC
      V(K,1)=0.
        V(K,2)=0.
        V(K,3)=0.
        IV(K,ICT)=0
        IV(K,IVT)=0
        IV(K,3)=0
        IV(K,6)=0
        IV(K,7)=0
      NC=IV(K,ICS)
      IF(NC.LE.0.OR.NC.GT.NCMX) GO TO 71
      UC(NC)=ABS(UC(NC))
      NV=IUC(NC,1)
      IF(NV.LE.0.OR.NV.GT.NVMX) GO TO 71
      IF(IUV(NV,1).LT.0)GO TO 71
      IF(IUV(NV,1) .NE. NC) GO TO 711
      V(K,1)=UC(NC)
      UV(NV)=-UV(NV)
      UC(NC)=-UC(NC)
      IV(K,3)=NV
      GO TO 71
  711 IMSM=IMSM+1
   71 CONTINUE
      IF(KV.LE.(KC-IMSM) .OR. IMSM.EQ.0) GO TO 75
C
C           FIRST ROUND MIN WEIGHT SELECTION
C
      KNC=0
      KNV=0
      DO 72 K=1,KC
      IF(IV(K,3) .NE. 0) GO TO 72
      NC=IV(K,ICS)
      IF(NC.LE.0.OR.NC.GT.NCMX) GO TO 72
      IF(UC(NC).LE.0.1) GO TO 72
```

```
      NV=IUC(NC,1)
      IF(NV.LE.0.OR.NV.GT.NVMX) GO TO 724
      IF(UV(NV).GT.0.1) GO TO 725
      IV(K,2)=NV
      KNC=KNC+1
  724 I=IUC(NC,2)
      IF(I.LE.0.OR.I.GT.IM) GO TO 721
      JX=IABS(ID(I,1))
      IF(JX.LE.0.OR.JX.GT.JM) GO TO 721
      NV=0
      DWT=999.
      DO 723 J=1,JX
      NVT=ID(I,JH+J)
      IF(NVT.LE.0.OR.NVT.GT.NVMX) GO TO 723
      IF(UV(NVT) .LE. 0.1) GO TO 723
      DELW=D(I,J)
      IF(DELW.LE.0.1) GO TO 723
      DWT=AMIN1(DWT,DELW)
      IF(DWT .EQ. DELW) NV=NVT
  723 CONTINUE
      IF(NV.LE.0.OR.NV.GT.NVMX.OR.DWT.GT.VMISW(IZTS).OR.DWT.LT..1)
     1 GO TO 721
      GO TO 726
  725 DWT=UC(NC)
  726 U(K,2)=DWT
      UV(NV)=-ABS(UV(NV))
      UC(NC)=-ABS(UC(NC))
      IV(K,ICT)=NV
      GO TO 72
  721 KNC=KNC+1
      IF(KNC.GT.KC) GO TO 72
      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
C           CASCADE REORDER OF COMPARE LIST
C
      J=0
  731 J=JH+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=IUC(NC,1)
      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,ICT).EQ.NV) GO TO 7382
  738 CONTINUE
      GO TO 739
 7381 NCT=IV(L,ICS)
      DELT=VMISW(IZTS)+V(L,1)
      GO TO 7383
 7382 NCT=IV(L,ICS)
      DELT=VMISW(IZTS)+V(L,2)
 7383 KT=1
      IF(NCT.LE.0.OR.NCT.GT.NCMX) GO TO 739
      IF(IUC(NCT,1).LE.0.OR.IUC(NCT,2).LE.0) GO TO 739
```

```
      DELW ABS(DC(NC))
      I IABS(IDC(NC),2))
      IF(I .LE. 0.OR.I.GT.IM) GO TO 239
      JX=IABS(IB(I,L))
      IF(JX .LE. 0.OR.JX.GT.JM) GO TO 239
      DWT=999.
      NVB=0
      DWT1=999.
      NV1=0
      DO 232 I=1,KC
      IF(DC(I),LE.0.) GO TO 232
      DWT=AMIN1(DWT,DC(I))
      NVT1=IABS(IB(I,1))
      IF(DWT .EQ. DC(I)) NVB=NVT1
      IF(NVT1.LE.0.OR.NVT1.GT.NVMX) GO TO 232
      IF(DV(NVT1).GT.0.) DWT1=AMIN1(DWT1,DC(I))
      IF(DWT1.EQ.DC(I)) NV1=NVT1
  232 CONTINUE
      IF(NVB.LE.0.OR.NVB.GT.NVMX.OR.DWT. [illegible] .OR.DWT.GT.VMIN(NVB))
     1  GO TO 239
      IF(NV1.LE.0.OR.NV1.GT.NVMX.OR.DWT1.GT.VMIN(WCTZTS).OR.DWT1.LE. [illegible]
     1 GO TO 235
      DELW1=DELW+DWT1
      DELW2=DELW+DWT
      IF(DELW1.GT.DELT) GO TO 235
      IV(K,6)=NV
      V(K,3)=DELW
      IV(KT,6)=NVT1
      V(KT,3)=DWT1
      IF(DELW2.GE.DELW1) GO TO 239
      IV(KT,7)=NVB
      GO TO 239
  235 DO 236 I=1,KC
      IF(IV(I,3).EQ.NVB) GO TO 239
      IF(IV(I,ICT).EQ.NVB) GO TO 237
  236 CONTINUE
      GO TO 239
  237 DELT=DELT+V(I,2)
      DELW2=DELW+DWT+VMIN(WCTZTS)
      IF(DELW2.GT.DELT) GO TO 239
      IV(K,6)=NV
      V(K,3)=DELW
      IV(KT,6)=NVB
      V(KT,3)=DWT
      IV(I,ICT)=0
      V(I,2)=0.
  239 IV(K,7)=0
      KNV=KNV-1
      GO TO 231
C
C       EXCHANGE PAIRS FOR MIN MEASURE
C
   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(IDV(NVB,2).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 I=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=IABS(IUV(NVB,2))
      JX=IC(I,1)
      IF(JX.LE.0.OR.JX.GT.JM) GO TO 801
      DO 807 J=1,JX
      IF(IC(I,J+1).EQ.NCB) GO TO 808
  807 CONTINUE
      GO TO 801
  808 DSET=C(I,J)
      DELWB=DSET+ABS(UC(NC))
      DELW1=V(K,3)
      IF(DELW1.LE..1) DELW1=V(K,2)
      IF(DELW1.LE..1) DELW1=V(K,1)
      DELW2=V(L,3)
      IF(DELW2.LE..1) DELW2=V(L,2)
      IF(DELW2.LE..1) DELW2=V(L,1)
      DELW=DELW1+DELW2
      IF(DELW.LE.DELWB) GO TO 801
      V(K,3)=ABS(UC(NC))
      V(L,3)=DSET
      IV(K,6)=NVB
      IV(L,6)=NV
  801 IV(K,7)=0
C
C       FINISHED ASSOCIATION, FIND CLUSTER ID
C
C
    75 ICLUST=0
      DO 761 K=1,KV
      NV=IV(K,IVS)
      IF(NV.LE.0.OR.NV.GT.NVMX) GO TO 761
      ICLUST=MAXO(ICLUST,IVCL(38,NV))
  761 CONTINUE
      IF(ICLUST .GT.0) GO TO 7642
      ICLUST=ICLN
      ICLN=ICLUST+1
      IF(ICLN.LE.ICLMX) GO TO 7641
      ICLN=ICLMX
      ICO=ICO+1
 7641 ICLIST(ICLUST)=ICLUST
      GO TO 764
 7642 ICL=ICLIST(ICLUST)
      IF(ICL.EQ.ICLUST) GO TO 764
      IF(ICL.LE.0.OR.ICL.GT.ICLN) GO TO 7641
      ICLUST=ICL
      GO TO 7642
  764 DO 762 K=1,KV
      NV=IV(K,IVS)
      IF(NV.LE.0.OR.NV.GT.NVMX) GO TO 762
      ICL=IVCL(38,NV)
      IF(ICL.GT.ICLN.OR.ICL.LE.0) GO TO 763
      ICLIST(ICL)=ICLUST
  763 IVCL(38,NV)=ICLUST
      IF(PRSCAN) WRITE(6,1029) ICLUST,NV,VCL(1,NV),VCL(2,NV)
 1029 FORMAT(1X,' CLST',2I5,2F10.2)
  762 CONTINUE
```

```
C
C       UPDATE ATTRIBUTES
C
        DO 78 K=1,KU
        IF(PRSCAN)
     1  WRITE(6,788) ICLUST,IV(K,ICS),IV(K,6),IV(K,ICT),
     1    IV(K,3),V(K,3),V(K,2),V(K,1)
  788 FORMAT(1X,5I5,3F8.2)
        NC=IV(K,ICS)
        IF(NC.LE.0.OR.NC.GT.NCMX) GO TO 78
        IF(IUC(NC,1).LE.0) GO TO 78
        NV=IV(K,6)
        IF(NV.LE.0.OR.NV.GT.NVMX) GO TO 810
        DWT=V(K,3)
        GO TO 820
  810 NV=IV(K,ICT)
        IF(NV.LE.0.OR.NV.GT.NVMX) GO TO 811
        DWT=V(K,2)
        GO TO 820
  811 NV=IV(K,3)
        IF(NV.LE.0.OR.NV.GT.NVMX) GO TO 79
        DWT=V(K,1)
  820 IF(DWT.LE.0.1.OR.DWT.GT.VMISW(IZTS)) GO TO 79
        IF(IUV(NV,1).LE.0) GO TO 79
        IF(INITAL.EQ.0) GO TO 77
        CALL ATRAK(NT,NV,NC,DWT,ECL,NCMX,IECL,IZTS)
        CALL RTRAK(NV,NC,ECL)
        GO TO 77
C
C    NO ASSN  FIND EMPTY NV AND START NEW CELL
C
    79 IF(NVMX.LT.NVMIN) GO TO 7911
        DO 791 I=NVMIN,NVMX
        IF(IVCL(53,I).EQ.0 .AND. IVCL(9,I).EQ.0) GO TO 792
  791 CONTINUE
 7911 I=NVMX+1
        IF(I.LT. NVARM) GO TO 7921
        NVO=NVO+1
        I=NVARM
        NV=1
        GO TO 77
 7921 NVMX=I
  792 NV=I
        NVMIN=I
        IVCL(28,NV)=ICLUST
        IF(INITAL.EQ.0) GO TO 77
        CALL ATRAK(NT,NV,NC,0.,ECL,NCMX,IECL,IZTS)
        CALL RTRAK(NV,NC,ECL)
C
C    FLAG NC/NV AS ASSOCIATED
C
    77 IUV(NV,1)=NC
        IUC(NC,1)=NV
        UV(NV)=0.
        UC(NC)=0.
    78 CONTINUE
C
        DO 99 K=1,KU
        NV=IV(K,IVS)
        IF(NV.LE.0.OR.NV.GT.NVMX) GO TO 99
```

```
      IF(IUV(NV,1).LE.0.) GO TO 99
      IUV(NV,1)=-IABS(IUV(NV,1))
      IF(VCL(10,NV).LT..9999.OR.ENSN.LT.1.1) GO TO 99
      HTC=VCL(7,NV)*SINEL+VCL(7,NV)*VCL(7,NV)*COSPI2/EARTH
      IF(HTC.LE.HM) GO TO 99
      VCL(10,NV)=VCL(10,NV)/(ENSN-1.)
   99 CONTINUE
  100 RETURN
      END
```

```
      SUBROUTINE COMBINE(N,IS,IT,K,J,IVMX)
      COMMON /IVC/ IV(512),IC(256),IUV(512,2),IUC(256,2),IVC(
C
C     INSERT N INTO ORDERED ARRAY IV(K,IT)
C     RETURN NEW ARRAY AS IV(K,IS)
C
      I=IS
      IS=IT
      IT=I
      I=0
      DO 10 I=1,K
      IF(IABS(IV(I,IT)).GT.N) 20,30,40
   20 IF(IV(I,IT).EQ.0) GO TO 40
   10 IV(I,IS)=IV(I,IT)
      J=K
   40 I=J
      IV(I,IS)=N
   30 DO 50 I=J,K
   50 IV(I+1,IS)=IV(I,IT)
      K=K+1
      IF(K.GE.IVMX) GO TO 70
      IV(K,IS)=0
      GO TO 80
   70 PRINT 100,K,IVMX
  100 FORMAT(' ERROR IN COMBINE ',I3,K,IVMX=',2X,2I10)
      K=IVMX-1
   80 RETURN
      END
```

