R-96-04

SNOW ACCUMULATION ALGORITHM FOR THE WSR-88D RADAR, VERSION 1

June 1996

U.S. DEPARTMENT OF THE INTERIOR
Bureau of Reclamation
Technical Service Center
Water Resources Services
River Systems and Meteorology Group

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave Blank)*

2. REPORT DATE
June 1996

3. REPORT TYPE AND DATES COVERED
Final

4. TITLE AND SUBTITLE
Snow Accumulation Algorithm for the WSR-88D Radar, Version 1

5. FUNDING NUMBERS
PR

6. AUTHOR(S)
Arlin B. Super and Edmond W. Holroyd, III

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Bureau of Reclamation
Technical Service Center
Denver CO 80225

8. PERFORMING ORGANIZATION REPORT NUMBER
R-96-04

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Bureau of Reclamation
Denver Federal Center
PO Box 25007
Denver CO 80225-0007

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
DIBR

11. SUPPLEMENTARY NOTES
Hard copy available at the Technical Service Center, Denver, Colorado

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Available from the National Technical Information Service,
Operations Division, 5285 Port Royal Road, Springfield, Virginia 22161

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This annual Bureau of Reclamation report describes the initial year of a 3-year effort to develop a Snow Accumulation Algorithm for the new network of WSR-88D (NEXRAD) radars. Snowfall measurements were made during the 1995-96 winter/spring within range of WSR-88Ds at Albany, NY, Cleveland, OH, and Denver, CO. Observations of S (snow water equivalent) from the latter two locations were related to Z_e (effective reflectivity factor) measurements to determine "best fit" α and β coefficients for the commonly-used equation $Z_e = \alpha S^{\beta}$. Recommended α and β values are 318 and 1.5 for Cleveland and 155 and 1.6 for Denver.

Observations near Lake Erie revealed a significant range effect. The Cleveland radar underestimated snowfall beyond about 60 km. Radar-estimated snowfall amounts were about 85, 61, 31, and 22 percent of gage observations at 61, 87, 115, and 146 km, respectively. A scheme should be developed to use the vertical profile of Z_e to adjust for mid- and far-range snowfall underestimates by WSR-88Ds.

Initial Snow Accumulation Algorithm code is described and presented. The code development borrows from the current NEXRAD rainfall algorithm but makes several important modifications, including an advection scheme, because snowflakes can be transported tens of kilometers from the lowest tilt radar beam to the ground.

14. SUBJECT TERMS--radar/ NEXRAD/ WSR-88D/ snow/ snowfall/ precipitation

15. NUMBER OF PAGES
133

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UL	UL	UL	UL

SNOW ACCUMULATION ALGORITHM FOR THE WSR-88D RADAR, VERSION 1

by

Arlin B. Super and Edmond W. Holroyd, III

River Systems and Meteorology Group
Water Resources Services
Technical Service Center
Denver, Colorado

June 1996

UNITED STATES DEPARTMENT OF THE INTERIOR ★ BUREAU OF RECLAMATION

ACKNOWLEDGMENTS

It is no surprise that the efforts of many individuals were required in various aspects of special data collection or analyses which culminated in this report. Jerry Klazura provided outstanding MOU monitoring, often suggesting sources for technical information and assistance. Other NEXRAD Operational Support Facility personnel who were helpful in particular aspects of the research include Joe Chrisman, Tim O'Bannon, Bill Urell, and Colonels Tim Crum and Andy White.

Several National Weather Service personnel from the Albany, NY, Cleveland, OH, and Denver, CO, WFOs made special efforts on behalf of collecting quality observations. Notable among these are John Quinlan (Albany), Bob LaPlante and Frank Kieltyka (Cleveland), and David Imy and Mike Holzinger (Denver). The three WFO MIC's deserve special thanks for allowing their staffs to assist in the snow algorithm efforts.

John Kobar and Neal Lott of the National Climatic Data Center promptly furnished the many Level II data tapes that were requested for snow algorithm development.

The people who were interested enough in this project to collect special snowfall observations all deserve acknowledgement. Those from the Albany area are too numerous to mention because about 90 volunteer observers were involved, but their efforts are certainly appreciated. Cleveland area observers, all of whom did a first-rate job, are Mike Bezoski, Doug Brady, David Henderson, Anthony Marshall, and Paul Mechling. The two Denver observers who deserve special mention for providing high quality hourly observations in addition to operating Belfort gages are Malcolm Bedell and Richard Kissinger. Steve Becker and Marc Jones provided quality gage measurements west of Denver.

Special appreciation is expressed to Dr. James Heimbach of the University of North Carolina at Asheville. Besides providing significant counsel and being the main source of rawinsonde data throughout the field season, Dr. Heimbach found time in his busy schedule to program and rigorously test the optimization scheme used to determine radar equation coefficients, all without any monetary compensation. His considerable and generous help is very much appreciated and will not be forgotten.

Dr. Paul Smith of the South Dakota School of Mines and Technology provided significant expert technical advice to Reclamation scientists. Dr. Smith carefully read the first draft of this report and made many helpful comments that improved it.

Jack McPartland of Reclamation was deeply involved in all aspects of site selection, gage calibration and installation, and observer training. Curt Hartzell of Reclamation coordinated the collection of data needed for future storm partitioning. Ra Aman developed much of the programming for dealing with Level II data on a Sun workstation. Besides writing several programs used in data handling, Anne Reynolds carefully reduced most of the numerous Belfort gage charts and then double-checked them all, an exceptionally tedious job requiring extraordinary care and patience.

Special thanks are due Dave Matthews and Jim Pierce of Reclamation for providing additional resources and a flexible environment for accomplishment of this work.

This work was primarily supported by the WSR-88D Operational Support Facility and the Next Generation Weather Radar (NEXRAD) Program, with additional support by Reclamation's Technical Service Center.

U.S. Department of the Interior
Mission Statement

As the Nation's principal conservation agency, the Department of the Interior has responsibility for most of our nationally-owned public lands and natural resources. This includes fostering sound use of our land and water resources; protecting our fish, wildlife, and biological diversity; preserving the environmental and cultural values of our national parks and historical places; and providing for the enjoyment of life through outdoor recreation. The Department assesses our energy and mineral resources and works to ensure that their development is in the best interests of all our people by encouraging stewardship and citizen participation in their care. The Department also has a major responsibility for American Indian reservation communities and for people who live in island territories under U.S. administration.

CONTENTS

APPENDIXES

Appendix

TABLES

Table

FIGURES

Figure

1. Introduction

The OSF (Operational Support Facility) of the WSR-88D Radar Program released a request for proposals in the fall of 1994 seeking development of a snow accumulation algorithm for the new national network of Doppler weather radars. Reclamation (Bureau of Reclamation) submitted a proposal in mid-October 1994, which was evaluated along with proposals from other Federal and Federally-supported agencies. An MOU (memorandum of understanding) was signed the end of May 1995 among the NEXRAD (NEXt generation RADar) Program, the WSR-88D OSF, and Reclamation, which called for Reclamation to develop an algorithm over a 3-year period to estimate both S (snow water equivalent) and SD (snow depth) from radar measurements. (Snow depth is sometimes called snow accumulation—in this report, snowfall accumulation refers to S accumulation for 1 hour or more.) This snow accumulation algorithm (hereafter Algorithm) is to be limited to *dry snowfall*. The complications of dealing with mixed rain and snow and/or melting snow with associated "bright band" effects is beyond the scope of the requested work and the resources available to accomplish it.

The original MOU was amended in August 1995 to include precipitation data collection parallel to the south shore of Lake Erie east-northeast of the Cleveland, Ohio, WSR-88D radar. Subsequent analysis of these snowfall and radar measurements was expected to evaluate the ability of the developing Algorithm to detect and quantify lake effect snowfall.

This report discusses progress during the first year of effort ending June 1, 1996. Three letter-type quarterly reports have been submitted to the OSF which provide more detail about some efforts.

This report is organized around the tasks to be performed by Reclamation scientists during the first year ending May 31, 1996, as spelled out in the MOU's SOW (statement of work). Briefly, the MOU tasks are:

1. Scrutinize existing precipitation gage observations of S from the 1994-95 winter within reasonable range of WSR-88D systems with Level II data. Level II data, recorded on 8-mm tape, are the most basic data available to researchers (Crum et al., 1993).

2. Obtain Level II data from selected WSR-88D systems and storm periods for the 1994-95 winter that have corresponding gage data. Also obtain supporting software from the OSF for manipulation of these data and hardware suitable for working with these data and software. Progress under tasks No. 1 and 2 is discussed in section 2 of this report.

3. Use the data, software, and hardware of tasks No. 1 and 2 above and write additional software as needed for development of a "simplified prototype" Algorithm for prediction of S from WSR-88D Level II data. The initial Algorithm will be based on comparisons of radar measurements of equivalent (also called effective) reflectivity factor, Z_e, with surface gage measurements of S. The Algorithm will incorporate radar-estimated horizontal wind speed and direction for advection of falling snow particles to match surface observations of S with radar bin observations of Z_e. A large number of Z_e-S pairs will be used to calculate the empirically-determined coefficients, α and β, for the commonly-used power-law model:

$$Z_e = \alpha S^{\beta} \tag{1}$$

Development of the initial prototype Algorithm is the subject of sections 10 to 13, although calculations of α and β coefficients is discussed in section 9.

4. Collect high-quality observations of *S* and *SD* during the winter/spring of 1995-96 near Denver, Colorado. In the context of the MOU, snow depth refers to the depth of freshly-fallen snow in the absence of melting, compaction by surface heating or other factors, or redistribution by the wind. These *S* measurements (5 locations) and *SD* measurements (1 location) will be used for further Algorithm testing and refinement. This task is discussed in section 3.3.

5. Obtain good-quality observations of *S* and *SD* in a climatological area with WSR-88D coverage other than the Denver area during the winter/spring of 1995-96. This observational program, called "more limited" (than Denver's program) in the SOW, ended up in the Albany, New York, area. The Albany WFO (Weather Forecast Office) of the NWS (National Weather Service) installed and operated a large volunteer observer network, which is discussed in section 3.2. Availability of the Albany *S* and *SD* measurements has been delayed by quality control checking. Observations of *S* from the Cleveland, Ohio, area, scheduled for November 1, 1996, delivery, have been substituted for Albany observations in this report.

6. Based on the MOU Supplement No. 1 of August 1995, install and maintain five Belfort Universal recording precipitation gages from mid-November 1995 through March 1996 between the near and far ranges of the Cleveland WSR-88D. The gages were deployed parallel to the south shore of Lake Erie. The main purpose of this line of gages was to investigate lake effect storms and the ability of the WSR-88D system to detect snow and estimate snowfall accumulation as a function of range. The Cleveland measurement program is discussed in section 3.4.

2. Observations from the 1994-95 Winter

As discussed in more detail in the quarterly reports, only a limited number of NEXRAD systems was equipped with functioning Level II recorders during the 1994-95 winter, and those systems were typically located in the southern half of the continental U.S. A key measurement provided on Level II data tapes is Z_e for each 1-km by 1° range bin out to a 230-km range. A range bin is the basic spatial unit for which these data are recorded. Doppler velocities and spectrum width are also provided, but at 0.25-km by 1° resolution.

Denver was the northernmost radar with Level II data during the winter in question (the Boise, Idaho, Level II recorder was not functional). Other radars with operational Level II recorders, located even farther south and at lower elevations than Denver, likely observed frequent bright band returns associated with melting snowflakes. Dry snowfall cases were probably infrequent at those locations and no effort was made to obtain data from them. Instead, it was decided to concentrate on data collection during the 1995-96 winter when many more WSR-88Ds would be equipped with Level II recorders.

The high-resolution (0.01-inch) Belfort Universal gage at the Denver (Stapleton Airport) WFO was discontinued with the March 1, 1995, opening of DIA (Denver International Airport). The convention in this nation of using English units for measurement of precipitation and snow depth is followed throughout this report. The DIA ASOS (automated surface observation station) non-heated tipping bucket gage does not provide suitable observations

for Algorithm development. Reclamation scientists did acquire the existing November through February Denver WFO hourly snowfall data, as well as daily snowfall amounts from all cooperative observer gages within range of the Denver radar.

Besides the problem of sparsity of Denver data during the dry winter of 1994-95, it was discovered that the standard mode of operation that winter was not to use the Clutter Bypass Map. Rather, because of frequent ground-based inversions and superrefraction, maximum suppression was routinely applied to the entire radar area of coverage. Communication with OSF specialists has revealed that suppression application might have commenced with radial wind speeds (i.e., toward or away from the radar) as great as 10 knots. Suppression is increased as the radial wind speed decreases (Chrisman et al., 1994). Radial winds less than 10 knots are not uncommon over large portions of the lowest tilt radar beam (0.5° elevation angle) during many Denver-area winter storm periods. Consequently, meteorological returns were likely often suppressed even in regions without ground clutter. With the uncertainty of how much suppression was applied at a given place and time and the scarcity of surface snowfall observations it is difficult to see how much use can be made of 1994-95 winter data from Denver for establishing a Z_e-S relationship. However, some attempts were made.

All hourly precipitation data from the Denver WFO were examined for the three snowstorms of the 1994-95 winter with associated Level II data. The only hourly values of record are Trace, 0.01, 0.02 and 0.06 (1 value only) inches, so the available range is very limited. Comparisons of these data with Z_e values directly above the WFO gage revealed large scatter. The scatter was possibly caused in part by over-application of clutter suppression and partially by the lack of range in the snowfall observations.

Twenty-four-hour precipitation totals from all area cooperative observer gages were also examined and compared to Z_e values directly overhead. This comparison resulted in even larger scatter than that observed with the Denver WFO hourly data. At that point, it was decided that resources would be better spent working with the upcoming 1995-96 winter observations than dealing further with the limited and uncertain measurements from the prior winter.

3. Observations from the 1995-96 Winter

3.1 Snowfall Measurement Considerations

Accurate snowfall measurement is difficult because wind effects can cause significant to severe gage undercatch as demonstrated by several authors over many decades. For example, Goodison (1978) reported that gage undercatch by an unshielded Belfort Universal gage can exceed 50 percent with wind speeds as low as 2.5 m s^{-1}, and 75 percent with a 7-m s^{-1} wind. Furthermore, Goodison showed that even a Belfort gage equipped with an Alter wind shield can exceed 50-percent undercatch with a 5 m s^{-1} wind speed. Goodison and others have shown that the degree of undercatch is even greater for some gage types, including the Fisher-Porter, because of their shape.

Another problem is that many existing recording gages in the national network are of the Fisher-Porter type with resolution of only 0.10-inch water equivalent. This resolution is an order of magnitude less than the 0.01 inch (or less) provided by a Belfort gage. In most regions of the U.S. which commonly experience snowfall, only a small fraction of all snowfall hours have a melted water equivalent of 0.10 inch or more. The infrequent occurrence of

higher rates is demonstrated for the Cleveland and Denver areas in section 5. Therefore, Fisher-Porter gages have little utility for relating radar observations to hourly snowfall observations.

Even if snowboards are used to manually measure snowfall and are set flush with the snow or ground surface, windy sites can result in drifting of additional snow onto them or scouring of snow off of them. Windy sites must be avoided for quantitative snowfall measurements.

Because of the problems noted and others discussed by Groisman and Legates (1994), the existing national precipitation gage network was determined at the onset to be inadequate for the purposes of this study. Most climatological gages are read daily and, therefore, do not have the needed time resolution. Many recording gages with hourly time resolution do not have the desired mass resolution for snowfall (0.01-inch melted water equivalent or less). Tipping bucket gages usually can resolve 0.01 inch of water, but most are unheated so they cannot measure snowfall with any reasonable accuracy. And even where gages with adequate mass resolution exist, they tend to be located near WFOs, typically at wide-open, windy airports. The undercatch of such gages can be serious and unknown in magnitude. Snowfall measurements from such locations can add considerable variability to attempts to relate snowfall accumulation to radar observations.

Small clearings in widespread conifer forest generally provide excellent snowfall measurement sites (e.g., Brown and Peck, 1962). Clearings in thick deciduous forest, especially if low brush is common, can also provide well-protected snowfall observing sites. Such forest clearings, together with gage wind shields, can almost eliminate gage undercatch caused by airflow around the gage orifice. Gages installed and operated for this study were placed in forest clearings wherever possible. Of course, most of the U.S. does not have widespread forest, and alternatives needed to be found for protecting gages from the wind in the absence of forest. As will be discussed, different approaches to attempting to solve the snowfall measurement problem were taken at each of at the three measurement areas (Albany, Cleveland, and Denver) during the 1995-96 winter.

It is desirable to locate snow observing sites intended for Z_e-S comparisons as near the radar as practical. Such locations minimize the vertical distance between the radar beam and the surface, which reduces uncertainties caused by wind advection of snow particles. In addition, the volume sampled by the radar increases with range as the beam broadens in width and height. So the representativeness of a surface point observation for the overlying range bin or bins becomes increasingly uncertain at more distant ranges. However, as a practical matter, these factors must be weighed against usually greater ground clutter contamination near the radar and whether suitable surface observing sites exist near the radar. The tradeoffs involved in selecting a snow observing site near a radar are perhaps best illustrated in the discussion of site selection in the Denver area in section 3.3.

3.2. Albany, New York, Observations

Unlike the other two sites, Reclamation had a limited role in data collection within range of the Albany, New York, WSR-88D. Reclamation supplied one Belfort gage with an Alter wind shield. John Quinlan, a forecaster at the Albany WFO, supplied another Belfort gage, and Reclamation provided an Alter shield for that gage as well. These two gages, noted in table 1, were installed by Albany WFO personnel. Reclamation also supported computer data entry of volunteer observations by university students.

Table 1. - Locations of two Belfort gage sites in the Albany area.

Gage Location (Operator)	Latitude (°-')	Longitude (°-')	m.s.l. Elevation (m)	Distance/Azimuth from Radar (km/°)*
Round Lake (J. Quinlan)	42-55.430	73-47.152	59	44/31°
East Durham (F. Stark)	42-20.095	74-03.038	136	28/178°

* All azimuths are given in degrees (°) true in this report.

A large volunteer network was established around Albany largely through the efforts of John Quinlan. After advertising for volunteers, Mr. Quinlan met with potential snowfall observers to explain the project during a series of meetings around the area of radar coverage. Mr. Quinlan visited those persons with both the interest and a suitable location and gave them instructions and the equipment necessary to make hourly observations of both *S* and *SD*. Mr. Quinlan also took GPS (global positioning system) readings at each location to document latitude and longitude. About 90 volunteers were trained and equipped with appropriate forms for logging data, snowboards (1 by 1 ft), rulers graduated to the nearest 0.1 inch, and Clear Vu Model 1100 rain gages. The latter were not used as gages, but the 4-inch-diameter outer shells with sharp tapered edges were used to core snow on boards after the *SD* measurement was made at the end of each hour. The cored sample was then melted indoors and the measurement of *S* was made in the usual manner by pouring the melted water into the 1-1/4-inch-diameter inner tube, which is graduated every 0.01 inch. Snowboard observations are generally more accurate than gage observations unless gages are well protected from the wind.

Efforts were made to locate Albany network snowboards in reasonably protected locations, but local wind measurements were only made at sites with pre-existing anemometers. Some drifting and scouring of snow from individual boards may have occurred during windier storm periods and network observers in most instances noted this occurrence on the observation form or stopped taking observations at that time. The use of a large number of sampling points should partially compensate for wind-caused errors in the hourly snowboard measurements of *S* and *SD*.

The Albany network was partially or fully activated by phone calls from the WFO on 13 occasions from December 1, 1995, through mid-April 1996 as noted in table 2 supplied by Mr. Quinlan. The start and stop times in table 2 include a couple of hours before and after actual snowfall as each storm approached and left the Albany area. Partial or full network activation depended upon whether the Albany WFO forecasters expected only a portion or all of the WSR-88D's area or coverage to be affected by an approaching storm. The network was activated only when forecasters expected dry snowfall without bright band effects.

Table 2. - Summary of snow storm periods sampled by the volunteer network near Albany, New York, during the 1995-96 winter. Activation refers to whether part or all of network was requested to make *S* and *SD* observations (see text).

Event No.	Start date-hour (UTC)*	Stop date-hour (UTC)	Activation
1	01 Dec-03	01 Dec-21	Partial
2	09 Dec-06	10 Dec-09	Full
3	14 Dec-09	15 Dec-06	Partial
4	19 Dec-18	21 Dec-06	Full
5	02 Jan-12	04 Jan-12	Full
6	07 Jan-12	09 Jan-00	Partial
7	12 Jan-15	13 Jan-15	Full
8	16 Feb-13	17 Feb-13	Full
9	02 Mar-09	02 Mar-21	Partial
10	05 Mar-06	06 Mar-06	Full
11	06 Mar-21	09 Mar-00	Full
12	29 Mar-03	29 Mar-18	Partial
13	09 Apr-20	11 Apr-00	Partial

* UTC = universal time coordinated

Not all volunteers were available during each storm, and the times during which each volunteer was able to take measurements varied from storm to storm to account for sleep time. During storms with full network activation, about 40 to 50 volunteers made hourly observations. Of course, specific measurement locations varied from storm to storm depending upon the availability of particular observers.

Volunteer observations were collected at the Albany WFO, entered into a computer data base by university students and, at this writing, are being double-checked for accuracy by comparison with nearby observations. In addition, two Belfort Universal gages were operated during the latter part of the winter at the locations shown in table 1. As with all Belfort gages used in this study, the two near Albany were equipped with Alter wind shields. The Albany gages were located in protected clearings in the forest.

The standard mode of clutter suppression with the Albany WSR-88D during storms of the past winter was to always use the Clutter Bypass Map and moderate suppression (personal communication with John Quinlan). No operator-designated boxes with zero suppression were attempted near Albany because of complex terrain and corresponding widespread ground clutter.

To date, no analysis of the Albany radar and snowboard data has been undertaken.

3.3. Denver, Colorado, Observations

Snow measurement sites within range of the Denver radar were selected by Reclamation meteorologists after consideration of several factors to be discussed, including examination of maps and considerable on-the-ground searching. The Denver area provides both prairie locations, chiefly affected by upslope storms, and mountain locations affected by various storm types.

It was hoped that at least one snow measurement site could be located within 20 km of the radar to provide the primary observations for calculation of α and β in equation (1). The area within more than a 50-km radius of the Denver NEXRAD is generally wide-open prairie with the metropolitan area built on the western portions. The "Denver" WSR-88D is actually located at the Front Range Airport, almost 40 km east of downtown Denver. Consequently, the ideal gage site of a small clearing within an extended forest does not exist near the radar.

Diligent searching of many stream bottoms, parks, cemeteries, military installations (Rocky Mountain Arsenal, Fitzsimons Medical Center, Lowry and Buckley Air Bases) and some light industrial complexes did not reveal adequately protected gage sites even within 30 km of the radar. A possible exception was that some shelterbelts were found consisting of rows of conifer trees partially or completely surrounding farm homes and buildings, some within 10 to 15 km of the radar. However, although such sites have considerable local protection, the areal extent of such protection is quite limited, rarely consisting of more than a few rows of trees. These shelterbelts, surrounded by large areas of open prairie and typically located on windy hilltops, likely would act as "snow traps." Shelterbelts probably catch well above average snowfall during the frequent periods with snow blown across the prairie by strong winds. In contrast, most blowing snow in an extended conifer forest would be trapped among the numerous tree tops so that any additional catch within small clearings should be minor.

After considerable examination of the land and structures within about 30 km of the Denver WSR-88D, the best snow observation sites were found in some extended subdivisions of homes. Some subdivisions consist of many 2-story houses with solid wood fences, typically 6 ft high, surrounding most back yards. Large established trees add to the general wind protection. Such sites may approximate the "ideal" clearing in conifer forest site. Literature evaluating the accuracy of snowfall measurement in fenced suburban yards is not known to exist, although the gage site classification scheme of Sevruk and Zahlavova (1994) considers just the vertical angle to the tops of obstacles, whether structures or trees. They found that sites surrounded by obstacles 20 to 25° above the horizon provided good protection.

Experience with suburban sites within 50 km of the Denver NEXRAD was encouraging during the 1995-96 winter. Hourly snowboard and wind observations were made at three carefully chosen sites in established neighborhoods. At each location, the large majority of hours with snowfall had winds less than 1 m s^{-1} at gage orifice level (~1 m a.g.l.).

Another major consideration in choosing snowfall observing sites is the presence or absence of ground clutter. This factor is especially important with the WSR-88D radar in its present configuration because no automatic record is made of the ground clutter suppression scheme in use at any time. Ground-clutter suppression is applied at the RDA (radar data acquisition) unit prior to data stream transmission to the RPG (radar product generator). Further description of the WSR-88D system and an overview of available analysis products is given by the Federal Meteorological Handbook No. 11 (1991) and Klazura and Imy (1993).

The ground clutter suppression scheme, described by Chrisman et al. (1994), can use a clutter bypass map, notch width map and operator-defined clutter suppression regions in various combinations to suppress returns in regions with potential or actual ground clutter. The particular clutter suppression scheme in use can affect both the resulting Z_e and Doppler velocity fields in unknown ways. MOUs were established between the OSF and the Albany, Cleveland, and Denver WFOs which provided, among other things, record-keeping of the suppression scheme in use. However, it was still judged best to avoid cluttered regions for

gage locations so the Z_e values over the gages were both uncluttered and uncontaminated by clutter suppression. One deliberate exception (Mt. Evans in table 3) provided data with which to evaluate the WSR-88Ds ability to detect snowfall over rugged mountains.

Table 3. - Locations of six snow observing sites in the Denver area. The three sites located nearest the radar had both Belfort gages and hourly snowboard readings. Latitudes and longitudes were measured by GPS, and elevations were estimated from a USGS (U.S. Geological Survey) data base or large-scale maps.

Gage Location (Operator)	Gage No.	Latitude (°-')	Longitude (°-')	m.s.l. Elevation (m)	Distance/Azimuth from Radar (km/°)*
SE Aurora (M. Bedell)	1	39-38.217	104-46.206	1758	25/229
NE Aurora (R. Kissinger)	2	39-45.609	104-49.322	1631	24/263
Lakewood (A. Super)	3	39-41.272	105-06.282	1703	49/257
Pine Valley (S. Becker)	4	39-24.400	105-20.843	2100	81/239
Mt. Evans (M. Jones)	5	39-39.316	105-35.642	3265	91/261
Black Forest (J. Bishop)	6	39-01.720	105-40.908 (no data until late January)	2295	85/188

Clutter bypass maps were obtained from the three radars in question and plotted on 1:500,000 scale topographic maps of each area. A clutter bypass map shows where ground clutter was detected by the radar during the particular few hours of special observation. The map is attempted to be made under well-mixed atmospheric conditions with no precipitation present. The bypass map is expected to approximate the clutter pattern during standard refraction. Actually, two maps are generated, one for the two lowest antenna tilts (0.5 and 1.5°) and the other for all higher scan angles. All future references will be to the former map because Z_e values from only the lower tilt were used in the analysis to be discussed.

Unfortunately, clutter bypass maps are not generated to the same 1° by 1-km resolution as Z_e data but are generated on a coarser 256-radial by 360° pixel grid. In addition to the problem of the mismatch in Z_e and clutter bypass map coordinates, the bypass map tends to "smear" potentially cluttered regions. Much clutter is "point-target" echo (Paul Smith, personal communication), which is probably over-emphasized by the bypass map pixel scale.

The Denver bypass map, generated in October 1995 and used throughout the 1995-96 winter, showed few continuous uncluttered areas within 30 km of the radar. The uncluttered areas were located to the east and southeast where prairie exists with almost no tree cover or housing developments to provide protection for snow measurements.

The extent of bypass map clutter at moderate ranges west of the Denver radar was surprising because a minor ridge about 12 km distant for azimuths between about 250 and 280° (all azimuths are given in degrees [°] true in this report) reaches the bottom of the 0.5° tilt beam (the beam is 0.95° wide). This ridge would be expected to absorb any sidelobe energy under the lowest elevation angle beams for which the lower clutter map is generated. Elevations decrease beyond the near ridge until about the 40-km range (South Platte River) and then

gradually increase into the foothills of the Rocky Mountains. The region between about 12 to 50 km from the radar for the 30° segment noted would be expected to be clutter free. However, the bypass map shows considerable clutter in that area, perhaps caused by edge diffraction over the intervening ridge.

The Denver clutter bypass map was compared with predictions from two computer programs. The OSF provided a software package (RDRHGT.FOR) written by an NWS meteorologist. The software package provides a visual presentation on a personal computer monitor of terrain underlying the radar beam for any selected 1° of azimuth. Both beam tilt and refraction can be varied. A Reclamation meteorologist wrote separate software to plot a PPI (plan-position indicator) format map of where clutter should exist under standard refraction, and this map is in very good agreement with RDRHGT indications. Both software packages use USGS (U.S. Geological Survey) elevation files to compute where the radar beam should be relative to the underlying terrain.

The October 1995 Denver clutter bypass map showed significantly more extensive clutter than predicted by the computer programs, and a bypass map generated in April 1995 showed even more clutter. The presence of tall buildings and towers in the Denver area could explain some but not all of the observed clutter. The clutter bypass maps may be more sensitive to clutter returns than necessary for effective clutter suppression. The clutter bypass map makes no distinction between very weak and very strong ground returns, and each pixel (somewhat larger than a 1-km by 1° range bin) is only designated as either cluttered or not.

After consideration of the disagreement between physical reasoning and the bypass map results, the snow measurement sites were located nearest the Denver radar where the RDRHGT program suggested lack of ground clutter. Even then, the nearest sites had to be established about 25 km from the radar, farther than desired, because protected snow measurement locations could not be found closer to the radar. As an additional approach, an operator-selected "box" was designated over one of the two measurement sites nearest the radar. Zero suppression was applied in the boxed area over gage No. 2, between 18 to 30 km in range and 248 to 278° in azimuth, from February 22 until the end of April 1996 when observations were terminated. This zero suppression box was used in both precipitation and clear air scanning modes. Future careful examination of bin-by-bin returns within this box should reveal whether ground clutter was present or not during snowfalls. If present, it is expected that particular range bins can be selected which avoid contaminated areas and, thereby, provide only meteorological returns from snowfall.

Besides use of the operator-designated box over gage No. 2, the standard mode of clutter suppression with the Denver WSR-88D during the past winter was to use the clutter bypass map and moderate suppression during periods when the radar was in precipitation mode scanning. In addition, two other operator-designated boxes were routinely used in more cluttered areas, neither of which affected the gages of table 3. These boxes were located from 270 to 330° and 60 to 180 km, where maximum suppression applied; and from 330 to 060° and 60 to 180 km, where moderate suppression was applied. When in clear air mode scanning, the entire area of Denver radar coverage was designated an operator-selected "box," and maximum suppression was applied everywhere because of frequent ground-based inversions and superrefraction (personal communication with Mike Holzinger, Denver radar focal point).

Besides the two measurement sites at about 25 km from the radar, four other sites were chosen at greater distances as shown in table 3. As with the two nearest sites, the observing location at the 49-km range was located in an established neighborhood with solid fenced backyards in a region that should not have had ground clutter. In addition to operation of the shielded Belfort gages at these three sites, hourly manual measurements were made of *S* and *SD* on 1- by 1-ft snowboards laid on the ground or snow cover near the gages. The same Clear Vu Model 1100 gages used in Albany were used to core snow on the boards after four depth measurements were made and averaged. Manual measurements were usually made from early morning until normal bedtime whenever snow was falling. Comparisons of all hourly snowboard and gage observations from gages No. 1 through 3 indicated that the 3 operators were conscientious. Resulting correlation coefficients between *S* pairs exceeded 0.96 for each data set.

Denver WFO personnel made special hourly observations of *S* and *SD* whenever snowfall occurred during the 1995-96 winter/spring. These measurements and the snowboard observations from the 3 gage sites nearest the Denver WSR-88D have yet to be analyzed.