```
      SUBROUTINE STRAK
      CHARACTER*8 INPUT
      CHARACTER*3 FILSTAT(2)
      INTEGER*2 KEY(13)
      REAL*8 SVA,SVB,SVC,SA2,SB2,SC2,SAB,SAC,SBC,SV2,SB,SC
      REAL*8 PVA,PVB,PVC,PA2,PB2,PC2,PAB,PAC,PBC,PV2,PB,PC,UNOM,W
      INTEGER ISEC,IM,IMI,IMX,ISTAT
      LOGICAL PRCELL,PRSTG,PRFIXC,PRCLUS,PRSCAN,PRHEAD,PRNOTS,NTEST,
     +        PROVLR
      REAL N2(10),N2S(10),N3(10),N3S(10),
     +     SUM2(10),SPD2(10),SUM3(10),SPD3(10)
      DIMENSION IVCL(53,460),IFCL(9,256)
      DIMENSION STG(2)
C
      COMMON /IMAX/     IM,IMI,IMX
      COMMON /HEADIC/   H1(3,5),H2(3,5),H3(3,5)
      COMMON /BUFFER/   A(5),B(5),NSM,ISR(20),ISHR
      COMMON /FLGS/     PRCELL,PRSTG,PRFIXC,PRCLUS,PRSCAN,PRHEAD,
     +                  PRNOTS,PROVLR
      COMMON /MULT/     VDPT(4),NACTT,NTEST
      COMMON /PNTRS/    NVMIN,NVMX,IFLSN,NSCAN,IESNL,NVSCN,NT
      COMMON /TIMS/     ISEC,JDAY,JHR,JMIN,JSEC,IDAY,THR,TMIN,TSEC
      COMMON /NVLTS/    NVARM,NLARM,NVO,NFO,ICO,IO,JO,JYR,IBL,KTL
      COMMON /NVLTT/    KTLT,NKNTO,NKDO,IZTH,NKDMX,ITHR,TEXC(10,4),IITST
      COMMON /CONST/    VMISW(2),DTV,VMAG,VMISWM,ZDTV,AIDTV,
     +                  A1,A2,A3,B1,B2,HDTV
      COMMON /CLST/     ICLN,ICLIST(256),ICLMX
      COMMON /CNT/      COSPHI,SINFI,COSFI2,ZMIN,ELAST,SPRM,IFXMX
      COMMON /INTL/     MHSN,MNSN,HM,FNSN,FNSRN,NCLN,NFLN,MZSN,NNMIN,FLA/
      COMMON /UVC/      UX(512),UY(512),UCX(128),UCY(128),UCVX(128),
     1                  UCVY(128),UCN(128),UCZ(128),UCXY(128),UCY2(128),
     2                  UCY2(128),ICFZ(128),ICFNT(128),UCHS(128),UCV(128),
     3                  ICTK(128),ICTK2(128),ICTNO(128),ICTSP(128)
      COMMON /UFC/      ICF(128),TCVN(128),FX(256),FY(256),
     1                  FVX(256),FVY(256),FZ(256),FV(256),FXY(256),
     2                  FX2(256),FY2(256),IFPZ(256),FHS(256),
     3                  IFSC(256),IFNC(256),IFVN(256),IFTNO(256),NFDR(256)
      COMMON /DATA1/    ECL(10,128),NCO,NCMX,NRJC
      COMMON /DATA2/    VCL(53,460)
      COMMON /DATA3/    VR(6,460)
      COMMON /DATA4/    FCL(9,256),AFCS,WFCS,NFMX,NFARM,KNIDC(1024),NFTA
      COMMON /DATA5/    NCL,NFL,JCL,JCTNO(128),WCX(256),WCY(256),JFT,
     1                  JFTNO(256),WFX(256),WFY(256),WF(256),IFMG(256),
     2                  ICMG(128),ICZC(128),NVMM
      COMMON /VPARM/    VX,VY,VX1,VY1,TMKTL,TMKTLL
      COMMON /FILTER/   TATRMN,AREAMN,CELMN(2),SVMX
      COMMON /AZENDS/   AZLO,AZHI,AZREF,ELOW,ELAVE,IFADAR
      COMMON /KNCTR/    LI,ITL(2),KLVL,JNTD,JNTDA(2,1024),
     +                  KNIDM(2),KNIDL(1,1024)
      COMMON /WIND/     SVA(14,8),SVB(14,8),SVC(14,8),
     +                  SA2(14,8),SB2(14,8),SC2(14,8),
     +                  SAB(14,8),SAC(14,8),SBC(14,8),
     +                  SV2(14,8),SB(14,8),SC(14,8),NUM(14,8)
C
      EQUIVALENCE(VCL(1,1),IVCL(1,1)),(FCL(1,1),IFCL(1,1))
C
      DATA KEY/3,2,1,65,4,2,1,61,4,2,0,69,4/,INPUT/'SRTD.SCR'/
      DATA FILSTAT/'NEW','OLD'/,IFILS/1/,STG/' ','*'/
      DATA ZERO/1/,ISTY/6/
      PARAMETER(LD=1000,FK=1000.,IZMX=60,DPR=57.29578)
```

```
      PARAMETER(RETA=57.29578,ARMX=99.9)
      PARAMETER(DAMAX=1000.,AMAX=1000.)
C
      OPEN(UNIT=3, FILE='INPUT', FORM='UNFORMATTED',
     +     STATUS='OLD',IOSTAT=IOFILS)
      IFILS=2
      REWIND 3
C
      NETA=0
      VXC=0
      VYC=0
      DNN=0
      DNS=0
      DCN=0
      DCS=0
      NSN=0
      NACT=0
      NACTV=0
      NCT=0
      NFT=0
      DFN=0
      DFS=0
      DCA=0
      DSC=0
        VXA=0
        VYA=0
      VXS=0
      VYS=0
        DNMN=0.
        DNMX=0.
        YNN=0.
        XNN=0.
        DCNM=0.
        DCNX=0.
        XCN=0.
        YCN=0.
        DCAM=0.
        DCAX=0.
        XSC=0.
        YSC=0.
        DFNM=0.
        DFNX=0.
        XFN=0.
      YFN=0.
      DO 79 I=1,10
      N2(I)=0
      N2S(I)=0
      N3(I)=0
      N3S(I)=0
      SUM2(I)=0.
      SUM3(I)=0.
      SPD2(I)=0.
   79 SPD3(I)=0.
      NVMM=NVMX
      IF(NVMX.EQ.NVARM) NVMM=NVMX-1
      DO 80 J=1,NFMX
        FX(J)=0.
        FY(J)=0.
        FVX(J)=0.
        FVY(J)=0.
```

```
          FZ(J)=0.
          FV(J)=0.
          FXY(J)=0.
          FX2(J)=0.
          FY2(J)=0.
          FHS(J)=0.
          IFSC(J)=0
          IFPZ(J)=0
          IFNC(J)=0
          IFVN(J)=0
          IFTNO(J)=0
   80   CONTINUE
        NGCT=0
        NTOT=0
        NFN=FNSN-1.
        DELTM=KTL-KTLL
        IF(DELTM.GT.0) VDM=1./DELTM
        IF(NFN.LE.0.OR.NVMX.LE.0) GO TO 559
C
C    LOOP THROUGH ACTIVE VOLUME CELLS
C
        DO 100 NV=1,NVMM
C
C    IF CELL NOT UPDATED THIS SCAN, GO AROUND
C
        ISO=0
        IF(IVCL(9,NV) .LE. 0) GO TO 102
        ICTR=IVCL(9,NV)
        JCTR=IVCL(29,NV)
        IF(IVCL(8,NV).LT.0) IVCL(8,NV)=-IVCL(8,NV)
        NTOT=NTOT+1
        IDTC=FLOAT(ICTR)/(FNSN-1.)*10.+.5
C
C    TEST QUALITY OF REFL ATTRIBS
C
        IF(IVCL(20,NV).GT.85) GO TO 102
        IF(VCL(11,NV).LE.0.) GO TO 101
C
C    COMPUTE MEAN REFL. CELL ATTRIBS
C
        DRC=1./VCL(11,NV)
        VCL(11,NV)=VCL(11,NV)/ICTR
        VCL(12,NV)=VCL(12,NV)*DRC
        VCL(13,NV)=VCL(13,NV)*DRC
        VCL(14,NV)=VCL(14,NV)*DRC
        VCL(15,NV)=VCL(15,NV)*DRC
        VCL(16,NV)=VCL(16,NV)*DRC
        VCL(17,NV)=VCL(17,NV)*DRC
        VCL(18,NV)=VCL(18,NV)*DRC
C
C    COMPUTE MEAN SHEAR CELL ATTRIBS
C
        IF(JCTR.LT.1000) GO TO 90
        JCTR=JCTR-1000
        IVCL(29,NV)=JCTR
        ISO=1
   90   IF(JCTR.LE.0) GO TO 95
        DTSC=1./VCL(30,NV)
        VCL(32,NV)=VCL(32,NV)*DTSC
        VCL(33,NV)=VCL(33,NV)*DTSC
```

```
      VCL(34,NV)=VCL(34,NV)*DTSC
      DTSC=1./(JCTR+ISO)
      VCL(30,NV)=VCL(30,NV)*DTSC
      VCL(31,NV)=VCL(31,NV)*DTSC
C
C     COMPUTE MEAN COMBINED CELL ATTRIBS
C
   95 DCC=1./(ICTR+JCTR)
      TSQNT=VCL(23,NV)*DCC*.1
      IF(TSQNT.GT.0.) VCL(23,NV)=10.**(TSQNT)
      V=VCL(24,NV)*DCC
      V2=VCL(25,NV)*DCC
      DS=VCL(26,NV)*DCC
      IF(ISHR.NE.5) DS=SQRT(DS)
      DELW=VCL(27,NV)*DCC
      VAR=ABS(V2-V*V)
      VCL(25,NV)=SQRT(VAR)
      VCL(26,NV)=DS
      VCL(24,NV)=V
C
      Z=VCL(11,NV)
      X=VCL(12,NV)
      Y=VCL(14,NV)
      XL=VCL(49,NV)
      YL=VCL(50,NV)
      VXYB=X*Y
      VXB2=X*X
      VYB2=Y*Y
      VVZB=VCL(23,NV)*Z
      IZPK=IVCL(20,NV)
      HSMT=VCL(22,NV)
C
C     IF CELL FOUND ON LT 40% OF SCANS, NO SPREAD
C
      IVCL(43,NV)=IDTC
      IF(ICTR.GE.4) VCL(42,NV)=VCL(13,NV)-VXB2+VCL(15,NV)-VYB2
      INTER=0
C
C     TEST FOR NOISE
C       1) CELL AREA LT CELMN
C       2) 1ST ELEV SCAN ONLY
C       3) 1ST OBSERVATION
C
      RES=VCL(4,NV)/VCL(7,NV)
      IADR=ISO+1
      IF(RES.LE.CELMN(IADR).AND.ICTR.LE.1.AND.
     1 IVCL(53,NV).LE.0) INTER=1
C
C     TEST FOR DOPPLER NOISE
C       1) DOP SPD GT 90% MAX
C       2) REFL LT 40 DBZ
C
      ZZZ=10.*ALOG10(Z)
      IF(VCL(26,NV).GE.SVMX.AND.ZZZ.LT.40.) INTER=2
      IF(VCL(19,NV).LT.10..AND.INTER.LT.1) GO TO 30
C
C     EXCLUDE REJECTED CELLS FROM CONTOUR ASSOCIATION
C
      IVCL(51,NV)=0
      GO TO 31
```

```
C
C     PREPARE FIXED CONTOUR TRACK DIRECTORY
C
   30 KNID=IVCL(37,NV)
      IF(KNID.LE.0.OR.KNID.GT.NKDMX) GO TO 31
      NF=KNIDC(KNID)
      IF(NF.LE.0.OR.NF.GT.NFMX) GO TO 31
      IVCL(37,NV)=NF
      MF=IVCL(51,NV)
      IF(MF.LE.0.OR.MF.GT.JFL) GO TO 32
      MF1=IFCL(9,NF)
      IF(MF.EQ.MF1) GO TO 32
      IF(MF1.GT.0.AND.MF1.LE.JFL) GO TO 33
      IFCL(9,NF)=MF
      GO TO 32
C
   33 IF(WF(MF).LT.WF(MF1)) GO TO 34
      IFCL(9,NF)=MF
      MF=MF1
   34 MF1=IFCL(7,NF)
      IF(MF1.GT.0.AND.MF1.LE.JFL) GO TO 35
      IFCL(7,NF)=MF
      GO TO 32
C
   35 NF1=IFMG(MF1)
      IF(NF1.LE.0.OR.NF1.GT.NFMX) GO TO 351
      IF(FCL(5,NF).LE.FCL(5,NF1)) GO TO 352
  351 IFMG(MF1)=NF
  352 IF(MF.EQ.MF1) GO TO 32
      IF(WF(MF).LT.WF(MF1)) GO TO 32
      IFCL(7,NF)=MF
      GO TO 32
C
C    CELL NOT IN CONTOUR, SET NF=0
C
   31 IVCL(37,NV)=0
C
C    NORMAL UPDATE, INITIALISE VELOCITY
C
   32 IVCL(44,NV)=0
      IVCL(45,NV)=0
      IVCL(46,NV)=0
      VTX=VCL(47,NV)
      VTY=VCL(48,NV)
      VXT=VTX
      VYT=VTY
      IVCAL=0
      IF(IVCL(53,NV).EQ.0 .OR. DELTM.EQ.0) GO TO 40
C
C    CELL UPDATED, MAKE POSITIVE SET ON CELL COUNTER INCREMENT
C
      IVCAL=1
      VXT=(X-XL)*VDM
      VYT=(Y-YL)*VDM
      VCL(45,NV)=VXT
      VCL(46,NV)=VYT
C
C    TEST VELOCITY AGAINST EXPECTED VELOCITY
C
      NTEST=.TRUE.
```