Three additional sites were chosen for operation of Belfort gages but not hourly snowboard observations. However, snowboards were sampled in the same manner just discussed each morning after snowfall had occurred. This procedure provided comparisons with gage measurements and identified days with snowfall. The same practice was used at the five Cleveland gages to be discussed.

All Belfort gages used in this study had accurate clocks that rotated once per 24 h, which meant the pen trace overwrote the same horizontal line until precipitation occurred. The need to identify days with precipitation is obvious. This identification was not difficult because gage operators filled out a worksheet each morning after snow or rain, noting current weather conditions including wind speed at gage orifice level. A small Taylor wind speed meter with a minimum indication of 2 mi h^{-1} was used to measure wind speed.

Little data resulted from the Black Forest gage listed in table 3 until after late January 1996 because of difficulties finding a reliable operator. The data quality from the other five gages in the Denver area was generally very good. The three most distant gages were located in well-protected clearings in conifer forest. The two gages well west of the radar were located in a large mountain valley, protected from ground clutter by upwind terrain, and in a mountainous area where the lowest tilt beam intersects the terrain. The latter site, at 3265 m altitude, was located at the Mt. Evans Research Station climatological station. Measurements from this location will be examined to determine if any useful snowfall accumulation estimates can be made by radar in a very cluttered region. The 1.5° tilt beam may be usable for this purpose, but this estimate has yet to be attempted.

Five gage sites were operational by November 1, 1995. The sixth gage was also established by then, but reliable observations were not obtained until after late January when a new operator was trained. Gages (all with Alter-type wind screens) were installed and calibrated, and antifreeze and other supplies were located at each site. Each observer received training concerning gage servicing. The two observers nearest the radar received additional training in making hourly measurements of snow depth and in observing sizes and types of the larger snowflakes that provide almost all of the meteorological radar returns during snowfall. The third site with special hourly observations (Lakewood) was maintained by a Reclamation meteorologist. A snow particle identification guide was prepared for use by hourly observers.

The observers mailed gage charts and weather observation forms on a weekly basis so quality checks could be made by Reclamation meteorologists. Meteorologists then called or visited observers to clear up any problems.

Table 4 lists the significant snowfall events detected by the Denver area gage network. Some events affected only the Mt. Evans gage in the mountains as noted. Although 28 snowfall events occurred, all but a few were minor precipitation producers near Denver.

Table 4. - Summary of significant snowstorm periods sampled by Belfort gages and snowboards near Denver, Colorado, during the 1995-96 winter. Seven storms affected only gage No. 5 in the mountains as noted.

Event No.	Start date-hour (UTC)	Stop date-hour (UTC)	Event No.	Start date-hour (UTC)	Stop date-hour (UTC)
1	01 Nov-19	02 Nov-14	15	18 Feb-07	18 Feb-17*
2	10 Nov-02	10 Nov-22	16	20 Feb-19	21 Feb-15*
3	27 Nov-02	28 Nov-20	17	26 Feb-14	26 Feb-23
4	05 Dec-08	05 Dec-21*	18	27 Feb-15	28 Feb-08
5	06 Dec-14	07 Dec-02*	19	05 Mar-19	06 Mar-19
6	17 Dec-22	18 Dec-08	20	14 Mar-05	14 Mar-21
7	31 Dec-15	02 Jan-17	21	17 Mar-16	18 Mar-02
8	03 Jan-22	05 Jan-22	22	18 Mar-07	18 Mar-16
9	17 Jan-17	18 Jan-05	23	24 Mar-00	24 Mar-20
10	19 Jan-18	20 Jan-11*	24	04 Apr-17	05 Apr-13
11	25 Jan-11	26 Jan-05	25	13 Apr-13	14 Apr-03
12	30 Jan-22	31 Jan-05	26	18 Apr-01	18 Apr-06*
13	31 Jan-21	01 Feb-07	27	18 Apr-20	19 Apr-13*
14	02 Feb-01	02 Feb-07	28	28 Apr-07	29 Apr-00

* - storms which affected only gage No. 5

3.4. Cleveland, Ohio, Observations

Cleveland WFO personnel provided major help in gage site selection. They broadcast an appeal on NOAA (National Oceanic and Atmospheric Administration) weather radio for persons interested in making snow observations who lived in the region of interest (parallel to the south shore of Lake Erie) and who had sites on their property that were well protected from the wind. Their appeal resulted in a list of several potential sites. A Reclamation meteorologist then visited all potential sites and selected the five which provided both good protection from wind and a network providing near, moderate, and far ranges from the radar in areas without ground clutter. Immediately after site selection, gages were installed and operators were trained. In addition, Reclamation hired Bob Paddock, a retired NWS forecaster, to assist in gage installation and to serve as the first point of contact whenever problems developed with the gage network. Mr. Paddock also visited all gages and operators on approximately a monthly basis to check gage level and calibration. As in the Denver area, Cleveland area gage charts and associated checklist forms were sent to Reclamation meteorologists on a weekly basis for quality control.

The Cleveland approach worked very well. Selecting gage operators who showed definite interest in the project by responding to the NOAA weather radio appeal and hiring a part-time local meteorologist to oversee the project resulted in a high quality data set. Few problems were encountered with the Cleveland area gage network. The enthusiasm and dedication of all five gage operators and Mr. Paddock were remarkable.

Table 5 lists the five Belfort gage locations and gage operators. As with Albany and Denver sites, latitudes and longitudes were determined by GPS, and altitudes were extracted from a terrain data base or large-scale maps.

Table 5. - Locations of five Belfort gage sites in the Cleveland area. Gages No. 1 to 4 were located in Ohio, and gage No. 5 was located in northwestern Pennsylvania.

Gage Location (Operator)	Gage No.	Latitude (°-')	Longitude (°-')	m.s.l. Elevation (m)	Distance/Azimuth from Radar (km/°)*
Highland Hts (A. Marshall)	1	41-33.126	81-28.348	292	36/64
Chardon (D. Brady)	2	41-36.712	81-10.382	359	61/69
Geneva (M. Bezoski)	3	41-46.552	80-56.317	247	87/62
Pierpont (P. Mechling)	4	41-44.330	80-32.780	317	115/71
Cranesville (D. Henderson)	5	41-55.299	80-14.301	373	146/67

All five Cleveland area gages were brought to operational status during the first week of November, well ahead of the mid-November schedule called for under this task. Snowstorms began affecting the gage network after the first three gages were installed. Because of record snowfalls in the area, many storm periods were observed by the gage network and by radar. Most were lake effect storms, but some general synoptic storms were also observed.

A listing of all storms that produced more than minor snowfall in the Belfort gage line is given in table 6. This table was prepared from a larger list supplied by Bob LaPlante and Frank Kieltyka of the Cleveland WFO which includes all snowfall events, some very minor along the gage line. Messrs. LaPlante and Kieltyka classified each storm into one of three categories: Major LES (lake effect storms), defined by production of at least 6 inches of snowfall in no more than 12 hours at 2 or more sites in a county; major synoptic storms; and other less prominent events which were not classified (mixture of minor lake effect storms and "other"). The latter are called "Minor Events" in table 6, although some produced significant snow at one or more gages.

The standard method of clutter suppression with the Cleveland WSR-88D during the past winter was to use the clutter bypass map and moderate suppression except from November 1 through December 4, 1995, when high suppression is believed to have been applied. In addition, an operator-designated box was established over the gage nearest the radar from 29 to 43 km and 53 to 75° azimuth. Zero suppression was applied within the box from December 4, 1995, until the end of the field season (personal communication with Bob LaPlante, Cleveland Science Operations Officer).

Table 6. - Summary of significant snowfall periods sampled by Belfort gages east-northeast of Cleveland, Ohio, during the 1995-96 winter.

Event No.	Start date -hour (UTC)	Stop date -hour (UTC)	Storm Type	Event No.	Start date -hour (UTC)	Stop date -hour (UTC)	Storm Type
1	04 Nov-00	05 Nov-05	Major LES	13	12 Jan-00	13 Jan-04	Minor Event
2	08 Nov-08	09 Nov-19	Major LES	14	24 Jan-12	25 Jan-05	Minor Event
3	15 Nov-12	16 Nov-18	Major LES	15	30 Jan-12	31 Jan-06	Minor Event
4	17 Nov-05	17 Nov-23	Minor Event	16	11 Feb-12	12 Feb-12	Minor Event
5	21 Nov-11	22 Nov-16	Major LES	17	14 Feb-00	15 Feb-07	Minor Event
6	09 Dec-00	12 Dec-17	Major LES	18	17 Feb-12	18 Feb-08	Minor Event
7	19 Dec-06	20 Dec-04	Major Synoptic	19	28 Feb-18	01 Feb-00	Major LES
8	20 Dec-10	21 Dec-20	Major LES	20	02 Mar-12	03 Mar-04	Minor Event
9	22 Dec-12	24 Dec-20	Minor Event	21	03 Mar-04	04 Mar-05	Major LES
10	25 Dec-17	26 Dec-13	Major LES	22	08 Mar-23	09 Mar-12	Major LES
11	02 Jan-14	04 Jan-00	Major Synoptic	23	01 Apr-00	02 Apr-03	Minor Event
12	09 Jan-21	10 Jan-14	Major LES				

4. Data Processing

4.1 Belfort Gage Chart Reduction

Considerable care has been given to the time-consuming task of reducing Belfort gage charts. Clock accuracy was generally within 5 minutes per week or less, and charts were changed at least weekly. Start and stop times were carefully noted, and an additional mid-week time check was made as a minimum. Clocks which did not maintain an accuracy of 5 minutes per week were adjusted (spring-wound clocks) or replaced (quartz-crystal electric clocks).

Charts were carefully read with at least 4-power magnification and good lighting to the nearest 0.005 inch at hourly intervals. The resolution claimed for the Belfort gage is 0.01 inch, but pen trace movements of 0.005-inch magnitude are readily apparent with careful examination.

The "raw" chart reading was penciled in by each hour's line on the chart. Later, a second check was made of each reading to detect any errors. After computer processing, hourly snowfall accumulations were compared among neighboring gages and any obvious outliers were checked yet again on the charts. Moreover, all high snowfall accumulations were given an additional special check because they are uncommon but important in calculation of the optimum Z_e-S relationship.

Special attention was paid to chart readings near the daily time of chart overlap (0700-0800) because the extra thickness of the overlapping paper causes the pen to rise then fall over a few-hour period. The Belfort Charts No. 5-4047-B have a triple paper thickness when normally installed (two end flaps with one folded over), but one flap was cut off to minimize the vertical pen motion. Even then, readings near the overlap can be misleading if one simply uses the chart readings; that is, the pen trace relative to the horizontal lines that are inked at 0.05-inch intervals on the charts. A better approach, which was used, is to compare the differences in chart readings near the overlap with days having no precipitation. Because 7 rotations (traces) were obtained on most charts, non-precipitation periods were almost always available to compare against. Changes in the differences hour by hour provided the best estimate of snowfall accumulation near times of chart overlap. The raw chart reading was used at other times. These raw readings were entered into a computer file and a program was written to calculate hourly differences. Because charts were operated in local

time to avoid confusing the operators, the program also listed both local and UTC (universal time coordinated) date and hours because radar data are recorded in UTC time.

Some data losses occasionally occurred, usually because of clock stoppages or gage malfunctions. The only major data loss was the previously discussed Black Forest gage south of Denver. Overall, the gage record is considered to be of very good quality, both because of careful locating of gages so they were protected from the wind and because of considerable care taken to calibrate and maintain the gages and in chart reduction.

4.2 Processing of Level II Radar Tapes

Copies of original Level II 8-mm Exabyte tapes from the Albany, Cleveland, and Denver radars have been, or will be, obtained from the NCDC (National Climatic Data Center) in Asheville, North Carolina, for all periods with significant snowfalls. Tape processing was done with a Sun Sparc 20 workstation under the Solaris 2.4 operating system. The first step was to use a routine that extracted each file's start and stop times and size and then closed the file. The resulting file directory from this initial processing provides the means to determine exactly which files are needed from each tape to match snowstorm periods in later processing.

File number 1 on Level II tapes contains header information and numbers 2 to 401 are data files unless recording is interrupted before the last file, 401, is written. One volume scan is usually equivalent to one file although occasionally two volume scans are erroneously written to a single file. Software has not yet been developed to extract the second file in such cases. Occasionally, a short file is written which may contain little but header information and perhaps part of a volume scan. All files less than 4 minutes in duration were ignored in later processing.

File size indicates scan strategy. For example, a typical precipitation mode volume scan (see Federal Meteorological Handbook No. 11, 1991) during snowfall requires about 5.75 minutes and produces a 9.8-megabyte file. Clear air volume scans take almost 10 minutes and produce somewhat smaller files.

The second step in processing is to open only the desired files on each tape and save the fraction of the total information needed for comparison with snow observations. The Z_e values, recorded to the nearest 0.5 dBZ, where dBZ = 10 LogZ_e, were extracted for *two* arrays of range bins over each snow measurement site from the lowest (0.5°) beam. One array was centered directly over the site (so-called "vertical" array) as though snow fell vertically without advection. The other array was advected by the wind (so-called "upwind" array) in an attempt to make a more realistic match between gage position and the region of the 0.5° radar beam from which the snow actually fell.

For each type of array and each snow measurement site, dBZ values were extracted over semi-equal areas. Each array consisted of a "box" exactly 3 km in range (depth) by at least 3 km in azimuth (width). A minimum of 3° of azimuth was always used so the azimuthal width was greater than 3 km at ranges exceeding 60 km where 1° equals 1 km in width (e.g., three 1° radials result in a 6-km width at a 120-km range). The smallest array was, therefore, 3 x 3 = 9 range bins. For ranges nearer to the radar than about 50 km, additional radials were added to the array to maintain an approximate 3-km width. Radials were added in steps of two to keep an odd number (5, 7, 9, . . .) of degrees azimuth by a 3-km range in

the array. Because the recorded radial azimuth values vary from scan to scan, the exact azimuthal position of the array over a gage also varied slightly. In contrast, range was always fixed with the center of each range bin exactly at 0.5, 1.5, 2.5, . . . km from the radar. In any event, all vertical arrays were chosen so that the gage was always positioned directly below the *center* range bin of the array.

For the purpose of extracting upwind arrays, VAD (vertical azimuth display) winds were calculated at 1000-ft intervals for each volume scan (file) as discussed in section 10.2. This process simulates the standard WSR-88D VAD product, which portrays winds in 1000-ft intervals. The wind information was used to advect the falling particles. Snowflake fall speed observations are not made with the WSR-88Ds. Although the Doppler shift could be used to estimate particle fall speeds, a different scan strategy would be required, including an almost vertical antenna tilt. In the absence of fall speed measurements, the larger snowflakes, which produce almost all the returned signal (Z_e), were assumed to fall at exactly 1.0 m s^{-1}. Of course, graupel (snow pellets) may fall more than twice that fast and snowfall consisting of individual, large, non-aggregated ice crystals may fall half that fast. So the constant fall speed used is only an approximation that may or may not improve the Z_e-S relationship in general. Comparisons between radar-estimated snowfall accumulations from both vertical and upwind arrays and underlying gages will be used to judge the improvement provided by the advection scheme. Some improvement might be anticipated because snow particles often can be advected for tens of kilometers between the lowest tilt radar beam and the ground, especially at long ranges where the beam is high above the ground.

The method of advection calculation starts at each gage location and elevation and then steps upward at 1000-ft intervals, calculating the horizontal motion of a particle falling at 1.0 m s^{-1} within each 1000-ft layer using the VAD wind velocity for that layer. This process is continued upward until the center of the 0.5° radar beam is reached. That point in space then becomes the center of the upwind array.

5. Comparison of Cleveland and Denver Snowfalls

The data from gages No. 1 to 5 in both tables 3 (Denver) and 5 (Cleveland) were summarized for general information. Radar and gage observations from all Denver area storms listed in Table 4 from November 1, 1995, through January 30, 1996 (3 months), are considered in this report. Similarly, all Cleveland area storms listed in table 6 from November 3, 1995, through January 3, 1996 (2 months), are considered. The later storms listed in tables 4 and 6 have yet to be analyzed.

Table 7 shows the summation of all observed snow water equivalent for the five gages in each area. Records were complete or almost complete at most gages. The two exceptions were Cleveland gages No. 1 (39 hours missing) and 4 (30 hours missing), which missed the season's first storm simply because they had yet to be installed. More hours with snowfall are considered in table 7 than in tables 8 or 10 because hours with missing radar data are included in table 7.

Significantly more snow fell in the Cleveland gages in 2 months than in the Denver gages over 3 months. Gage totals east-northeast of Cleveland ranged between 2.86 and 4.88 inches. In contrast, the gages located within 50 km of Denver received only 1.23 to 1.84 inches, the mountain valley gage (No. 4) only 0.52 inch and the mountain gage (No. 5) 3.01 inches. Although the 1995-96 winter had record snowfalls northeast of Cleveland, the Denver area

Table 7. - Summary of snow water equivalents for 5 Cleveland gages for the first 2 months of the winter, and for 5 Denver gages for the first 3 months of the winter. Units are inches or inch h^{-1} snow water equivalent.

Cleveland gage number	1	2	3	4	5
Total Snow Water Equivalent	2.86	4.88	4.78	4.72	4.35
Hours with Detectable (0.005 inch) Snowfall	146	241	190	205	225
Half Total *S* Produced at Hourly Accumulations ≤	0.030	0.035	0.050-	0.040+	0.030+
Median Hourly Accumulation	0.010	0.010	0.010	0.010	0.010
Average Hourly Accumulation	0.020	0.020	0.025	0.023	0.019
Maximum Hourly Accumulation	0.100	0.140	0.195	0.120	0.155
Denver Gage Number	**1**	**2**	**3**	**4**	**5**
Total Snow Water Equivalent	1.23	1.25	1.84	0.52	3.01
Hours with Detectable (0.005 inch) Snowfall	71	55	92	35	136
Half Total *S* Produced at Hourly Accumulations ≤	0.030	0.035	0.030-	0.025	0.035
Median Hourly Accumulation	0.010-	0.010	0.010	0.010-	0.010
Average Hourly Accumulation	0.017	0.023	0.020	0.015	0.022
Maximum Hourly Accumulation	0.095	0.115	0.080	0.065	0.175

experienced a very dry winter. Numerous storms occurred near Denver (table 5), but they typically produced light snowfall accumulations over limited periods. Even the mountain gage west of Denver received fewer hours of detectable snowfall than any of the Cleveland gages. The gages near Denver had only 55 to 92 hours with measurable snowfall as compared to 146 to 241 hours for the Cleveland gages. The frequency of snowfall is the main difference between the Cleveland and Denver area data of table 7; average and median snowfall accumulations are similar. Because of the lack of any major storms near Denver, maximum snowfall accumulations tended to be higher in the Cleveland area, with the expected exception of the Denver mountain gage (No. 5).

Calculation of the median hourly rates provided almost identical values at all 10 gages—near 0.01 inch. No doubt the medians would be even lower if gage resolution was less than 0.005 inch. Average accumulations were also similar at most gages, near 0.02 inch h^{-1}. Moreover, half of the total snow water equivalent at each gage occurred at similar low hourly accumulations—from 0.025 inch h^{-1} or less at the dry mountain valley site to 0.050 inch h^{-1} or less at Cleveland's gage No. 3. The other 8 gages all produced half of their total snowfall at accumulations less than or equal to 0.030 to 0.040 inch h^{-1}.

These measurements illustrate that most hours with snowfall in both geographical locations had relatively low accumulations. Half the hours had accumulations of 0.01 inch or less. Yet relatively low accumulations are important as indicated by half the snowfall totals usually occurring at hourly accumulations less than or equal to 0.03 to 0.04 inch. These similarities are especially interesting because most hours of snowfall in the Cleveland area are from lake effect storms and most in the Denver area are from upslope storms.

Cleveland gage No. 1 had the fewest hours with detectable snowfall in that area, partly because it was not installed in time for the first early-November storm and had one later period of missing data. But had no hours been missed, gage No. 1 still would have received fewer hours of snowfall than gages located farther from the radar which are more influenced by lake effect storms. When it is realized the gage No. 4 had 30 h of missing data because it also was not operational for the first storm, the snowfall frequency did not vary much for gages No. 2 to 5. The Cleveland line of gages was located along relatively flat terrain as shown by their limited range of elevations in table 5.

Denver gage No. 1 received fewer hours of snowfall than gage No. 2 at almost the same range. This result likely occurred because gage No. 2 is located 127 m higher in elevation on a minor ridge which may produce some local orographic uplift. Gage No. 3 had a higher snowfall frequency than higher gage No. 2, probably because gage No. 3 is located farther west, near the foothills of the Rocky Mountains. Snowfall was infrequent at gage No. 4, in a broad mountain valley, likely because of "rain shadow" effects. Even gage No. 5, located high in the mountains, received only 136 hours of snowfall during the dry 3-month period.

6. Calculation of Radar-Estimated Snowfall Accumulations

Once arrays of range bins were obtained for each volume scan, the next step was to provide any needed adjustment to the Level II Z_e data. Adjustments were provided by William Urell, OSF radar engineer, after he made careful calibration checks and studied the hourly records of overall WSR-88D performance from Albany, Cleveland, and Denver over the entire winter. For example, an offset of +0.3 dB was applied to all Denver data before January 31, 1996, the date of a major calibration, and no correction was needed thereafter. Higher adjustments were decided upon for the period of Cleveland data reported herein, ranging between +0.7 and +1.1 dB depending on the storm period. Mr. Urell estimated the RMS (root mean squared) error of the radar estimates at about 0.7 dB.

Corrected Z_e values from the *lowest available tilt* (0.5°) were always used to estimate snowfall accumulations using particular values of α and β for equation (1). It is important to always convert recorded values in dBZ to snowfall rate before any averaging is done. Although often done, averaging values of Z_e in $mm^6\ mm^{-3}$ or in dBZ over time or space can cause significant bias. For example, assume $\alpha = 300$, $\beta = 1.4$, and 3 adjoining range bins are averaged which have Z_e values of 20, 25, and 30 dBZ. Use of the average of 25 dBZ would lead to an estimated snowfall of 0.041 inch h^{-1}. However, first converting each range bin's Z_e to snowfall rate leads to an average of 0.051 inch h^{-1}, a 24-percent higher value in this example. The approach used throughout this study always involved converting each range bin's Z_e value to snowfall rate before averaging all range bins in an array.

In like manner, precipitation rates, not values of Z_e, were averaged over time. In the results presented here, average array snowfall rate for each volume scan was weighted equally with every other volume scan that had a start time (near the time of 0.5° scanning) within the

hour of interest, and a simple average accumulation was calculated for the hour. Occasional hours had scans from both clear air and precipitation mode scan strategies, representing about 10- and 6-minute samples, respectively. But weighing each volume scan by the time interval it represents would have little influence on the results because hours rarely had different scan strategies.

Because each precipitation gage chart was read at the end of each hour, the radar snowfall estimates were for the same hours. Future work will investigate whether any significant improvement results from shifting the hourly radar-estimated snowfalls forward to allow time for the snowflakes to fall from the 0.5° beam to the surface. The present results are based on zero time lag between the two observational sets.

Future work will partition storms by types (upslope, lake effect, general synoptic, etc.) and other means. However, the results herein are based on *all* available data extracted so far with no partitioning. A few minor storms were ignored at both Cleveland and Denver if they were of limited duration (few hours) and if hourly totals did not exceed 0.02 inch. Many other storms provided an abundance of hours with accumulations of 0.02 inch h^{-1} or less in addition to higher accumulations.

Because gage charts were reduced at the "top" of each hour, hourly snowfall accumulations are for 1200 to 1300, 1300 to 1400, etc. UTC on a 24-hour per day basis. Individual radar volume scans were extracted for all the same hours for which gage charts were reduced; that is, from the beginning to the end of any detectable snowfall by any gage during a storm episode. This approach resulted in many hours with zero detectable snowfall by some or all of the gages. However, unless otherwise stated, only hours with detectable snowfall (at least 0.005 inch) were used in the analyses to follow. This approach ignores many hours with light radar-estimated accumulations. Such hours may have had virga, snowfall that missed the gage, or snowfall that was too light to be detected by the gage.

In general, WSR-88D antennas never stop except for problems, maintenance, etc. When a volume scan is completed by making the highest tilt scan, the antenna is moved to the lowest elevation tilt and the next volume scan starts at 0.5° tilt. Volume scans do not take exactly 5, 6, or 10 minutes as suggested by Federal Meteorological Handbook No. 11 (1991), but slightly shorter times, like 5.75 or 9.67 minutes, which might vary from scan to scan. Scans were separated by several seconds as the antenna moved from highest to lowest tilt. As a result, most hours had from 9 to 11 volume scans with the precipitation mode scan strategy usually used during snowfall. As few as 4 volume scans were accepted as adequate for the occasional hours where continuous scanning was interrupted for whatever reason. Four volume scans require about 23 minutes in precipitation mode scanning and about 39 minutes in clear weather mode scanning, which is sometimes used at the beginning and ending portions of storm periods. But the large majority of hours had continuous radar coverage.

7. *Z-S* and Z_e-*S* Relationships for Snowfall

7.1 Introduction

The fact that sensitive radars can *detect* and *track* precipitation echoes is of major importance to weather forecasting and other weather-related activities such as aircraft operations. But use of well-calibrated radars is also important to provide *quantitative* precipitation accumulation estimates over large areas. To do so clearly requires that a relationship be

established between what a radar measures, Z_e, and R, the rainfall accumulation, or S, the snowfall accumulation.

Many studies have been published concerning Z-R relationships for rain, usually of the form of equation (1) with the α and β coefficients empirically determined, using units of mm^6 per m^3 for Z and mm h^{-1} for R. For a variety of reasons involving microphysics, vertical air motions and other factors, no unique relationship exists for either rain or snow. Progress has been made in optimizing values of α and β for different types of rain and different geographical regions. However, the current practice with the network of WSR-88s is to use $\alpha = 300$ and $\beta = 1.4$ as the default values in equation (1) for rainfall estimation nationwide.

Less work has been done with Z-S relationships for snowfall. Reasons for more emphasis on rain include greater importance of rainfall to society in general, and the increased difficulty of detecting snowfall because of its generally lighter rates, typically producing melted water equivalent values in the range less than 0.10 inch h^{-1}. Such accumulations are well below common rain accumulations, especially from convective storms. Therefore, snowfall requires sensitive radars for detection at even moderate ranges. Moreover, because the signal-to-noise ratio is often relatively low for snowfall intensities, and because snowfall often develops in shallow clouds, ground clutter returns can more easily hinder attempts to quantify snowfall.

Discussing the merits of various ground clutter suppression schemes is beyond the scope of this report. Section 3.3 discusses the difficulties of not having an automatic electronic record of how, when, and where ground clutter suppression was applied with WSR-88D radars. It is certainly recommended that steps be taken to record suppression methods in use on future Level II tapes. The general approach taken in development of this Algorithm has been to attempt to measure S in areas around Cleveland and Denver which presumably have little or no ground clutter, even when that approach meant locating snow measurements farther from the radar than desired for development of Z_e-S relationships. The alternative was to measure S within areas shown to have potential ground returns by the clutter bypass maps. This practice would result in Z_e observations that may have been suppressed to unknown degrees, depending on the radial wind speed, the degree of suppression applied (notch width map setting), and other factors. That approach could only add unknown variance to development of a Z_e-S relationship because the "true" Z_e could never be determined.

Two basic, but fundamentally different, approaches can be used to relate radar measurements to precipitation in general, and snowfall in particular. With the most common approach, the reflectivity factor is *calculated* from surface observations of snow particles without involvement of radar measurements. The definition of the reflectivity factor, Z, is the summation of the sixth powers of melted drop diameters divided by the contributing volume V_c; that is:

$$Z = \sum D^6 / V_c \tag{2}$$

This approach has a number of problems, including uncertainties in the density and fall speeds of individual snowflakes needed to estimate the volume each particle represents. These uncertainties are especially acute when often-present aggregation and/or riming are important to the snowfall progress. Snowflake densities and fall velocities are usually estimated from empirical functions with particle size, introducing the uncertainty of how representative these functions are for the particular snowfall being sampled. Another problem is obtaining sufficient particle observations to be representative of the large volumes

sampled by radar. Although several uncertainties exist, this method has produced the majority of published Z-S relationships.

The alternative method, with its own set of uncertainties, is to relate radar measurements of Z_e to surface observations of snowfall. Equivalent reflectivity factor, Z_e, is essentially equal to Z for spherical drops with diameters small compared to the radar wavelength (i.e., rain). However, Smith (1984) makes the important and often overlooked point that Z_e and Z are *unequal* for snowfall. Smith shows that if melted drop diameters are used as the particle sizes in calculating Z (a common practice), then:

$$Z_e = 0.224\ Z \tag{3}$$

Therefore, Z values are over a factor of 4 higher than Z_e for the same snowfall conditions, and the two quantities should not be grouped together. But they often are, accounting for some of the published wide ranges of values for the coefficient α in equation (1) (β is unaffected).

7.2 Review of Some *Z-S* Relationships

For simplicity, the common approach of using the term "reflectivity" for measured equivalent reflectivity factor (Z_e) will be used in the remainder of this report. It should be understood that reflectivity is not the same as reflectivity factor (Z), a calculated quantity.

Published values of α and β for equation (1) applied to snowfall are scarce. Sekhon and Srivastava (1970) reanalyzed four previous studies based on snow particle data in arriving at one commonly used expression for snowfall:

$$Z = 1780\ R^{2.21} \tag{4}$$

The pioneering study by Wilson (1975) compared 5-cm radar observations of Z_e, converted to snowfall by equation (4), with measurements from a special network of Universal gages generally located in small clearings in coniferous forests near Oswego, New York. Unfortunately, Wilson did not attempt to develop a "best fit" Z_e-S relationship for the specially-collected snowfall observations, which are probably some of the best ever obtained for relating to radar measurements. Climatological gage data were also used which clearly produced more scatter than the special network. Equation (4) does not appear to have been converted from Z to Z_e which, using equation (3), would have resulted in:

$$Z_e = 399\ R^{2.21} \tag{5}$$

Wilson found a marked range dependence in gage/radar ratios using a 1.7° beamwidth tilted at a 0.9° elevation angle. Ratios increased from about 3.4 for the nearest gages, about 30 km from the radar, to values well over 20 at ranges beyond 100 km. Besides making it obvious that some sort of range correction scheme is needed with snowfall, Wilson pointed out the importance of keeping the radar beam narrow and close to the ground.

The ratios reported by Wilson (1975) (see his fig. 2) would be reduced by about half if equation (5) were applied rather than equation (4). Nevertheless, the radar estimates would still have been significantly lower than the protected gage observations. This difference suggests that one or both coefficients in equations (4) and (5) may be too high for (mostly lake effect) snowfall near Oswego, New York.

Boucher and Wieler (1985) compared *SD* observations in cm h^{-1} against Z_e measurements made by a 3.2-cm radar for 6 Massachusetts snowstorms. Their Z_e-*SD* relationship, when converted to snow water equivalent by assuming the commonly-used 10:1 *SD*:*S* ratio and using conventional units of mm h^{-1} results in:

$$Z_e = 227\ S^{1.65} \tag{6}$$

SD:*S* ratios vary considerably with snowfall type, degree of riming, and other factors. Therefore, using an equation like (6) developed for *SD* may provide only a crude estimate of *S*. For example, the α coefficient increases dramatically for low density snow. If the *SD*:*S* ratio is 15:1, the α coefficient in equation (6) becomes 442; at 20:1, it becomes 711.

Smart and McGinley (1989) used hourly and 3-hourly observations of both *SD* and *S* from a large volunteer network in the Boulder-Denver, Colorado, area. Radar observations were made with a 10-cm radar with characteristics similar to the WSR-88D. Two case studies were presented, and the measurements of *SD* expressed in cm h^{-1} were compared to a reference relationship:

$$Z_e = 200\ SD^{1.6} \tag{7}$$

The α value in equation (7) would have the same value for *S* expressed in the usual mm h^{-1} if the *SD*:*S* ratio was 10:1. However, if the ratio was 20:1, the α value would be 606.

Plots of observed *SD* versus average Z_e (in dB*Z*) showed that equation (7) rose about as rapidly as the observations with increasing Z_e in about the 25- to 35-dB*Z* range in both case studies. However, the observations made their steep exponential rise at about 5-dB lower values than predicted by (7) in one case with "wet" snow averaging 12:1 *SD*:*S* ratios. In the other case, with "dry" snow and *SD*:*S* ratios of about 25:1, the observations rose rapidly in the 15- to 20-dB*Z* range, well lower than predicted by (7).

Plots of observed *S* versus Z_e were compared to three similar reference curves including equation (4). Smart and McGinley (1989) concluded that the reference relationships did not provide enough reflectivity to satisfy the data. But no attempt was made to develop a better-fitting Z_e-*S* or Z_e-*SD* relationship with their data set.

Further comparisons with equation (7) were presented by Smart and Albers (1991) for two upslope storms which affected a large volunteer network from Colorado Springs to Fort Collins, Colorado, and far eastward onto the prairie. A 10.4-cm radar was used, similar to a WSR-88D. It was concluded that equation (7) in general underestimates snow depths by about a factor of 3 or 4, and that the error became much worse for the case study with more snow. Their results suggested that snow intensity in eastern Colorado is a strongly dependent function of Z_e between about 20 and 30 dB*Z*. Examination of figures 1 and 2 would thereby suggest a need for relatively low α and β values in equation (1). Further work was recommended to better define a Z_e-*S* relationship, taking into account ice crystal types, degree of riming, and the *SD*:*S* ratio.

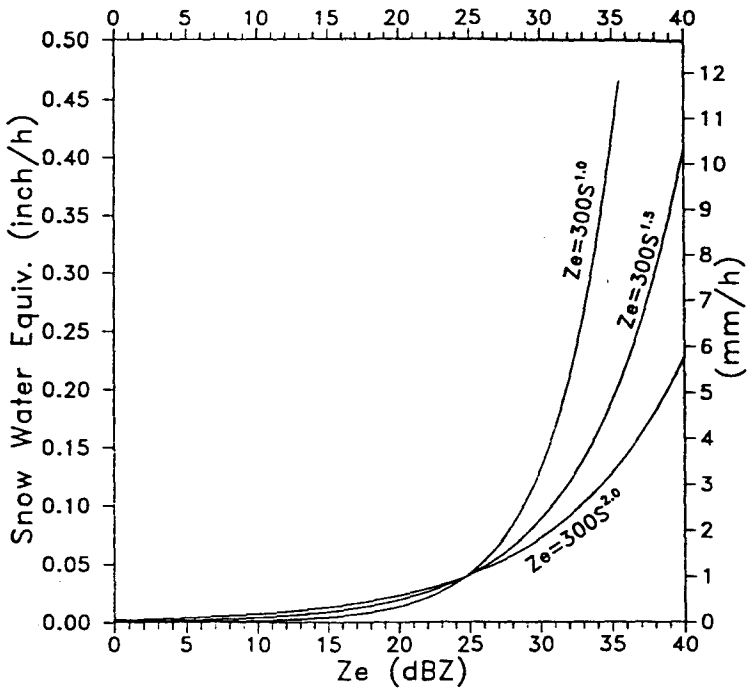

Figure 1. - Plots of three different Z_e-S relationships showing the effect of varying the β coefficient while the α coefficient remains constant.

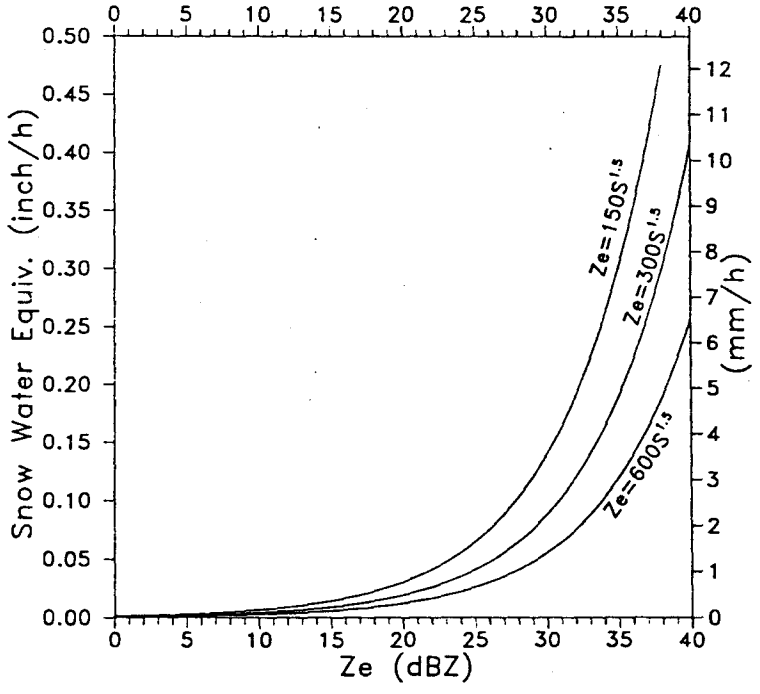

Figure 2. - Plots of three different Z_e-S relationships showing the effect of varying the α coefficient while the β coefficient remains constant.

Fujiyoshi et al. (1990) used 3-minute radar measurements and 1-minute sensitive electrobalance observations of snowfall intensity only 8.7 km from a 3.2-cm radar. They arrived at values of $\alpha = 427$ and $\beta = 1.09$ for the range 0.1 to 3.0 mm h^{-1} (0.004 to 0.118 inch h^{-1}). Unfortunately, in this generally well-done study, the accuracy of snowfall measurements is questionable. The authors note that the snowfall amount was 50 percent lower at the western end of their line of three measurement sites where the wind "blew harder." The three sites were spaced only 100 m apart, and only data from the center site were finally used. Although a windbreak was used to protect the electrobalance, how this windbreak affected snowfall onto the balance in the reported 5- to 6-m s^{-1} winds is not clear. As discussed in section 3.1, wind speeds of that magnitude can cause significant undercatch by shielded gages. The question of snowfall accuracy is not well addressed in numerous studies.

Fujiyoshi et al. (1990) group together values based on both calculations of Z and measurements of Z_e in their figure 7, which shows published α values for snowfall ranging from as low as 50 to almost 4000. Not all the references cited are readily available, but it seems likely that most if not all of the highest values are from studies which calculated Z (e.g., equation (4) is included). But even limiting α values to those based on measured Z_e would result in a large range. Values of β in the same figure range from 0.9 to 2.3, also a considerable range.

Published values of both α and β vary considerably. Although no unique Z-S relationship can be expected, what is not clear is how much of the published variation in coefficients is caused by natural differences in snowfall characteristics and how much is related to experimental shortcomings. Developing the "best" values for α and β for snowfall in any given geographical region is clearly a challenge. And adding to the challenge is the need for large numbers of data pairs for calculation as discussed by Krajewski and Smith (1991) (see their fig. 6 simulating synchronous observations). Ideally, hundreds of pairs of observations would be used to establish Z_e-S relationships, a difficult and expensive undertaking at any single location much less at many geographic regions.

One of the reasons that determining the "best" α and β coefficients for snowfall is difficult can be illustrated by figures 1 and 2. Figure 1 shows the effect of varying the β value between 1.0 and 2.0 while holding the α value constant at 300, all reasonable values according to the literature. The lower the β value, the more rapidly the curve rises with increasing radar returns. The 3 curves barely deviate for values of Z_e lower than the 25-dBZ crossover point where all 3 predict 0.04 inch h^{-1}. As discussed in section 5, most S observations in this study were below 0.04 inch h^{-1}. Only for Z_e values above 30 dBZ do the curves deviate markedly from one another. But this region of relatively high snowfall rates is where data points become infrequent because of the highly skewed nature of precipitation intensities. So the Z_e-S relation is relatively insensitive to changes in β for the large majority of hours with snowfall, a point made by Wilson (1975).

On figure 2, the α value of equation (1) is doubled from 150 to 300 and then doubled again to 600, while β is left constant at 1.5, again reasonable values according to the literature. Lower α values cause the curves to rise more rapidly as Z_e increases. So a rapid rise in predicted snowfall with increasing Z_e can be achieved with either a decrease in α or β or both.

In the case of figure 2, the curves begin to deviate from one another above 20 dBZ, so the chance of determining the "best" α value may be better than the chance of determining the "best" β value with typical snowfalls.

8. Optimization Scheme for Determination of α and β Coefficients

An optimization scheme was developed for determining the best α and β coefficients for equation (1), repeated below, for any given data set; that is, from the Z_e values in an array of range bins over a particular gage and simultaneous hourly S observations by that gage.

$$Z_e = \alpha S^{\beta} \tag{1}$$

This scheme is based on the work of Smith et al. (1975) and personal communication with Dr. Paul Smith. The approach is discussed in more detail in appendix A by Dr. James Heimbach, who programmed the scheme for use with the Cleveland and Denver data sets.

The average estimated snowfall accumulation for each vertical (or upwind) array of 9 or more range bins was repeatedly calculated as described in section 6. At the suggestion of Dr. Paul Smith, for any specified value of β in equation (1), a unique value of α was established by forcing the average radar-derived snowfall accumulation to be equal to the average gage hourly snowfall accumulation for each data set (e.g., all pairs of hourly accumulations observed by a particular gage and radar estimates over that gage). This constraint simplified the optimization process by limiting the solution to one of a set of β values, each with only one α. A more complex scheme might vary both α and β. Moreover, as pointed out by Smith et al. (1975), the optimization scheme is not restricted to equation (1) because higher order term equations could be tested. But the work herein has been limited to equation (1) with the constraint of using only a unique α value for each β value.

It is again emphasized that radar estimates of S, *not* Z_e values, were calculated for each range bin using equation (1), summed, and averaged in each iteration of this optimization process. This approach avoids the biases resulting from averaging Z_e over time and space.

Calculations were made for any input range of β and β increment. A CTF (criterion function) was used to judge the fit of each pair of α and β. Smith et al. (1975) discuss the difficulties of selecting the most appropriate CTF. The CTF used here is the simple sum of absolute differences between the gage measurement of precipitation and that estimated by the radar, again as suggested by Dr. Paul Smith. The combination of α and β accepted for any optimization run was that which yielded the smallest CTF value.

Operator-input values to this scheme include:

- The particular gage to be tested.
- The range of β and its increment. Unless otherwise stated, the range was from 0.7 to 3.0 in steps of 0.05.
- The minimum number of volume scans required for 1 hour of data to be accepted, always set at 4.
- The minimum number of range bins needed, always set at 5. But in fact, no missing range bins have been noted so this requirement appears unnecessary.

Range bins commonly had no detectable signal during the beginning and ending portions of storms and/or over distant gages. The estimated snowfall rates for all such bins were set to

zero and they were used in calculating average values for the vertical and upwind arrays. The radar did scan the volumes of space in question. With the sensitivity of the WSR-88D, non-detection of any return is equivalent to trivial snowfall rates within a 100-km range for any reasonable combination of α and β values in equation (1). Although the current software cannot handle groups of gages, a future version will be modified to do so. For example, gages at similar ranges could be grouped

9. Application of Optimization Scheme to Cleveland and Denver Data

The optimization scheme just discussed was applied to the Cleveland and Denver data sets presently available. As noted in section 5, the data sets consist of 2 wet months from the Cleveland area and 3 dry months from the Denver area.

9.1 Cleveland Observations

The Belfort gage nearest the Cleveland radar was located at the 36-km range and the 64° azimuth (table 5), just beyond the ground clutter region according to the clutter bypass map used all winter. A total of 143 hours with detectable hourly snowfall (at least 0.005 inch) was available with simultaneous radar observations. The optimization scheme produced its lowest CTF with an α value of 318 and a β value of 1.5. These values are similar to the default α = 300 and β = 1.4 used in the WSR-88D rainfall algorithm.

A plot of the resulting hourly data is given on figure 3 with a 1:1 (solid) line and linear least squares regression (dashed) line. The two lines almost overlap, indicating the optimization scheme produces radar-estimated hourly snowfall accumulations that are well matched with hourly gage accumulations. Inspection of figure 3 shows the high frequency of hours with only 0.005 or 0.010 inch of snowfall as previously discussed. Radar-estimated snowfall accumulations ranged from just above zero to about 0.03 inch h^{-1} with gage observations of 0.005 or 0.010 inches. This result suggests the radar would not seriously overpredict in the Cleveland area during the many hours with very light snowfall using the relationship:

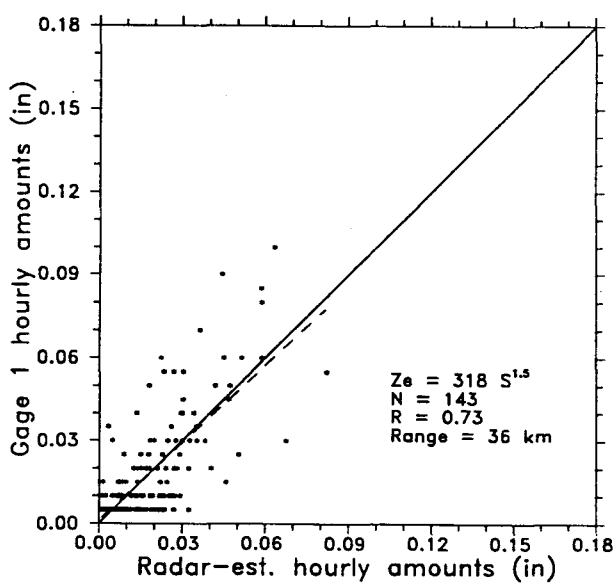

Figure 3. - Scatter plot of 143 pairs of hourly-observed snowfall for Cleveland gage No. 1 versus radar-estimated snowfall directly overhead using the relation $Z_e = 318\ S^{1.5}$. The regression line is dashed, and the 1:1 line is solid.

$$Z_e = 318\ S^{1.5} \tag{8}$$

The best estimates of α and β should be from gage No. 1, nearest the radar. However, the optimization scheme was applied to all 5 gages east-northeast of Cleveland as summarized in table 8. Applying the optimization scheme at more distant ranges provides some insight into expected underestimation of S by the radar as range increases and the beam becomes broader and higher above the terrain. At some far range, the radar beam will be above snow-producing clouds, and returns will be negligible even when heavy snowfall is occurring below the beam.

Both the "optimum" values of α and β change with range in table 8. The optimization scheme caused α to decrease and β to increase with range as the calculations attempted to match the observations with the constraint that the average radar estimate must equal the average S value at each gage. Eventually, absurd values resulted at gage No. 5, 146 km from the radar.

Table 8. - Summary of results of applying the optimization scheme to the five gages located east-northeast of Cleveland. Only hours with gage amounts of 0.005 or more inches are included.

Gage No.	Distance from Radar (km)	Beam Height* (m)	Hours of Observation	α	β	R	Standard Error Estimate (in h^{-1})	Average S (in h^{-1})
1	36	390	143	318	1.50	0.73	0.0136	0.0196
2	61	750	235	248	1.45	0.81	0.0139	0.0204
3	87	1205	187	147	1.85	0.67	0.0226	0.0260
4	115	1780	202	48	2.20	0.72	0.0179	0.0232
5	146	2530	218	8	8.50	0.50	0.0169	0.0196

* Height of 0.5° tilt beam center above gage assuming standard refraction.

The average hourly S values were consistent along the line of gages, ranging only between 0.0196 and 0.0260 inch. Table 7 showed that the median was near 0.01 inch h^{-1} at each gage, and average accumulations varied only between 0.019 and 0.025 inch h^{-1}. The Cleveland line of gages appears to have produced a reasonably similar data set, providing the opportunity to examine range effects on radar estimates.

A linear least-squares regression line (hereafter regression line), linear correlation coefficient, R, and standard error of estimate were calculated for each gage using the α and β values of table 8 and the same data sets to which the optimization scheme was applied. The standard error of estimate measures the scatter in the Y-direction of the observed points about the regression line and is analogous to the standard deviation. For a large normal distribution, 68 percent (95 percent) of the population would fall within plus or minus 1 (2) standard error of estimate of the regression line. Limited variation of the standard error appears to occur with range although the two gages closest to the radar have the smallest standard errors.

The R values in table 8 are encouraging. In spite of generally low precipitation accumulations typical of snowfall, the correlations between the hourly gage observations and radar estimates are generally at least as high as the 0.68 reported by Smith et al. (1975) for hourly rainfall on the northern Great Plains. The notable exception is the most distant gage at the 146-km range. No time delay to allow for snowflakes to settle from the illuminating radar beam to the surface has yet been applied to these data. Neither has the advection

scheme discussed in section 11 been applied. The relatively high R values also suggest that gage measurements were reasonably accurate. More variability might be expected with unshielded gages in windy locations.

It is reasonable to assume that equation (8), derived from the gage nearest the radar, offers the best available Z_e-S relation for the Cleveland area. In order to provide additional insight into range effects, figures 4 through 7 were prepared by applying equation (8) to each gage's data and the vertical array radar data. (The radar array was 3 by 3 range bins for each of these gage locations.) A regression line was also calculated for reference as shown by the dashed lines. Gage values that appeared to be outliers on these and all similar plots, and all hourly totals of 0.10 inch h^{-1} or more, were verified by checking yet again against the original charts.

Table 9 summarizes the results of the various calculations.

Table 9. - Summary of applying equation (8) to the data set from each Cleveland area gage. Also noted are the intercept (A) and slope (B) of regression lines (gage = A + B x radar), correlation coefficients (R), standard error of estimate, average hourly gage-observed and radar-estimated snowfall accumulations, and the gage/radar estimate ratios. The number of pairs is the same as given in table 8.

Gage No.	Distance from Radar (km)	A (in)	B	R	Standard Error Estimate (in)	Gage Observation (in)	Radar Estimation (in)	Gage/ Radar
1	36	0.001	0.985	0.73	0.0136	0.0196	0.0196	1.00
2	61	0.000	1.187	0.80	0.0140	0.0204	0.0173	1.18
3	87	0.004	1.400	0.68	0.0224	0.0260	0.0160	1.63
4	115	0.009	1.856	0.72	0.0177	0.0232	0.0074	3.14
5	146	0.013	1.607	0.55	0.0162	0.0196	0.0043	4.56

Reference to figures 4 through 7 and table 9 shows a number of interesting features. Most important, the gage/radar ratio increases with range as anticipated. The ratios suggest that radar estimates are 85 percent (1/1.18) of gage average snowfall by a 61-km range and 61 percent by 87 km. By a 115-to 146-km range, the radar predictions have fallen to only 32 to 22 percent of the gage amounts. As shown in table 8, the center height of the 0.5° tilt beam increases from about 390 m above gage No. 1 to over 2500 m above gage No. 5, so underestimates with increasing range should be expected.

Although radar underestimation is serious beyond 60 km, these results are not as drastic as those reported by Wilson (1975) for a nearby region of the country. Gage:radar ratios were estimated from the curve on his figure 2 for the ranges of Cleveland gages No. 1 to 4. These ratios, divided by the 36-km ratio, yielded results compatible with table 9 (all of Wilson's ratios exceeded 3.0 because of use of an inappropriate Z-S relation). Resulting values for ranges of 61, 87, and 115 km are 1.7, 3.7, and 18.0, much higher than the ratios of table 9. At least three factors probably improve the results given here. The WSR-88D radar beam is narrower (0.95 versus 1.7°) and tilted nearer the ground (0.5 versus 0.9°) than the radar used by Wilson. In addition, the WSR-88D is more sensitive than radars available at the time of Wilson's study.

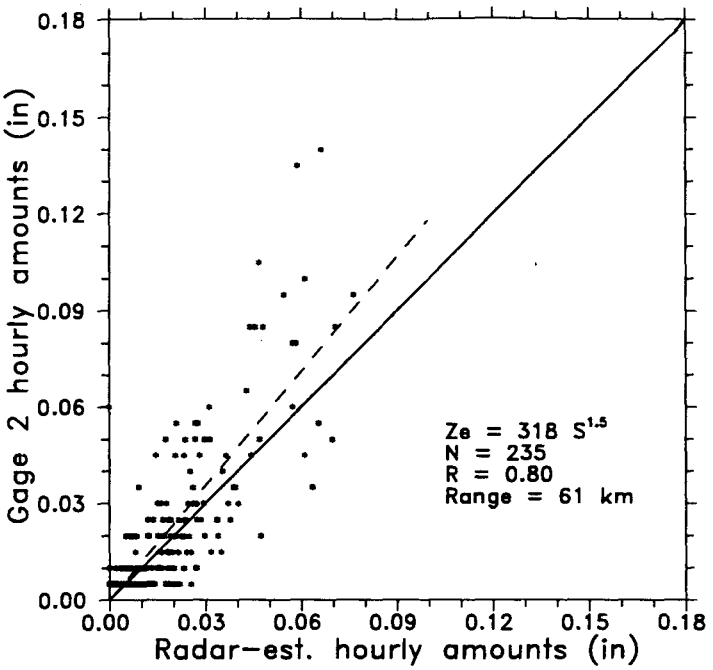

Figure 4. - Similar to figure 3 except for 235 pairs for Cleveland gage No. 2.

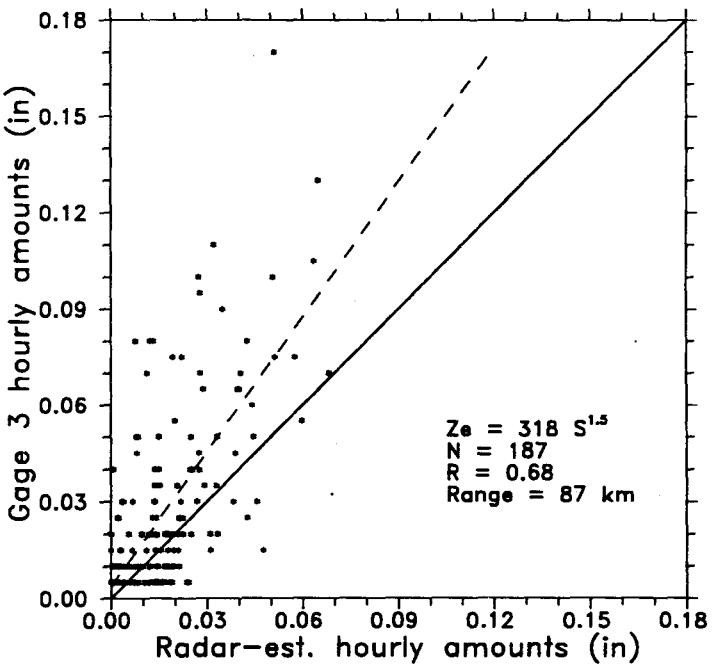

Figure 5. - Similar to figure 3 except for 187 pairs for Cleveland gage No. 3.

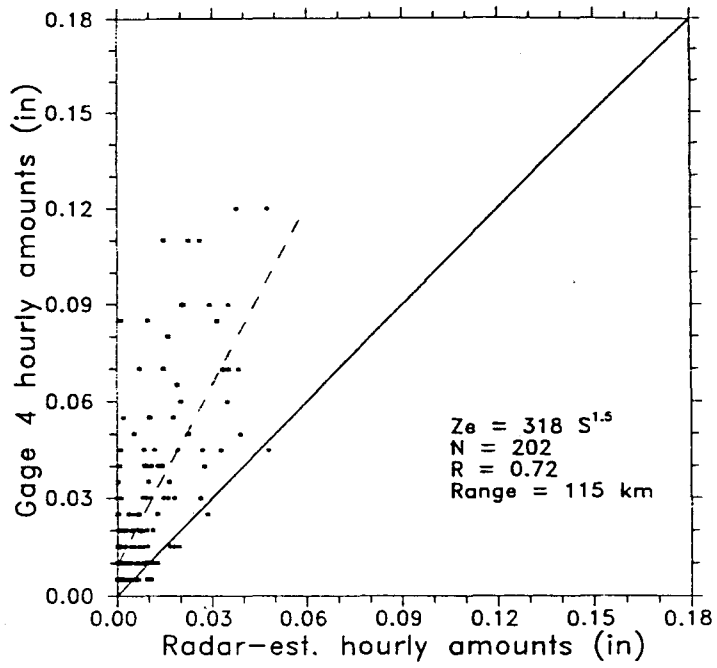

Figure 6. - Similar to figure 3 except for 202 pairs for Cleveland gage No. 4.

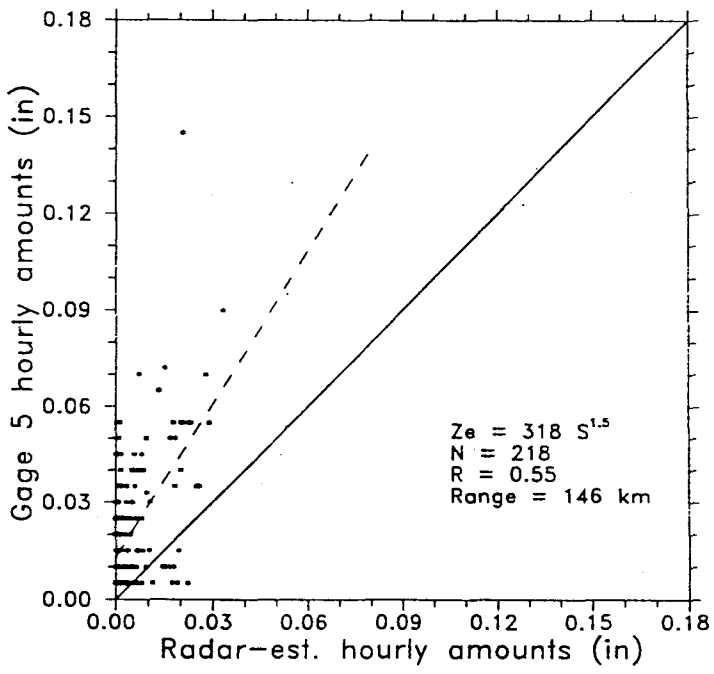

Figure 7. - Similar to figure 3 except for 218 pairs for Cleveland gage No. 5.

Although use of the WSR-88D with a 0.5° tilted narrow beam has reduced the range effect reported by Wilson (1975), further improvement is still needed, especially beyond the 60-km range. Extrapolation of radar measurements from beam altitudes to ground level using the vertical profile of Z_e appears to be a viable approach for significantly improving radar-estimated precipitation at moderate-to-far range (Joss and Waldvogel, 1990; Joss and Lee, 1995). A vertical profile extrapolation scheme will be developed for a future version of the snow algorithm if resources permit.

The R values of tables 8 and 9 are similar even though the optimization scheme was separately applied to each gage in table 8, and a single Z_e-S relation was used for table 9. This finding is comparable to the results of Smith et al. (1975) who noted that their similar optimization scheme, "does not noticeably improve the correlation between the radar and gage estimates of the hourly point rainfall amounts." Although the basic scatter of the data appears to remain the same no matter what Z_e-S relation is applied (within reason), the optimization scheme provides an objective and effective means for estimating the "best fit" α and β values to quantify radar-estimated snowfall.

The table 9 R values are similar for gages No. 1 to 5 and the same for gage No. 1 at the 36-km range and gage No. 4 at the 115-km range. The standard error of estimate values are also very similar to those of table 8. These results suggest that the basic radar "signal" is not significantly diminished to at least a 115-km range. This finding provides hope that a method can be developed to improve WSR-88D estimates of snowfall at ranges to and beyond 100 km. However, a note of caution is in order. The data set used herein combined *all* storm types for the 2-month period, including at least two major synoptic storms which produced the heavier snowfalls. Results may not be as encouraging when lake effect storms are considered separately in the future because such storms tend to have shallow cloud systems and light to moderate snowfall accumulations at the Cleveland gage sites.

The optimization scheme was applied to upwind arrays of range bins (sec. 4.2) used with gage No. 1 observations. The resulting α value was 354, and the β value was 1.25. However, these results likely were contaminated by ground clutter in the operator-designated box with zero suppression imposed around gage No. 1, the only Cleveland gage with nearby clutter. Whenever the advection scheme "moved" the upwind array a few kilometers west of the gage, a region of known clutter likely increased the Z_e values in some range bins. Software to exclude this cluttered area has not yet been developed. However, a partial test of the advection scheme was applied as follows. Equation (8) was applied to the upwind arrays, and correlation coefficients were calculated for the resulting pairs of radar-estimated snowfall and gage-observed snowfall. Only gage No. 2 had a slight improvement in R value (0.82) over table 9. The lesser value for gage No. 1 (0.68) might be explained by clutter contamination. Gage No. 3's R value was almost unchanged (0.65), but the gage No. 4 value of 0.56 and gage No. 5 value of 0.10 are well below the R values of table 9. These preliminary results do not indicate any improvement at near and moderate ranges with application of the advection scheme, and indicate a definite degrading of predictability at far ranges. This result suggests that the current advection scheme, using the VAD wind profile and a fixed 1.0-m s^{-1} fall speed for all crystals, offers no advantage over simply assuming that snowflakes fall vertically and using Z_e measurements directly over the gages.

One common feature of figures 3 through 7 is the lack of a linear relationship between radar-estimated and gage-observed snowfall. Of course, the predominance of data points would be expected to lie above the 1:1 line for gages No. 3 through 5 because of the range effect.

However, for a linear relationship with a reasonably high R value, an approximately equal number of points would be expected on either side of the *regression line* for any given range of snowfall accumulations. But the large majority of radar-estimated data points lie below the regression lines for figures 3 through 7 for light snowfalls (radar tends to overestimate). The converse is true for the higher snowfalls where the radar tends to underestimate. One possible explanation for the radar underestimating higher snowfall amounts and overestimating lower amounts is related to spatial averaging. The radar's minimum spatial unit is the range bin, 1° by 1 km in size. Using Cleveland's gage No. 1 as an example, the single range bin directly over the gage at a 36-km range has an area of 6.3×10^5 m^2. But arrays of range bins approximating 3 by 3 km of area were used in these analyses. For the same gage, a 5° by 3-km array (15 bins) was used, which has a total area of 9.4×10^6 m^2. In contrast, the area of an 8-inch-diameter Belfort gage orifice is 0.03 m^2, more than 7 orders of magnitude smaller than the single range bin in this example.

Clearly, the Belfort gage measures *point* values of snowfall accumulations which have highly skewed temporal distributions. For example, the frequency distribution for Cleveland gage No. 1 hourly totals ranged from 49 h (34 percent) with 0.005 inch, the minimum detectable amount, to 1 h with the a maximum observed total of 0.100 inch. The number of hours with amounts between 0.005 and 0.050 inch was 130 (91 percent). Only 13 hours (9 percent) had amounts between 0.055 and 0.100 inch. But the real question is the degree to which spatial gradients in the snowfall pattern cause these observed skewed temporal distributions. It is well known that convective rainfall has strong spatial gradients, but snowfall patterns would be expected to be much more homogeneous. But dense snow gaging networks are rare, so not much can be said about snow gradients at resolutions smaller than the radar range bin.

A test was performed with the existing data by comparing snowfall estimates from the single range bins that make up the averaging array with the array's estimate. This test was done for the array of 15 range bins over Cleveland's gage No. 1, used for the radar-estimated hourly amounts of figure 3.

For reference, table 8 shows that the R value between the hourly array estimates and gage No. 1 observations was 0.73, the average radar-estimated snowfall amount was 0.0196 inch, and the standard error of estimate was 0.0136 inch. These values can be compared with the ranges resulting from performing similar calculations with each of the 15 range bins in the array which follow: R-values ranged between 0.69 and 0.73, average radar-estimated snowfall amounts ranged between 0.0190 and 0.0212 inch, and standard errors of estimate ranged between 0.0135 and 0.0144 inch. Even within these limited ranges, the larger R values and smaller standard errors of estimate tended to be located near the array center. These comparisons suggest limited spatial variation over the area of the array. However, although any of the bins within the array would provide reasonable estimates of snowfall, those nearest the array center had the highest association with gage No. 1 observations.