```
      IF(NTEST) GO TO 40
      KTV=IVCL(53,NV)
      IF(KTV.GT.4) KTV=4
      IF(DELW.LE.3.) GO TO 40
      AV=ABS(VCL(45,NV)-VCL(47,NV))
      IF(AV.LE.VDM) GO TO 322
      IF(AV.GT.ABS(VCL(47,NV))*VDPT(KTV)) GO TO 321
  322 AV=ABS(VCL(46,NV)-VCL(48,NV))
      IF(AV.LE.ABS(VCL(48,NV))*VDPT(KTV)) GO TO 40
C
C   IF VELOCITY TOO LARGE, DROP TRACK
C
  321 VXT=VX
      VCL(47,NV)=VX
      VYT=VY
      VCL(48,NV)=VY
      VCL(45,NV)=0.
      VCL(46,NV)=0.
      VCL(49,NV)=0.
      VCL(50,NV)=0.
      IVCL(53,NV)=0
      IVCL(44,NV)=3
      VCL(41,NV)=TVCL(8,NV)
      TVCAL=0
      NT=NT+1
      IVCL(8,NV)=NT
C
C   UPDATE SMOOTHED TRACKING VELOCITY
C   IF NO CELL UPDATE DEFAULT TO PRIOR SCAN VELOCITY
C
   40 VCL(47,NV)=A1*VXT+A2*VCL(47,NV)+A3*VX
      VCL(48,NV)=A1*VYT+A2*VCL(48,NV)+A3*VY
      VXN=VCL(45,NV)
      VYN=VCL(46,NV)
C
C   REJECT DOPPLER NOISE
C
      IF(INTER.EQ.2) GO TO 580
C
C   REJECT SURFACE HEIGHT GT 10 KM
C
      IF(VCL(19,NV).GE.10.) GO TO 582
C
C   REJECT NOISE
C
      IF(INTER.GE.1) GO TO 583
      IF(NVSCN.LE.1) GO TO 57
C
C   GROUND CLUTTER TEST
C     1) VELOCITY NEAR 0
C     2) FOUND ON LT 20% OF SCANS
C     3) VELOCITY UPDATED (TRACKED)
C
      IF(ABS(VXN).LE..0002.AND.ABS(VYN).LE..0002.AND
     1  .IDTC.LE.2.AND.IVCAL.EQ.1)  GO TO 581
C
C   EXCLUDE CELLS WITH EXTREAM ATTRIBUTES FROM AVG VEL
C
      IF(IVCAL.EQ.0) GO TO 57
      IF(IZPK.GT.IZMX) GO TO 57
```

```
C      IF(IVCL(38,NV).NE.0.OR.IVCL(52,NV).NE.0) GO TO 57
C      IF(VCL(19,NV).GT.HM) GO TO 57
       IF(VCL(42,NV).GT.SPRM) GO TO 57
C
C    SUM VELOCITY VALUES
C
       NSN=NSN+1
       VXC=VXC+VXN
       VYC=VYC+VYN
       IVCL(44,NV)=1
C
C    NORMAL PROCESSING ON ACTIVE (UPDATED) CELLS
C
    57 NACT=NACT+1
       NACTV=NACTV+IVCAL
       IF(IVCL(44,NV).LT.2) IVCL(44,NV)=IVCL(44,NV)+1
       UX(NACT)=X
       UY(NACT)=Y
       VXA=VXA+VXN
       VYA=VYA+VYN
       ISET=0
C
C    SUM SPREAD AND TRACK ERROR ON UPDATTED CELLS
C
       IREJ=IVCL(44,NV)
       IF(IREJ.EQ.1) GO TO 610
       SPD=VCL(42,NV)
       DX=(XL+VTX*DELTM)-X
       DY=(YL+VTY*DELTM)-Y
       DXY=DX*DX + DY*DY
       IAGE=IVCL(53,NV)+1
       IF(IAGE.LE.1 .OR. IAGE.GT.10) GO TO 610
       IF(IREJ.EQ.2) GO TO 609
       IF(IREJ.NE.3) GO TO 610
C
       SUM3(IAGE)=SUM3(IAGE)+DXY
       SPD3(IAGE)=SPD3(IAGE)+SPD*SPD
       IF(SPD.GT.0.) N3S(IAGE)=N3S(IAGE)+1
       N3(IAGE)=N3(IAGE)+1
       GO TO 610
C
   609 SUM2(IAGE)=SUM2(IAGE)+DXY
       SPD2(IAGE)=SPD2(IAGE)+SPD*SPD
       IF(SPD.GT.0.) N2S(IAGE)=N2S(IAGE)+1
       N2(IAGE)=N2(IAGE)+1
C
   610 CONTINUE
C
C     DEFINE A SIGNIFICANT CELL (IVCL(8,NV).LT.0)
C
       IF(ICTR.LE.1) GO TO 611
       IF((IDTC.GT.MNSN.AND.IVCL(10,NV).GT.MHSN
      +         .AND.VCL(42,NV).LE.SPRM)
      +      .OR.(IZPK.GT.MZSN.AND.IDTC.GE.3
      +         .AND.IVCL(10,NV).GT.0))
      +     IVCL(8,NV)=-IABS(IVCL(8,NV))
C
C    FIND BASE CLUSTER ID (ICLT)
C
   611 ICL=IVCL(38,NV)
```

```
   61 IF(ICL.LE.0.OR.ICL.GT.ICLN) GO TO 62
      ICLT=ICLIST(ICL)
      IF(ICLT.EQ.0) GO TO 62
      ICLA=IABS(ICLT)
      IF(ICLA.EQ.ICL) GO TO 63
      ICL=ICLA
      GO TO 61
C
C    TEST IF CELL: ASSIGNED CLUSTER ID ON PRIOR SCAN(ICLT.LT.0)
C                  NOT IN A CLUSTER(ICLT.EQ.0)
C                  ASSIGNED CLUSTER ID ON CURRENT SCAN(ICLT.GT.0)
C
   63 IF(ICLT) 64,62,65
C
C    CELL NOT IN CLUSTER, TEST FOR ISO SIG CELL
C
   62 IVCL(52,NV)=0
      IVCL(38,NV)=0
      ISET=1
      IF(IVCL(8,NV).LT.0) GO TO 67
      GO TO 70
C
C    1ST TIME, SET CLUSTER ATTRIBUTE VALUES POSITIVELY
C
   65 ICLIST(ICL)=-ICLIST(ICL)
   67 NCL=NCL+1
      IVCL(38,NV)=0
      IF(ISET.EQ.0) IVCL(38,NV)=NCL
      UCX(NCL)=X
      UCY(NCL)=Y
      UCN(NCL)=1-ISET
      UCZ(NCL)=Z
      UCVX(NCL)=VXN
      UCVY(NCL)=VYN
      ICVN(NCL)=IVCAL
      UCXY(NCL)=VXYB
      UCX2(NCL)=VXB2
      UCY2(NCL)=VYB2
      ICPZ(NCL)=IZPK
      IF(ISET.EQ.0) ICPNT(ICL)=NCL
      UCHS(NCL)=HSMT
      UCV(NCL)=VVZB
      ICTK(NCL)=IVCL(52,NV)
      ICTK2(NCL)=0
      ICF(NCL)=IVCL(37,NV)
      GO TO 70
C
C    INCREMENT CELL COUNTER AND SUM ATTRIBUTES
C
   64 NC=ICPNT(ICL)
      IVCL(38,NV)=0
      IF(NC.LE.0.OR.NC.GT.ICLN) GO TO 70
      UCN(NC)=UCN(NC)+1.
      IVCL(38,NV)=NC
      IF(ICF(NC).LE.0) ICF(NC)=IVCL(37,NV)
      IF(IVCL(37,NV).LE.0) IVCL(37,NV)=ICF(NC)
      UCX(NC)=UCX(NC)+X
      UCY(NC)=UCY(NC)+Y
      UCZ(NC)=UCZ(NC)+Z
      UCVX(NC)=UCVX(NC)+VXN
```

```
      UCVY(NC)=UCVY(NC)+VYN
      ICVN(NC)=ICVN(NC)+IVCAL
      ICPZ(NC)=MAXO(ICPZ(NC),IZPK)
      IF(ICPZ(NC).EQ.IZPK) ICF(NC)=IVCL(37,NV)
      UCXY(NC)=UCXY(NC)+VXYB
      UCX2(NC)=UCX2(NC)+VXB2
      UCY2(NC)=UCY2(NC)+VYB2
      UCHS(NC)=AMAX1(UCHS(NC),HSMT)
      UCV(NC)=UCV(NC)+VVZB
      MC=IVCL(52,NV)
      IF(MC.LE.0.OR.MC.GT.JCL) GO TO 70
      MC1=ICTK(NC)
      IF(MC1.EQ.MC) GO TO 70
      IF(MC1.GT.0.AND.MC1.LE.JCL) GO TO 661
      ICTK(NC)=MC
      GO TO 70
  661 IF(ICZC(MC).LT.ICZC(MC1)) GO TO 662
      ICTK(NC)=MC
      MC=MC1
  662 MC1=ICTK2(NC)
      IF(MC1.GT.0.AND.MC1.LE.JCL) GO TO 663
      ICTK2(NC)=MC
      GO TO 70
  663 ICMG(MC1)=NC
      IF(MC.EQ.MC1) GO TO 70
      IF(ICZC(MC).LT.ICZC(MC1)) GO TO 70
      ICTK2(NC)=MC
C
C     FIXED CONTOUR ATTRIBUTES
C
   70 NF=IVCL(37,NV)
      IF(NF.LE.0.OR.NF.GT.NFMX) GO TO 100
      IF(IFCL(8,NF).GT.0) GO TO 71
C
C    1ST OCCURENCE, STE ATTRIBUTES POSITIVELY
C
      IFCL(8,NF)=1
      FX(NF)=VCL(12,NV)
      FY(NF)=Y
      FVX(NF)=VXN
      FVY(NF)=VYN
      IFVN(NF)=IVCAL
      FZ(NF)=Z
      FV(NF)=VVZB
      FXY(NF)=VXYB
      FX2(NF)=VXB2
      FY2(NF)=VYB2
      IFPZ(NF)=IZPK
      FHS(NF)=HSMT
      IFSC(NF)=0
      IF(IVCL(8,NV).LT.0.AND.IVCL(38,NV).EQ.0) IFSC(NF)=1
      IFNC(NF)=0
      GO TO 100
C
C    INCREMENT CELL COUNTER AND SUM ATTRIBUTES
C
   71 IFCL(8,NF)=IFCL(8,NF)+1
      FX(NF)=FX(NF)+X
      FY(NF)=FY(NF)+Y
      FVX(NF)=FVX(NF)+VXN
```

```
      FVY(NF)=FVY(NF)+VYN
      IFVN(NF)=IFVN(NF)+IVCAL
      FZ(NF)=FZ(NF)+Z
      FV(NF)=FV(NF)+VVZB
      FXY(NF)=FXY(NF)+VXYB
      FX2(NF)=FX2(NF)+VXB2
      FY2(NF)=FY2(NF)+VYB2
      IFPZ(NF)=MAXO(IFPZ(NF),IZPK)
      FHS(NF)=AMAX1(FHS(NF),HSMT)
      IF(IVCL(8,NV).LT.0.AND.IVCL(38,NV).EQ.0)
     +     IFSC(NF)=IFSC(NF)+1
      GO TO 100
C
C   DOP SPREAD
C
  580 IVCL(44,NV)=6
      IVCL(38,NV)=0
      GO TO 100
C
C   GROUND CLUTTER
C
  581 IVCL(44,NV)=7
      IVCL(38,NV)=0
      NGCT=NGCT+1
      GO TO 100
C
C   BASE TOO HIGH
C
  582 IVCL(44,NV)=4
      IVCL(38,NV)=0
      GO TO 100
C
C   NOISE FLAG
C
  583 IVCL(44,NV)=5
      IVCL(38,NV)=0
      GO TO 100
C
C   REFL VALUES OUT OF RANGE, DO NOT UPDATE CELL THIS SCAN
C
  101 IVCL(9,NV)=0
C
C   CELL NOT UPDATED, CLEAR CLUSTER POINTER
C
  102 IVCL(37,NV)=0
      IVCL(38,NV)=0
C
C   END VOLUME CELL LOOP, FLAG ISOLATED TS CELL ASSOCIATION
C
  100 IVCL(40,NV)=ISO
```

```
C
C    COMPUTE CLUSTER ATTRIBUTES AND NEAREST NEIGHBOR DISTANCES
C
       CALL CTRAK
       IF(NACT.GE.NNMIN) CALL NEARN(UX,UY,NACT,DNN,DNS,XNN,YNN,DNMN,DNMX
C
C    COMPUTE AND OUTPUT FIXED CONTOUR ATTRIBUTES
C
       CALL FTRAK
```