A plot was prepared (not shown) similar to figure 3 but using the Z_e observations from the single range bin directly over gage No. 1; that is, the bin in the center of the array. The plot appeared almost identical to figure 3, which might be expected with an R value of 0.99 between this range bin's hourly amounts and the array amounts. (The R values between the array hourly estimates and those from single bins ranged between 0.96 and 0.99).

The above discussion suggests that averaging over an array of range bins may be unnecessary, presuming the data from and over Cleveland's gage No. 1 are representative.

Using the single bin directly above each gage may provide essentially the same results as averaging over approximately 3- by 3-km range bin arrays. Furthermore, use of a single range bin will not eliminate the problem of radar overestimation of small snowfall amounts and underestimation of larger amounts. If that estimation problem is related to spatial variations, the variations are on space scales smaller than a single range bin.

Future analysis will give further consideration to the radar overestimation/underestimation problem. Perhaps a more complex relationship than equation (1) would be useful, or some improvement to the optimization scheme of section 8 may be in order.

9.2 Denver Observations

The optimization scheme was applied to 3 months of observations by the Denver WSR-88D and gages No. 1 to 5 of table 3 in the same manner as done for the Cleveland area measurements. (Gage No. 6 provided no useful data during this period.) As previously noted, the Denver area experienced a dry winter, and no major storms occurred during the period being considered.

Table 10. - Summary of results of applying the optimization scheme to five gages in the Denver area. Only hours with gage amounts of 0.005 or more inches are included.

Gage No.	Distance from Radar (km)	Beam Height* (m)	Hours of Observation	α	β	R	Standard Error Estimate (in)	Average S (in h^{-1})
1	25	205	65	152	1.20	0.82	0.0119	0.0182
2	24	320	48	148	1.70	0.71	0.0170	0.0238
3	49	575	83	164	1.90	0.78	0.0120	0.0205
4	81	700	34	134	2.50	0.63	0.0113	0.0150
5	91	-275	108	1	2.20	0.89	0.0133	0.0251

* Height of 0.5° tilt beam center above gage assuming standard refraction.

Table 10 presents the results of applying the optimization scheme plus the correlation coefficients resulting from using the noted α and β values with equation (1) to calculate radar-estimated snowfall accumulations. The α values for gages No. 1 to 3 are tightly grouped between 148 and 164. The similar value for gage No. 4 may be a fluke, being based on only 34 pairs. The absurdly low α value for gage No. 5 is discussed below.

The relatively high R values of table 10 are similar to those within 115 km of the Cleveland radar (table 8) with the possible exception of gage No. 4. These R values again suggest good gage accuracy because more variability would otherwise be expected. In the absence of forest cover, gages No. 1 through 3 were located in backyards of established neighborhoods with wind protection primarily provided by wooden fences and houses. This approach apparently produced decent snowfall measurements.

It is encouraging that similar α values resulted from three independent runs of the optimization scheme using the 3 gages within 49 km of the radar. This result is in spite of the limited hours with snowfall detected by each gage. Although the β values vary from 1.2 to 1.9, this range of exponent has limited influence for S below 0.10 inch h^{-1} as shown on figure 1. Only 1 percent of the combined 196 hours with snowfall at gages No. 1 through 3

exceeded this value. The suggestion of table 10 is that the α coefficient for the Denver area should be about half that of the Cleveland value of 318, but the exponents should be similar, near 1.5. Of course, this preliminary conclusion is based on limited data without heavy snowfalls.

The optimization scheme software cannot presently handle input data from more than one gage with its corresponding radar observations. However, it was decided to apply the scheme to gages No. 1 and 2 at 24- and 25-km ranges using their average α and β values from table 10; that is, 150 and 1.45, respectively. The resulting radar estimates and gage observations were then grouped together to increase the population to 113 pairs. The resulting R value was a respectable 0.76 and the regression line essentially matched the 1:1 line. The average $\alpha = 150$ and $\beta = 1.45$ were used with equation (1) to estimate snowfall with the vertical array over gage No. 3, which resulted in an average radar-estimated snowfall of 0.0211 inch h^{-1}, slightly above the value of table 10. The corresponding R value was 0.79. These results indicate no significant decrease in radar estimation to at least a 49-km range; that is, no range effect.

With the absence of any discernible range effect out to at least gage No. 3, the average α and β values for gages No. 1 to 3 of table 10 were used in equation (9) to estimate snowfall accumulations from the vertical arrays over gages No. 1 to 3.

$$Z_e = 155\ S^{1.6} \tag{9}$$

The resulting 196 data pairs are plotted on figure 8 along with the regression line, which almost matches the 1:1 line. The correlation coefficient is 0.76. The average radar-estimated and gage-observed hourly snowfall proved to be identical at 0.0205 inch. These results suggest that equation (9) is a good first approximation to use with winter storms in the Denver area. But as with the similar Cleveland area figures, a marked tendency is seen on figure 8 for the radar estimates to be too high for low snowfalls and too low for higher observed accumulations.

Little can be said about gage No. 4 observations with only 34 hours of detectable snowfall at a low average value of 0.015 inch. This broad mountain valley site was chosen because it was surrounded by mountains but had no ground clutter for several kilometers around it. This location seemed to offer a good opportunity to evaluate mountain valley sites where people tend to live. However, it has become apparent that the location is in a distinct "rain (and snow) shadow" because of downslope motion from higher terrain.

Gage No. 5 was deliberately chosen as a mountain site in the midst of a very cluttered area according to the clutter bypass map used by the Denver WSR-88D all winter. Its data can be used to evaluate whether the Denver radar might be used to estimate snowfall over the Rocky Mountains. Examination of PPI radar displays from Denver generally reveals a large area with very little reflectivity from low tilts over the Rocky Mountains because of beam blockage and clutter suppression. The altitude of gage No. 5 is above the center of the 0.5° beam (for standard refraction), the only reflectivity data so far examined, and little useful information was expected from this lowest tilt beam. Future plans include examination of radar returns from higher tilts.

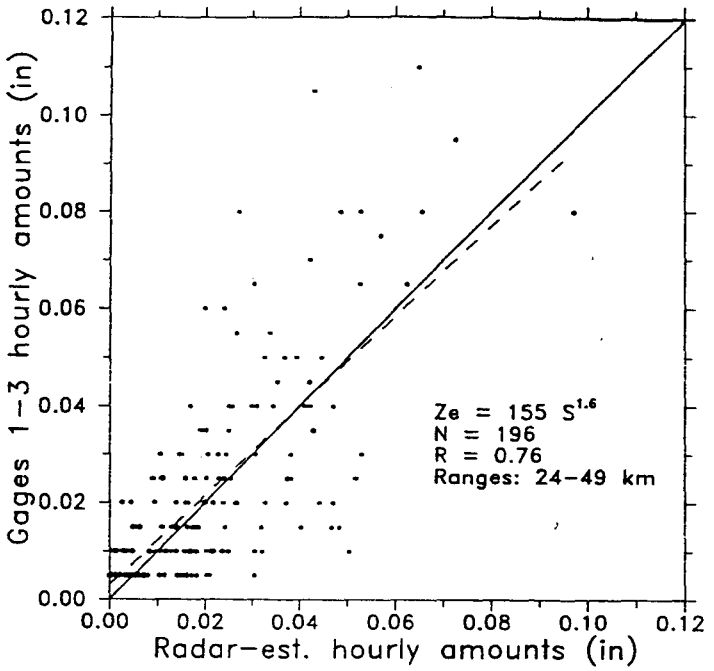

Figure 8. - Scatter plot of 196 pairs of hourly-observed snowfall for Denver gages No. 1 to 3 versus radar-estimated snowfall directly overhead using the relation $Z_e = 155\ S^{1.6}$. The regression line is dashed, and the 1:1 line is solid.

The R value for gage No. 5 is the highest in table 10 and the α value is extremely low at 1.0. Table 10 shows that the center of the 0.5° tilted beam was below the elevation of gage No. 5, so ground clutter and beam blockage would certainly be expected near this gage. The 9 range bins in the vertical array over gage No. 5 were briefly inspected for many volume scans. Range bins along about the 260 and 261° radials for ranges of 90 to 92 km rarely had any detectable returns. Such bins were assumed to represent zero snowfall in the processing software. However, the 3 bins along the 262° radial frequently had returns, albeit with relatively low Z_e values. The RDRHGT program discussed in section 3.3 shows almost total blockage of the 0.5° beam near gage No. 5 for the 260 and 261° radials, and a large fraction of the beam is unblocked for the 262° radial.

Because the snowfall rate was calculated for each bin in an array and then averaged, the usual inclusion of about 6 zero values over gage No. 5 resulted in very low snowfall estimates. The optimization scheme, in forcing radar estimates and gage observations to have the same average values, produced the very low α value and relatively high β value. A plot of the resulting radar estimates and gage measurements (not shown) shows limited scatter, in agreement with the R value of 0.89. But when the reasonable α and β values of equation (9) were used, most radar-estimated hourly snowfalls became trivial.

The 3 range bins along the 262° radial over gage No. 5 contained enough signal to give rise to some optimism. These results suggest that radar returns that are not totally blocked or suppressed may be used over mountainous terrain to provide some estimates of snowfall intensity. Further work is warranted on the topic of radar estimation of snowfall on mountainous terrain, including consideration of the 1.5° beam.

Upwind arrays were also used with gages No. 1 to 5 in the optimization scheme. Some R values increased slightly, but no more than 0.02 over those values in table 10. These results suggest the current advection scheme offers no significant increase in predictive value in the Denver area over simply using the range bins directly above the location of interest.

Finally, equation (9) was applied to all five gages with the result that the R values were basically unchanged from table 10, similar to the Cleveland findings, with the exception of gage No. 5 which had its R value reduced from 0.89 to 0.82. Moreover, the radar-estimated snowfall accumulations were reasonable except at gage No. 5. The average radar-estimated hourly snowfall accumulation over gage No. 5 was 0.002 inch, and the gage-observed value was 0.025 inch h^{-1}. So, with the exception of the blocked and cluttered mountainous regions, use of equation (9) appears justified in the Denver area, at least until testing with additional data is possible.

9.3 Significance of Different Z_e-S Relations for Cleveland and Denver

The recommended initial α and β values are 318 and 1.5 for Cleveland and 155 and 1.6 for Denver. These values are very similar to the top two curves of figure 2.

It is reasonable to ask whether these curves produce sufficiently different snowfall accumulations to make a practical difference. One way to address that question is to apply the Cleveland relation of equation (8) to Denver data and the Denver relation of equation (9) to Cleveland data. This application produced the results shown on figures 9 and 10, respectively.

Examination of figure 9 reveals Cleveland equation (8) significantly underpredicts Denver hourly snowfall accumulations. The dashed regression line is well above the 1:1 line, and the average radar-estimated snowfall accumulation is 0.0126 inch, 61 percent of the gage-observed 0.0205 inch. Conversely, figure 10 shows use of Denver equation (9) with Cleveland radar measurements seriously overpredicts snowfall accumulations as shown by the regression line well below the 1:1 line. The average radar-estimated snowfall accumulation in this case is 0.0289 inch, 144 percent of the gage-observed average value of 0.0201 inch. Use of a Z_e-S relationship appropriate to the geographical region clearly is important for quantitative snowfall estimation.

It is not known why the different Z_e-S relations resulted from the Cleveland and Denver area data. The difference may be related to microphysical differences between the lake effect storms that predominated over the Cleveland gages and the upslope storms that affected the Denver area. Ice crystal observations were routinely made near Denver but not near Cleveland. However, microphysical observations of lake effect storms have been published which may be compared with the past winter's Denver observations in a future report.

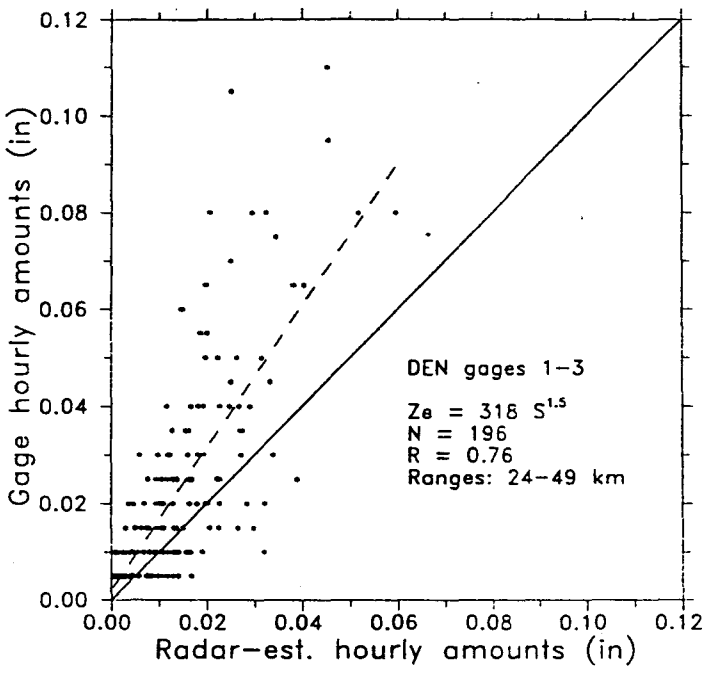

Figure 9. - Similar to figure 8 except the Cleveland Z_e-S relationship of figure 3 is deliberately misapplied to Denver gages No. 1 to 3.

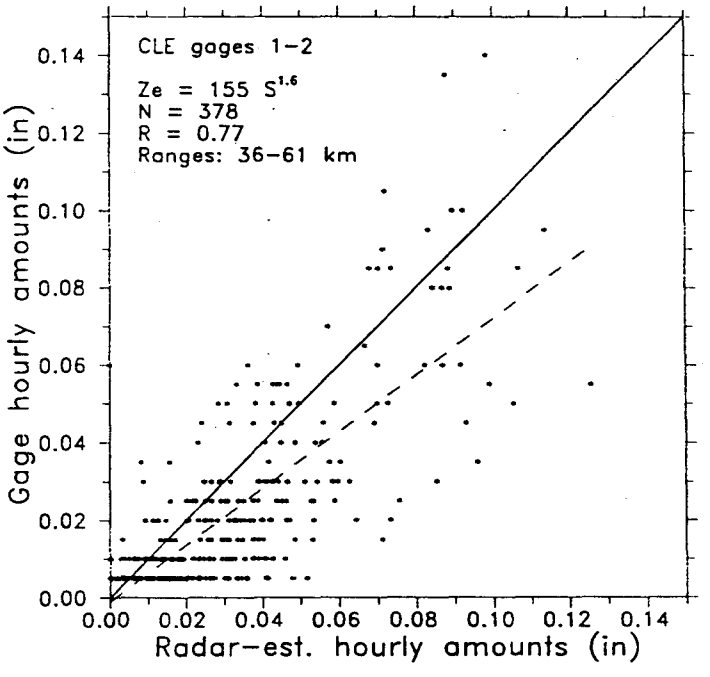

Figure 10. - Similar to figure 9 except that the Denver Z_e-S relationship of figure 8 is deliberately misapplied to Cleveland gages No. 1 and 2.

10. Development of a Snow Accumulation Algorithm for the NEXRAD Radar System, First Version: June 1996

10.1 Introduction

The MOU specifies several milestones for the delivery of results in the development of the Snow Accumulation Algorithm (hereafter Snow Algorithm) for the NEXRAD radar system. The following sections and appendices describe the Snow Algorithm itself and not the associated data gathering activities (secs. 2 to 6), or their results in specifying a relationship between effective reflectivity factor (Z_e) and snow water equivalent (S) on the ground (secs. 7 to 9). The appendixes give specific details for the various specialists who will be involved in the further programming or use of the Snow Algorithm.

- Appendix B contains appendix B of the MOU: Algorithm Documentation Requirements. It specifies a document containing nine sections:
    1. Source.
    2. Bibliography
    3. Functional Description
    4. Functional Comparison
    5. Statistical Performance
    6. Operational Performance
    7. Adaptable Parameters
    8. Strengths and Limitations
    9. Future Enhancements

  However, this report has a different structure. The rest of appendix B identifies the sections of this report that fulfill those requirements.

- Appendix C is the same as appendix C of the MOU: Criteria for Acceptance of Deliverables. It specifies numerous characteristics to be included in the Snow Algorithm.

- Appendix D summarizes the tasks specified in the SOW given in the MOU and gives the delivery timetable and preliminary results.

10.2 Relationship to Other OSF Software Programs

This Snow Algorithm is partly based on procedures already in use with the NEXRAD system, specifically the PPS (precipitation processing subsystem), intended for rain, and Doppler winds (VAD) portions. Where appropriate, the same PPS or VAD subroutine names have been used where they are identical or accomplish the same thing. Likewise, a few of the variables are the same. The program structure differs in that it does not use the same shared memory as the PPS and VAD routines because of the complicated addresses in those routines and the reduced program memory space in the development versions. The enclosed simpler program structure is expected to be rewritten into the NEXRAD software style.

The programs use reflectivity (Z_e) data from Level II tapes consisting of dBZ, offset, multiplied by 2, and offset again to yield integer numbers called "biased dBZ" in the PPS software ranging from 0 to 255; 0 indicates that Z_e is below the detection level for the WSR-88D. Velocity data are also rescaled to a range of 0 to 255. Resulting numbers are therefore not "biased" in the normal dictionary sense but are rescaled into the byte range of numbers.

10.2.1 The PPS Algorithm

The PPS algorithm is described in section 3.3 of the Federal Meteorological Handbook No. 11 (1991), Doppler Radar Meteorological Observations, part C, WSR-88D Products and Algorithms. The algorithm produces maps of accumulated precipitation (intended for rain, not snow) for 1-hour, 3-hour, and storm totals. Five basic steps are listed: 1) preprocessing, 2) rate, 3) accumulation, 4) adjustment, and 5) products.

1) Preprocessing:

The preprocessing algorithm prepares Z_e data for input into the rate algorithm. Quality control steps include: blockage correction, isolated bin check, outlier check, vertical echo continuity check, and bi-scan maximization. The beam blockage correction adds 1, 2, 3, or 4 dBZ (or 2, 4, 6, or 8 to the biased dBZ) to radial data beyond ranges at which the beam is estimated to be blocked by particular percentages. The occultation file has azimuth resolution of 0.2° and is calculated from terrain data specific to each radar antenna site. Isolated bin values of reflectivity are set to a code for "less than minimum detectable signal." Bins with unreasonably large reflectivities are set to a near-minimum level or else to the average of the eight adjacent bins. Bins that are completely occulted (>60-percent blockage) are sometimes replaced by the values of adjacent bins having less blockage. An attempt is made to remove spurious echoes in the lowest tilt ("tilt test") if such echoes tend to disappear in the second tilt. A hybrid-sector file then specifies which tilt to use for precipitation calculations. At far range, it is one of the two lowest tilts giving the greatest echo (bi-scan maximization). The reflectivities are then ready for those further calculations.

The Snow Algorithm, presented here, uses a new hybrid-sector file, developed by Tim O'Bannon of the OSF, which attempts to keep the bottom of the beam closer to the terrain than previous versions while still maintaining 500 feet vertical separation. It does not yet attempt to use adjacent bins for substitution for beam blockages. It does not use the tilt test because snow may occur from shallow clouds that are observed only in the lowest tilt at far range. Though the preprocessing software coding in this version of the Snow Algorithm has differences in style from that in the PPS version, most of the existing PPS routines are acceptable for snow estimation.

Five significant differences are needed for work with snow:

(a) All radials with data, including overlaps, are stored in an array along with their precise azimuth angles. (Each tilt of a volume scan typically has 367 radials of data, including the overlap of the start and end of each tilt. The Snow Algorithm allows up to 370.) *No* radials are averaged together when their integerized azimuth angles are the same. Such averaging requires the conversion of reflectivities to precipitation rates before averaging. The integerization of azimuths may leave some angles with no data even though no break occurred in the scan. Instead, a "nearest neighbor" routine is used later when data for a particular azimuth angle are needed. That routine avoids both averaging and gaps.

(b) Reflectivity values are kept in units of biased dBZ rather than precipitation rate. The conversion back and forth between units adversely quantizes reflectivities at the low precipitation rates typical of snow. Averaging of radials is no longer needed in the Snow Algorithm preprocessing. The only other PPS preprocessing averaging is to

replace outliers by the simple average of its eight neighbors. The Snow Algorithm does the conversion back and forth only for those neighbors in order to replace only the outlier itself; as in the PPS version, the neighbors are not affected.

(c) The new hybrid scan file is used to keep the bottom of the beam close to, but at least 500 feet above, the terrain.

(d) The tilt test is not done because snow may be falling from shallow clouds illuminated only by the 0.5° tilt beam at far ranges.

(e) The biscan maximization for the two lowest tilts is not done. Instead, the lowest acceptable tilt is used as a partial protection against virga.

2) Rate:

The PPS uses only a *Z-R* relation for rain. The constants for the relation are stored in an adaptable parameters file. This Snow Algorithm has found different constants for snow that depend on geographic (climate) region. The observed range dependency for snow has not yet been addressed.

3) Accumulation:

The PPS sums rainfall for 1° by 2-km sample volumes and also for 1/4 LFM (limited fine mesh) grid boxes. Accumulations are made by 1-hour, 3-hour, and storm total intervals. This version of the Snow Algorithm sums snowfall in polar coordinate arrays (230 km x 360°) at the basic resolution of 1° by 1 km. They may be resampled into any other grid style. The accumulation grids presently used by the PPS are entirely acceptable for snow. The three time periods for accumulations are the same as in the PPS version. The Snow Algorithm does not yet contain the continuity and quality checks done by the PPS version, but no changes are anticipated.

4) Adjustment:

The PPS algorithm contains coding to compare radar-derived accumulations with those from precipitation gages. No such adjustments are yet included in the Snow Algorithm.

5) Products:

The PPS algorithm produces 2- by 2-km grids of rainfall totals at 1-hour, 3-hour, and storm total intervals. The accumulations from the Snow Algorithm can be transformed to the same grids that are presently in use.

10.2.2 The VAD Algorithm

The VAD algorithm is described in section 3.11 in part C of Federal Meteorological Handbook No. 11 (1991) and in chapter 6 of part B of the Handbook. Upper winds are needed at 1000-foot increments above sea level. A range, default 30 km, is assigned for data gathering purposes. For each 1000-foot level, the range at which each tilt's beam crosses that level is calculated. The tilt yielding the closest range to the 30 km is then used. All valid velocity measurements at that tilt and range are fitted to a sine wave as a function of azimuth. The

phase of the sine wave gives the azimuth of the winds. The amplitude of the sine wave, divided by the cosine of the tilt, gives the horizontal wind speed. The fitting is done in successive iterations; outlier values are discarded between iterations.

The Snow Algorithm uses nearly the same processes for determining the vertical wind profile and the same VAD subroutines for the fitting itself. The first difference, apart from coding style, is that more data points are used. Four velocity measurements reside within each reflectivity bin of 1-km length. The Snow Algorithm uses the velocity measurements within 0.5 km of the calculated range. That distance increases the data density for the fitting routine and should thereby improve the fit. Secondly, the 30-km nominal range has been changed to 10 km, a value used by some radar sites. The radar beam is therefore smaller when it passes through each 1000-foot interval. Comparing the 10-km with the 30-km results shows that the latter are systematically biased toward greater speeds. The top of the beam normally samples greater speeds than the bottom under typical wind profiles. The taller beam at 30 km sees a greater spread of velocities than the shorter beam at 10 km. It is therefore expected that the 10-km wind profiles are more accurate for use in the advection scheme of the Snow Algorithm. The wind profile is not used in the PPS algorithm for rain because rain is not advected as severely as snow.

11. Functional Description of the Snow Algorithm

The Snow Algorithm, RADAR7.F (seventh in a series of development programs), performs the calculations for the algorithms leading to the integration of snowfall, based on radar reflectivities and wind velocities. The geometry of the Snow Algorithm is nested cylindrical coordinates. For each tilt (elevation angle), the view is 230 km by 360° in 1 km and 1° resolution for most data. The Algorithm was adapted to a Sun (Solaris 2.4) UNIX environment. Rather than use the complexities of the PPS shared memory, this version makes use of direct access files. This usage generally affects only coding style and execution speed but not the functioning of the Snow Algorithm. The Snow Algorithm retains a shared memory, /STORE/, for the data used most frequently in the VAD wind and reflectivity calculations. The Snow Algorithm will be rewritten to conform with NEXRAD program standards. The Snow Algorithm is in modules, some of which are very much like those already in use in the PPS algorithm and with similar names. Others can be readily converted. An abundance of internal comments have been placed in the FORTRAN code so that programmers can understand what the Snow Algorithm is doing.

Three terrain-based files are needed that are site-specific. The programs generating these files are described in appendix E. The programs themselves are listed in appendixes F, G, and H. The first program simply trims the occultation file of its first four bytes and all of its trailing bytes in order to create a file that is easy to read by direct access. The second program takes the new hybrid sector file, which is an attempt to have the bottom of the radar beam be at least 500 ft above the terrain, and greatly rearranges its structure. The result is a simple direct access file that indicates which elevation tilt is to be used for each azimuth/range location. The third program generates a file in radar coordinates indicating the ground elevation in meters. That terrain file is needed in the advection routine to help determine where falling snow should land on the ground.

11.1 General Functions and Subroutines

NCLOCK, ICLOCK, HHMMSS, DATEJ are time and date conversion conveniences. SIND, COSD, and TAND are trigonometric functions using inputs in degrees (°) rather than radians.

BEAMHT(R,E,A) calculates the height of the center of the radar beam at input radius R (km), elevation angle E (degrees), and antenna altitude A (km). The formula uses a 4/3 earth radius relationship. Setting A=0, gives radar beam height above the radar antenna.

The following is a "tree" showing, by indentations, which program units call the others. GETRADAR is called by three program units, twice by the main program; the data that are subsequently used are different with each call. The above program units in the "tree" are omitted. Program units with names starting with "A31" are equivalents to PPS and VAD routines of the same names. The specific differences in each unit between this Snow Algorithm and those provided by the OSF are listed in appendix J.

```
PROGRAM RADAR7      ! The main program that calls the other units
   A3133D           ! _INIT_ADAPT loads the adaptable parameters
      A3133F        ! _FILL_PRECIP_TABLE biased dBZ to mm/h * 100
   GETRADAR         ! The unpacker of Level II data: header
   GETPROF          ! Calculates the vertical wind profile to 15k ft
      GETRADAR      ! The unpacker of Level II data: (dBZ and velocity)
      A317G2        ! _VAD_PROC
         A317H2     ! _VAD_LSF
         A317I2     ! _VAD_RMS
         A317J2     ! _FIT_TEST
         A317K2     ! _SYM_CHK
   GETRADAR         ! The unpacker of Level II data: header
   GETARI           ! Loads azimuth-range arrays
      GETRADAR      ! The unpacker of Level II data: dBZ (and velocity)
   OCCULT           ! Performs occultation adjustment
   NOSPIKES         ! Cleans reflectivities of isolateds and outliers
   PRECIP           ! Integrates snowfall rates for both falling styles
      ADVECT        ! Calculates volume scan bins for advected snow
   FILENAME         ! Increments input file name
```

11.2 Program Unit Description for the Snow Algorithm

GETRADAR:

The /RADIAL/ list has indices such that an index of 0 indicates a location centered at the range of the radar antenna itself. The subroutine loads all data for each radial into two named commons whether they are ever used or not. The /CLOCK/ list has all of the dates and times in "yymmdd" and "hhmmss" format. The VAL array has floating point numbers and the NUM has integer numbers for what might be considered housekeeping numbers: VAL(2), the center azimuth angle; VAL(3), the center elevation angle; NUM(7), the radial status; and NUM(8), the RDA elevation number, are used extensively by the rest of the Snow Algorithm. The other arrays present the reflectivities, velocities, and spectrum widths in

both raw and decoded format. The menu in the comment section identifies all variables in the named common lists. This style of having an unpacker of Level II data that loads the named common arrays with full radial data for each call is very convenient. It only takes a CALL GETRADAR to get whatever one might want to know about the next radial's data.

PROGRAM RADAR7

After initiating the adaptable parameters, the Snow Algorithm calls up three files specific to the sites being considered. The terrain file was generated by "nearest neighbor" sampling from 30-arc-second data prepared by the Defense Mapping Agency, which has an original resolution near 1 km, similar to the radar resolution. A file could be prepared from similar 3-arc-second data if greater precision is desired, but that task is not a part of the work described here. The occultation file is the same as that used in the PPS Algorithm except that the first four bytes and all trailing bytes have been stripped to make a file that is easy to examine under direct access. The hybrid scan file is related to that produced by Tim O'Bannon for having the bottom of the beam near the terrain but at least 500 feet above it. The expanded sector portion is being ignored; this version of the Snow Algorithm does not try to substitute for reflectivity data in range bins at the edges of cluttered regions. The remaining hybrid sector data for the four lowest tilts were merged into one 230- by 360-array with numbers 1,2,3, and 4 indicating which tilt (0.5, 1.5, 2.5, or 3.5°) to use for each range bin. All three site files (terrain, occultation, hybrid scan) are then easily examined by direct access when needed, avoiding the need to reserve program memory for their full arrays.

The Snow Algorithm uses a file name convention that expects an increment of the file name extension from 2 to 401, representing the file numbers on a Level II tape that are found after the header file. The Snow Algorithm expects a set of files available on a removable or fixed hard disk. The Snow Algorithm calls each file in sequence until the series has no more files left. The algorithm then asks for a new file name for a continuation of the processing. The operator can thereby step over a gap in the series or insert a different removable disk or Level II tape when the next expected file is not found. The rewrite of the Snow Algorithm can use whatever scheme is convenient for obtaining the next file of the series.

After loading and initializing adaptable parameters and site-specific files, the Snow Algorithm first calculates the vertical wind profile, needed for the advection scheme. The Snow Algorithm operates basically the same as the current version used by all WSR-88Ds. The software modifications are merely a condensation by removing white space and trivial comment lines and a substitution of numbers for named constants so that a reader might better understand the processes. The functional modifications allow 4 velocity measurements per radial rather than 1 and use 10 km rather than 30 km for the default range.

The Level II tape data file on disk is then rewound. The entire file must be checked to determine the wind profile and the ending time for the scan, all before considering the reflectivities. Both the wind data and the reflectivity data occupy the same /STORE/ named common and therefore must be loaded sequentially. The process could probably be speeded up in the rewritten version that does not have memory limits by loading the four reflectivity arrays while the wind data are being examined. That procedure would prevent the necessity for a rewind and a rereading could be avoided.

The Snow Algorithm then calls GETARI to load the reflectivity arrays. Notice that the dimensions are to 370°, of which 367 are typically needed. Unlike the PPS version, this Snow

Algorithm does not average two radials that happen to have the same integerized azimuth angle. Instead, it keeps track of the exact azimuth angles in an array and scans the array each time a particular azimuth is needed. The nearest azimuth to the nominated azimuth is selected. That technique produces no gaps in coverage as can happen with simple integerized azimuth angles. The range has been limited to 115 km for the 2.5° tilt and to 10 km for the 3.5° tilt. The ranges needed for those tilts for the three sites (Denver, Cleveland, Albany) are within those values according to the expanded hybrid sector files. That range limitation saves memory space in the computer environment. It does not affect any results.

The reflectivity storage arrays are in named common rather than in a direct access file. The advection scheme needs rapid access to the stored data in almost a random access fashion. Reserving space in named common makes the search go faster than disk access. The Snow Algorithm differs from the PPS algorithm by saving the biased reflectivity numbers (byte or $I*2$) rather than floating point dBZ's or precipitation rates. Converting all of the reflectivities to precipitation rates before the rest of the manipulations takes time, and converting them all back takes additional time. Furthermore, the RATE_TABLE has the same precipitation rate for a variety of reflectivities. If nothing is ever done to a range bin's reflectivity (averaging, occultation boost), the conversion from biased reflectivity to precipitation rate and back to biased reflectivity almost always results in a different number for weak echoes. The PPS conversion thereby substitutes adversely quantized numbers for the real ones.

The subroutine OCCULT is called to add 1, 2, 3, or 4 dBZ to the radial beyond particular ranges indicated by the occultation file. This addition amounts to adding twice that number (or 2, 4, 6, or 8) to the biased reflectivity. The PPS routine has the same adjustment but is coded differently. Both versions have the same 0.2° azimuth resolution.

Then NOSPIKES is called to "weed out" isolated bins above a minimum threshold for precipitation detection and outlier bins above a maximum threshold. The former are replaced by zeros (in biased dBZ units) and the latter by either zeros if the outlier is not isolated from other outliers or averages of the neighbors if the outlier is isolated. The Snow Algorithm uses similar logic to that in the PPS routines but has a smaller array for corrected values. NOSPIKES replaces PPS routines A3133S, A3133O, A3133G, and A3133N.

Once the biased reflectivity data are thereby cleaned, the three snowfall totals files are opened. The names are automatically calculated from the date and time. The 1-hour and 3-hour files are always closed after a Level II file has been processed, but the storm total file remains open. The 1-hour and 3-hour files may be reopened if necessary, or a new file may be created. Separate files record snow accumulation for vertically-falling and advected snow, respectively. Only the advected snow routine is required by the MOU for this version of the Snow Algorithm. The assumption of vertically-falling snow is included here because of its simplicity and speed of calculation. Moreover, as discussed in section 9, initial results indicate the present advection scheme does not improve snowfall estimates over the assumption of vertically-falling snow.

The PRECIP subroutine is then called. This version does not make any adjustment for the time delay between reflectivity observations and the arrival of the snow at the surface. It integrates the snowfall under two assumptions. The fastest calculations are for assuming no wind speed (vertically falling snow). The routine then calls ADVECT to determine the part of the sky from which advected snow is arriving at each ground location. It is assumed that the snow has a vertical fall speed of 1.0 m s^{-1}, typical of aggregated snow. Future revisions

may want to use 0.5 m s^{-1} for single crystals and 2.0 m s^{-1} or more for graupel. The totals files are *I**2 integers in units of mm*100, giving a resolution much finer than can be determined with gages on the ground. The *I**2 integer has space for totals in excess of a foot of water equivalent.

The advection scheme takes most of the processing time. Its value has not yet been determined compared to using a vertically falling snow assumption or extrapolating a vertical gradient of reflectivity. Preliminary inspections show differences between the vertical and advected snow assumptions to result in about a 10-percent difference in total snow accumulation for one particular snowstorm that was the most intense in Denver in early January 1996.

Finally, the Snow Algorithm closes the input file and the four short-term precipitation files. It tries to open the next input file by incrementing the file name. If it finds such a file, then the processing continues automatically.

A3133D

This subroutine provides the Snow Algorithm with the array of adaptable parameters. This PPS subroutine contains a change that loads the /A3133CA/ named common of adaptable parameters by direct definitions rather than by reading a file. Returning to a file-reading version in the eventual rewrite of the Snow Algorithm presents no problem.

A3133F

This subroutine loads the array by which biased dB*Z* is converted to snowfall rate according to the constants in the adaptable parameters list. This version differs significantly in that the snowfall rate (mm h^{-1}) is multiplied by 100 rather than only by 10 to get a better precision at low rates characteristic of snowfall.

GETPROF

The vertical wind profile is measured by this subroutine. The "front end" of the official VAD routine package was too awkward for the rapid development of the Snow Algorithm, so it was replaced. Notice that four initial processing numbers must be determined. The first, VAD_RNG, is assigned to 10 km rather than the default 30 km used elsewhere. Such a value forces the wind algorithm to select greater tilt angles. A byproduct is that the beam is not as broad where it is measuring the winds as it is when the 30 km range is used. The 30 km range winds are systematically of greater speed, most likely because the beam averages over a greater span of altitudes than at 10 km and is biased by the faster winds at the higher altitudes. Therefore, the 10-km range produces more believable wind speeds.

The best NFIT, TH_RMS, and TSMY values have not been supplied by the OSF. The values that this Snow Algorithm assigns essentially let the Snow Algorithm do whatever it wants. Three cycles through the fitting (NFIT) seem adequate for convergence to a reasonable value. Values of the other two variables have been selected so that they are never exceeded and are therefore not necessary.

This Snow Algorithm then uses an inverted beam height routine and quadratic formula to calculate the range closest to the 10 km at which the beam is at a whole thousand feet above

sea level. All valid velocities within 0.5 km of that range (4 velocity readings are expected per kilometer of range) are then used for a fitted sine curve whose amplitude and phase are converted into horizontal wind speed and direction. The call to A317G2 gets all of the official VAD routines with only insignificant editing changes. The Snow Algorithm thereby seems to match the intent of the VAD routines. It differs in its means of finding the appropriate range and then in the use of 4 measurements of Doppler speed rather than only 1. It is hoped that these modifications will provide a slightly superior measurement of the vertical wind profile.

GETARI

This subroutine, which loads reflectivity data into an azimuth-range array, can probably be merged into GETPROF in a future rewrite to avoid a rewind and second reading of the input file. It loads the reflectivity data into program memory for rapid random access by the advection routine. Note that all data are loaded and their precise azimuths are stored in an accompanying array. No radials are averaged. The rest of the Snow Algorithm seeks the *nearest* radial for its snowfall rate integrations. That procedure avoids both averages and gaps.

OCCULT

This subroutine performs occult corrections to reflectivity data at 0.2° azimuth resolution. With one exception, only the first tilt receives occult corrections for each of the three sites. The Denver site needs a correction at the 1.5° tilt for only one radial when it passes Mt. Evans, a high mountain located to the west-southwest. Rather than waste memory space with an increased array size, that exception is corrected by software whenever that specific radial is encountered. If future sites need 1.5° or higher tilt corrections, then the coding can be expanded to do so. The occultation corrections are made for all (about 367) radials separately, even if they fall into the same integerized azimuth angle as duplicate entries.

NOSPIKES

This subroutine combines the elements of A3133S, A3133O, A3133G, and A3133N from the PPS routines. The search for isolated bins and non-isolated outlier bins is the same. When isolated outlier bins are found (if ever), then the snowfall rates of the eight surrounding bins are averaged and converted back to a biased dB*Z* value. This process is done for all (about 367) radials. The updates are made only after the processing of the subsequent radial is completed. The update for the first radial of a scan is delayed until the last radial has been processed. This delay is similar to what is done in the PPS routines. However, this Snow Algorithm deals with the actual stored values of biased reflectivity and has a much smaller corrections array, dimension 3 rather than 9, for each of the 230 range bins.

PRECIP

The Snow Algorithm calculates the time span of the volume scan, extending it to the end of the previous scan if it looks like the series was not interrupted with a large time gap. It tries to read the time periods and durations listed at the end of any previous snowfall totals files. If unsuccessful, it initiates a new set of times for that file. If successful, it updates them. The time and duration numbers can be used by applications programs (written by others) to normalize the snowfall accumulations from possibly partial hours to full hours.

Calculating the snowfall contributions from the assumption of vertically falling snow simply means reaching up to the tilt level specified by the hybrid sector file for each range bin. The calculations go quickly, using the rate look-up table to convert biased reflectivity to precipitation rate. This part of the Snow Algorithm was not required. However, it provides a basis for assessing how much the advection routine improves (if at all) estimation of snowfall totals.

ADVECT

Under the advection version, it is first assumed that winds are calm below the altitude of the radar antenna. The radar cannot measure the winds there anyway. The coding has been left in for dealing with advection below the radar antenna altitude in case a different assumption is used (like using the lowest calculated winds or substituting surface winds from some source). The snow particle trajectory calculation then integrates the horizontal wind profile from the surface or the radar antenna altitude, whichever is higher, and reaches up to within 1 m of the altitude of each successive beam center. If that location is underground (according to the terrain file), then the integration stops and LEVEL is set to 5 to indicate that no snow can be coming from that location. If the trajectory integration location exceeds the 230-km range of the radar, LEVEL is set to 6 to indicate that the radar reflectivity and precipitation rate are not knowable. The integration correctly stops if the tilt sequence number matches the number in the hybrid sector file for that location (LEVEL is set equal to that number). That way, the advected snow always comes from an acceptable location. The snow particle trajectory can twist and turn according to the wind profile. At far ranges and upper tilts, the beam can be rising faster than the snow can fall, leading to an integration that goes beyond the 230-km range limit.

11.3 Outputs

The storm totals files are being provided 2 years before the due date required by the MOU. Six files have been written for precipitation integration. Three are for vertically-falling snow and three are for advected snow. Two are for 1-hour totals, two for 3-hour totals, and two for storm totals (the entire Level II tape or fraction thereof). All are simple arrays in units of mm*100 of water equivalent precipitation. The first 230 *I**2 values are for azimuth 1.0° from a 1- to 230-km range. The 360th set of 230 values is for azimuth 360.0°. A final 361st line of 460 bytes consists of times. JSTART is the start time for the integration and JSTOP is the stop time, both in hhmmss format. JDATE is the stop date in yymmdd format. SUMT (hours) integrates the precise time spans (of each volume scan) over which the precipitation was totaled. SPANT is the time difference (hours) between the JSTART and JSTOP and should be different than SUMT. The time spans of several volume scans never total an exact hour and are always slightly out of phase. These times should allow the extrapolation of the precipitation integrations to a full hour or 3-hour interval if gaps exist.

Specific details in the coding changes for equivalent PPS and VAD subroutines are listed in appendix J. They are intended for programmers and not for general reading.

Appendix K contains the entire FORTRAN code for PROGRAM RADAR7 and its subroutines as written for the Sun UNIX environment. Also included is a C routine for file access.

12. Strengths and Limitations

The Snow Algorithm is presently designed to work on the assumptions of both vertically falling snow and snow being advected by the measured wind and shear profiles. The precipitation accumulation files are at a finer resolution in location and in precipitation amount than the PPS products for rain. This difference allows these files to be converted to the coarser grids used by other PPS products. Some parts of the Algorithm are simpler than comparable PPS code, making it easier to determine the functional styles used. Other parts directly use or only slightly modify PPS and VAD code, making the conversion of the Snow Algorithm into another OSF product that much easier.

Preliminary calculations suggest that snow falling from echoes at ranges greater than about 100 km will advect major distances. Some such snow may be indistinguishable from virga. Analyses of gage data from sites at far range are expected to refine the estimate of a range beyond which the data should be considered unreliable.

This study specifically excludes from consideration all snow that is partially melted. Studies of "bright band" echoes are not considered here.

Advection results from high wind speeds and strong wind shear are likely to be unreliable.

This Snow Algorithm version assigns a fall speed of 1 m s^{-1} for all snow. That generic value is typical of aggregates. Single crystals may have a generic fall speed of about 0.5 m s^{-1} and graupel or graupel-like-snow may have fall speeds of 2 m s^{-1} or greater. Future Snow Algorithm versions may have selectable fall speeds based on storm type and sector of the storm.

This Snow Algorithm has been tested only for proper functioning and not for accuracy in estimating precipitation totals. No indication of the sizes of the errors is available at this time. The users of this Snow Algorithm should treat the outputs with caution.

13. Future Enhancements

The following enhancements to this initial SNOW Algorithm are required during the second and third years of effort under the MOU. However, enhancements No. 1 to 3 listed below have already been accomplished in the initial Snow Algorithm discussed in this report.

1. Produce 1-h, 3-h, and storm-total snowfall accumulation products.

2. Make use of the WSR-88D hybrid scan/occultation data file.

3. Include a sectorized hybrid scan scheme similar to that used for rainfall estimation.

4. Refine the Snow Algorithm with the observations collected during the 1995-96 winter that have not yet been analyzed. This refinement includes the Albany, New York, data set which has not yet been examined (*S* and *SD* observations have not yet been received for 11 of 13 storm events). The MOU calls for using data from a "more limited" (than Denver) observational program, and Denver was specified to have 5 measurement sites. The Albany volunteer network provides up to 90 potential *S* and

SD measurement points. This large data set would be desirable to work with in examination of range effects, ground clutter, and blockage problems, etc. But currently available resources may limit work to a small subset of the available data.

5. Expand the Snow Algorithm to include predictions of *SD*. Observations of *SD* were made at the 3 gage sites nearest the Denver radar. Again, most of the data from the large Albany network of *SD* (and *S*) measurements would be desirable to work with. Twelve-hour *SD* data exist from the Cleveland volunteer network, and it would be desirable to incorporate analysis of these data into the Snow Algorithm.

6. Investigate the means of meaningfully partitioning Z_e-S data and snow density data, and incorporate any useful results into the Snow Algorithm.

7. Optimize the Snow Algorithm for lake effect storms, with special emphasis on the WSR-88D's ability to detect and quantify snowfall (*S* only) as a function of range. Optimize the adaptable parameters for the Cleveland area.

8. Design the Snow Algorithm to be able to use all current WSR-88D scanning strategies, including clear air, and also future modifications (e.g., elevation angles less than 0.5°). The Snow Algorithm presently is appropriate for the clear air and precipitation mode scanning strategies. It will never be appropriate for any severe thunderstorm mode because of the expected presence of wet ice surfaces and therefore bright band reflectivities. Modifications are needed for elevation angles that are of a different series than those currently used by the WSR-88D network.

9. Include the use of adaptable parameters wherever appropriate. Suggested default values and ranges should be given. Specific adaptable parameters mentioned in the OSF request for proposal are coefficients in the Z_e-S relationship, precipitation rate threshold, and minimum reflectivity to convert to rate. The coefficients were already defined in the PPS software. The change from rain to snow merely involves a substitution in the adaptable parameter file. The minimum reflectivity recommended for snow is -10 dB*Z*, which corresponds to a precipitation rate much smaller than can be measured with conventional gages.

10. Provide an adjustment scheme using real-time gages as called for by the MOU during the second year of effort. A simple gage/radar ratio scheme was envisioned. However, Reclamation has recently funded application of an already-developed objective analysis scheme by the University of Oklahoma for adjustment of NEXRAD rainfall fields by real-time gages. This scheme could be expanded for snowfall.

11. Obtain observations of *S* and *SD* in several additional climatological areas with WSR-88D coverage during the winter/spring of 1996-97 (sec. 4.b.(4) of MOU) and continue to test and refine the Snow Algorithm with all data sets (sec. 4.c.(1) of MOU). No funding of special observations has been planned beyond those observations already collected the past winter. Suitable *S* and *SD* measurements will be very difficult to obtain in other geographic areas without significant interaction between Reclamation meteorologists and local WFOs. Funding of such interaction to include site visits by Reclamation personnel is strongly recommended. Even if funding for such interaction becomes available, concern exists regarding the collection of accurate *S* and *SD* observations for all regions of the U.S. As discussed in section 3, the collection of

accurate snowfall measurements, which requires special siting and observations, is difficult because the national precipitation network is inadequate for snow measurements with the resolution needed for Algorithm Development.

12. Pursue a modification to measure and use a vertical gradient of a Z_e adjustment scheme for the Snow Algorithm. The MOU already calls for the enhancements noted above, at least to some degree. However, the work of others has shown that the vertical gradient adjustment rivals the Z_e-S relationship in importance. It is strongly recommended that resources be provided to pursue a vertical gradient of Z_e adjustment scheme for the Snow Algorithm. Such a scheme would also have considerable utility for radar rainfall estimates, especially from shallow clouds.

    As Jurg Joss and colleagues of the Swiss Meteorological Institute have pointed out in several papers (e.g., Joss and Waldvogel, 1990; Joss and Lee, 1995), even a crude seasonal estimate of Z_e's vertical gradient can be used to significantly reduce underestimates of precipitation (especially from shallow storms) at farther ranges from the radar. The vertical gradient of Z_e is used to extrapolate the Z_e measured at the lowest beam tilt with valid data down to the underlying surface. This extrapolation requires use of a terrain elevation file (already generated for Denver, Cleveland, and Albany for use with the Snow Algorithm) for each WSR-88D as well as Z_e's vertical gradient, but elevation data are readily available. At present, Reclamation meteorologists have no plans to deal with this important problem, but strongly believe that it should be part of the Snow Algorithm. This extrapolation would markedly improve snow estimates at middle and far ranges. Development of a scheme to calculate and use the vertical gradient of Z_e is considered to be of similar priority to the use of additional Z_e and S data discussed above. The current PPS scheme is seriously lacking in this important area.

13. It would be desirable to determine the relative merits of two schemes of measuring snow by radar: Use the reflectivities at the tilts given by the new hybrid scan (>500 ft above terrain)—

    (a) *directly over* each ground location (fastest computationally but suspect at far ranges).
    (b) *upwind* from each ground location with perhaps 3 to 4 different snowflake fall speeds (slowest computationally with some range effects).

14. Summary and Recommendations

14.1 Project Overview and Summary

This report discusses the progress of Reclamation meteorologists and support personnel during the first year of effort on a 3-year program to develop a Snow Accumulation Algorithm (hereafter Algorithm). The Algorithm is intended for use by the network of WSR-88D radar systems recently deployed throughout the United States. The Algorithm is intended to predict for *dry snow* both S, the melted water equivalent, and SD, the depth of recently fallen snow prior to settling. An initial prototype Algorithm discussed in this report includes a wind advection scheme for calculating snowflake trajectories and S accumulations over 1- and 3-hour periods as well as the entire storm period. The next version of the Algorithm will include SD predictions based on special observations made during the 1995-96 winter.

Data on Level II tapes, which contain the most basic WSR-88D measurements routinely recorded, were used to provide all radar measurements used in this study. Some attempts were made to acquire suitable snowfall and Level II radar observations from the winter of 1994-95. Nothing of significance came from these efforts because a limited number of Level II recorders (the northernmost at Denver) were operational during that winter. Denver hourly snowfall accumulations were limited from November 1994 through February 1995, at which time routine *S* and *SD* observations were terminated. Comparison of available *S* measurements against available Denver radar data resulted in considerable scatter. Data were not obtained from the 1994-95 winter from more southern and lower-elevation radars than Denver because of likely contamination of most snow events by "bright band" returns from at least partial melting of falling snowflakes.

A major portion of the first year's efforts was collection of suitable *S* and *SD* observations on an hourly basis during the 1995-96 winter and early spring. Data collection took place in three areas observed by WSR-88Ds operating at Albany, New York; Cleveland, Ohio; and Denver, Colorado. Reclamation installed and operated five Belfort Universal gages in a line east-northeast of Cleveland and five gages in the Denver area (a sixth Denver area gage produced data during February through April 1996). In addition, snowboards were used for hourly manual observations of both *S* and *SD* at four sites near Denver including the WFO. Hourly *SD* measurements were not made near Cleveland, but 12-hour observations are available from a volunteer network established by the local WFO.

A large volunteer network obtained manual *S* and *SD* snowboard observations in the Albany, New York, area during 13 dry snowfall events of the 1995-96 winter. This network was established and monitored by personnel from the Albany WFO. Two Belfort gages provided additional *S* data near Albany. Reclamation will analyze these hourly *S* and *SD* observations once they have been quality checked by Albany WFO personnel and made available.

The problems involved in acquiring accurate snowfall measurements are discussed in some detail. It may not be commonly recognized that the national network of precipitation gages generally cannot provide the accurate hourly *S* observations needed for Algorithm development. Many gages are located in open locations (e.g., airports) unprotected from the wind. Wind shields can reduce wind-caused undercatch by exposed gages but cannot overcome the problem. Moreover, many recording gages have a resolution of only 0.10 inch, greater than the large majority of hours with snowfall. Heated tipping bucket gages usually offer sufficient *S* resolution (0.01 inch), but unheated units will not operate during snowfall. Heated tipping bucket gages cause other serious errors with snowfall. Special observations of *S* and *SD* are usually needed to develop and test a snow algorithm.

The importance of empirically estimating α and β coefficients as near the radar as practical was discussed and it was seen that compromises are inevitable. Nearness to the radar reduces uncertainties related to beam distance above the ground and increased beam spreading with range. However, for non-rugged terrain, ground clutter is most common near the radar where protected snow observing sites may or may not exist.

The WSR-88D employs a clutter suppression scheme that can be operator-changed at any time. No record exists on Level II tapes (and often anywhere else) concerning what suppression scheme was in use at any time. Even if the suppression scheme is known, original Z_e values from regions to which suppression was applied cannot be recreated because clutter suppression uses pulse-by-pulse information, and each recorded Z_e value per range bin

is based upon hundreds of pulses. For the above reasons, snow measurement locations were sought out that appeared to be located outside ground cluttered regions.

A literature survey revealed limited information on Z-S or Z_e-S relationships for snowfall. Moreover, broad ranges of published values exist for associated α and β coefficients. Some of the range of α values is caused by improper grouping of Z and Z_e experimental results because radar-observed Z_e is much smaller than calculated Z for snow (Smith, 1984).

Processing of data from Level II tapes is discussed in detail. A "range bin" is the basic spatial unit for which these data are recorded. For Z_e data, which can be related to snowfall rates, the range bin size is 1 km in distance (range) away from the radar by 1° in azimuth (width). Arrays of range bins, at least 3 km by 3 km in size, were extracted from each scan of the radar antenna at its lowest (0.5°) elevation angle (tilt). The scan closest to the ground that is not contaminated by ground clutter must be used to minimize the effects of the vertical profile of Z_e, known to be pronounced in many snow storms.

Extracted range bin arrays were of two types, either directly above individual gages or upwind from the gages as predicted by the Algorithm advection scheme. The commonly used Z_e-S relation with empirically-determined α and β coefficients was used to estimate S for each range bin. Then S was averaged over range bin arrays and finally over 1-hour time intervals calculated directly over or upwind from each gage. This approach avoids the potentially serious biases of averaging Z_e over space and time.

Considerable care was used in reduction of the Belfort gage charts to the nearest 0.005 inch melted water equivalent. This process involved meticulous manual reading using magnification and good lighting by persons with a good appreciation of the potential errors involved with chart reading. Original readings were all double-checked at a later time. Finally, any outlier points found in comparisons of radar estimates, gage observations, and hourly accumulations of 0.10 inch or greater were checked yet again against the original charts. This tedious reduction process, together with the care taken to select protected gage sites and maintain the gages, is believed to have produced a high-quality data base of S observations.

Cleveland and Denver area snowfall observations were compared. Hourly S values were very similar with median accumulations near 0.01 inch at all 10 gages and average values ranging only between 0.015 and 0.025 inch. The main difference between the two data sets was that the Cleveland area had a much higher frequency of snowfall during the early 1995-96 winter. Yet the Z_e-S relations were different, as will be discussed.

An optimization scheme was developed, based on the work of Smith et al. (1975), which uses many iterations of α and β value combinations to select the best fitting values for a given set of hourly radar and gage observations. One α value was established for each β value to be tested by the requirement that average radar-estimated hourly snowfall amounts be equal to the average gage amounts. Each iteration of the scheme requires that the snowfall rate be recalculated for each individual range bin in each array because no averaging of Z_e values was permitted in order to avoid bias.

The optimization scheme was applied to all Cleveland and Denver data available to date, consisting of 2 wet months from the Cleveland area and 3 dry months from around Denver. No time lag was used in this initial analysis to allow snowflakes to fall from the lowest tilt

radar beam to the gage. That is, radar and gage data were synchronized for each hour. Future work will explore the benefits of using time lags.

The closest gage east-northeast of the Cleveland radar was located at a 36-km range. The optimization scheme was applied to the 143 hours of data from this gage and the array of range bins directly over the gage. The resulting relationship was:

$$Z_e = 318\ S^{1.5} \tag{8}$$

using the usual units of mm^6 per m^3 for Z_e and mm h^{-1} for S. The optimization scheme produced α (β) coefficients, which generally decreased (increased) with range over the remaining 4 gages, providing evidence of a range effect.

Equation (8) was applied to the other 4 Cleveland area gages located along a line parallel to the south shore of Lake Erie extending to a 146-km range in northwest Pennsylvania. A range effect was obvious, resulting in ever-greater underestimates by the radar with increasing range. This range effect is related to the increasing height of the radar beam and beam spreading with increasing range. Based on equation (8), average radar-estimated hourly snowfall accumulations for ranges of 61, 87, 115 and 146 km were 85, 61, 32 and 22 percent, respectively, of gage-observed snowfall.

Future work incorporating the vertical profile of Z_e into the Algorithm could probably compensate for the range effect, which appears to become serious beyond about a 60-km range for the current WSR-88D scan mode. Use of a lower tilt scan of about 0.3° could also help reduce far-range underestimates (Smith, 1995).

Correlation coefficients (R values) between radar-estimated and gage-observed hourly accumulations were relatively high (0.7 to 0.8) for the line of Cleveland gages out to a 115-km range. This result suggests adequate signal usually exists with snowfall to beyond 100 km of the radar. The consequence may be that improved schemes, like incorporation of the vertical profile of Z_e, can be developed to quantitatively estimate snowfall to beyond the 100-km range. The relatively high correlations also suggest reasonably accurate snowfall observations by the gages.

Additional observations were collected in the Cleveland area during January to early April 1996 which will be used to further test equation (8), range effects, and the Algorithm.

Application of the optimization scheme to the Denver gage and radar data revealed no evidence of range effects for the 3 Denver gages located within 49 km of the radar. The α coefficient was quite consistent among the three gages in spite of limited available data, ranging between 148 and 164. The β coefficient varied from 1.2 to 1.9, but the influence of β is limited with the relatively light hourly snowfall accumulations contained in the available data set. Higher accumulations are needed to permit better specification of β.

Data from and over the 3 Denver gages within 49 km range were combined using an average relationship of:

$$Z_e = 155\ S^{1.6} \tag{9}$$

Equation (9) appears to be appropriate for the 3 months of Denver area observations analyzed to date. Existing measurements from the February through April 1996 period will be used to further test equation (9).

Use of the optimization scheme with Denver gage No. 5 located in the Rocky Mountains yielded surprising and encouraging results. A high correlation resulted between gage and radar-estimated hourly accumulations in spite of considerable terrain blockage and ground clutter in the 0.5° tilt radar scans over this gage. Apparently, enough signal existed in the (usually) single unblocked radial in the range bin array to have useful predictive value. The possibilities of using the Denver WSR-88D to estimate snow over the Rockies will receive further attention, including use of data from the 1.5° scan.

Use of equation (9) produced reasonable snowfall estimates over gages No. 1 to 4 but, not unexpectedly, resulted in a gross underestimation over gage No. 5. Use of optimized α and β values from table 10 or use of equation (9) produced similar R values. As found with the Cleveland data, use of an optimized relationship does not reduce basic scatter in the data.

An important feature of both equations (8) and (9) is that they tend to overpredict light snowfall accumulations and underpredict heavier accumulations. This problem may be partially caused by the spatial averaging scheme used with radar reflectivity, in which at least 3- by 3-km areas were used. The area sampled by a gage is, of course, only a tiny fraction of a single range bin. This potential problem will be examined further. It is also possible that the optimization scheme may require some modification or that a more complex relationship than equation (1) should be used.

Comparison of radar estimates from advected upwind arrays to those from range bins directly over the gages revealed no improvement. In fact, the advection scheme increased variability beyond 100 km of the Cleveland radar. It is disappointing that the advection scheme, based on the VAD wind profile, did not improve agreement between radar estimates and ground snowfall observations. Perhaps the use of a single snowflake fall speed is too great a simplification. No useful information concerning snowflake fall speeds is available from the WSR-88D because the highest scan is at a 19° elevation angle. Near-vertical scanning would be needed to estimate particle fall speeds with any accuracy using the Doppler shift.

To illustrate that marked differences exist between equations (8) and (9), Cleveland equation (8) was used with Denver data, and Denver equation (9) was applied to Cleveland data. This approach clearly demonstrated significant underestimates and overestimates, respectively, in radar-estimated snowfall accumulations by not applying appropriate Z_e-S relations. Use or the appropriate relation for each geographical region (and perhaps storm type) appears important to achieve quantitative snowfall estimation with radar.

The first version of the Snow Algorithm was produced. It builds upon and uses parts of the PPS and VAD algorithms. Simpler styles than PPS preprocessing are suggested for making occult, isolated, and outlier corrections. A simpler style for the use of the hybrid sector file is introduced.

The Snow Algorithm uses the newly derived (eqs. 8 and 9) relationships between Z_e and S. Precipitation is integrated into bins having the same resolution as the radar beam: 1 km by 1°, for a polar coordinate array of 230 km by 360°. These arrays are easily converted into the grids used by the PPS products. Precipitation files are of 1-hour, 3-hour, and storm-total

duration. One set assumes that snow falls vertically (no wind). This set is calculated the fastest. The other uses the VAD wind profile measured by each volume scan to estimate the snow particle trajectories backward (upward) from each surface location to the beam center of the tilt specified by the hybrid scan for the upwind location. This advection scheme requires considerable computational time.

The Snow Algorithm expects a series of Level II data files with a name having an incrementing extension. The Algorithm accesses each file successively until a break in the series is found. While thereby running unattended, it gives progress reports on the computer display, indicating VAD wind profiles, dates, times, tilt angles, and azimuths. The Algorithm, in spite of the advection scheme and some coding inefficiencies, operates significantly faster than real time in a Sun workstation environment; 6 hours of radar data, for example, take a few hours to process, with the exact time depending on scan mode. Precipitation total files (except for storm-total) are closed at the end of each volume scan, allowing at least temporary access to them by other programs.

Future enhancements and testing of the accuracy of the Snow Algorithm are needed. Users should therefore treat current numerical results with caution.

14.2 Recommendations for Further Study

Several courses of action are recommended for future improvement of the initial prototype Algorithm. The most obvious need is to rigorously test the Algorithm with the data discussed in this report and the remainder of the 1995-96 winter data set. As yet unanalyzed data sets include Cleveland area observations from early January to early April, Denver observations from late January to the end of April, and all the Albany area storm measurements noted in table 2.

The optimization scheme needs to be applied to the remaining measurements to determine the best α and β values to be used in the Algorithm for each geographic area. Moreover, the storm data should be partitioned by storm types and by other means to test whether significant and consistent differences exist in Z_e-S relations.

The optimization scheme was applied to all existing snowfall regardless of rate and including the minimum detectable hourly accumulation: 0.005 inch h^{-1}. Such accumulations are of little importance for many applications. The optimization scheme should be applied to different ranges of snowfall accumulations to determine how the α and β values are influenced. For example, preliminary testing with the data presented, but ignoring accumulations less than 0.01 inch h^{-1}, showed a tendency to decrease the α values and perhaps increase the β values. More optimum relations may be found if only moderate and heavy snowfall accumulations are considered important.

The Algorithm needs to be used to predict the snowfall at several measurement points within range of each radar. Predictions should be attempted both with the advection scheme turned on and off to test whether this scheme offers any advantage. It would be well to test the advection scheme with perhaps 3 to 4 different snowflake fall speeds to determine the range of predicted snowfall rates and how they compare to ground observations.

All observations in this report were hourly and correlations were relatively high. Snowfall predictions should be integrated over longer time periods, such as 3 hours and entire storm

events, to test whether expected improvements are found between radar estimates and gage observations.

Special attention needs to be given to range effects and how to compensate for known radar underestimation at ranges beyond 60 km. An adjustment based on the range raised to some power offers the simplest approach. But a much better technique would use, as a minimum, a climatological vertical profile of Z_e to extrapolate from the lowest beam with valid data to the surface, incorporating a terrain elevation file in the process. Use of real-time vertical profile measurements likely would offer further improvement. The entire vertical profile of Z_e is measured near each radar, which can provide real-time gradients when precipitation echoes exist near the radar. In the absence of nearby echoes, as when a storm is approaching or leaving a radar, a climatological profile could be substituted.