```
C
      IF(NTOT.LE.0) GO TO 541
      IF(PRCELL.OR.PRSIG) WRITE(6,1001) (H1(I,ISHR),I=1,3),
     +                                  (H2(I,ISHR),I=1,3),
     +                                  (H3(I,ISHR),I=1,3)
C
 1001 FORMAT(//,1X,' VOLUME CELL OUTPUT'//
     1 ,1X,'    CENTROID -  - Z -  -  HGT  '
     2   ,'VBAR CELL SPACIAL  (',A3,') ',A3,2X,A3,' RAD CS CN D F R'/
     3 ,1X,' TRK  E.  N. AV PK LW HI L M H '
     4   ,'EM/S NM/S SPRD  A  (',A3,') ',A3,2X,A3,' SPD TR TR O E E'/
     5 ,1X,'  NO  KM  KM DB DB DB DB W N I '
     6   ,'   OLD ID  KM  KM2 (',A3,') ',A3,2X,A3,' M/S NO NO P T J')
C
C   OUTPUT VOLUME CELL ATTRIBUTES
C
      DO 150 NV=1,NVMM
      IF(IVCL(9,NV).LE.0) GO TO 200
      IF(.NOT.PRNOIS .AND. IVCL(44,NV).GT.3) GO TO 58
      VX1=VCL(45,NV)*FK
      VY1=VCL(46,NV)*FK
      VX2=VCL(47,NV)*FK
      VY2=VCL(48,NV)*FK
      IZVAL=10.*ALOG10(VCL(11,NV))
      AREA=VCL(17,NV)
C
C   LOOKUP TRACK ID'S
C
      IFXNO=0
      NF=IVCL(37,NV)
      IF(NF.GT.0.AND.NF.LE.NFMX) IFXNO=IFTNO(NF)
      ICXNO=0
      ICL=IVCL(38,NV)
      IF(ICL.GT.0.AND.ICL.LE.NCL) ICXNO=ICTNO(ICL)
      ITRKNO=IABS(IVCL(8,NV))
C
C   HANG SIGNIFICANCE FLAG ON AGE
C
      IVCL(53,NV)=IVCL(53,NV)+1
      IAGE=IVCL(53,NV)
      IF(IVCL(8,NV).LT.0) IAGE=-IAGE
C
C   HANG CONTOUR TRACK EXTRAPOLATION FLAG ON ASSN COUNTER
C
      NTSHT=IVCL(29,NV)+IVCL(40,NV)
      NREHT=IVCL(9,NV)-IVCL(40,NV)
      NASSN=NREHT*LD+NTSHT
      IF(IVCL(51,NV).LT.0) NASSN=-NASSN
C
C   OUTPUT VOLUME CELL DATA TO SORT ROUTINE
C
      SHRT=VCL(23,NV)
      SHR2=VCL(26,NV)
      IF(ISHR.NE.5) GO TO 56
      SHR2=SHRT
      SHRT=VCL(26,NV)
C
   56 WRITE(3) KTL,IZVAL,VCL(12,NV),VCL(14,NV),VX2,VY2,
     1          AREA,VCL(18,NV),VCL(42,NV),
     2          SHRT,VCL(24,NV),SHR2,
```

```
     3          NASSN,IAGE,ITRKNO,ICXNO,IFXNO,
     4          ISIX
C
C    OUTPUT ALL CELLS ON PRCELL
C
       IF(PRCELL) GO TO 59
C
C    OUTPUT ONLY SIG CELLS ON PRSIG
C
       IF(.NOT.PRSIG) GO TO 58
       IF(IVCL(8,NV).GE.0) GO TO 58
   59  ISIG=1
C
       IF(IVCL(8,NV).LT.0) ISIG=2
       IX=VCL(12,NV)
       IY=VCL(14,NV)
       IHB=VCL(19,NV)
       IHM=VCL(18,NV)
       IHS=VCL(22,NV)
       IF(AREA.GT.ARMX) AREA=ARMX
C
C    HANG CONTOUR TRACK EXTRAPOLATION FLAG ON IFXNO
C
       IF(IVCL(51,NV).LT.0) IFXNO=-IFXNO
       IF(ITRKNO.LT.IAMAX) GO TO 53
       ITR=ITRKNO/IAMAX
       ITRKNO=ITRKNO-ITR*IAMAX
   53  CONTINUE
C
C    IF CELL TRACK DROPPED DUE TO EXCESSIVE VELOCITY
C    TAG CELL WITH PRIOR TRACK ID
C
       IF(IVCL(44,NV).NE.3) GO TO 55
       VY2=VCL(41,NV)
       VX2=IFIX(VY2/AMAX)
       VY2=VY2-VX2*IAMAX
   55  CONTINUE
C
C    OUTPUT VOLUME CELL SUMMARY
C
       WRITE(6,1005) ITRKNO,SIG(ISIG),IX,IY,
     1               IZVAL,IVCL(20,NV),IVCL(3,NV),IVCL(21,NV),
     2               IHB,IHM,IHS,VX2,VY2,VCL(42,NV),AREA,
     3               SHRT,SHR2,VCL(24,NV),VCL(25,NV),
     4               ICXNO,IFXNO,NTSHT,NREHT,IVCL(44,NV)
C
 1005  FORMAT(1X,I3,A1,2I4,4I3,3I2,2F5.1,F5.2,F4.1,3F5.1,F4.1,2I3,3I2)
C
C    SAVE PHYSICAL ATTRIBS AND TRACK ID'S FOR NEXT SCAN ASSN.
C
   58  VCL(49,NV)=VCL(12,NV)
       VCL(50,NV)=VCL(14,NV)
       IVCL(51,NV)=IVCL(37,NV)
       IVCL(52,NV)=IVCL(38,NV)
C
       VR(1,NV)=VCL(12,NV)
       VR(2,NV)=VCL(14,NV)
       VR(3,NV)=IZVAL
       VR(4,NV)=VCL(17,NV)
       VR(5,NV)=VCL(18,NV)
```

```
      VR(6,NV)=VCL(6,NV)
C
C   CLEAR CELL ACCUMULATORS
C   LEAVE CELL BASE AND TRACK ATTRIBUTES
C
      DO 41 I=9,46
   41 IVCL(I,NV)=0
      GO TO 150
C
C   CELL NOT UPDATED, CLEAR NV SLOT FOR A NEW CELL
C
  200 CONTINUE
      IF(IVCL(53,NV) .LE. 0) GO TO 150
      DO 160 I=1,53
  160 IVCL(I,NV)=0
      NVMIN=MIN0(NV,NVMIN)
  150 CONTINUE
C
C   OUTPUT CLUSTER ATTRIBUTES
C
      CALL COUT
C
C   COMPUTE AVG CELL VELOCITY OF ENTIRE SCAN
C
      IF(NSN .EQ. 0) GO TO 541
      VN=NSN
      IF(NSN.LE.10) GO TO 541
      IF(NACT.LT.10) GO TO 541
      IF(VN/FLOAT(NACT).LT.FNSRN) GO TO 541
      VX=B1*VXC/VN+B2*VX
      VY=B1*VYC/VN+B2*VY
C
C   UPDATE DEFAULT VELOCITY ON SCAN WITH MAXIMUM NUMBER ACTIVE CELLS
C
      IF(NACT.LT.NACTT) GO TO 54
      NACTT=NACT
      VXI=VX
      VYI=VY
      NTEST=.FALSE.
      GO TO 54
C
C   NO CELLS UPDATED CURRENT SCAN, RESET TO DEFAULT VELOCITY
C
  541 VX=VXI
      VY=VYI
      NTEST=.TRUE.
C
C   INITIALISE CELL TRACK WITH DEFAULT VELOCITY
C
   54 DO 43 I=1,NVARM
      IF(IVCL(53,I).GT.0)GO TO 43
      VCL(47,I)=VX
      VCL(48,I)=VY
   43 CONTINUE
      VXP=VX*FK
      VYP=VY*FK
C
C   COMPUTE MEAN SQUARE SPREAD & TRACKING ERROR
C
      DO 46 IA=1,10
```

```
      IF(N2(IA).EQ.0) GO TO 45
      SUM2(IA)=SQRT(SUM2(IA)/N2(IA))
      IF(N2S(IA).GT.0) SPD2(IA)=SQRT(SPD2(IA)/N2S(IA))
C
   45 CONTINUE
      IF(N3(IA).EQ.0) GO TO 46
      SUM3(IA)=SQRT(SUM3(IA)/N3(IA))
      IF(N3S(IA).GT.0) SPD3(IA)=SQRT(SPD3(IA)/N3S(IA))
C
C    COMPUTE NEAREST NEIGHBOR DISTANCES ON:
C       1) SC'S
C       2) CLUSTERS
C
   46 CONTINUE
      IF(NCL.GE.NNMIN) CALL NEARN(WCX,WCY,NCL,DCN,DCS,XCN,YCN,DCNM,DCNX
      IF(JCL.GE.NNMIN) CALL  NEARN(UCX,UCY,JCL,DCA,DSC,XSC,YSC,DCAM,DCAX)
C
C    OUTPUT SCAN SUMMARY
C
  559 CONTINUE
      IF(.NOT.PRHEAD) GO TO 549
      WRITE(6,1003)
 1003 FORMAT(///1X,' VOL HHMM AREA WFLUX NEAR NEIGHBOR '
     1           ,' ACT NO NO VELOCITY  TRK CLS CNT G   OVER'/
     2          ,1X,'SCAN      KKM2 KMT/H CELL CLST CONT'
     3           ,' VCL CS FC EM/S NM/S  NO CTR CTR C')
C
      IJ=(IO+JO)*.1+.9
      NVO=NVO*.1+.9
      NCO=NCO*.1+.9
      NFO=NFO*.1+.9
      ICO=ICO*.1+.9
C
      WRITE(6,1004) NVSCN,JHR,JMIN,AFCS,WFCS,DNN,DCN,
     1 DFN,NACT,NCL,NFL,VXP,VYP,NT,NCLN,NFLN,NGCT,NVO,NCO,NFO,
     2 ICO,IJ
 1004 FORMAT(1X,I4,I3.2,I2.2,F5.1,F6.2,3F5.1,I4,2I3,2F5.1,
     1       3I4,6I2)
C
      WRITE(6,1006)
 1006 FORMAT('0','AGE       UPDATTED         REJECTED'/
     +            7X,'CELL  TRK  AVG    CELL  TRK  AVG'/
     +            7X,'CNTR  ERR  SPD    CNTR  ERR  SPD')
C
      DO 49 IA=1,10
      WRITE(6,1007) IA,N2(IA),SUM2(IA),SPD2(IA),
     +                 N3(IA),SUM3(IA),SPD3(IA)
   49 CONTINUE
 1007 FORMAT(1X,I3,2(3X,F4.0,F5.1,F5.2))
C*
C*   COMPUTE BACKGROUND WIND
C*
      IF(PRHEAD) GO TO 549
      WRITE(6,44)
   44 FORMAT('0 AZMUTH HT  DIR  MAG  VX  DEV   VY  DEV  CNTR     DEL'/
     #       2X,'RANGE  KM  DEG  M/S  MS  M/S   MS  M/S')
C
C   LOOP THROUGH EACH AZMUTH OCTANT
C
      DO 47 J=1,8
```

```
      IOH=J*45
      IOL=IOH-45
C
C     LOOP THROUGH EACH ALTITUDE STEP
C
      DO 48 N=1,14
      NP=NUM(N,J)
      IF(NP.LT.10) GO TO 52
      ANP=NP-2
      DNP=1./NP
C
C     A=SIN(EL)
C     B=COS(EL)*COS(AZ)
C     C=COS(EL)*SIN(AZ)
C     V=RADIAL VELOCITY
C
      PA2=SA2(N,J)
      PB2=SB2(N,J)
      PC2=SC2(N,J)
      PVA=SVA(N,J)
      PVB=SVB(N,J)
      PVC=SVC(N,J)
      PAB=SAB(N,J)
      PAC=SAC(N,J)
      PBC=SBC(N,J)
      PV2=SV2(N,J)
      PB=SB(N,J)
      PC=SC(N,J)
C
C     SOLVE FOR 3 WIND COMPONENTS
C
C       DEL1 = PA2*PB2*PC2 + PAB*PBC*PAC + PAC*PAB*PBC
C      #      - PB2*PAC*PAC - PBC*PBC*PA2 - PC2*PAB*PAB
C       WPR = PVA*PB2*PC2 + PAB*PBC*PVC + PAC*PVB*PBC
C      #     - PVC*PB2*PAC - PBC*PBC*PVA - PC2*PVB*PAB
C       VPR = PA2*PVB*PC2 + PVA*PBC*PAC + PAC*PAB*PVC
C      #     - PAC*PVB*PAC - PVC*PBC*PA2 - PC2*PAB*PVA
C       UPR = PA2*PB2*PVC + PAB*PVB*PAC + PVA*PAB*PBC
C      #     - PAC*PB2*PVA - PBC*PVB*PA2 - PVC*PAB*PAB
C
C     SOLVE FOR HORIZ WIND COMPONENTS ONLY
C
      DEL=PB2*PC2 - PBC*PBC
      IF(ABS(DEL).LT.0.001) GO TO 52
      VPR=PVB*PC2 - PVC*PBC
      UPR=PB2*PVC - PBC*PVB
      WPR=0.
      DDEL=1./DEL
      WPR=WPR*DDEL
      VPR=VPR*DDEL
      UPR=UPR*DDEL
C
C     COMPUTE ERROR BOUNDS
C
      R=PV2 + VPR*VPR*PB2 + UPR*UPR*PC2
     #      - 2.0*VPR*PVB - 2.0*UPR*PVC
     #      + 2.0*VPR*UPR*PBC
      R=R*DNP
      R=DSQRT(R)/SQRT(ANP)
      SIGV=-99.
```