Development of a vertical profile of Z_e adjustment scheme appears at least as important to snowfall estimation as determining the best Z_e-S relationship for various geographical regions according to the work of Jung Joss and others. Such a scheme would provide major improvements to rainfall estimation at farther ranges as well, especially from shallow cloud systems. Development of a vertical profile adjustment scheme would require considerable computations and more resources than are currently available for Snow Algorithm development. But the data already exist to develop, test, and improve such a scheme at Albany, Cleveland, and perhaps Denver. Moreover, any WSR-88D which records Level II data (soon to be all of them) is collecting measurements from which climatological vertical profiles can be calculated. It is strongly recommended that a vertical profile of a Z_e adjustment scheme be developed for the WSR-88D for both rain and snowfall estimation.

15. References

Boucher, R.J. and J.G. Wieler, 1985: "Radar determination of snowfall rate." *Journal of Climate Applied Meteorology*, (24) pp. 68-73.

Brown, M.J. and E.L. Peck, 1962: "Reliability of precipitation measurements as related to exposure." *Journal of Applied Meteorology*, (1) pp. 203-207.

Chrisman, J.N., D.M. Rinderknecht and R.S. Hamilton, October 1994: "WSR-88D clutter suppression and its impact on meteorological data interpretation." *The First WSR-88D Users' Conference Postprints,* Norman OK. pp. 11-14.

Crum, T.D., R.L. Alberty, and D.W. Burgess, 1993: "Recording, archiving, and using WSR-88D data." *Bulletin of the American Meteorological Society*, (74) pp. 645-653.

Federal Meteorological Handbook No. 11 (1991): Doppler radar meteorological observations, part C, WSR-88D products and algorithms. Office of the Federal Coordinator for Meteorological Services and Supporting Research, Rockville, Maryland, 210 pp.

Fujiyoshi, Y., T. Endoh, T. Yamada, K. Tsuboki, Y. Tachibana and G. Wakahama, 1990: "Determination of a *Z-R* relationship for snowfall using a radar and high sensitivity snow gauges." *Journal of Applied Meteorology,* (29) pp. 147-152.

Goodison, B.E., 1978: "Accuracy of Canadian snow gage measurements." *Journal of Applied Meteorology,* (17) pp. 1542-1548.

Groisman, P.Y. and D.R. Legates, 1994: "The accuracy of United States precipitation data." *Bulletin of the American Meteorological Society*, (75) pp. 215-227.

Joss, J. and A. Waldvogel, 1990: "Precipitation measurements and hydrology." *Radar in Meteorology* (D. Atlas, ed.), American Meteorological Society, pp. 577-606.

Joss, J. and R. Lee, 1995: "The application of radar-gauge comparisons to operational precipitation profile corrections." *Journal of Applied Meteorology,* (24) pp. 2612-2630.

Klazura, G.E. and D.A. Imy (1993): "A description of the initial set of analysis products available from the NEXRAD WSR-88D system." *Bulletin of the American Meteorology Society,* (74) pp. 1293-1311.

Krajewski, W.F. and J.A. Smith (1991): "On the estimation of climatological *Z-R* relationships." *Journal of Applied Meteorology,* (30) pp. 1436-1445.

Sekhon, R.S. and R.C. Srivastava (1970): "Snow size spectra and radar reflectivity." *Journal of Atmospheric Sciences*, (27) pp. 299-307.

Sevruk, B. and L. Zahlavova (1994): "Classification system of precipitation gauge site exposure: evaluation and application." *International Journal of Climatology*, (14) pp. 681-689.

Smart, J.R. and J.A. McGinley (1989): "The use of radar reflectivity to determine snowfall over Northeast Colorado." *Preprints, 12th Conference on Weather Analysis and Forecasting,* American Meteorological Society, pp. 390-395.

Smart, J.R. and S.C. Albers (1991): "Evaluation of an objective technique using radar reflectivity and other data to diagnose snowfall." *Preprints, 25th International Conference on Radar Meteorology,* Paris, France, American Meteorological Society, June 24-28, pp. 103-106.

Smith, P.L., D.E. Cain and A.S. Dennis (1975): "Derivation of an *R-Z* relationship by computer optimization and its use in measuring daily areal rainfall." *Preprints, 16th Radar Meteorology Conference,* Houston Texas, American Meteorological Society, April 22-24, pp. 461-466.

Smith, P.L., 1984: "Equivalent radar reflectivity factors for snow and ice particles." *Journal of Climate and Applied Meteorology*, (23) pp. 1258-1260.

Smith, P.L., 1995: "On the minimum useful elevation angle for weather surveillance radar scans." *Preprints, 27th Conference on Radar Meteorology,* Vail CO, October 9-13, pp. 669-671.

Wilson, J.W. (1975): "Measurement of snowfall by radar during the IFYGL." *Preprints, 16th Radar Meteorology Conference,* Houston TX, American Meteorological Society, April 22-24, pp. 508-513.

APPENDIX A

Optimization Technique Used to Derive the Z_e-S Algorithm

by James A. Heimbach, University of North Carolina at Asheville

The method used to derive an optimal Z_e-S relationship follows the techniques described by Smith et al. (1975). This method has the advantage that any assumed Z_e-S relationship can be applied; however, the familiar form:

$$Z_e = \alpha S^{\beta} \tag{1}$$

is used in the current analysis, where Z_e is the reflectivity of the volume irradiated by the radar, and its units are $mm^6\ m^{-3}$. The S term is liquid equivalent snowfall accumulation normally expressed in $mm\ h^{-1}$. The α and β coefficients are what need to be optimized.

Hourly surface gage measurements of snowfall, g, to the nearest 0.005 inch were employed in this analysis, where g_i represents the ith hour. The corresponding radar-derived estimate is indicated by s_i, which has contributions from several volume scans and from several range bins over each gage during each volume scan;

$$s_i = \frac{1}{JK}\sum_J \left(\sum_K s_{i,j,k}\right) \tag{2}$$

The volume scan and bin are denoted by j and k, respectively, and $s_{i,j,k}$ is found from equation (2).

A CTF (criterion function) was used to judge the fit. In this analysis, the CTF is the simple sum of absolute differences between the hourly gage measurement of snowfall and that estimated by the radar:

$$CTF = minimum \sum_I |s_i - g_i| \tag{3}$$

Smith et al. (1975) included a sum of squared residuals term in their criterion function, which, combined with the sum of absolute residual term, gave a Z_e-R relation which had the averages of the gage measurements and radar-derived estimates approximately equal.

Subsequently (personal communication with Paul Smith, January 1996), it was suggested that forcing the average gage accumulation to equal the average radar-derived value would be desirable:

$$\bar{g} \equiv \bar{s} \tag{4}$$

This constraint is easily imposed because for any β applied in equation (1), a corresponding α exists which produces this equality. This constraint simplifies the optimization process by limiting the solution to one of a set of β's, each of which has only one α, rather than requiring both α and β to be varied in the process. In the optimization scheme, a series of β's is selected for which α's are derived using the assumption of equation (4). It should be stressed that the radar estimates of snowfall are summed and averaged, *not* the Z_e values. Also, all S's are recalculated for each α and β pair tested by the CTF.

Numerical Techniques

A program written in C code was used to find optimal α's and β's. The hierarchy of indices used in the program is as follows:

i = hourly index
j = volume scan within the hour
k = bin within vertical or upwind array

Because the number of volume scans and bins varied, the total number of bins, K, is a function of j, and the total number of volume scans, J, is a function of hour, i. For example, the term $K_{i,j}$ in the equations below implies that the jth volume scan of the ith hourly pair had $K_{i,j}$ bins whose snowfall rates were averaged to find that volume scan's contribution to the radar-estimated hourly snowfall. Analogously, J_i volume scans existed whose radar-derived snowfall rates were averaged to estimate the hourly snowfall, s_i.

To be included in the optimization, an array of range bins over a particular gage was required to have a minimum of 4 non-missing volume scans (about 40 minutes of data for clear air scanning and about 23 minutes of precipitation scanning). The minimum number of bins was set to whatever was required to approximate a 3- by 3-km or larger box over each gage as previously discussed. That setting means that no less than 9 range bins were averaged for gages at ranges greater than about 50 km. The maximum number of range bins was 27 for some upwind arrays located between the Denver radar's nearest gages and the radar.

Range bins with Z_e below the level of detectability of the radar, a function of range, were set to zero-estimated snowfall rate. These bins were used in calculation of averages for arrays of range bins because they were scanned by the radar. The radar simply could not detect Z_e above its noise level. For any reasonable values of α and β, the resulting rate of S for such low Z_e (typically less than 0 dBZ) would be negligible.

All the gage and radar data were kept memory resident for computational efficiency. All data arrays were initialized to a missing code, then as data were read in, the memory locations were changed accordingly. Snowfall data with respective dates/times were input first. The radar data were then input and matched to the gage data by the dates/times. Each hourly snowfall accumulation estimate could be based on up to 11 volume scans and from 9 to 27 range bins per array. One of the available gage numbers and either vertical or upwind bin ensembles could be selected by the user.

The radar data were input as hectoBells reflectivity (hBZ). The conversion to reflectivity is:

$$Z_e = 10^{hBz/100} \tag{5}$$

Minimum, maximum, and interval-step values of the β coefficient were specified by a user-defined setup. For each β, an α was found using the assumption of equation (4):

$$\alpha = \left[\frac{\sum_I \frac{1}{J_i} \sum_{J_i} \frac{\left[\sum_{K_{i,j}} \left(z_{i,j,k} \right)^{1/\beta} \right]}{k_{i,j}}}{\sum_I g_i} \right]^{\beta} \tag{6}$$

The CTF was found for each α and β pair by combining equations (2) and (3);

$$CTF = \sum_{I} \left| \left[\frac{1}{\alpha}\right]^{1/\beta} \frac{1}{J_i} \sum_{J_i} \frac{1}{K_{i,j}} \sum_{K_{i,j}} \left[Z_{i,j,k}\right]^{1/\beta} - g_i \right| \tag{7}$$

The minimum CTF defined the optimal α and β. In general, the search was constrained to β's ranging from 0.7 to 3.0 with an increment of 0.05.

Because the snowfall data, g_i, had units of inches per hour, the optimal α defined by equation (7) was converted by:

$$\alpha' = \alpha \left[\frac{1}{25.4}\right]^{\beta} \tag{8}$$

to have S be expressed as mm h^{-1}.

APPENDIX B

Algorithm Documentation Requirements

The Technical Service Center of the Bureau of Reclamation shall tailor its algorithm documentation to the three OSF branches who will be using it. First, for the Applications Branch, Reclamation shall describe the meteorological principles behind the algorithm including theory and formulas. Second, for the Engineering Branch, Reclamation shall provide information needed for successful implementation of the WSR-88D. Finally, for the Operations Training Branch, Reclamation shall provide information needed for course work development.

Each documentation delivery shall consist of nine sections. Following is general guidance regarding the contents of each section and a cross reference or description to fulfill these requirements. See the MOU task descriptions and consult with the OSF POC (point of contact) for detailed requirements.

1. Source

This section shall provide the name, organization, and phone number of the source scientist.

Principal Investigators for the Snow Algorithm for NEXRAD

Dr. Arlin B. Super
Bureau of Reclamation
P.O. Box 25007, D-8510
Denver, CO 80225-0007

(303) 236-0123 x232
fax (303) 236-0199
E-mail: asuper@do.usbr.gov

For program development and Z_e-S questions

Dr. Edmond W. Holroyd, III
Bureau of Reclamation
P.O. Box 25007, D-8260
Denver, CO 80225-0007

(303) 236-4301 x244
fax (303) 236-4697
E-mail: eholroyd@do.usbr.gov

For software development questions

2. Bibliography

This section shall include both the source scientist's papers and papers in closely related areas by other scientists.

This report is the first substantial report on this subject by the source scientists. Relevant papers by other scientists are listed in the references at the end of this report.

3. Functional Description

This section shall describe the meteorological functions the algorithm performs using a plain English language format. The format shall be similar to the Functional Descriptions in the NEXRAD Algorithm Report; however, it shall provide more detail. Detail shall be of such

level that the Applications Branch POC level of expertise will be comparable to the source scientist's. It shall also be of such level that the Training Branch can prepare training materials.

Functional descriptions are presented in section 11 of the main report.

4. Functional Comparison

If a similar WSR-88D algorithm exists, then this section shall discuss the similarities and differences between it and the Reclamation Algorithm. If no similar WSR-88D algorithms exist, Reclamation shall enter "Not applicable."

Functional comparisons are identified in section 11 of the main report and in appendix J.

5. Statistical Performance

This section shall provide statistical and/or quantitative performance of the algorithm. Typically, this section will consist of skill scores.

No statistical evaluation is available at this time and is not due until May 1, 1998.

6. Operational Performance

This section shall evaluate performance in the operational environment and shall include lessons learned from real-time experiments. It shall also recommend how to use the Snow Algorithm in the operational environment.

A personal computer version of the Snow Algorithm was tested on a 486/66. The Level II files had been transferred from tape to disk. When running the Snow Algorithm on data recorded in a clear-air mode, 5 hours of computations were required to process just over 5 hours of real-time data. When the volume scans were more frequent in precipitation mode, 5 hours of computations were required to process about 3 hours of real-time data. These times improve greatly with the use of a Sun workstation (or presumably a Pentium or HP workstation) environment, but timing runs have not yet been made in that environment.

The Snow Algorithm is simple to operate. The opening dialog asks for the site identification and a starting file name. Thereafter, it operates by itself until an incremented file name is not found. Then the operator can possibly perform manual operations, like inserting the next data disk or tape. If continuation is desired, the operator types in the name of the next file to consider, possibly stepping over a gap. If no more files are available to process, the operator can stop the Snow Algorithm by so declaring. The resulting precipitation totals files can then be accessed by other programs, including those that can transform them into images for viewing.

During processing, messages are written to the display to show the progress of the Snow Algorithm. Useful data, such as date, time, key azimuths, and tilts are made known. The wind profile is displayed as soon as it is calculated.

7. Adaptable Parameters

This section shall provide adaptable parameter information to include the parameter's name, a description of the parameter, justification for making it adaptable, rationale for change (i.e., what conditions necessitate change), algorithmic response to change, the range of expected values, and default values.

These parameters have been changed from those in the file:

ZR_MLT_COEFF	=399.	(for equation (5))
	=318.	(for Cleveland, Ohio, equation (8))
	=155.	(for Denver, CO, equation (9))
ZR_PWR_COEF	=2.21	(for equation (5))
	=1.5	(for Cleveland, Ohio, equation (8))
	=1.6	(for Denver, CO, equation (9))

MINTHRFL=-10.
MIN_DBZ_AR_WGTD=-10.
MIN_DBZ=-10.

The Z_e-S parameters are expected to change as a result of the new data being obtained in this study. Precipitation rates resulting from reflectivities less than -10 dBZ are calculated to be insignificant. Three adaptable parameters are therefore changed to -10.

8. Strengths and Limitations

This section shall list the assumptions which must be met for the algorithm to work. It shall describe the environments in which the algorithm performs well and the environments in which the algorithm performs poorly. It shall describe artifacts which are indicators of failure.

Strengths and limitations are described in section 12 of the main report.

9. Future Enhancements

Source scientist's plans or recommendations for future enhancements.

Future enhancements are described in section 13 of the main report.

Deliverable Format

Reclamation shall deliver documentation in hard copy and in WordPerfect for Windows 5.1/5.2 format on 3.5-inch floppy disks.

APPENDIX C

Criteria for Acceptance of Deliverables

1. The Snow Algorithm Code (hereafter code) must be written in FORTRAN and must make extensive use of comments, prologues, and other maintenance-friendly software practices to assist understanding of the code.

2. The code should be structured by functional area whenever possible to allow the insertion of additional functionality (e.g., new scanning strategy).

3. The code must run on a SUN Sparc 20 Unix workstation and should be easily adaptable to HP Unix workstations (models 715, 725, and 755). The code must create files that can be accessed by the NSSL (National Severe Storms Laboratory) RADS (radar algorithms and display system) so that appropriate graphics products can be generated.

4. The code should be able to use all current WSR-88D scanning strategies, including clear air, and also possible future modifications in scanning strategy (e.g., elevation angles less than 0.5°).

5. The code should make use of the WSR-88D hybrid scan/occultation adaptation data file which is part of the precipitation processing system on the RPG.

6. The code must include use of adaptable parameters wherever appropriate. Documentation will include a listing of all adaptable parameters and a suggested default value and ranges for each parameter.

7. The code should produce 1-hour, 3-hour, and storm total snowfall accumulation products.

8. Documentation will include verification statistics demonstrating comparisons between the Snow Algorithm products and ground truth measurements (gage observations of snow water equivalent and snow depth observations).

APPENDIX D

Tasks of the MOU:

The MOU contained a statement of work, partially edited here to give a project overview of the tasks and preliminary results.

July 1, 1996, Due Date

1. Scrutinize existing precipitation gage observations of S from the 1994-95 winter that are within reasonable range of WSR-88D systems with corresponding Level II data. (See sec. 2: insufficient and inadequate data were found for the study.)

2. Obtain Level II data from selected WSR-88D systems and storm periods for the 1994-95 winter that have corresponding gage data, supporting software for manipulation of these data from the OSF, and hardware suitable for working with these data and software. (See sec. 2: all storm data were adversely affected by clutter suppression practices and are not suitable for detailed analysis. All software was obtained from OSF. Most needed hardware was purchased before the beginning of the work, with some later equipment purchased at no cost to this project.)

3. Use the data, software, and hardware of tasks No. 1 and 2 above, and write additional software as needed, for development of a simplified prototype Algorithm for prediction of S from WSR-88D Level II data as further discussed in the Proposal. (This task is the subject of secs. 10 to 13.) The initial Snow Algorithm will be based on comparisons of radar measurements of equivalent reflectivity factor (Z_e) with surface gage measurements of S. The Snow Algorithm will incorporate radar-estimated (VWP [vertical wind profile]) horizontal wind speed and direction for advection of falling snow particles to match surface observations of S with radar bin observations of Z_e. A large number of Z_e-S pairs will be used to calculate the coefficients a and b for the equations $Z_e = aS^b$ (see also secs. 3 to 9).

Deliverable: Hard copy printout and ASCII files of Snow Algorithm source code and documentation suitable for the OSF to modify, recompile, and execute the Snow Algorithm without having to rekey the code. **Due: June 1, 1996 (extended to July 1, 1996, by MOU amendment of August 1995).**

4. Collect high-quality precipitation gage observations of S and observations of SD during the winter/spring of 1995-96 near Denver (see secs. 3 to 6: done).

5. Obtain good-quality observations of S and SD in a climatological area with WSR-88D coverage other than the Denver area during the winter/spring of 1995-96. This observational program, referred to as "more limited" (than Denver) in the MOU, was originally proposed for the second winter of work. However, it was decided to attempt both the Denver and "other climatological area" field programs the first winter in order to speed development of the Snow Algorithm. (See secs. 3 to 6: data have been collected for the area about Albany, New York, well in excess of these requirements.)

6. Based on the MOU modification of August 1995, collect snow accumulation data at five gage sites to be installed and operated in the Cleveland, Ohio, area during the 1995-96 winter. Use the hourly S data from five Belfort Universal gages to prepare the prototype Snow Algorithm that Reclamation is developing for testing in the Cleveland area the following winter. (See secs. 3 to 6: data have been collected. The initial Z_e-S relation is discussed in sec. 9.)

November 1, 1996, Due Date:

1. Refine the Snow Algorithm with the observations from Denver, Cleveland, and Albany, and expand the Snow Algorithm to include predictions of *SD*. Deliverable: the refined Snow Algorithm with supporting documentation, suitable for testing at selected WSR-88D sites with the OSF "proof of concept" UNIX workstations.

2. Reduce precipitation charts from the five Belfort gages operated east-northeast of Cleveland, Ohio, during the 1995-96 winter for several lake effect storm periods of particular interest (e.g., periods with high *S*; periods with low radar tops). With these and Cleveland WSR-88D data, analyze and report on the ability of the developing Snow Algorithm to detect and quantify *S* for lake effect storms.

3. Optimize the Snow Algorithm for lake effect storms in the Cleveland area with special emphasis on the ability of the WSR-88D to detect and quantify *S* as a function of range.

June 1, 1997, Due Date:

1. Develop means to automatically adjust radar-estimated *S* by incorporation of real-time surface observations of *S*. Assimilate this methodology into the November 1, 1996, version of the Snow Algorithm. Provide code and documentation suitable for testing at selected NEXRAD sites with OSF "proof of concept" UNIX workstations.

2. Summarize results of attempts to partition Z_e-*S* and Z_e-*SD* data by rawin and surface observations of temperature, moisture and stability, storm type and phase, cloud top height, and ice crystal types.

3. Obtain observations of *S* and *SD* in several additional climatological areas during the 1996-97 winter/spring. As stated in the MOU, it is assumed that good-quality observations will be identified from existing sources *without* the need for special observations by Reclamation, which are costly. Provide raw Z_e, *S*, and *SD* data from these areas and plots and statistical summaries of the Z_e-*S* pairs. (Note: incorporation of these data into Snow Algorithm development is scheduled for the third year's effort.)

November 1, 1997, Due Date:

Further Snow Algorithm refinements.

May 1, 1998, Due Date:

Final Snow Algorithm refinements, including items of unspecified date:

1. Usability for all scanning strategies, including clear air and elevation angles of less than 0.5°.

2. Use of hybrid scan/occultation adaptation data files (already included in June 1996 version).

3. Production of 1-hour, 3-hour, and storm totals of *S* (already included in June 1996 version).

4. Section 5 verification statistics.

June 1, 1998, Due Date:

1. Further refinements of adaptable parameters, especially for other climate regimes.

2. Section two bibliography (unspecified date) (partly included in this report).

APPENDIX E

Production of the Site-Specific Files Used by the Snow Algorithm in the June 1996 Version

These programs examine terrain-based data for the three radar sites currently under consideration, Denver (KFTG), Cleveland (KCLE), and Albany (KENX). All of the output files for the precipitation integrator are *I**2 in the IBM order of least significant byte first. A simple C-language routine, FLIPBYTE.C, is supplied to reverse the order of bytes in the *I**2 files. The program codes appear in the following appendixes:

Appendix F—PROGRAM OCCTRIM
Appendix G—PROGRAM HYSTRIMT
Appendix H—PROGRAM POLAR230
Appendix I—PROGRAM FLIPBYTE

PROGRAM OCCTRIM reads the entire K---.OCC file (where --- is the site identifier) into the named common /A3133C4/, derived from a rearranged A3133C4.inc file and inserted into this program. The K---.OCC file begins with 4 bytes to be stored in a variable START_C4. Thereafter is an array OCC_DATA(0:5, 1800, 4) of *I**2 data. Once that array is filled, the rest of the file is ignored. The first index, OCCODE, running from 0 to 5, is a code for the amount of occultation of the beam beyond a particular range. In later processing, an amount equal to OCCODE is added to the dB*Z*; alternatively, an amount equal to OCCODE * 2 is added to the biased dB*Z*. The subroutine comments identify the blockage thresholds.

The second index, OCINDX, is the azimuth in 0.2° increments (1800 of them). The third index, ELINDX, is for the four lowest tilt angles.

The table itself contains the last ranges (km) at which an occultation code is valid. Once the file is read into the named common, only the OCC_DATA array is written to a file in 12-byte records. The Snow Algorithm then calculates the particular azimuths and elevations needed and reads the appropriate 12-byte record to get the 6 ranges. It then applies the corrections if they are needed.

The purpose of this simple conversion was to get rid of the 4 leading bytes in the file and to ignore all trailing bytes. That conversion let the file be read in the Snow Algorithm in the DIRECT access mode. Apart from the shift in data, the resulting trimmed file is identical to the original.

PROGRAM HYSTRIMT reads the three new hybrid sector files, KFTGT.HYS for Denver, KCLET.HYS for Cleveland, and KENXT.HYS for Albany. They attempt to specify which tilt to use so that the bottom of the radar beam is at least 500 feet above the terrain. An unpacker reads the entire file into the /A3133C3/ named common. That common is described in A3133C3.inc and was rearranged and inserted into this program.

The original arrays reserve space for all 360 azimuths in the two giant arrays of dimension (51,360,4) for *I**2 data. Rather than loading data at all azimuth locations, the giant array has sector pointer arrays telling where to look in the big arrays for the data. That feature is an unnecessary complication that does nothing to save space. This program therefore makes the second (1 to 360) index be the actual azimuth angle and fills the array completely.

The first index in the giant arrays ranges from 1 to 51. Looking at the data for the three sites, only a range of 1 to 15 is ever needed. Again, valuable space is wasted. The third array index is the elevation angle index, ELINDX, ranging 1 to 4, for the 4 lowest tilts.

The first giant array, EL_AZ_RNG(51,360,4), mostly contains pairs of ranges between which the reflectivities for the elevation angle indicated by ELINDX are to be used for precipitation measurements. When the first index = 1, the array gives the number of range pairs to follow under the rest of the first index. The largest value of EL_AZ_RNG(1,*i*,*j*) ever seen was 7, indicating that 14 range values would be found as far as EL_AZ_RNG(15,*i*,*j*). The first dimension was therefore shortened from 51 to 15 in the Snow Algorithm to provide a great savings in space.

The second giant array, XP_EL_AZ_RNG(51,360,4), had the same structure. However, the areas of tilt usage were extended from the EL_AZ_RNG array locations by 2 km and 2° in all directions. Therefore, the marked regions overlapped for the various tilts. The purpose was to allow the extrapolation of reflectivities from areas of no clutter to a safe distance into cluttered regions. In the May 1996 version of the snow precipitation integrator, this extrapolation ability is ignored.

Part of this program, reflecting earlier development, wrote a file with 720-byte records that was more easily indexed for direct access reading. The giant arrays were reduced to dimension (15,360,4). To verify that the code was working correctly, the rewritten file was read back in and converted to an image for visualization. These parts of the program are not really needed any more. However, the final modification is located at the end of those adjustments.

One can use the data in the EL_AZ_RNG array to fill 4 arrays (for each tilt) of 230 (km range) by 360° (degrees azimuth) with a binary indicator specifying whether to use that particular tilt. Every azimuth-range location will have one and only one of the four tilts declared acceptable for use. The four arrays can then be condensed into a single array with values of 1, 2, 3, and 4 indicating the tilt (ELINDX) to use for the precipitation calculation at a particular location. The resulting array (165,600 bytes) is much easier to use than the many variables in the original /A3133C3/ named common (299,528 bytes). In this smaller version, the named common is still in an *I**2 format, but an *I**1 format (82,800 bytes) would be adequate and would save additional space. The Snow Algorithm accesses the resulting file (HYBRDDEN.INT, HYBRDCLE.INT, or HYBRDALB.INT) as direct access. Given a desired range and azimuth, the file record number is calculated (azimuth), the record read, and the proper elevation index is read from the range-indexed array in that record.

Unfortunately, the XP_EL_AZ_RNG array cannot be written with just the tilt indices to use because of the overlap in areas. Adding an index for the 4 tilts would make the file 331,200 bytes in the *I**1 format. That size may be suitable for direct access reading but would take up too much space in program memory.

PROGRAM POLAR230 converts 30-arc-second digital terrain data to polar coordinates from a radar out to a range of 230 km. MIPS (map and image processing system) software was used to splice and extract an array of terrain data from a set of files organized by 5° of longitude in width. The program expects the extraction to be in multiples of whole degrees (°) in latitude and longitude. Some files are 6° of longitude wide and others are 7° wide. The program asks for top and bottom latitudes and left and right edge longitudes.

The terrain data, in meters above mean sea level, are written to a polar coordinate array of width 230 km and length 360° of azimuth about the specified radar site coordinates. The array is initialized with -999 default values.

On the first pass all of the latitude/longitude coordinates of the terrain, data points are converted to the nearest radar polar coordinates by means of a great circle navigation subroutine. The terrain elevation is then assigned to that coordinate. The process leaves gaps in the array, indicated by the -999 code.

On the second pass, all of the unused range/azimuth coordinates are converted to the nearest latitude/longitude coordinates in 30-arc-second resolution by means of another great circle navigation subroutine. The terrain elevation from the input file is then transferred to the radar coordinate file. All gaps are thereby filled. The file is then written to disk in 460-byte *I**2 records for the ranges 1 to 230 km. There are 360 records (azimuths) written. The result is a file specifying the terrain elevation at every radar coordinate.

The program can be rewritten to use the higher resolution 3-arc-second terrain data. However, the general 1-km resolution of the 30-arc-second data was considered adequate for this test version of the software.