```
      DNOM=DSQRT(PB2-PB*PB*DNP)
      IF(DNOM.GT.0.) SIGV=R/DNOM
C
      SIGU=-99.
      DNOM=DSQRT(PC2-PC*PC*DNP)
      IF(DNOM.GT.0.) SIGU=R/DNOM
C
      IDIR=0
      IF(UPR.NE.0..AND.VPR.NE.0.) IDIR=ATAN2(UPR,VPR)*DPR
      IF(IDIR.LT.0) IDIR=IDIR+360
      AMAG=SQRT(UPR*UPR + VPR*VPR)
C
      IF(ABS(WPR).GE.1000.) WPR=-99.
      IF(ABS(VPR).GE.1000.) VPR=-99.
      IF(ABS(UPR).GE.1000.) UPR=-99.
      IF(AMAG.GE.1000.) AMAG=-99.
C
      WRITE(6,51) IDL,IOH,N,IDIR,AMAG,UPR,SIGU,VPR,SIGV,NP,DEL
   51 FORMAT(2(1X,I3.3),I3,I5,F5.0,2(F5.0,F5.1),I5,E13.5)
   52 NUM(N,J)=0
      SB(N,J)=0.
      SC(N,J)=0.
      SA2(N,J)=0.
      SB2(N,J)=0.
      SC2(N,J)=0.
      SV2(N,J)=0.
      SVA(N,J)=0.
      SVB(N,J)=0.
      SVC(N,J)=0.
      SAB(N,J)=0.
      SAC(N,J)=0.
   48 SBC(N,J)=0.
   47 CONTINUE
C
  549 CONTINUE
C
      IF(NACTV.LE.0) GO TO 550
      VN=FK/NACTV
      VXS=VXA*VN
      VYS=VYA*VN
  550 KNCL=NCL+LD
      KNFL=NFTA+LD
C
      ITLS=LD*ITL(2)+ITL(1)
C   OUTPUT VOLUME SCAN SUMMARY TO SORT ROUTINE
C
      WRITE(3) KTL,ITLS,NVSCN,AZLO,AZHI,VXS,VYS,AFCS,
     1         WFCS,DNN,DCN,DCA,AZREF,IRADAR,NACT,KNCL,KNFL,
     2         IONE
C
      CLOSE(3)
C
C   SORT TRACK DATA ON:
C     1)  CONTOUR ID
C     2)  CLUSTER ID
C     3)  DATA TYPE
C
      CALL SORT(NVSCN,INPUT,1,1,KEY,13)
C
C   RESET COUNTERS FOR NEXT VOLUME SCAN
```

```
C
      JCL=NCL
      NVO=0
      ICO=0
      IO=0
      JO=0
      NRJC=0
      ICLN=1
      FNSN=1.009
      NTL=NT
C
C    SET REFERENCE TIME TO CURRENT SCAN TIME
C
      JDAY=IDAY
      JHR=IHR
      JMIN=IMIN
      JSEC=ISEC
      TML=TSEC
      KTLL=KTL
      RETURN
      END
```

```
      SUBROUTINE FTRAK
      LOGICAL PRCELL,PRSIG,PRFIXC,PRCLUS,PRSCAN,PRHEAD,PRNOIS,ALIT
     +        PRINT1,COPLOT,CEPLOT,CONTRZ,CONTRV,PROVER,
     +        COMPLT
      CHARACTER*3 FILSTAT(2)
      CHARACTER*8 INPUT
      INTEGER*2 KEY(17),KEY1(13)
      INTEGER TL,BEGINT,ENDIT,BDAY,EDAY
      DIMENSION IFCL(9,256),UCFX(128),UCFY(128),
     +          IVCL(53,460),IWAGF(256),IFAGL(256),
     +          DAT(12,256),NUM(256),XY(4),
     +          CCFX(256),CCFY(256)
C
      COMMON /FDIR/   IFCDIR(512)
      COMMON /DATA2/ VCL(53,460)
      COMMON /PNTRS/ NVMIN,NVMX,IELSN,NSCAN,IESNL,NVSCN,NT
      COMMON /DATA5/ NCL,NFL,JCL,JCTNO(128),WCX(256),WCY(256),JET,
     1               JETNO(256),WFX(256),WFY(256),WF(256),IFMG(256),
     2               ICMG(128),ICZC(128),NVMM
      COMMON /NVLIS/ NVARM,NCARM,NVO,NFO,ICO,IO,JO,JYR,IBL,KTL
      COMMON /NVLIT/ KTLL,NKNID,NKDO,IZTH,NKDMX,ITHR,IFXC(1024),HTST
      COMMON /DATA4/ FCL(9,256),AFCS,WFCS,NFMX,NFARM,KNIDC(1024),NFTA
      COMMON /FLGS/   PRCELL,PRSIG,PRFIXC,PRCLUS,PRSCAN,PRHEAD,
     +                PRNOIS,PROVER
      COMMON /INTL/   MHSN,MNSN,HM,FNSN,FNSRN,NCLN,NFLN,MZSN,NNMIN,ICA7
      COMMON /UVC/    UX(512),UY(512),UCY(128),UCY(128),UCVX(128),
     1               UCVY(128),UCN(128),UCZ(128),UCXY(128),UCX2(128),
     2               UCY2(128),ICPZ(128),ICPNT(128),UCHS(128),UCV(128),
     3               ICTK(128),ICTK2(128),ICTNO(128),ICTSP(128)
      COMMON /UFC/    ICF(128),ICVN(128),FX(256),FY(256),FVX(256),
     1               FVY(256),FZ(256),FV(256),FXY(256),FX2(256),
     2               FY2(256),IFFZ(256),FHS(256),IFSC(256),
     3               IFNC(256),IFVN(256),IFTNO(256),NFDP(256)
      COMMON /ZLOOK/ JZOFF,ZARY(91),RRATE(91)
      COMMON /KNCTR/ LT,ITL(2),KLVL,JNID,JNIDA(2,1024),
     +               KNIDM(2),KNTDL(1,1024)
      COMMON /PARM/   PRINT1,COPLOT,CEPLOT,CONTRZ,CONTRV,CALIBO,
     +               NUMF,NUMR
      COMMON /DECODE/UP(9),HEIGHT,DLONG,DLAT
C
      EQUIVALENCE (FCL(1,1),IFCL(1,1)),(VCL(1,1),IVCL(1,1))
C
C    SET JNMX TO DIMENSION SIZE OF DAT AND NUM
C    SET NCTR TO 4X NUMBER OF XY PAIRS TO BE OUTPUT IN ONE RECORD
C
      PARAMETER(RTD=57.29578,FK=1000.,LD=1000,JNMX=256,NCTR=12)
      DATA IZERO/0/,ITWO/2/,ITHREE/3/,IFOUR/4/,ISEVEN/7/
      DATA FILSTAT/'NEW','OLD'/,IFILS/1/,INPUT/'CNTR.SCR'/
      DATA KEY/4,2,0,61,4,2,1,65,4,2,0,5,4,2,0,57,4/
      DATA KEY1/3,2,1,65,4,2,0,5,4,2,0,57,4/
C
      NFL=0
      IF(NFMX.LE.0) RETURN
      IF(KTL.GT.KTLL) VKAL=FK/(KTL-KTLL)
C
      IF(PRFIXC) WRITE(6,1000)
 1000 FORMAT(1X,' FIXED CONTOUR OUTPUT'//
     1 1X,'    CENTROID AV CELL    Z     N N N SPR SPR  O '
     2 ,'  WTR AREA VELOCITY NEAR MX MR SP'/
     3 1X,' TRK  E.   N.   E.   N. AV PK  V S C   X   L   R '
```

```
     4 ,' FLUX XSCN  AV CELL DIST HT ID 1'/
     5 1X,'  NO  KM  KM  KM  KM DB DB  C (  L  KM  KM  I '
     6 ,' MT/H KKM2 EM/S NM/S KM  KM NO NO')
C
C    ZERO COMPLETE CONTOUR ACCUMULATORS
C
      NCC=0
      NCI=0
      NCV=0
      NCS=0
      ARCC=0.
      WAB=0.
      WSB=0.
      ARB=0.
      WFB=0.
      CI=0.
C
C    LOOP THROUGH ALL ACTIVE CONTOURS
C
      DO 10 I=1,NFMX
      COMPLT=.TRUE.
      FSIG=FCL(1,I)*FCL(2,I)
      IF(IFCL(6,I).LT.0 .OR. FSIG.LT.FCAZ) COMPLT=.FALSE.
C     IF(FSIG.LT.FCAZ) COMPLT=.FALSE.
      IFINO(I)=0
      IZ=FCL(2,I)
      WFC=FCL(5,I)
      AREA=FCL(1,I)
      IFTSP=0
      NSTG=0
      VFX=0.
      VFY=0.
      NFV=0
      NSC=0
      SPRDC=0.
      SPRDL=0.
      SPRDA=0.
      ANGC=0.
      DFN=0.
      DFS=0.
      DVFN=0.
      DVFS=0.
      XVFN=0.
      YVFN=0.
      DVFM=0.
      DVFX=0.
C
C    COMPUTE AVG CELL VELOCITY ON IFUN(UPDATTED CELL COUNTER)
C
      IFLAG=0
      IF(IFCL(8,I).LE.0.OR.FZ(I).LE.0.) IFLAG=1
      IF(IFLAG.EQ.1) GO TO 90
      IF(IFUN(I).LE.0) GO TO 12
      FN1=FK/IFUN(I)
      FVX(I)=FVX(I)*FN1
      FVY(I)=FVY(I)*FN1
   12 CONTINUE
      FN=1./IFCL(8,I)
      FV(I)=FV(I)/FZ(I)
      FZ(I)=10.*ALOG10(FZ(I)*FN)
```

```
      IZ=FZ(I)
C
C    TEST IF CONTOUR HAS SPLIT FROM ANOTHER CONTOUR
C    DETERMINE BASE CONTOUR ID, USE CONTOUR WITH MAX WATER FLUX
C
      MF=IFCL(9,I)
      IF(MF.EQ.0) GO TO 40
      IF(MF.LE.0.OR.MF.GT.JFL) GO TO 30
      IP=I+1
      IF(IP.GT.NFMX) GO TO 21
      DO 20 J=IP,NFMX
      MF1=IFCL(9,J)
      IF(MF1.NE.MF) GO TO 20
      IF(WFC.LT.FCL(1,J)) GO TO 30
      IFCL(9,J)=-IFCL(9,J)
   20 CONTINUE
C
C    COMPUTE CONTOUR CENTROID VELOCITY AND
C    UPDATE AGE AND TRACK ID FROM PRIOR SCAN
C
   21 IFTNO(I)=JFTNO(MF)
      NFDR(MF)=I
      IF(IFTNO(I).LE.0) GO TO 41
      VFX=VKAL*(FCL(3,I)-WFX(MF))
      VFY=VKAL*(FCL(4,I)-WFY(MF))
      IFAGE(I)=IWAGE(MF)+1
      GO TO 90
C
C    SPLIT, SET SPLIT POINTER
C
   30 IF(MF.LT.0) MF=-MF
      IF(MF.LE.0.OR.MF.GT.JFL) GO TO 40
      IFTSP=JFTNO(MF)
      GO TO 41
   40 IFTSP=0
C
C    NEW CONTOUR, INCREMENT COUNTER AND UPDATE CONTOUR DIRECTORY
C
   41 NFLN=NFLN+1
      IFAGE(I)=1
      IFTNO(I)=NFLN
      IFCDIR(NFLN)=NFLN
      IF(IFTSP.NE.0) IFCDIR(NFLN)=IFTSP
C
C    TEST IF CONTOUR HAS MERGED WITH ANOTHER CONTOUR
C
   90 MF=IFCL(7,I)
      IFMGE=0
      IF(MF.LE.0.OR.MF.GT.JFL) GO TO 901
      IFMGE=JFTNO(MF)
      NFDR(MF)=I
  901 CONTINUE
      IF(IFMGE.LE.0.OR.IFMGE.GT.NFLN) GO TO 92
      IFT=IFTNO(I)
   94 IFTO=IFCDIR(IFT)
      IF(IFTO.EQ.IFT) GO TO 93
      IF(IFTO.LE.0.OR.IFTO.GT.NFLN) GO TO 92
      IFT=IFTO
      GO TO 94
C
```