APPENDIX F

PROGRAM OCCTRIM

```
      PROGRAM OCCTRIM
C   Reads Norman's occultation files *.OCC, flips bytes, stores without
C   header and trailer; get 86,400 bytes ready for direct access, *T.OCC
C   version of 17 April 1996, by Ed Holroyd
      INTEGER   START_C4
      INTEGER*2 OCC_DATA(0: 5, 1800, 4)
      COMMON /A3133C4/ START_C4, OCC_DATA
      INTEGER*2 LINE(6)
      CHARACTER*1 MSG,UCY,LCY
      CHARACTER*12 OCCULT1(3),OCCULT2(3)
      DATA OCCULT1/'KFTG.OCC    ','KENX.OCC    ','KCLE.OCC    '/
      DATA OCCULT2/'KFTGT.OCC   ','KENXT.OCC   ','KCLET.OCC   '/
      DATA UCY,LCY/'Y','y'/
C     OPEN(UNIT=4,FILE='PRINTER.DMP',STATUS='UNKNOWN')
    1 WRITE(6,*)' Which site? 1=Denver, 2=Albany, 3=Cleveland'
      READ(5,*)IDSITE
      OPEN(UNIT=1,FILE=OCCULT1(IDSITE),STATUS='OLD'
     +,RECL=256,FORM='BINARY',IOSTAT=IOERROR)
      IF(IOERROR.EQ.1)WRITE(6,*)' Error opening occultation file'
      OPEN(UNIT=2,FILE=OCCULT2(IDSITE),STATUS='UNKNOWN'
     +,RECL=12,FORM='BINARY',IOSTAT=IOERROR)
      IF(IOERROR.EQ.1)WRITE(6,*)' Error opening occult output file'
      WRITE(6,*)' Reading the array'
      CALL GETOCC ! to load the file into /A3133C4/: 86,404 bytes
      WRITE(6,*)' Writing the array'
C   output the array
      DO 50 K=1,4 ! tilts
      DO 40 J=1,1800 ! azimuths
      IF(MOD(J,100).EQ.0)WRITE(6,*)J
      DO 30 I=0,5 ! occultation codes
   30 LINE(I+1)=OCC_DATA(I,J,K)
   40 WRITE(2)LINE
   50 CONTINUE
C   ask for another site
      WRITE(6,*)' Do you want another site?'
      READ(5,81)MSG
   81 FORMAT(A1)
      IF(MSG.NE.UCY.AND.MSG.NE.LCY)GO TO 90
      CLOSE (1)
      CLOSE (2)
      GO TO 1 ! for a new input file selection
   90 WRITE(6,*)' Normal stop'
      STOP
      END
C
      SUBROUTINE GETOCC
C   To unpack occultation files for NEXRAD radars and load A3133C4 array
      INTEGER*1 IINBUF(256) ! input buffers
      INTEGER*2 INBYTE(256)
**A3133C4.inc
C** Named common definitions for the occultation data, modified for unpacking
      INTEGER   START_C4
      INTEGER*2 OCC_DATA(0: 5, 1800, 4),OCCWORD(43200)
      EQUIVALENCE (OCC_DATA,OCCWORD)
      COMMON /A3133C4/ START_C4, OCC_DATA
C   The values in OCC_DATA are the range bins at which the occultation reaches
C   the indicated percentage.
C   The first index, ranging from 0 to 6 is the occultation code:
C   Code   Occultation %
C     0       0-10, >60 when sample volume is part of a completely obscurred
C               region extending over more than 2 degrees in azimuth
C     1      11-29
C     2      30-43
C     3      44-55
C     4      56-60
```

```
C     5      >60 when sample volume is part of a completely obscured region
C                 extending over no more than 2 degrees in azimuth
C   The second index is for 0.2 degree resolution data, 360 degrees * 5.
C   The third index is for 4 elevation levels.
C   convert bytes to positive integers
C*********************************************************************
      WRITE(6,*)' Reading occultation file'
      READ(1)IINBUF ! 256 bytes at a time
      DO 10 I=1,256
      INBYTE(I)=IINBUF(I)
      IF(INBYTE(I).LT.0)INBYTE(I)=INBYTE(I)+256
   10 CONTINUE
      START_C4=((INBYTE(1)*256+INBYTE(2))*256+INBYTE(3))*256+INBYTE(4)
      NWORDS=1
      DO 15 I=5,255,2 ! merge the bytes into 2-byte words
      OCCWORD(NWORDS)=INBYTE(I)*256+INBYTE(I+1)
   15 NWORDS=NWORDS+1
      DO 30 J=1,336 ! read in most of the rest of the file
      IF(MOD(J,30).EQ.0)WRITE(6,*)' (336):',J
      READ(1)IINBUF ! 256 bytes at a time
      DO 20 I=1,256
      INBYTE(I)=IINBUF(I)
      IF(INBYTE(I).LT.0)INBYTE(I)=INBYTE(I)+256
   20 CONTINUE
      DO 25 I=1,255,2 ! merge the bytes into 2-byte words
      OCCWORD(NWORDS)=INBYTE(I)*256+INBYTE(I+1)
   25 NWORDS=NWORDS+1
   30 CONTINUE
      READ(1)IINBUF ! 256 bytes at a time; last buffer
      DO 40 I=1,132 ! need only part of it
      INBYTE(I)=IINBUF(I)
      IF(INBYTE(I).LT.0)INBYTE(I)=INBYTE(I)+256
   40 CONTINUE
      DO 45 I=1,131,2 ! merge the bytes into 2-byte words
      OCCWORD(NWORDS)=INBYTE(I)*256+INBYTE(I+1)
   45 NWORDS=NWORDS+1
      RETURN
      END
```

APPENDIX G

PROGRAM HYSTRIMT

```
      PROGRAM HYSTRIMT
C   to trim the size of the .HYS files for three sites from 51 to 15 in one
C   set of dimensions.  Also will fill out the EL_AZ_RNG tables to avoid the
C   need for a sector pointer, though that will be produced anyway.  The
C   output image in AIRMAGE.DTA is for visualization.
C   This version also makes an I*2 file of 230 km x 360 degrees indicating
C   which tilt to use for hybrid sector.
C   version of 22 April 1996, by Ed Holroyd
      CHARACTER*1 MSG,UCY,LCY
      CHARACTER*12 HYSECTRS(3),HYSTRIMS(3),HYBRDSIT(3)
      DATA HYSECTRS/'KFTGT.HYS   ','KENXT.HYS   ','KCLET.HYS   '/
      DATA HYSTRIMS/'KFTGTRMT.HYS','KENXTRMT.HYS','KCLETRMT.HYS'/
      DATA HYBRDSIT/'HYBRDDEN.INT','HYBRDALB.INT','HYBRDCLE.INT'/
      DATA UCY,LCY/'Y','y'/
      OPEN(UNIT=3,FILE='ARIMAGE.DTA',STATUS='UNKNOWN',FORM='BINARY'
     +,BLOCKSIZE=460) ! azimuth-range initial display
      OPEN(UNIT=4,FILE='PRINTER.DMP',STATUS='UNKNOWN')
  100 WRITE(6,*)' Which site? 1=Denver, 2=Albany, 3=Cleveland'
      READ(5,*)IDSITE
      OPEN(UNIT=1,FILE=HYSTRIMS(IDSITE),STATUS='UNKNOWN',ACCESS='DIRECT'
     +,RECL=720,FORM='UNFORMATTED',IOSTAT=IOERROR)
      IF(IOERROR.EQ.1)WRITE(6,*)' Error opening sector trim file'
      OPEN(UNIT=2,FILE=HYSECTRS(IDSITE),STATUS='OLD',ACCESS='DIRECT'
     +,RECL=256,FORM='UNFORMATTED',IOSTAT=IOERROR)
      IF(IOERROR.EQ.1)WRITE(6,*)' Error opening sector file'
      OPEN(UNIT=7,FILE=HYBRDSIT(IDSITE),STATUS='UNKNOWN',FORM='BINARY'
     +,BLOCKSIZE=460) ! hybrid sectors in polar coordinates
      CALL GETHYS ! to load the file into /A3133C3/: 299,524 bytes
      CALL PUTHYS ! to write the trimmed file
      CALL GETHYS2 ! to read back trimmed file
      WRITE(6,*)' Do you want another file?'
      READ(5,81)MSG
   81 FORMAT(A1)
      IF(MSG.NE.UCY.AND.MSG.NE.LCY)GO TO 90
      GO TO 100 ! for a new input file selection
   90 WRITE(6,*)' Normal stop'
      STOP
      END
C
      SUBROUTINE GETHYS
C   To unpack occultation files for NEXRAD radars and load A3133C3 array
      INTEGER*1 IINBUF(256) ! input buffers
      INTEGER*2 INBYTE(256)
**A3133C3.inc
C** Named common for the elevation, range, and azimuth sector adaptation
C** tables
      INTEGER   START_C3,AZBOUNDS(360),SECTOR_PNTR(360)
      INTEGER*2 EL_AZ_RNG(51,360,4),ELWORD(73440)
      INTEGER   XP_AZBOUNDS(360),XP_SECTOR_PNTR(360)
      INTEGER*2 XP_EL_AZ_RNG(51,360,4),XPWORD(73440)
      EQUIVALENCE(EL_AZ_RNG,ELWORD),(XP_EL_AZ_RNG,XPWORD)
      COMMON /A3133C3/ START_C3, AZBOUNDS, SECTOR_PNTR, EL_AZ_RNG,
     $            XP_AZBOUNDS, XP_SECTOR_PNTR, XP_EL_AZ_RNG
C   convert bytes to positive integers
C***********************************************************************
      WRITE(6,*)' Reading .HYS file into /A3133C3/'
      READ(2)IINBUF ! 256 bytes at a time
      DO 10 I=1,256
      INBYTE(I)=IINBUF(I)
      IF(INBYTE(I).LT.0)INBYTE(I)=INBYTE(I)+256
   10 CONTINUE
      START_C3=((INBYTE(1)*256+INBYTE(2))*256+INBYTE(3))*256+INBYTE(4)
      NWORDS=1
      WRITE(6,*)' Loading AZBOUNDS'
      DO 12 I=5,253,4 ! merge the bytes into first 63 4-byte words
```

```
      AZBOUNDS(NWORDS)=((INBYTE(I)*256+INBYTE(I+1))*256+INBYTE(I+2))*256
     ++INBYTE(I+3)
   12 NWORDS=NWORDS+1
      DO 18 J=1,4 ! read in most of the rest of the AZBOUNDS array (256 more)
      READ(2)IINBUF ! 256 bytes at a time
      DO 14 I=1,256
      INBYTE(I)=IINBUF(I)
      IF(INBYTE(I).LT.0)INBYTE(I)=INBYTE(I)+256
   14 CONTINUE
      DO 16 I=1,253,4 ! merge the bytes into 4-byte words
      AZBOUNDS(NWORDS)=((INBYTE(I)*256+INBYTE(I+1))*256+INBYTE(I+2))*256
     ++INBYTE(I+3)
   16 NWORDS=NWORDS+1
   18 CONTINUE
      READ(2)IINBUF ! 256 bytes at a time
      DO 20 I=1,256
      INBYTE(I)=IINBUF(I)
      IF(INBYTE(I).LT.0)INBYTE(I)=INBYTE(I)+256
   20 CONTINUE
      DO 22 I=1,161,4 ! merge the bytes into last 41 4-byte words
      AZBOUNDS(NWORDS)=((INBYTE(I)*256+INBYTE(I+1))*256+INBYTE(I+2))*256
     ++INBYTE(I+3)
   22 NWORDS=NWORDS+1
      NWORDS=1
      WRITE(6,*)' Loading SECTOR_PNTR'
      DO 24 I=165,253,4 ! merge the bytes into first 23 4-byte words
      SECTOR_PNTR(NWORDS)=((INBYTE(I)*256+INBYTE(I+1))*256+INBYTE(I+2))
     +*256+INBYTE(I+3)
   24 NWORDS=NWORDS+1
      DO 30 J=1,5 ! read in most of the rest of the SECTOR_PNTR array (320
                  ! more)
      READ(2)IINBUF ! 256 bytes at a time
      DO 26 I=1,256
      INBYTE(I)=IINBUF(I)
      IF(INBYTE(I).LT.0)INBYTE(I)=INBYTE(I)+256
   26 CONTINUE
      DO 28 I=1,253,4 ! merge the bytes into 4-byte words
      SECTOR_PNTR(NWORDS)=((INBYTE(I)*256+INBYTE(I+1))*256+INBYTE(I+2))
     +*256+INBYTE(I+3)
   28 NWORDS=NWORDS+1
   30 CONTINUE
      READ(2)IINBUF ! 256 bytes at a time
      DO 32 I=1,256
      INBYTE(I)=IINBUF(I)
      IF(INBYTE(I).LT.0)INBYTE(I)=INBYTE(I)+256
   32 CONTINUE
      DO 34 I=1,65,4 ! merge the bytes into last 17 4-byte words
      SECTOR_PNTR(NWORDS)=((INBYTE(I)*256+INBYTE(I+1))*256+INBYTE(I+2))
     +*256+INBYTE(I+3)
   34 NWORDS=NWORDS+1
      NWORDS=1
      WRITE(6,*)' Loading EL_AZ_RNG'
      DO 36 I=69,255,2 ! merge the bytes into first 94 2-byte words
      ELWORD(NWORDS)=INBYTE(I)*256+INBYTE(I+1)
   36 NWORDS=NWORDS+1
      DO 42 J=1,573 ! read in most of the rest of the ELWORD array (73344
                    ! more)
      IF(MOD(J,100).EQ.0)WRITE(6,*)J,'(573)'
      READ(2)IINBUF ! 256 bytes at a time
      DO 38 I=1,256
      INBYTE(I)=IINBUF(I)
      IF(INBYTE(I).LT.0)INBYTE(I)=INBYTE(I)+256
   38 CONTINUE
      DO 40 I=1,255,2 ! merge the bytes into 2-byte words
      ELWORD(NWORDS)=INBYTE(I)*256+INBYTE(I+1)
```

```
40 NWORDS=NWORDS+1
42 CONTINUE
   READ(2)IINBUF ! 256 bytes at a time
   DO 44 I=1,256
   INBYTE(I)=IINBUF(I)
   IF(INBYTE(I).LT.0)INBYTE(I)=INBYTE(I)+256
44 CONTINUE
   DO 46 I=1,3,2 ! merge the bytes into last 2 2-byte words
   ELWORD(NWORDS)=INBYTE(I)*256+INBYTE(I+1)
46 NWORDS=NWORDS+1
   NWORDS=1
   WRITE(6,*)' Loading XP_AZBOUNDS'
   DO 48 I=5,253,4 ! merge the bytes into first 63 4-byte words
   XP_AZBOUNDS(NWORDS)=((INBYTE(I)*256+INBYTE(I+1))*256+INBYTE(I+2))
  +*256+INBYTE(I+3)
48 NWORDS=NWORDS+1
   DO 54 J=1,4 ! read in most of the rest of the XP_AZBOUNDS array (256
               ! more)
   READ(2)IINBUF ! 256 bytes at a time
   DO 50 I=1,256
   INBYTE(I)=IINBUF(I)
   IF(INBYTE(I).LT.0)INBYTE(I)=INBYTE(I)+256
50 CONTINUE
   DO 52 I=1,253,4 ! merge the bytes into 4-byte words
   XP_AZBOUNDS(NWORDS)=((INBYTE(I)*256+INBYTE(I+1))*256+INBYTE(I+2))
  +*256+INBYTE(I+3)
52 NWORDS=NWORDS+1
54 CONTINUE
   READ(2)IINBUF ! 256 bytes at a time
   DO 56 I=1,256
   INBYTE(I)=IINBUF(I)
   IF(INBYTE(I).LT.0)INBYTE(I)=INBYTE(I)+256
56 CONTINUE
   DO 58 I=1,161,4 ! merge the bytes into last 41 4-byte words
   XP_AZBOUNDS(NWORDS)=((INBYTE(I)*256+INBYTE(I+1))*256+INBYTE(I+2))
  +*256+INBYTE(I+3)
58 NWORDS=NWORDS+1
   NWORDS=1
   WRITE(6,*)' Loading XP_SECTOR_PNTR'
   DO 60 I=165,253,4 ! merge the bytes into first 23 4-byte words
   XP_SECTOR_PNTR(NWORDS)=((INBYTE(I)*256+INBYTE(I+1))*256
  ++INBYTE(I+2))*256+INBYTE(I+3)
60 NWORDS=NWORDS+1
   DO 66 J=1,5 ! read in most of the rest of the XP_SECTOR_PNTR array (320
               ! more)
   READ(2)IINBUF ! 256 bytes at a time
   DO 62 I=1,256
   INBYTE(I)=IINBUF(I)
   IF(INBYTE(I).LT.0)INBYTE(I)=INBYTE(I)+256
62 CONTINUE
   DO 64 I=1,253,4 ! merge the bytes into 4-byte words
   XP_SECTOR_PNTR(NWORDS)=((INBYTE(I)*256+INBYTE(I+1))*256
  ++INBYTE(I+2))*256+INBYTE(I+3)
64 NWORDS=NWORDS+1
66 CONTINUE
   READ(2)IINBUF ! 256 bytes at a time
   DO 68 I=1,256
   INBYTE(I)=IINBUF(I)
   IF(INBYTE(I).LT.0)INBYTE(I)=INBYTE(I)+256
68 CONTINUE
   DO 70 I=1,65,4 ! merge the bytes into last 17 4-byte words
   XP_SECTOR_PNTR(NWORDS)=((INBYTE(I)*256+INBYTE(I+1))*256
  ++INBYTE(I+2))*256+INBYTE(I+3)
70 NWORDS=NWORDS+1
   NWORDS=1
```

```
      WRITE(6,*)' Loading XP_EL_AZ_RNG'
      DO 72 I=69,255,2 ! merge the bytes into first 94 2-byte words
      XPWORD(NWORDS)=INBYTE(I)*256+INBYTE(I+1)
   72 NWORDS=NWORDS+1
      DO 78 J=1,573 ! read in most of the rest of the ELWORD array (73344
                    ! more)
      IF(MOD(J,100).EQ.0)WRITE(6,*)J,'(573)'
      READ(2)IINBUF ! 256 bytes at a time
      DO 74 I=1,256
      INBYTE(I)=IINBUF(I)
      IF(INBYTE(I).LT.0)INBYTE(I)=INBYTE(I)+256
   74 CONTINUE
      DO 76 I=1,255,2 ! merge the bytes into 2-byte words
      XPWORD(NWORDS)=INBYTE(I)*256+INBYTE(I+1)
   76 NWORDS=NWORDS+1
   78 CONTINUE
      READ(2)IINBUF ! 256 bytes at a time
      DO 80 I=1,256
      INBYTE(I)=IINBUF(I)
      IF(INBYTE(I).LT.0)INBYTE(I)=INBYTE(I)+256
   80 CONTINUE
      DO 82 I=1,3,2 ! merge the bytes into last 2 2-byte words
      XPWORD(NWORDS)=INBYTE(I)*256+INBYTE(I+1)
   82 NWORDS=NWORDS+1
      RETURN
      END
C
      SUBROUTINE PUTHYS
C   writes a truncated version of /A3133C3/ to disk using 720 byte I*2 records
      INTEGER*2 OUTWORD(360),STA(2),AZB(720),SEC(720),XPA(720),XPS(720)
      EQUIVALENCE (START_C3,STA),(AZBOUNDS,AZB),(SECTOR_PNTR,SEC)
     +,(XP_AZBOUNDS,XPA),(XP_SECTOR_PNTR,XPS)
**A3133C3.inc
C** Named common for the elevation, range, and azimuth sector adaptation
C** tables
      INTEGER   START_C3,AZBOUNDS(360),SECTOR_PNTR(360)
      INTEGER*2 EL_AZ_RNG(51,360,4)
      INTEGER   XP_AZBOUNDS(360),XP_SECTOR_PNTR(360)
      INTEGER*2 XP_EL_AZ_RNG(51,360,4)
      COMMON /A3133C3/ START_C3, AZBOUNDS, SECTOR_PNTR, EL_AZ_RNG,
     $             XP_AZBOUNDS, XP_SECTOR_PNTR, XP_EL_AZ_RNG
C***********************************************************************
      OUTWORD(1)=STA(1)
      OUTWORD(2)=STA(2)
      DO 1 J=3,360
    1 OUTWORD(J)=0 ! blank fill on first record to
      WRITE(1,REC=1)OUTWORD ! all 720 bytes
      DO 10 J=1,360
   10 OUTWORD(J)=AZB(J)
      WRITE(1,REC=2)OUTWORD
      DO 12 J=1,360
   12 OUTWORD(J)=AZB(J+360)
      WRITE(1,REC=3)OUTWORD
      DO 14 J=1,360
   14 OUTWORD(J)=SEC(J)
      WRITE(1,REC=4)OUTWORD
      DO 16 J=1,360
   16 OUTWORD(J)=SEC(J+360)
      WRITE(1,REC=5)OUTWORD
      NREC=6
      DO 30 K=1,4
      DO 30 I=1,15 ! not the 51 available
      DO 20 J=1,360
      JJ=SECTOR_PNTR(J) ! to eliminate the need for this array
   20 OUTWORD(J)=EL_AZ_RNG(I,JJ,K) ! fill out the 360 dimension
```

```
      WRITE(1,REC=NREC)OUTWORD
   30 NREC=NREC+1
      DO 40 J=1,360
   40 OUTWORD(J)=XPA(J)
      WRITE(1,REC=NREC)OUTWORD
      NREC=NREC+1
      DO 42 J=1,360
   42 OUTWORD(J)=XPA(J+360)
      WRITE(1,REC=NREC)OUTWORD
      NREC=NREC+1
      DO 44 J=1,360
   44 OUTWORD(J)=XPS(J)
      WRITE(1,REC=NREC)OUTWORD
      NREC=NREC+1
      DO 46 J=1,360
   46 OUTWORD(J)=XPS(J+360)
      WRITE(1,REC=NREC)OUTWORD
      NREC=NREC+1
      DO 60 K=1,4
      DO 60 I=1,15 ! not the 51 available
      DO 50 J=1,360
      JJ=XP_SECTOR_PNTR(J) ! to eliminate the need for this array
   50 OUTWORD(J)=XP_EL_AZ_RNG(I,JJ,K) ! fill out the 360 dimension
      WRITE(1,REC=NREC)OUTWORD
   60 NREC=NREC+1
      RETURN
      END
C
      SUBROUTINE GETHYS2
C   To unpack trimmed hybrid sector files for NEXRAD radars and load /A3133C3/
C   from disk using 720 byte I*2 records
      INTEGER*2 INWORD(360),STA(2),AZB(720),SEC(720),XPA(720),XPS(720)
     +,LINE(230)
      EQUIVALENCE (START_C3,STA),(AZBOUNDS,AZB),(SECTOR_PNTR,SEC)
     +,(XP_AZBOUNDS,XPA),(XP_SECTOR_PNTR,XPS)
**A3133C3.inc
C** Named common for the elevation, range, and azimuth sector adaptation
C** tables
      INTEGER   START_C3,AZBOUNDS(360),SECTOR_PNTR(360)
      INTEGER*2 EL_AZ_RNG(51,360,4) ! 51 changes to 15 in other program
      INTEGER   XP_AZBOUNDS(360),XP_SECTOR_PNTR(360)
      INTEGER*2 XP_EL_AZ_RNG(51,360,4) ! 51 changes to 15 in other program
      COMMON /A3133C3/ START_C3, AZBOUNDS, SECTOR_PNTR, EL_AZ_RNG,
     $            XP_AZBOUNDS, XP_SECTOR_PNTR, XP_EL_AZ_RNG
C***********************************************************************
      WRITE(6,*)' Reading back hybrid sector file'
      READ(1,REC=1)INWORD ! all 720 bytes
      STA(1)=INWORD(1)
      STA(2)=INWORD(2)
      READ(1,REC=2)INWORD
      DO 10 J=1,360
   10 AZB(J)=INWORD(J)
      READ(1,REC=3)INWORD
      DO 12 J=1,360
   12 AZB(J+360)=INWORD(J)
      READ(1,REC=4)INWORD
      DO 14 J=1,360
   14 SEC(J)=INWORD(J)
      READ(1,REC=5)INWORD
      DO 16 J=1,360
   16 SEC(J+360)=INWORD(J)
      NREC=6
      DO 30 K=1,4
      DO 30 I=1,15 ! not the 51 originally in the named common
      READ(1,REC=NREC)INWORD
```

```
      DO 20 J=1,360
   20 EL_AZ_RNG(I,J,K)=INWORD(J)
   30 NREC=NREC+1
      READ(1,REC=NREC)INWORD
      DO 40 J=1,360
   40 XPA(J)=INWORD(J)
      NREC=NREC+1
      READ(1,REC=NREC)INWORD
      DO 42 J=1,360
   42 XPA(J+360)=INWORD(J)
      NREC=NREC+1
      READ(1,REC=NREC)INWORD
      DO 44 J=1,360
   44 XPS(J)=INWORD(J)
      NREC=NREC+1
      READ(1,REC=NREC)INWORD
      DO 46 J=1,360
   46 XPS(J+360)=INWORD(J)
      NREC=NREC+1
      DO 60 K=1,4
      DO 60 I=1,15 ! not the 51
      READ(1,REC=NREC)INWORD
      DO 50 J=1,360
   50 XP_EL_AZ_RNG(I,J,K)=INWORD(J)
   60 NREC=NREC+1
C   Write an image file showing the large arrays for hybrid sectors
      DO 70 K=1,4
      DO 70 J=1,360
      DO 61 I=1,230
   61 LINE(I)=1 ! clean the output line
      N=EL_AZ_RNG(1,J,K) ! count of range pairs to follow
      IP=1
      DO 64 NP=1,N ! pairs loop
      IP=IP+1
      IBEG=EL_AZ_RNG(IP,J,K) ! beginning range
      IP=IP+1
      IEND=EL_AZ_RNG(IP,J,K) ! ending range
      DO 62 I=IBEG,IEND ! fill loop
   62 LINE(I)=K+1 ! load with the elevation index + 1
   64 CONTINUE ! pairs loop
   70 WRITE(3)LINE ! image output
C   Write a hybrid sector file in polar coordinates
      DO 80 J=1,360 ! azimuth loop
      DO 71 I=1,230
   71 LINE(I)=0 ! clean the output line; all zeros should be replaced
      DO 76 K=1,4 ! tilts loop
      N=EL_AZ_RNG(1,J,K) ! count of range pairs to follow
      IP=1
      DO 74 NP=1,N ! pairs loop
      IP=IP+1
      IBEG=EL_AZ_RNG(IP,J,K) ! beginning range
      IP=IP+1
      IEND=EL_AZ_RNG(IP,J,K) ! ending range
      DO 72 I=IBEG,IEND ! fill loop
      IF(LINE(I).EQ.0)LINE(I)=K ! update with the elevation index
   72 CONTINUE ! fill loop
   74 CONTINUE ! pairs loop
   76 CONTINUE ! tilts loop
   80 WRITE(7)LINE ! hybrid sector output, end of azimuth loop
C   Write an image file showing the large arrays for expanded sectors
      DO 90 K=1,4
      DO 90 J=1,360
      DO 81 I=1,230
   81 LINE(I)=1 ! clean the output line
      N=XP_EL_AZ_RNG(1,J,K) ! count of range pairs to follow
```

```
      IP=1
      DO 84 NP=1,N ! pairs loop
      IP=IP+1
      IBEG=XP_EL_AZ_RNG(IP,J,K) ! beginning range
      IP=IP+1
      IEND=XP_EL_AZ_RNG(IP,J,K) ! ending range
      DO 82 I=IBEG,IEND ! fill loop
   82 LINE(I)=K+1 ! load with the elevation index
   84 CONTINUE ! pairs loop
   90 WRITE(3)LINE ! image output
      RETURN
      END
```

APPENDIX H

PROGRAM POLAR230

```
      PROGRAM POLAR230
C   Converts 30 arc-sec DEM elevations to polar coordinates from a radar.
C   Version of 8 April 1996 by Ed Holroyd
      COMMON /LATLON/ ALAT,ALON,AD,R,RLAT,RLON
      DOUBLE PRECISION ALAT,ALON,AD,R,RLAT,RLON
      INTEGER*2 INBUF6(721),INBUF7(841) ! for arrays 6 and 7 degrees wide
      EQUIVALENCE (INBUF6,INBUF7)
      INTEGER*2 LAND(360,230),OUTBUF(230)
      CHARACTER*30 INFILE
C     CHARACTER*1 MSG,UCY,LCY
C     DATA UCY,LCY/'Y','y'/
      DECD(ND,NM,SEC)=FLOAT(ND)+FLOAT(NM)/60.+SEC/3600.
      OPEN(UNIT=2,FILE='POLARDEM.DMP',STATUS='UNKNOWN',FORM='BINARY'
     +, BLOCKSIZE=460) ! I*2 elevations, meters, to 230 km
      OPEN(UNIT=4,FILE='PRINTER.DMP',STATUS='UNKNOWN')
      WRITE(6,*)' Type input DEM data file name'
      READ(5,1)INFILE
    1 FORMAT(A30)
      WRITE(6,*)' Type N latitude of top of array (whole degrees)'
      READ(5,*)LATT
      WRITE(6,*)' Type N latitude of bottom of array'
      READ(5,*)LATB
      WRITE(6,*)' Type W longitude of left of array (whole degrees)'
      READ(5,*)LONL
      WRITE(6,*)' Type W longitude of right of array'
      READ(5,*)LONR
      LATN=(LATT-LATB)*120+1 ! number of lines in array
      LONN=(LONL-LONR)*120+1 ! number of columns in array
      WRITE(4,*)' LATN,LONN=',LATN,LONN
      WRITE(6,*)' LATN,LONN=',LATN,LONN
      NBYTES=LONN*2 ! for I*2 elevation data, meters
      OPEN(UNIT=1,FILE=INFILE,STATUS='OLD',MODE='READ',FORM='BINARY'
     +,ACCESS='DIRECT',RECL=NBYTES,BLOCKSIZE=NBYTES)
      DO 2 I=1,360 ! degrees
      DO 2 J=1,230 ! km
    2 LAND(I,J)=-999 ! initialization to nearly 1 km below sea level
      WRITE(6,*)' Type radar N latitude,   dd,mm,ss.'
      READ(5,*)LATD,LATM,SLAT
      RLAT=DECD(LATD,LATM,SLAT)
      WRITE(6,*)' Type radar W longitude, ddd,mm,ss.'
      READ(5,*)LOND,LONM,SLON
      RLON=DECD(LOND,LONM,SLON)
    5 LINES=0
C   Begin sequential reading through entire file
   10 IF(LONN.EQ.721)READ(1,ERR=90)INBUF6
      IF(LONN.EQ.841)READ(1,ERR=90)INBUF7
      LINES=LINES+1
      ALAT=FLOAT(LATT)-FLOAT(LINES-1)/120.
      IF(MOD(LINES,12).EQ.1)WRITE(6,*)' Latitude now',ALAT
      DO 20 K=1,LONN ! scan along all longitudes, eastward
      ALON=FLOAT(LONL)-FLOAT(K-1)/120.
      CALL GTCIRC ! get R, AD radar coordinates for that location
      I=IDNINT(AD) ! rounded azimuth, degrees
      J=IDNINT(R)  ! rounded range, km
      IF(I.LT.1.OR.I.GT.360)I=360
      IF(J.LT.1)J=1
      IF(J.GT.230)GO TO 20 ! too far out, skip it
C   nearest neighbor assignment of elevations:
      IF(LONN.EQ.721)LAND(I,J)=INBUF6(K)
      IF(LONN.EQ.841)LAND(I,J)=INBUF7(K)
   20 CONTINUE ! eastward scan
      IF(LINES.LT.LATN)GO TO 10 ! for another line of data
C   search for data voids
      DO 30 J=1,230
      R=J ! km
```

```
      N=0
      DO 25 I=1,360
      AD=I ! degrees
      IF(LAND(I,J).EQ.-999.)THEN ! calculate nearest DEM pixel location
        CALL LALONG ! for latitude ALAT, longitude ALON of that range bin
        L=(FLOAT(LATT)-ALAT)*120.+1.5 ! rounded line number
        IF(LONN.EQ.721)READ(1,REC=L,ERR=90)INBUF6
        IF(LONN.EQ.841)READ(1,REC=L,ERR=90)INBUF7
        K=(FLOAT(LONL)-ALON)*120.+1.5 ! rounded column number
        LAND(I,J)=INBUF6(K) ! nearest neighbor assignment
        N=N+1
      ENDIF
   25 CONTINUE
      IF(N.GT.0)THEN
        WRITE(4,*)' Voids filled at range',J,N
        WRITE(6,*)' Voids filled at range',J,N
      ENDIF
   30 CONTINUE
      DO 80 I=1,360
      DO 70 J=1,230
   70 OUTBUF(J)=LAND(I,J)
      WRITE(2)OUTBUF
      IF(MOD(I,10).EQ.0)WRITE(6,*)' Output, azimuth=',I
   80 CONTINUE
   90 WRITE(6,*)' normal stop'
      STOP
      END
C
      SUBROUTINE GTCIRC
C   Converts latitude, longitude (fractional degrees) of ALAT, ALON into
C   range and azimuth from origin at RLAT, RLON (also fractional degrees).
C   Range R will be in km, azimuth AD in degrees.
      COMMON /LATLON/ ALAT,ALON,AD,R,RLAT,RLON
      DOUBLE PRECISION ALAT,ALON,AD,R,RLAT,RLON
      DOUBLE PRECISION OLAT,OLON,DLAT,DLON,DR,SCS,SC,X,DA,PR,C,S
      DATA PR/.017453293D+00/
C
      OLAT=PR*RLAT
      OLON=PR*RLON
      DLAT=PR*ALAT
      DLON=PR*ALON
      C=DSIN(OLAT)*DSIN(DLAT)+DCOS(OLAT)*DCOS(DLAT)*DCOS(DLON-OLON)
      IF(C.GT.1.)C=1.
      IF(C.LT.-1.)C=-1.
      DR=DACOS(C)
      SCS=DSIN(DLAT)-C*DSIN(OLAT)
      SC=DSIN(DR)*DCOS(OLAT)
      IF(SC.EQ.0.)GO TO 10
      X=SCS/SC
      IF(X.GT.1.)X=1.
      IF(X.LT.-1.)X=-1.
      A=DACOS(X)
      R=DR*6377.59D+00 ! or whatever earth radius (km) you want to use.
      IF(DLON.GT.OLON)A=-A
      AD=A/PR
      IF(AD.LT.0.)AD=AD+360.
      RETURN
   10 A=0.
      AD=0.
      RETURN
C
      ENTRY LALONG
C   Converts azimuth AD (degrees) and range R (km) from origin RLAT, RLON
C   (fractional degrees) into destination latitude ALAT and longitude ALON
C   (fractional degrees).
```

```
OLAT=PR*RLAT
OLON=PR*RLON
DR=R/6377.59D+00 ! from earth radius in km
IF(AD.LT.0.)AD=AD+360.
A=AD*PR
DA=A
IFLG=0
IF(AD.LE.180)IFLG=1
S=DCOS(DA)*DSIN(DR)*DCOS(OLAT)+DCOS(DR)*DSIN(OLAT)
IF(S.GT.1.)S=1.
IF(S.LT.-1.)S=-1.
DLAT=DASIN(S)
C=(DCOS(DR)-DSIN(OLAT)*DSIN(DLAT))/(DCOS(OLAT)*DCOS(DLAT))
IF(C.GT.1.)C=1.
IF(C.LT.-1.)C=-1.
DLON=DACOS(C)
IF(IFLG.EQ.1)DLON=-DLON
ALON=(OLON+DLON)/PR
ALAT=DLAT/PR
RETURN
END
```

APPENDIX I

PROGRAM FLIPBYTE

```
#include <stdio.h> /*input-output*/
/* FLIPBYTE.C    a program to interchange I*2 bytes */

/* DEFINE GLOBAL VARIABLES */
char inbyte[16384], outbyte[16384], infile[32], outfile[32];
int count=0, i=0;
long m=0, n=0, count1=0, count2=0;
FILE *fp1, *fp2;

main()
{
questions();   /* OPENING DIALOGUE */
while((count = fread(inbyte,1,16384,fp1)) != 0){
   i=0;
   while(i<count){
      outbyte[i+1]=inbyte[i];
      outbyte[i]=inbyte[i+1];
      n += 1;
      count2 += 1;
      if(n == m){
         printf("byte pairs flipped: %ld\r",count2);
         n=0;
         }
      i+=2;
      }       /* END OF WHILE FLIP LOOP */
   count1 = fwrite(outbyte,1,count,fp2)/2;
   }          /* END OF WHILE READ LOOP */
printf("Final byte pairs flipped: %ld\n",count2);
}             /* END OF PROGRAM */

questions()  /* FUNCTION FOR OPENING DIALOGUE */
{
printf("Type input file name: ");
scanf("%s",infile);
fp1=fopen(infile, "rb");
if(fp1 == NULL){
   printf("Couldn't open %s\n",infile);
   exit(1);
   }
printf("Type output file name: ");
scanf("%s",outfile);
fp2=fopen(outfile, "wb");
if(fp2 == NULL){
   printf("Couldn't open %s\n",outfile);
   exit(2);
   }
printf("How many bytes for message frequency: ");
scanf("%ld",&m);
}
```

APPENDIX J

Comparison Between OSF and Reclamation Routines

Changes in PPS and VAD routines that appear in the Snow Algorithm (May 1996 version).