```
C    FIND BASE CONTOUR ON MAX WATER FLUX
C
   93 IFTM=IFMGE
   96 IFTO=IFCDIR(IFTM)
      IF(IFTO.EQ.IFTM) GO TO 95
      IF(IFTO.EQ.IFT) GO TO 92
      IF(IFTO.LE.0.OR.IFTO.GT.NFLN) GO TO 92
      IFTM=IFTO
      GO TO 96
   95 IFCDIR(IFTM)=IFT
C
C    FINISHED DIRECTORY ENTRY
C
   92 CONTINUE
      IF(IFTNO(I).EQ.0) COMPLT=.FALSE.
C*
C*   COMPUTE ENCLOSED CELL ATTRIBUTES (USE NFV COUNTER)
C*
      NFV=IFCL(8,I)
      IF(NFV.LT.NNMIN) GO TO 65
      NFV=0
      DO 66 N=1,NVMM
      IF(IVCL(44,N).LT.1.OR.IVCL(44,N).GT.3) GO TO 66
      IF(IVCL(37,N).NE.I) GO TO 66
      NFV=NFV+1
      UX(NFV)=VCL(12,N)
      UY(NFV)=VCL(14,N)
C
C    SAVE VC ENCLOSED IN COMPLETE CONTOURS
C
      IF(.NOT.COMPLT) GO TO 66
      NCV=NCV+1
      WCX(NCV)=VCL(12,N)
      WCY(NCV)=VCL(14,N)
   66 CONTINUE
C
C    COMPUTE CELL AVG NEAREST SPACING
C
      IF(NFV.LT.NNMIN) GO TO 65
      CALL NEARN(UX,UY,NFV,DVFN,DVFS,XVFN,YVFN,DVFM,DVFX)
C
C    COMPUTE AVG CELL POSITION ON IFCL(8,I)(ALL CELLS)
C
   65 CONTINUE
      IF(IFLAG.EQ.1) GO TO 70
      FX(I)=FX(I)*FN
      FY(I)=FY(I)*FN
      FXY(I)=FXY(I)*FN
      FX2(I)=FX2(I)*FN
      FY2(I)=FY2(I)*FN
C
C    COMPUTE SPREAD AND MAKE A LINE FIT TO ALL ENCLOSED CELLS
C
      CALL LINFIT(FX(I),FY(I),FX2(I),FY2(I),FXY(I),IFCL(8,I),
     +            DFX,DFY,SPRDC,SPRDL,SPRDA,ANGC,RCOEF)
C*
C*   COMPUTE ENCLOSED SC ATTRIBUTES (USE NF COUNTER)
C*
   70 CONTINUE
      IF(NFV.LE.3) GO TO 76
```

```
      FXI=0.
      FYI=0.
      FXYI=0.
      FX2I=0.
      FY2I=0.
      XFN=0.
      YFN=0.
      DFNM=0.
      DFNX=0.
      SPRDS=0.
      SPDL=0.
      SPDA=0.
      ANGS=0.
   69 DO 61 N=1,NCL
      IF(ICF(N).NE.I) GO TO 61
      NSC=NSC+1
      X=UCX(N)
      Y=UCY(N)
      UCFX(NSC)=X
      UCFY(NSC)=Y
C
C   SUM POSITION DATA FOR LINE FIT
C
      FXI=FXI+X
      FYI=FYI+Y
      FXYI=FXYI+X*Y
      FX2I=FX2I+X*X
      FY2I=FY2I+Y*Y
C
C   SAVE SC ENCLOSED IN COMPLETE CONTOURS
C
      IF(.NOT.COMPLT) GO TO 61
      NCS=NCS+1
      CCFX(NCS)=X
      CCFY(NCS)=Y
   61 CONTINUE
      IF(NSC.LE.3) GO TO 76
      FN=1./NSC
C
C   COMPUTE MEAN SC POSITION VALUES
C
      FXI=FXI*FN
      FYI=FYI*FN
      FXYI=FXYI*FN
      FX2I=FX2I*FN
      FY2I=FY2I*FN
C
C   COMPUTE SPREAD AND MAKE LINE FIT TO ENCLOSED SC'S
C
      CALL LINFIT(FXI,FYI,FX2I,FY2I,FXYI,IFCL(8,I),
     +            DFX,DFY,SPRDS,SPDL,SPDA,ANGS,RCOEF)
C
C   COMPUTE SC AVG NEAREST SPACING
C
      IF(NSC.LT.NNMIN) GO TO 76
      CALL NEARN(UCFX,UCFY,NSC,DFN,DFS,XFN,YFN,DFNM,DFNX)
   76 CONTINUE
C
C   PASS FIXED CONTOUR DATA SET TO SORT ROUTINE
C
```

```
      IF(IFTNO(I).LE.0) GO TO 10
      NETA=NETA+1
      NSIG=IFNC(I)+IFSC(I)
      KNSC=NSIG+1000
      WRITE(3) KTL,IZ,FCL(3,I),FCL(4,I),FVX(I),FVY(I),
     1         FCL(1,I),FCL(5,I),IZERO,IZERO,IZERO,
     2         IFMGE,IFTSP,IFAGE(I),NFV,KNSC,IFTNO(I),
     3         ITHREE
C
C    OUTPUT CONTOUR AND ENCLOSED CELL SUMMARY
C
      IF(.NOT.PRFIXC) GO TO 77
      IX1=FCL(3,I)
      IY1=FCL(4,I)
      IX2=FX(I)
      IY2=FY(I)
      IH=FHS(I)
      IORNT=ANGC
      WRITE(6,1001) IFTNO(I),IX1,IY1,IX2,IY2,IZ,IFF(I)
     1 ,IFCL(8,I),NSIG,IFNC(I),SPRDA,SPRDL,IORNT,FCL(5,I),FCL(1,I
     2 ,FVX(I),FVY(I),DVFN,IH,IFMGE,IFTSP
 1001 FORMAT(1X,5I4,3I3,2I2,2F4.1,I3,F6.2,F5.2,2F5.1,F4.1,I3,2I4)
C
C    PASS ENCLOSED CELL AND SC ATTRIBUTES TO SORT
C
      WRITE(3) KTL,IZERO,
     1         FX(I),FY(I),DVFN,DVFS,ANGC,SPRDC,
     2         FXI,FYI,DFN,DFS,ANGS,SPRDS,
     3         NFV,KNSC,IFTNO(I),
     4         IFOUR
C
   77 CONTINUE
      IF(FSIG.GE.FCAZ .AND. IFTNO(I).NE.0) NFL=NFL+1
C*
C*   ACCUMULATE COMPLETE CONTOUR ATTRIBUTES
C*
      IF(.NOT.COMPLT) GO TO 10
      NCC=NCC+1
      FX(NCC)=FCL(3,I)
      FY(NCC)=FCL(4,I)
      ARCC=ARCC+AREA
      WAB=WAB+WFC/AREA
      IF(NSC.GT.0) WSB=WSB+ALOG10(WFC/NSC)
C
      IF(NFV.LT.NNMIN) GO TO 10
      NCI=NCI+1
      A=SQRT(AREA)/DVFN
      ARB=ARB+ALOG10(A)
      WFB=WFB+ALOG10(WFC)
C
   10 CONTINUE
C
C    COMPUTE COMPLETE CONTOUR ATTRIBUTES
C
      IF(NCI.EQ.0) GO TO 9
      CI=ARB/NCI
      CI=10.**CI
      WFB=WFB/NCI
      WFB=10.**WFB
    9 CONTINUE
```

```
      IF(NCS.EQ.0) GO TO 8
      WSB=WSB/NCS
      WSB=10.**WSB
    8 CONTINUE
      IF(NCC.GT.0) WAB=WAB/NCC
C
C   COMPUTE AVG SPACINGS ON ENCLOSED V.CELLS, SC'S AND THE COMPLETE CONTRS
C
      IF(NCV.GE.NNMIN) CALL NEARN(WCX,WCY,NCV,DVFN,DVFS,
     +                            XVFN,YVFN,DVFM,DVFX)
      IF(NCS.GE.NNMIN) CALL NEARN(CCFX,CCFY,NCS,DFN,DFS,
     +                            XFN,YFN,DFNM,DFNX)
      IF(NCC.GE.NNMIN) CALL NEARN(FX,FY,NCC,DCN,DCS,
     +                            XCN,YCN,DCNM,DCNX)
      KNCL=NCL+LD
      KNFL=NFL+LD
C
C   PASS COMPLETE CONTOUR ATTRIBUTES TO SORT
C
      WRITE(3) KTL,DLONG,DLAT,DVFN,DFN,DCN,ARCC,CI,
     +         WFB,WSB,WAB,NCV,NCS,NCI,NCC,KNCL,KNFL,
     +         ITWO
C*
C*  END CONTOUR LOOP, PREPARE CONTOUR PLOTS
C*
      IF(.NOT.CEPLOT) GO TO 200
      OPEN(UNIT=9, FILE=INPUT, FORM='UNFORMATTED',
     +     STATUS= FILSTAT(IFILS))
      IFILS=2
      REWIND 9
      REWIND 8
      IEOF=0
C
      IF(JNID.LE.JNMX) GO TO 114
      WRITE(6,113) NVSCN,JNID
  113 FORMAT(1X,'V.SCAN',I4,2X,'TOO MANY CONTOURS',I5)
      JNID=JNMX
C
C   CLEAR OUTPUT REGISTERS AFTER EACH WRITE
C
  114 ID=1
  115 DO 120 J=5,NCTR
      DAT(J,ID)=-999.
  120 CONTINUE
      NUM(ID)=0
      IF(INIT.EQ.1) GO TO 150
      ID=ID+1
      IF(ID.LE.JNID) GO TO 115
      INIT=1
C
C   INPUT AN XY PAIR AND LOOKUP CONTOUR BASE ID
C
  150 READ(8,END=190) KNID,K,XY
      IF(KNID.LE.0.OR.KNID.GT.KNIDM(K)) GO TO 150
      ID=JNIDA(K,KNID)
      IF(ID.LE.0.OR.ID.GT.JNID) GO TO 150
C
C   UPDATE OUTPUT REGISTER WITH CURRENT XY PAIRS
C
      N=NUM(ID)
```

```
      NL=N+1
      NH=N+4
      J=0
      DO 160 I=NL,NH
      J=J+1
  160 DAT(I,ID)=XY(J)
      NUM(ID)=NH
      IF(NH.LT.NCTR) GO TO 150
C
C   REGISTER FULL, LOOK UP LOW THRESH ENCLOSING CONTOUR ID
C
  165 NF=0
      IDB=KNID
      IF(K.GT.1) IDB=KNIDL(K-1,IDB)
C
C   LOOKUP CONTOUR TRACK ID ON CELL DETECTION THRESH LEVEL (KLVL)
C
      IF(KLVL.NE.1) GO TO 170
      NF=KNIDC(IDB)
      GO TO 180
C
  170 CONTINUE
      IF(K.NE.KLVL) GO TO 185
      NF=KNIDC(KNID)
C
  180 CONTINUE
      IF(NF.LE.0.OR.NF.GT.NFMX) GO TO 175
      NF=IFTNO(NF)
      IF(NF.EQ.0) GO TO 175
C
C   PASS 3 XY PAIRS AS ONE RECORD TO SORT
C
  185 IDB=JNIDA(1,IDB)
      WRITE(9) KTL,ITL(K),(DAT(I,ID),I=1,NCTR),ID,IDB,NF,ISEVEN
C
  175 CONTINUE
      IF(IEOF.EQ.1) GO TO 195
      GO TO 115
C
C   END OF PLOT FILE, PASS REMAINING XY PAIRS TO SORT
C
  190 IEOF=1
      K=1
      KNID=0
C
  195 KNID=KNID+1
      IF(KNID.GT.KNIDM(K)) GO TO 205
      ID=JNIDA(K,KNID)
      IF(ID.LE.0.OR.ID.GT.JNID) GO TO 195
      IF(NUM(ID).LE.0) GO TO 195
      NUM(ID)=0
      GO TO 165
C
  205 KNID=0
      K=K+1
      IF(K.LE.LT) GO TO 195
      CLOSE(9)
      IF(KLVL.EQ.1) GO TO 210
C
C   SORT ON:  1) ENCLOSING CONTOUR ID
```

```
C               2) CONTOUR TRACK ID
C               3) THRESHOLD LEVEL
C               4) CONTOUR ID
C
        CALL SORT(NVSCN,INPUT,1,2,KEY,17)
        GO TO 200
C
C    TRACKED CONTOUR IS BASE ENCLOSING CONTOUR
C    SORT ON:  1) CONTOUR TRACK ID
C               2) THRESHOLD LEVEL
C               3) CONTOUR ID
C
  210 CALL SORT(NVSCN,INPUT,1,2,KEY1,13)
C*
C*   CLEAN UP SPLIT/MERGE DIRECTORY
C*
  200 CONTINUE
        IF(IFT.LE.0) GO TO 84
        DO 85 I=1,JFL
        NF=IFMG(I)
        IFMG(I)=0
        IF(NF.LE.0.OR.NF.GT.NFMX) GO TO 85
        IFT=IFTNO(NF)
        IF(IFT.LE.0.OR.IFT.GT.NFLN) GO TO 85
        IFMGE=JFTNO(I)
        IF(IFT.EQ.IFMGE.OR.IFMGE.LE.0) GO TO 85
        INXT=4
        IF(NFDR(I).EQ.0) NFDR(I)=NF
   87   IFTO=IFCDIR(IFT)
        IF(IFTO.EQ.IFT) GO TO 88
        IF(IFTO.LE.0.OR.IFTO.GT.NFLN) GO TO 85
        IFT=IFTO
        GO TO 87
C
C    HAVE ROOT THIS CONTOUR
C
   88   IFTM=IFMGE
   89   IFTO=IFCDIR(IFTM)
        IF(IFTO.EQ.IFTM) GO TO 83
        IF(IFTO.EQ.IFT) GO TO 85
        IF(IFTO.LE.0.OR.IFTO.GT.NFLN) GO TO 85
        IFTM=IFTO
        GO TO 89
   83   IFCDIR(IFTM)=IFT
   85   CONTINUE
        DO 81 NV=1,NVMM
        IF(IVCL(53,NV).EQ.0.OR.IVCL(37,NV).NE.0) GO TO 81
        MF=IVCL(51,NV)
        IF(MF.LE.0.OR.MF.GT.JFL) GO TO 81
        IVCL(37,NV)=NFDR(MF)
        IVCL(51,NV)=-IVCL(51,NV)
   81   CONTINUE
C
C    SAVE POSITION, WATER FLUX, AGE AND TRACK ID FOR NEXT SCAN
C
   84   JFL=NFMX
        DO 80 I=1,NFMX
        JFTNO(I)=IFTNO(I)
        WFX(I)=FCL(3,I)
        WFY(I)=FCL(4,I)
```

```
      WF(I)=FCL(1,I)
      IWAGE(I)=IFAGE(I)
      NFDR(I)=0
   80 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CTRAK
      LOGICAL FRCELL,FRSIG,FRFIXC,FRCLUS,FRSCAN,FRHEAD,FRNOTS,FROVER
      DIMENSION UVX(128),UVY(128),ID(15),LD(10)
      DIMENSION IVCL(53,460),UDIV(128),UROT(128),ICR(128)
      DIMENSION XPOS(128),YPOS(128),IWAGE(128),ICAGE(128)
      COMMON /DATA2/ VCL(53,460)
      COMMON /DATA5/ NCL,NFL,JCL,JCTNO(128),WCX(256),WCY(256),JFL,
     1               JFTNO(256),WFX(256),WFY(256),WF(256),
     2               IFMG(256),ICMG(128),ICZC(128),NVMM
      COMMON /NVLIS/ NVARM,NCARM,NVO,NFO,ICO,IO,JO,JYR,LBL,KTL
      COMMON /NVLIT/ KTLL,NKNID,NKDO,IZTH,NKDMX,ITHR,IFXC(1024),HTST
      COMMON /DATA4/ FCL(9,256),AFCS,WFCS,NFMX,NFARM,KNIDC(1024),NFTA
      COMMON /FLGS/  FRCELL,FRSIG,FRFIXC,FRCLUS,FRSCAN,FRHEAD,
     +               FRNOTS,FROVER
      COMMON /INTL/  MHSN,MNSN,HM,FNSN,FNSRN,NCLN,NFLN,MZSN,NNMIN,FCAZ
      COMMON /UVC/   UX(512),UY(512),UCX(128),UCY(128),UCVX(128),
     +               UCVY(128),UCN(128),UCZ(128),UCXY(128),UCX2(128),
     +               UCY2(128),ICFZ(128),ICFNT(128),UCHS(128),UCV(128),
     +               ICTK(128),ICTK2(128),ICTNO(128),ICTSP(128)
      COMMON /UFC/   ICF(128),ICVN(128),FX(256),FY(256),FVX(256),
     +               FVY(256),FZ(256),FV(256),FXY(256),FX2(256),
     +               FY2(256),IFFZ(256),FHS(256),IFSC(256),
     +               IFNC(256),IFVN(256),IFTNO(256),NFDR(256)
C
      EQUIVALENCE(VCL(1,1),IVCL(1,1))
C
      PARAMETER(RTD=57.29578,FK=1000.)
      DATA ID/15*0/,LD/10000,9*0/,IFIVE/5/
C
C    NCL IS ACTIVE CLUSTER COUNTER
C
      IF(NCL.LE.0) RETURN
C
C    COMPUTE TIME INTERVAL BETWEEN ARRAYS
C
      IF(KTL.GT.KTLL) VKAL=FK/(KTL-KTLL)
C
C    GENERATE SC ATTRIBUTE ARRAYS
C
      DO 10 I=1,NCL
      UDIV(I)=0
      UROT(I)=0
      ICR(I)=0
      ICTNO(I)=0
      ICTSP(I)=0
C
C    COMPUTE AVG CELL VELOCITY ON ICVN(UPDATTED CELL COUNTER)
C    COMPUTE AVG CELL POSITION AND REFL ON UCN(ALL CELLS)
C
      IF(UCN(I).LE.0.1) GO TO 10
      DUC=1./UCN(I)
      UVX(I)=0.
      UVY(I)=0.
      IF(ICVN(I).LE.0) GO TO 12
      DUC1=FK/ICVN(I)
      UCVX(I)=UCVX(I)*DUC1
      UCVY(I)=UCVY(I)*DUC1
   12 UCX(I)=UCX(I)*DUC
      UCY(I)=UCY(I)*DUC
      UCXY(I)=UCXY(I)*DUC
```

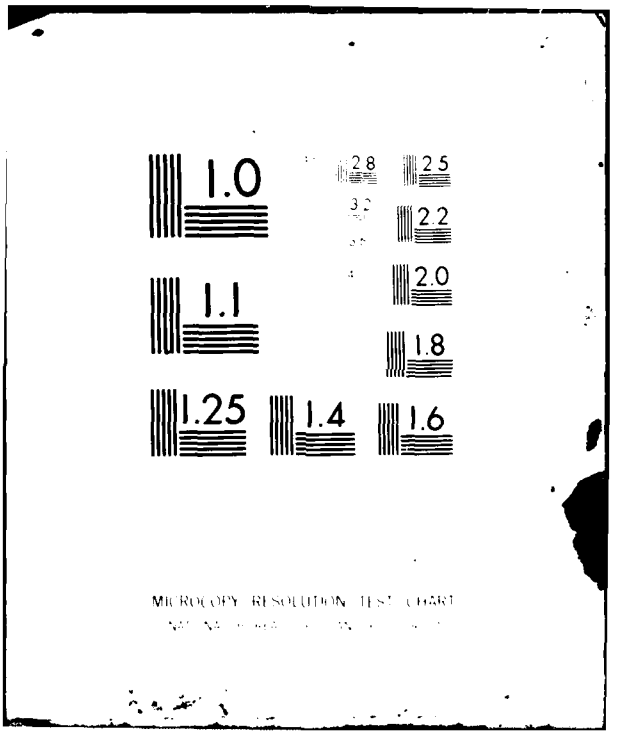
```
      UCX2(I)=UCX2(I)*DUC
      UCY2(I)=UCY2(I)*DUC
      UCV(I)=UCV(I)/UCZ(I)
      UCZ(I)=10.*ALOG10(UCZ(I)*DUC)
      IZVAL=UCZ(I)
      NF=ICF(I)
C
C   COUNT CLUSTERS ENCLOSED WITHIN EACH CONTOUR
C
      IF(NF.GT.0.AND.NF.LE.NFMX) IFNC(NF)=IFNC(NF)+1
C
DC    TEST IF CLUSTER HAS SPLIT FROM ANOTHER CLUSTER
C   DETERMINE BASE CLUSTER ID, USE CLUSTER WITH MOST HITS
C
      MC=ICTK(I)
      IF(MC.EQ.0) GO TO 40
      IF(MC.LE.0.OR.MC.GT.JCL) GO TO 30
      IZC=ICFZ(I)
      IP=I+1
      IF(IP.GT.NCL) GO TO 21
      DO 20 J=IP,NCL
      MC1=ICTK(J)
      IF(MC1.NE.MC) GO TO 20
      IF(IZC.LT.ICFZ(J)) GO TO 30
      ICTK(J)=-ICTK(J)
   20 CONTINUE
C
C   UPDATE VELOCITY FROM PRIOR SCAN
C   INCREMENT AGE
C
   21 ICTNO(I)=JCTNO(MC)
      IF(ICTNO(I).LE.0) GO TO 40
      UVX(I)=VKAL*(UCX(I)-WCX(MC))
      UVY(I)=VKAL*(UCY(I)-WCY(MC))
      ICAGE(I)=IWAGE(MC)+1
      GO TO 101
C
C   SPLIT, START NEW CLUSTER
C
   30 IF(MC.LT.0) MC=-MC
      IF(MC.LT.0.OR.MC.GT.JCL) GO TO 40
      ICTSP(I)=JCTNO(MC)
C
C   NEW CLUSTER, INCREMENT COUNTER
C
   40 NCLN=NCLN+1
      ICTNO(I)=NCLN
      ICAGE(I)=1
C
C   SUM ENCLOSED CELL POSITION VALUES
C
  101 IF(UCN(I).LE.2) GO TO 10
      BC=0.
      RC=0.
      SC=0.
      CC=0.
      DO 102 J=1,NVMM
      IF(IVCL(38,J).NE.1.OR.IVCL(53,J).LE.0) GO TO 10
      ICR(I)=ICR(I)+1
      SC=SC+VCL(12,J)
```

```
      RS=RC+VCL(14,J)
      BC=BC+VCL(49,J)
      CC=CC+VCL(50,J)
  102 CONTINUE
C
C   COMPUTE CELL ROTATION AND DIVERGENCE ON CLUSTERS OF GT 2 CELLS
C
      IF(ICR(I).LE.2) GO TO 10
      AC=1./ICR(I)
      SC=SC*AC
      XPOS(I)=SC
      RC=RC*AC
      YPOS(I)=RC
      BC=BC*AC
      CC=CC*AC
      DO 103 J=1,NVMM
      IF(IVCL(38,J).NE.I.OR.IVCL(53,J).LE.0) GO TO 103
      IVCL(39,J)=LU(I)
      DC=SQRT((VCL(12,J)-SC)**2+(VCL(14,J)-RC)**2)
      SPRD=SQRT((VCL(49,J)-BC)**2+(VCL(50,J)-CC)**2)
      UROT(I)=UROT(I)+ATAN2(VCL(12,J)-SC,VCL(14,J)-RC)
     :    -ATAN2(VCL(49,J)-BC,VCL(50,J)-CC)
      UDIV(I)=UDIV(I)+(DC-SPRD)/(DC+SPRD)
  103 CONTINUE
      UROT(I)=UROT(I)*AC*VKAL
      UDIV(I)=UDIV(I)*AC*VKAL*2.0
   10 CONTINUE
      RETURN
C
C   PREPARE CLUSTER DATA FOR OUTPUT
C
      ENTRY COUT
C
      IF(NCL.LE.0) RETURN
C
      IF(PRCLUS) WRITE(6,1000)
 1000 FORMAT(1X//1X,' CLUSTER OUTPUT'//
     1 1X,'     CENTROID    Z     N SPR SPR ORT CNT'
     2 ,' VELOCITY SHEAR MX MR SP  CELL   CELL  NO'/
     3 1X,' TRK  E.   N. AV PK  V  X   L  ANG  ID'
     4 ,'  AV CELL  MSKM HT ID ID  ROT.   DIV.  RD'/
     5 1X,'  NO  KM  KM DB DB  C  KM  KM DEG    '
     6 ,' EM/S NM/S       KM NO NO  MSKM   MSKM  CS')
C
C   LOOP THROUGH ALL ACTIVE CLUSTERS
C
      DO 100 I=1,NCL
      IN=UCN(I)
      IF(IN.LE.0) GO TO 100
C
C   COMPUTE SPREAD AND MAKE LINE FIT TO ENCLOSED CELLS
C
      CALL LINFIT(UCX(I),UCY(I),UCX2(I),UCY2(I),UCXY(I),IN,
     1           DCX,DCY,SPRD,SPRDL,SC,BC,RC)
  120 IFXNO=0
      NF=ICF(I)
      IF(NF.GT.0.AND.NF.LE.NFMX) IFXNO=IFTNO(NF)
      ICMGE=0
      MC=ICTK2(I)
      IF(MC.GT.0.AND.MC.LE.JCL) ICMGE=JCTNO(MC)
```

```
      IZ=UCZ(I)
C
C   PASS CLUSTER DATA SET TO SORT ROUTINE
C
      WRITE(3) KTL,IZ,,UCX(I),UCY(I),UCVX(I),UCVY(I),
     1         SPRD,UCHS(I),XPOS(I),YPOS(I),BC,ICMGE,
     2         ICTSP(I),ICAGE(I),IN,ICTNO(I),IFZRO,
     3         IFIVE
C
      IF(.NOT.PRCLUS) GO TO 100
      IX=UCX(I)
      IY=UCY(I)
      IORNT=BC
      IH=UCHS(I)
C
      WRITE(6,1001) ICTNO(I),IX,IY,IZ,ICPZ(I),IN,SC,SPRDL
     1 ,IORNT,IFXNO,UCVX(I),UCVY(I),UCV(I),IH
     2 ,ICMGE,ICTSP(I),UROT(I),UDIV(I),ICR(I)
 1001 FORMAT(1X,3I4,3I3,2F4.1,2I4,3F5.1,3I3,2F7.2,I3)
C
  100 CONTINUE
      IF(JCL.LE.0) GO TO 140
      DO 130 I=1,JCL
      NC=ICMG(I)
      IF(NC.LE.0.OR.NC.GT.NCL) GO TO 130
      ICMGE=ICTNO(NC)
  130 CONTINUE
  140 JCL=0
C
C   SAVE POSITION AND AGE FOR NEXT SCAN
C
      DO 150 I=1,NCL
      JCTNO(I)=ICTNO(I)
      WCX(I)=UCX(I)
      WCY(I)=UCY(I)
      IWAGE(I)=ICAGE(I)
      ICZC(I)=ICPZ(I)
      ICMG(I)=0
C
C   SEGREGATE SIG CELL DATA FROM CLUSTER
C
      IF(ICTNO(I).EQ.0) GO TO 150
      JCL=JCL+1
      UCX(JCL)=UCX(I)
      UCY(JCL)=UCY(I)
  150 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE NEARN(X,Y,IM,DNN,DNS,XNN,YNN,DNMN,DNMX)
C
C     ***************************************************
C
C     NAME:      NEARN
C     PROJECT:   ERT 8035-620 (FAA/NOAA)
C
C     PURPOSE:   COMPUTE THE AVERAGE NEAREST SEPARATION, THE SPREAD,
C                THE AVERAGE X AND Y NEAREST SEPARATION AND THE
C                MIN AND MAX NEAREST SEPARATION OVER THE SAMPLE
C                INPUT THROUGH THE ARGUMENT LIST.
C
C     INTERFACES:
C        CALLING MOD.   STRAK,FTRAK,CTRAK
C        CALLED MODS.   NONE
C        INPUT PARM.    X = E-W CENTROID LOCATIONS OF INPUT ENTITIES
C                       Y = N-S CENTROID LOCATIONS OF INPUT ENTITIES
C                       IM = NUMBER OF INPUT ENTITIES
C        OUTPUT PARM.   DNN = AVG NEAREST NEIGHBOR DISTANCE
C                       DNS = SPREAD ABOUT DNN
C                       XNN = AVG E-W NEAREST NEIGHBOR DISTANCE
C                       YNN = AVG N-S NEAREST NEIGHBOR DISTANCE
C                       DNMN = MIN NEAREST SEPARATION
C                       DNMX = MAX NEAREST SEPARATION
C
C     COMMENTS:   SPREAD IS THE STANDARD DEVIATION OF THE SET
C                 OF NEAREST SEPARATIONS FROM THE AVG NND.
C
C     VERSION:   1.0 DEC/VAX-11
C     DATE:      4/17/81
C     DESIGN:    RKCRANE
C     PROGMR:    GBGUSTAFSON
C
C     ***************************************************
C
      DIMENSION X(1),Y(1)
      PARAMETER(DM=1.E8)
C
C     INITIALISE SUMMATION PARAMETERS
C
      DVAR=0.
      DNS=0.
      DAVG=0.
      DNN=0.
      XNN=0.
      YNN=0.
      DNMX=0.
      DNMN=999.
      NN=0
C
C     LOOP THROUGH ENTIRE SAMPLE
C
      DO 10 I=1,IM
      DCOMP=DM
C
C     COMPARE EACH ENTITY TO EVERY OTHER ENTITY
C
      DO 20 J=1,IM
      IF(I.EQ.J) GO TO 20
C
```

AD-A109 517 ENVIRONMENTAL RESEARCH AND TECHNOLOGY INC CONCORD MA F/G 1/2
DETECTION AND TRACKING ALGORITHM REFINEMENT.(U)
OCT 81 G B GUSTAFSON, R K CRANE DTFA01-81-Y-10521
UNCLASSIFIED ERT-P-B035 FAA/RD-81/80 NL


1.0
2.8
2.5
3.2
2.2
2.0
1.1
1.8
1.25
1.4
1.6
MICROCOPY RESOLUTION TEST CHART

```
C   COMPUTE SEPARATION
C
      DX=X(I)-X(J)
      DY=Y(I)-Y(J)
      D2=DX*DX+DY*DY
C
C   FIND MINIMUM SEPARATION
C
      DCOMP=AMIN1(D2,DCOMP)
      IF(D2.NE.DCOMP) GO TO 20
      DXJ=ABS(DX)
      DYJ=ABS(DY)
   20 CONTINUE
C
C   SUM MINIMUM SEPARATION VALUES
C
      IF(DCOMP.GE.DM.OR.DCOMP.LE.0.) GO TO 10
      ND=ND+1
      DXA=SQRT(DCOMP)
      DAVG=DAVG+DXA
      DVAR=DVAR+DCOMP
      XNN=XNN+DXJ
      YNN=YNN+DYJ
C
C   FIND MAX AND MIN OF NEAREST NEIGHBOR DIST FOR SAMPLE
C
      DNMN=AMIN1(DNMN,DXA)
      DNMX=AMAX1(DNMX,DXA)
   10 CONTINUE
C
C   COMPUTE AVG NEAREST NEIGHBOR VALUES AND SPREAD
C
      IF(ND.LE.0) RETURN
      DN=1./ND
      DNN=DAVG*DN
      XNN=XNN*DN
      YNN=YNN*DN
      DVAR=DVAR*DN-DNN*DNN
      IF(DVAR.GT.0.) DNS=SQRT(DVAR)
C
      RETURN
      END
```

```
      SUBROUTINE LINFIT(X,Y,X2,Y2,XY,N,
     +                  DFX,DFY,SPRD,SPRDL,SPRDA,ANG,RCOEF)
C
C   ****************************************************************
C
C   NAME:     LINFIT
C   PROJECT:  ERT 8035-620 (FAA/NOAA)
C
C   PURPOSE:  TO FIT A LINE TO THE ARRAY OF POINTS X,Y.  TO
C             COMPUTE THE SPREAD, SPREAD ALONG THE LINE, AND
C             SPREAD PERPENDICULAR TO THE LINE.  TO COMPUTE
C             THE REGRESSION COEF OF THE FIT.
C
C   INTERFACES:
C      CALLING MOD.  FTRAK,CTRAK
C      CALLED MODS.  NONE
C      INPUT PARA.   X,Y  DATA PAIRS
C                    X2,Y2  SQUARE OF X,Y
C                    XY  PRODUCT OF X,Y
C                    N  NUMBER OF DATA PAIRS
C      OUTPUT PARA   DFX  VARIENCE OF X COMP.
C                    DFY  VARIENCE OF Y COMP.
C                    SPRD  TOTAL SPREAD
C                    SPRDL  SPREAD ALONG LINE
C                    SPRDA  SPREAD PERP TO LINE
C                    ANG  ORIENTATION OF LINE
C                    RCOEF  REGRESSION COEF OF FIT
C
C   COMMENTS:  MUST BE AT LEAST NMIN DATA POINTS TO MAKE FIT.
C
C   VERSION:  1.0 DEC/VAX 11-780
C   DATE:     5/18/81
C   DESIGN:   RKCRANE
C   PROGMR:   GBGUSTAFSON
C
C   ****************************************************************
C
      PARAMETER(RTD=57.29578,NMIN=3,QUAD=90.)
C
C   INITIALISE AND COMPUTE SPREAD
C
      SPRDL=0.
      SPRDA=0.
      ANG=0.
      RCOEF=0.
      DFX=X2-X*X
      DFY=Y2-Y*Y
      SPRD=DFX+DFY
C
      IF(N.LT.NMIN) RETURN
C
C   MAKE LINE FIT
C
      CF=XY-X*Y
      BFX=ATAN2(CF,DFX)
      BFY=ATAN2(DFY,CF)
      BF=(BFX+BFY)*.5
      TANA=TAN(BF)
      TANA2=TANA*TANA
      ANG=QUAD - BF*RTD
```

```
      DF=DFX*DFY
      IF(DF.GT.0.) RCOEF=CF/SQRT(DF)
      AF=Y-TANA*X
      SF= Y2 + AF*AF + TANA2*X2 + 2. * (AF*TANA*X - AF*Y - TANA*XY
      SPRDA = SF / (1. + TANA2)
      SPRDL=SPRD-SPRDA
      IF(SPRDL.GT.0.) SPRDL=SQRT(SPRDL)
      IF(SPRDA.GT.0.) SPRDA=SQRT(SPRDA)
C
C   ADJUST ORIENTATION SUCH THAT SPRDL IS ALONG MAJOR AXIS
C
      IF(SPRDA.LE.SPRDL) GO TO 10
      CF=SPRDL
      SPRDL=SPRDA
      SPRDA=CF
      ANG=ANG-QUAD
C
   10 CONTINUE
      IF(ANG.LT.0.) ANG=ANG+360.
      RETURN
      END
```

```
      SUBROUTINE SORT(N,INPUT,ITYP,IORD,KEY,KY)
C
C  ***************************************************************
C
C   NAME:      SORT
C   PROJECT:   ERT 8035-620 (FAA/NOAA)
C
C   PURPOSE:   TO SORT THE SCRATCH FILE 'INPUT' ACCORDING TO THE
C              PARAMETERS IN THE KEY ARRAY.  TO CREATE A NEW FILE
C              CONTAINING THE SORTED DATA NAMED 'OUTPUT'.
C
C   INTERFACES:
C      CALLING MOD.   FTRAK,STRAK
C      CALLED MODS.   SOR$PASS_FILES,SOR$INIT_SORT,SOR$SORT_MERGE,
C                     SOR$END_SORT
C      INPUT PARA.    N  VOLUME SCAN NUMBER, USED TO LABEL 'OUTPUT'
C                     INPUT  NAME OF INPUT(UNSORTED) DATA FILE
C                     ITYP  NUMBER OF OUTPUT FILES GENERATED EACH SCAN
C                     IORD  VERSION NUMBER ASSIGN TO OUTPUT FILE
C                     KEY  ARRAY CONTAINING THE SORT KEY
C                     KY  SIZE OF KEY
C
C   COMMENTS:  OUTPUT FILE LABEL WIL BE 'S' FOLLOWED BY THE
C              VOLUME SCAN NUMBER (IE S003.TEM FOR V.SCAN 3).
C              FOR 2 DIFFERENT DATA FILES ALTERNATE 1 AND 2 FOR ITYP.
C
C   VERSION:   1.0 DEC/VAX 11-780
C   DATE:      5/19/81
C   DESIGN:    GBGUSTAFSON
C   PROGMR:    GBGUSTAFSON
C
C  ***************************************************************
C
      INTEGER*2 KEY(KY)
      INTEGER SOR$PASS_FILES,SOR$INIT_SORT,SOR$SORT_MERGE,
     +        SOR$END_SORT
      CHARACTER*3 STRING
      CHARACTER*8 INPUT
      CHARACTER*10 OUTPUT
      CHARACTER*4 FILE(2)
      CHARACTER*2 ORD(2)
      DATA FILE/'.TEM','.CTR'/,ORD/'2','1'/,NO/0/
C
      IF(N.EQ.NO) GO TO 10
      ENCODE(3,15,STRING) N
   15 FORMAT(I3.3)
   10 OUTPUT='S'//STRING//FILE(ITYP)//ORD(IORD)
      ISORT=SOR$PASS_FILES(INPUT,OUTPUT)
      IF(.NOT.ISORT) WRITE(6,550) ISORT,N
      ISORT=SOR$INIT_SORT(KEY)
      IF(.NOT.ISORT) WRITE(6,550) ISORT,N
      ISORT=SOR$SORT_MERGE( )
      IF(.NOT.ISORT) WRITE(6,550) ISORT,N
      ISORT=SOR$END_SORT( )
      IF(.NOT.ISORT) WRITE(6,550) ISORT,N
  550 FORMAT(1X,'ERROR IN SORT =',Z8,' ON SCAN',I5)
      NO=N
      RETURN
      END
```

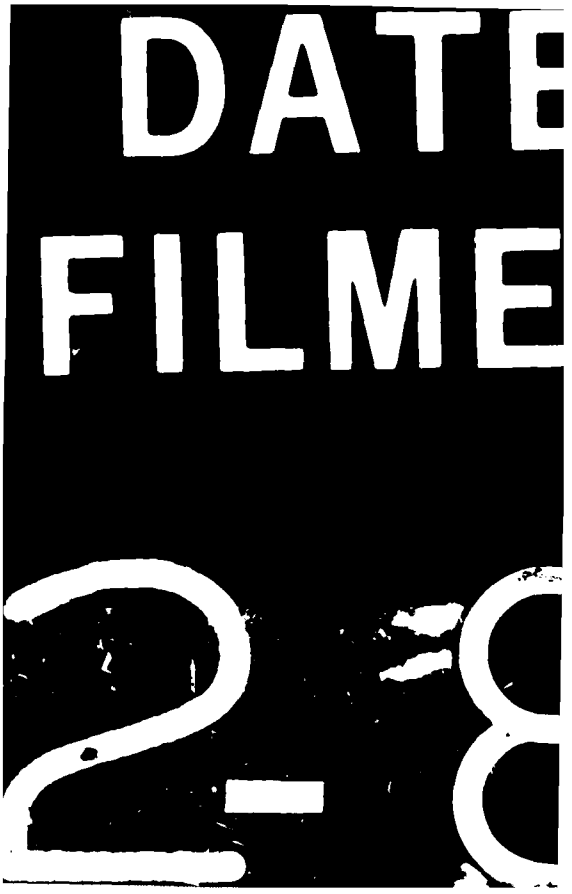

$

REFERENCES

Crane, R.K. (1979): "Automatic Cell Detection and Tracking", IEEE Trans. Geoscience Elect., GE-17, 250-262.

Crane, R.K. (1981): "Thunderstorm Turbulence Hazard Detection", ERT Doc. P-2832-F, Environmental Research & Technology, Inc., Concord, Mass.

Gustafson, G.B. (1980): "Software Routines for the Automatic Processing of Weather Radar Data", ERT Doc. P-1552-IP-5, Environmental Research & Technology, Inc., Concord, Mass.


DATE
FILME
2-8