A3133D_INIT_ADAPT

Removed (ADPT_PARAMS) from passed parameters. Replaced all (CONSTANT) pointers and used adaptable parameters list from a file with a direct definition (not read in) instead of using =ADPT_PARAMS(pointer). Removed the include lines that obtained those constant pointers. Rearranged A3133CA.inc and inserted it. Removed the REAL ADPT_PARAMS and DATA lines. Removed all UPDATE_TABLE lines. Removed excessive comments and debug statements.

These parameters have been changed from those in the file:

ZR_MLT_COEFF	=399.	(for equation (5))
	=318.	(for Cleveland, Ohio, equation (8))
	=155.	(for Denver, CO, equation (9))
ZR_PWR_COEF	=2.21	(for equation (5))
	=1.5	(for Cleveland, Ohio, equation (7))
	=1.6	(for Denver, CO, equation (8))

MINTHRFL=-10.
MIN_DBZ_AR_WGTD=-10.
MIN_DBZ=-10.

NOTE: The original PPS version of A3133D is acceptable when the program is rewritten by OSF. The five variables are more properly changed in the file for the adaptable parameters according to OSF style and specifications.

A3133F_FILL_PRECIP_TABLE

Removed debug statements and excessive comment lines. Rearranged A3133CA.inc and inserted it. Replaced all (CONSTANT) values with their numbers and eliminated the related include statements. Retained only the REAL MIN_DBZ, MAX_DBZ line of the type declarations and distributed the parameter values where needed.

IMPORTANT CHANGE: Changed from 10ths of mm/h to 100ths of mm/h for better resolution. See line 200 RATE_TABLE(I)=.... Also changed MIN_NONZERO_RATE to .005 mm/h for snow.

A317G2_VAD_PROC

None of the include files is needed. The /A317VA/ and /A317VD/ common blocks are not referenced. The debug statements and excessive comments have been removed. The REAL and INTEGER declarations are removed. Most of the PARAMETER definitions are distributed if needed.

Added the /STORE/, /TESTS/, and /WINDV/ named commons.

Substituted everywhere (old version --> new version):

- FIT_TESTS --> NFIT (arbitrarily defined as 3 elsewhere)
- CELV --> K (index equal to kft level)
- NPT --> NK (point count)
- HWD --> DK (horizontal wind direction)
- SHW --> SK (horizontal wind speed)
- RMS --> RMSK (r.m.s.)
- 7 --> 17 (DO loop index)
- MINPTS --> 15 (arbitrary 15-point requirement)

- Added DO 40 K=1,15 ! kft to cycle through first 15 kft levels above sea level.
- Removed from all called subroutine parameters: NAZIMS, NRADIALS, MISSING, AZM, VE and added K.
- In IF(CF3.NE.0....) changed AND to OR. Then changed HWD(CELV) to DKR in four statements except the last: HWD(CELV)=HWD(CELV)/DTR --> DK(K)=DKR/DTR.
- Removed references to WMODE, HT_SEA, and PFV.
- Removed /COS(ELV_RAD) because the speeds were already adjusted to horizontal in subroutine GETPROF.
- Added SK(K)=-99. to indicate bad or missing wind speeds.
- Removed call to A317L2.

A317H2_VAD_LSF

Removed NAZIMS, NRADIALS, MISSING, AZM, VE from call parameters and added K. Removed debug statements and excessive comments. Retained only DNPT and TWO_N as INTEGER. Removed REAL statements. Added /STORE/ named common. Removed DTR everywhere by using trig functions in degrees (°):

- SIN --> SIND (sine in degrees [°])
- COS --> COSD (cosine in degrees [°])
- AZM_RAD --> A (azimuth)
- VE(I) --> FLOAT(IV(K,J,I))/10. ! /10. to get back to m s^{-1} (wind speed)

- Added CF1=CF2=CF3=-9999. for missing values and removed the many =MISSING lines later.
- Added loop DO 12 J=1,360 ! azimuths to cycle through all azimuths.
- Moved SIN_AZ=SIND(A) and COS_AZ=COSD(A) forward and added SIN_2AZ=SIND(2.*A) and COS_2AZ=COSD(2.*A) for substitution later.
- Added ILIM lines to work through all data entries.
- Changed DO 10 I=1,NRADIALS to 1,ILIM.
- Changed IF(VE(I).GT.MISSING)THEN structure to IF(IV(K,J,I).EQ.-9999)GO TO 10.

The important complex calculations are not changed at all.

A317I2_VAD_RMS

Removed NAZIMS, NRADIALS, MISSING, AZM, VE from call parameters and added K. Removed debug statements and excessive comments. Removed all DTR lines. Removed all type declarations except INTEGER DNPT. Added /STORE/ named common.

- Added loop DO 12 J=1,360 ! azimuths to cycle through all azimuths.
- Added A=FLOAT(J)-.5 as a substitute for AZM(I).
- Added ILIM lines to work through all data entries.
- Changed DO 10 I=1,NRADIALS to 1,ILIM.
- Changed VE(I) to IV(K,J,I) and MISSING to -9999. Defined VE=FLOAT(IV(K,J,I))/10. to convert speed to m s^{-1}.
- Changed COS to COSD for angles in degrees (°).

A317J2_FIT_TEST

Removed NAZIMS, NRADIALS, MISSING, AZM, VE from call parameters and added K. Removed debug statements and excessive comments. Removed all DTR lines. Removed all type declarations. Added /STORE/ named common.

- Added loop DO 12 J=1,360 ! azimuths to cycle through all azimuths.
- Added A=FLOAT(J)-.5 as a substitute for AZM(I).
- Added ILIM lines to work through all data entries.
- Changed DO 10 I=1,NRADIALS to 1,ILIM.
- Changed VE(I) to IV(K,J,I) and MISSING to -9999. Defined VE=FLOAT(IV(K,J,I))/10. to convert speed to m s^{-1} and substitute for VE(I).
- Changed COS to COSD for angles in degrees (°).
- Shortened some IF statements.

A317K2_SYM_CHK

Removed debug and excessive comments. Removed type declarations except for LOGICAL SYM.

APPENDIX K

PROGRAM RDNX.C

RDNX.C

```
/* open and read NEXRAD file as unformated reads function on a DOS PC.  If it is
found that C2432 starts the 24 characters "Please open NEXRAD file ", then what
follows up to a null byte is taken to be the name of a file to open and read
both the first 24 bytes starting "ARCHIVE2." into C24, and the first whatever it
is of expected standard radial size 2432 bytes into C2432.  Other calls to here
will continue from where last left off and read in the next 2432 bytes (or of
whatever specified size *2 +8*2) into C2432.      */
#include<stdio.h>  /* standard I/O functions etc. */
#include<ctype.h>  /* standard character & string processing routines */
struct comCBUF
{ unsigned char C24[24]; unsigned char C2432[2432]; } ;
  extern struct comCBUF cbuf_;
/* for COMMON /CBUF/ C24, C2432 */
void rdnexrad ( Msg )
int *Msg;          /* set = bytes read in, or -1 if read error */
{ static FILE *pIn;          /* prt to input file to read */
  int I,J,L=0,M,   /* for various indices etc.  */
    IFEOF=0;       /* to remember if EoF hit    */
  static int IsOpnd=0;        /* to remember if file has been opened */
  static char PlzOpn[]={"Please open NEXRAD file ",
              ARCHIVE2[]={"ARCHIVE2.");
  for (I=0; I<24; I++) if (PlzOpn[I]!=cbuf_.C2432[I]) goto NxtRec;
/* caller has indicated a new file is to be opened so first close old (if any)*/
  if (IsOpnd) fclose(pIn);
  if ((pIn=fopen(cbuf_.C2432+24,"r"))==NULL)      /* if bad named file to read */
  { printf("unable to open %s\n", cbuf_.C2432+24);  exit(1);}  IsOpnd=1;
  for (I=0; I<24; I++) cbuf_.C24[I]=getc(pIn);   /* read file header*/
  for (I=0; I<8; I++) if (cbuf_.C24[I]!=ARCHIVE2[I])
  { printf("1st read from new NEXRAD file not ARCHIVE2."); exit(1);}
  *Msg=24;  if (cbuf_.C24[8]!='.') return;       /* apparently short header   */
/* assume first strange radial is standard size 2432 and read that much in    */
  for (I=0; I<2432; I++) cbuf_.C2432[I]=getc(pIn);  return;
NxtRec: ;          /* read start of next block to find the size to read in    */
  for (I=0; I<14; I++) cbuf_.C2432[I]=getc(pIn); /* read start of next block  */
  L=(cbuf_.C2432[12]<<8)+cbuf_.C2432[13];
  L=2*(L+8);       /* 2* to convert to bytes    */
/* normally 04B8=1208 as halfwords after this to end of block, + the 6 halfwds.
before this, this halfword itself, and the final halfword =6+1+1208+1=1216    */
  *Msg=14;              /* change to bytes read in, or -1 if read error  */
  for (I=14; I<L; I++)        /* now read in rest of this block      */
  { cbuf_.C2432[I]=getc(pIn);  if ((IFEOF=feof(pIn))!=0) {*Msg=-1; break; } }
  if (0<*Msg) *Msg=I; } /*


/* Sun Fortran examples
BYTE X        = char x       REAL X        = float x     LOGICAL X     = int x
CHARACTER X   = char x       INTEGER X     = int x       LOGICAL*2 X   = short x
CHARACTER*n X = char x[n]    INTEGER*2 X   = short x     LOGICAL*1 X   = char x
simref ( t, f, c, i, r, d, si, w, z)  LOGICAL*1 T, F
char   *t, *f, *c, *c8;               CHARACTER C
int    *i;                            DOUBLE PRECISION D
float  *r;                            INTEGER*2 SI
double *d;                            COMPLEX W
short  *si;                           DOUBLE COMPLEX Z
struct  complex ( float  r, i; ) *w;  EXTERNAL SimRef !$pragma C( SimRef )
struct dcomplex ( double r, i; ) *z;  CALL SimRef ( T, F, C, I, R, D, SI, W, Z )
{ *t=1;  *f=0;  *c='z';  *i=9;  *r=9.9;  *d=9.9;  *si=9;
  w->r=6;  w->i=7;  z->r=8;  z->i=9; }
compiled and linked using     cc -c Samp.c     then     f77 Samp.o Sampmain.f
strref ( c8, c8, L3, L8 )     after regular arguments come char. length values
char *c3, *c8;  int L3, L8;     CHARACTER C3*3, C8*8
{ static char s3[4] = "123";    EXTERNAL StrRef !$pragma C( StrRef )
  static char s4[9] = "12345";  CALL StrRef ( C3, C8 )  !! also passes L3,L8=3,8
  strncpy ( c3, s3, 11);
  strcpy ( c8, s8, 6 ); }
fixvec ( V, Sum, IJ )  INTEGER IV(9), IJ(3,2)
int *Sum;              DATA IV/1,2,3,4,5,6,7,8,9/
int V[9], IJ[2][3];    EXTERNAL FixVec !$pragma C( FixVec )
{ int i;               CALL FixVec ( IV, ISUM)
  *Sum=0;  for (i=0; i<9; i++) *Sum=*Sum+V[i]; }
for a function value r to be returned, in C end with    return ( r );
#include<stdio.h>
extern struct comtype                COMMON /LAB/ P, Q, R
{ float p; float q; float r; } ;
extern struct comtype lab_ ;
{ lab_.p = 1.;  lab_.q = 2.;  lab_.r = 3.; ...

other C misc.:  for (i=0, j=100; i<11; ++i, J-=10) ...
                if i==0 then (++i)==1 and i==1 while (i++)==0 and i==1        */
```

RADAR7.F

```
      PROGRAM RADAR7
C   to work with NEXRAD radar data in polar coordinates in an attempt to read
C   the Level II tape data (from disk), make adjustments for real terrain
C   effects, determine wind profile, select appropriate radar bins over and
C   upwind of every ground location for conversion into precipitation, which
C   is then summed in 1-hr, 3-hr, and storm totals by advected or vertical
C   schemes.
C   version of 29 April 1996, by Ed Holroyd
C 96/5 -Ra Aman: modified to run on Sun with C fundtion RDNEXRAD to read
C disk copies of tape files produced under Unix with commands like
C mt -f /dev/rmt/0mn fsf $nfskip        # load tape after file $nfskip
C dd if=/dev/rmt/0mn of=$fn ibs=31616  # copy next file from tape to $fn
      COMMON /CBUF/ C24, C2432
        CHARACTER C1(2432), C24*24, C2432*2432, C56*56, C80*80
        INTEGER*2 I2NBUF(1216)
        INTEGER   LABEL(608)
        EQUIVALENCE (I2NBUF,IINBUF,LABEL,C1,C2432,C80), (LABEL(7),C56)
      COMMON/CLOCK/IDATE,ITIME,KDATE,KTIME,MDATE,MTIME
      COMMON/FILES/NUMF,IMAGE,SN1HA,SN3HA,SNSTA,SN1HV,SN3HV,SNSTV
      CHARACTER*14 IMAGE
      CHARACTER*12 SN1HA,SN3HA,SNSTA,SN1HV,SN3HV,SNSTV
      COMMON/RADIAL/VAL(6),NUM(20),DBZ(0:459),VEL(-3:916),WID(-3:916)
     +,IFLAG,NDBZ(0:459),NVEL(-3:916),NWID(-3:916)
      INTEGER*2 NDBZ,NVEL,NWID
      COMMON/SITE/IDSITE,RH(3),MS(3) ! radar site elev. m, for DEN, ALB, CLE
      CHARACTER*3 MS
      COMMON/SPAN/LASTSCAN,INITSCAN,IENDSCAN ! times for volume scan
      CHARACTER*1 MSG,UCY,LCY, C48TAP*48
      LOGICAL LOGICL          !! to inquire if next NEXRAD file exixts
```

```
      DATA LASTSCAN,INITSCAN,IENDSCAN/3*0/
      DATA UCY,LCY/'Y','y'/
      DATA RH/1707.,581.,262./ ! elevations, meters, for DEN, ALB, CLE
      DATA MS/'DEN','ALB','CLE'/
      DATA SN1HA/'SN1A1234.TTL'/ ! 1-hr total precip, advected
      DATA SN3HA/'SN3A1234.TTL'/ ! 3-hr total precip, advected
      DATA SNSTA/'SNSA1234.TTL'/ ! storm total precip, advected
      DATA SN1HV/'SN1V1234.TTL'/ ! 1-hr total precip, vertical
      DATA SN3HV/'SN3V1234.TTL'/ ! 3-hr total precip, vertical
      DATA SNSTV/'SNSV1234.TTL'/ ! storm total precip, vertical
      OPEN(UNIT=4,FILE='PRINTER.DMP',STATUS='UNKNOWN')
      CALL A3133D ! INIT_ADAPT to load /A3133CA/, including RATE_TABLE
      WRITE(6,*)' Extracts dBz, vel over and upwind, gets precip. '
    1 WRITE(6,*)' Which radar location? 1=DEN, 2=ALB, 3=CLE'
      READ(5,*)IDSITE
      GO TO (2,4,6)IDSITE
C   Open appropriate site files: terrain, occult, hybrid files
C   The occult files, like KFTGT.OCC, are derived from the official files,
C   like KFTG.OCC, by flipping the bytes to IBM order and throwing away the
C   leading I*4 word and 31 trailing I*4 words (all zeros).  They are then
C   easily addressed by direct access.  Notice that no corrections are needed
C   for tilts 3 and 4.  Only KFTG needs a correction at the second tilt and
C   only for the radial with OCINDX=1287 beyond range 97 km (Mt. Evans).
C
C   All terrain files are derived from 30 arc-second files, resampled into
C   polar coordinates about the radars.  Units: meters above mean sea level.
    2 OPEN(UNIT=3,FILE='TERRADEN.INT',STATUS='OLD'
     +,FORM='UNFORMATTED',RECL=460,ACCESS='DIRECT',ERR=8)
      OPEN(UNIT=7,FILE='KFTGT.OCC',STATUS='OLD'
     +,FORM='UNFORMATTED',RECL=12,ACCESS='DIRECT',ERR=8)
      OPEN(UNIT=8,FILE='HYBRDDEN.INT',STATUS='OLD'
     +,FORM='UNFORMATTED',RECL=460,ACCESS='DIRECT',ERR=8)
      GO TO 10
    4 OPEN(UNIT=3,FILE='TERRAALB.INT',STATUS='OLD'
     +,FORM='UNFORMATTED',RECL=460,ACCESS='DIRECT',ERR=8)
      OPEN(UNIT=7,FILE='KENXT.OCC',STATUS='OLD'
     +,FORM='UNFORMATTED',RECL=12,ACCESS='DIRECT',ERR=8)
      OPEN(UNIT=8,FILE='HYBRDALB.INT',STATUS='OLD'
     +,FORM='UNFORMATTED',RECL=460,ACCESS='DIRECT',ERR=8)
      GO TO 10
    6 OPEN(UNIT=3,FILE='TERRACLE.INT',STATUS='OLD'
     +,FORM='UNFORMATTED',RECL=460,ACCESS='DIRECT',ERR=8)
      OPEN(UNIT=7,FILE='KCLET.OCC',STATUS='OLD'
     +,FORM='UNFORMATTED',RECL=12,ACCESS='DIRECT',ERR=8)
      OPEN(UNIT=8,FILE='HYBRDCLE.INT',STATUS='OLD'
     +,FORM='UNFORMATTED',RECL=460,ACCESS='DIRECT',ERR=8)
      GO TO 10
    8 WRITE(6,*)' Problem opening one of the site files'
      GO TO 1 ! to try again
   10 WRITE(6,'(45H Name initial NEXRAD file name, like c11078.2)')
      READ(5,'(A)') C56
      I = INDEX(C56,'.')
      READ(C56(I+1:I+3),'(BN,I3)') NUMF
C     OPEN(UNIT=1,FILE=C56,STATUS='OLD',MODE='READ',FORM='BINARY'
C    +,BLOCKSIZE=2432,ERR=80) ! raw NEXRAD data from Level II tape or disk
      NSTM=-1
C   get next volume scan (recycle point)
      C48TAP = C56 !! save initial tape file
   20 IFLAG=0
      I = INDEX(C56,' ')      !! C string to be ended by null byte
      C56(I:I) = CHAR(0)      !! prepare for C to open and read
      C80(1:24) = 'Please open NEXRAD file '      !! for Ra's RDNEXRAD
      WRITE(*,'(21H OPENing NEXRAD file ,A)') C56(1:I-1)
      CALL GETRADAR ! for expected header
      WRITE(6,'(I7.6,6H=KDATE,I7.6,6H=KTIME/21H Getting wind profile)')
     ,  KDATE, KTIME
      LASTSCAN=IENDSCAN ! remember the end time of the last volume scan
      INITSCAN=KTIME
      CALL GETPROF ! for wind profile; loads PPI files, gets last time
      KM=MOD(IDATE/100,100) ! month
      KD=MOD(IDATE,100) ! day
      KH=IENDSCAN/10000 ! hour; use end of scan time for file names
      K3=(KH/3)*3 ! 3-hour
      C80(1:24) = 'Please open NEXRAD file '
      C56 = C48TAP !! recall current tape file
      C56(I:I) = CHAR(0)      !! prepare for C to open and read
      IFLAG = 0
      CALL GETRADAR ! for expected header
      CALL GETARI ! to load azimuth-range array with 0.5 and 1.5 degree data
      CALL OCCULT ! apply occult corrections
      CALL NOSPIKES ! to remove lingering clutter and blemishes if possible
C   Only the storm total files remain the same while cycling through data files
      IF(NSTM.LT.0)THEN
        NSTM=KM*100+KD ! month and day, I4
        WRITE(SNSTA(5:8),'(I4.4)')NSTM
        WRITE(SNSTV(5:8),'(I4.4)')NSTM
        OPEN(UNIT=11,FILE=SNSTA,STATUS='UNKNOWN',FORM='UNFORMATTED'
     +  ,RECL=460,ACCESS='DIRECT',ERR=80) ! storm total precip, advected
        OPEN(UNIT=14,FILE=SNSTV,STATUS='UNKNOWN',FORM='UNFORMATTED'
     +  ,RECL=460,ACCESS='DIRECT',ERR=80) ! storm total precip, vertical
      ENDIF
C   Always reopen the 1-hr and 3-hr precip files, using new ones if necessary.
      N1HR=KD*100+KH ! day and hour, I4
      N3HR=KD*100+K3 ! day and 3-hour, I4
      WRITE(SN1HA(5:8),'(I4.4)')N1HR
      WRITE(SN1HV(5:8),'(I4.4)')N1HR
      WRITE(SN3HA(5:8),'(I4.4)')N3HR
      WRITE(SN3HV(5:8),'(I4.4)')N3HR
      OPEN(UNIT= 9,FILE=SN1HA,STATUS='UNKNOWN',FORM='UNFORMATTED'
     +,RECL=460,ACCESS='DIRECT',ERR=80) ! 1-hour total precip, advected
      OPEN(UNIT=12,FILE=SN1HV,STATUS='UNKNOWN',FORM='UNFORMATTED'
     +,RECL=460,ACCESS='DIRECT',ERR=80) ! 1-hour total precip, vertical
      OPEN(UNIT=10,FILE=SN3HA,STATUS='UNKNOWN',FORM='UNFORMATTED'
     +,RECL=460,ACCESS='DIRECT',ERR=80) ! 3-hour total precip, advected
      OPEN(UNIT=13,FILE=SN3HV,STATUS='UNKNOWN',FORM='UNFORMATTED'
     +,RECL=460,ACCESS='DIRECT',ERR=80) ! 3-hour total precip, vertical
      CALL PRECIP  ! calculate precip and integrate
C   Close files 9, 10, 12, 13
      CLOSE (9)
```

```
      CLOSE (10)
      CLOSE (12)
      CLOSE (13)
      NUMF=NUMF+1
      C56 = C48TAP            !! to automatically step to next file
      I = INDEX(C56,'.')
      IF (99.LT.NUMF) WRITE(C56(I+1:I+3),'(I3)') NUMF
      IF (NUMF.LT.100) WRITE(C56(I+1:I+3),'(I2,1H )') NUMF
      IF (NUMF.LT.10) WRITE(C56(I+1:I+3),'(I1,2H  )') NUMF
      INQUIRE(FILE=C56, EXIST=LOGICL, IOSTAT=IER)
      WRITE(*,'(9H INQUIRE(,A,L2,7H=EXIST))') C56(1:I+3), LOGICL
      IF (.NOT.LOGICL .OR. IER.NE.0) GOTO 80
      C48TAP = C56 !! save tape file
      GO TO 20
   80 WRITE(6,*)' Could not find ',C56
      WRITE(6,*)' Do you want a different data file?'
      READ(5,81)MSG
   81 FORMAT(A1)
      IF(MSG.NE.UCY.AND.MSG.NE.LCY)GO TO 90
      GO TO 10 ! for next data file
   90 WRITE(6,*)' Normal stop'
      STOP
      END

      SUBROUTINE A3133D
C   simulates A3133D__INIT_ADAPT
*.  MODULE FUNCTION:
*.    This routine initializes local copies of the Preprocessing
*.  RPG adaptation parameters
*.
*.  MODULES CALLED: A3133F__FILL_PRECIP_TABLE
*.
*.  PARAMETERS:            (*:  G = GLOBAL, C = COMMON, P = PASSED)
*.  C  ADPT_PARAMS      R*4  Array of RPG adaptation parameters
*.  C  IN_RNG_TILT      I*4  Inner range limit for BI-Scan maximization
*.                           procedure.
*.  C  MAX_AR_PCT_REDUC R*4  Maximum reduction in area between lowest two
*.                           elevations allowed to perform Bi-Scan
*.                           maximization.
*.  C  MAX_BISCAN_RNG   I*4  Maximum range for BI-Scan maximization procedure.
*.  C  MAX_THRESH_RATE  I*4  Maximum threshold rate used in isolated bin test.
*.  C  MIN_AR_LO_ECHO   R*4  Minimum area of precipitation echoes in the lowest
*.                           Echo Region.
*.  C  MIN_DBZ_AR_WGTD  R*4  Minimum reflectivity weighed area of precipitation
*.                           echoes in the lowest elevation needed to do area
*.                           reduction tilt test.
*.  C  MIN_THRESH_RATE  I*4  Minimum threshold rate used in isolated bin test.
*.  C  OUT_RNG_TILT     I*4  Outer range limit for BI-Scan maximization tests.
*.  C  RATE_TABLE       I*2  Look-up table for converting biased reflectivity
*.                           to precipitation rate.
*.  C  RATE_TLTEST      I*4  A low value of precipitation rate used as a
*.                           replacement and threshold value.
*.  C  ZR_MLT_COEFF     R*4  Multiplicative coefficient for converting
*.                           reflectivity to precipitation rate.
*.  C  ZR_PWR_COEFF     R*4  Power coefficient for converting reflectivity to
*.                           precipitation rate.
*.
*.  INTERNAL TABLES/WORK AREA:
*.  BDBZ          I*4  Biased DBZ variable
*.  MAX_DBZ       R*4  Maximum DBZ for converting to rate
*.  MIN_DBZ       R*4  Minimum DBZ for converting to rate
*.  UPDATE_TABLE L*4  Flag used to indicate whether to recal-
*.*****************************************************************
**A3133CA.inc
C** Local versions of adaptation parameters
      INTEGER        MIN_THRESH_RATE, MAX_THRESH_RATE, RATE_TLTEST
     +,IN_RNG_TILT,  OUT_RNG_TILT,    MAX_BISCAN_RNG
      REAL           MIN_AR_LO_ECHO,  MIN_DBZ_AR_WGTD, MAX_AR_PCT_REDUC
     +,ZR_PWR_COEFF, ZR_MLT_COEFF
      INTEGER*2 RATE_TABLE(0:256)
      COMMON /A3133CA/ MIN_THRESH_RATE, MAX_THRESH_RATE,RATE_TLTEST
     $,IN_RNG_TILT,    OUT_RNG_TILT,    MAX_BISCAN_RNG, MIN_AR_LO_ECHO
     $,MIN_DBZ_AR_WGTD,MAX_AR_PCT_REDUC,ZR_PWR_COEFF,   ZR_MLT_COEFF
     $,RATE_TABLE
C*****************************************************************
      INTEGER   BDBZ
      REAL      MIN_DBZ, MAX_DBZ
C*****************************************************************
C     WRITE(6,*)' Entering A3133D__INIT_ADAPT'
C** Initialize values and compute rate-table
C   Z=1780.*R**2.21, from Sekhon and Srivastava (1970)
C   Ze=0.224*Z, so Ze=399.*R**2.21, from Arlin and Paul Smith
      ZR_MLT_COEFF    = 399. ! Z-P multiplicative coeff. for snow
      ZR_PWR_COEFF    = 2.21 ! Z-P power coefficient for snow
C     ZR_MLT_COEFF    = 155. ! Z-P multiplicative coeff. for snow for DEN
C     ZR_PWR_COEFF    = 1.60 ! Z-P power coefficient for snow for DEN
C     ZR_MLT_COEFF    = 318. ! Z-P multiplicative coeff. for snow for CLE
C     ZR_PWR_COEFF    = 1.50 ! Z-P power coefficient for snow for CLE
      MINTHRFL        = -10. ! (1) isolated bin threshold (dBz) for snow
      MAXTHRFL        =  53. ! (2) outlier bin threshold (dBz)
      REFLECTLT       =   1. ! (3) tilt test threshold (dBz)
C** Set distance thresholds
      IN_RNG_TILT     =  40  ! (4) inner range tilt test (km)
      OUT_RNG_TILT    = 150  ! (5) outer range tilt test (km)
      MAX_BISCAN_RNG  = 230  ! (6) max range biscan (km)
C** Set biscan maximization test thresholds
      MIN_AR_LO_ECHO  = 600  ! (7) min area echo (km**2)
      MIN_DBZ_AR_WGTD = -10. ! (8) min refl. area averaged (dBz) for snow
      MAX_AR_PCT_REDUC=  50  ! (9) max area % reduced (%)
      MIN_DBZ=-10.
      MAX_DBZ= 53.
      CALL A3133F__FILL_PRECIP_TABLE(MIN_DBZ, MAX_DBZ)
C** Compute the rate thresholds by first converting dBz to biased dBz and
C** using the rate look-up table
      BDBZ=NINT(2.*(MINTHRFL +32.))+2
      MIN_THRESH_RATE=RATE_TABLE(BDBZ)
      IF (MIN_THRESH_RATE.LE.0) MIN_THRESH_RATE=1
      BDBZ=NINT(2.*(MAXTHRFL +32.))+2
      MAX_THRESH_RATE=RATE_TABLE(BDBZ)
```

```
      BDBZ=NINT(2.*(REFLECTLT+32.))+2
      RATE_TLTEST    =RATE_TABLE(BDBZ)
      IF (RATE_TLTEST.LE.0) RATE_TLTEST=1
      RETURN
      END

      SUBROUTINE A3133F__FILL_PRECIP_TABLE(MIN_DBZ, MAX_DBZ)
*.  MODULE FUNCTION:
*.    This routine initializes look-up table for the conversion of
*.  Biased DBZ to Precipitation rate in 100ths of mm/hr
*.  ---> changed by Ed Holroyd from 10ths to 100ths for better resolution
*.
*.  PARAMETERS:          (*:  G = GLOBAL, C = COMMON, P = PASSED)
*. P  MAX_DBZ       R*4  Maximum DBZ for converting to rate
*. P  MIN_DBZ       R*4  Minimum DBZ for converting to rate
*. C  RATE_TABLE    I*2  Look-up table for converting biased reflectivity
*.                       to precipitation rate.
*. C  ZR_MLT_COEFF  R*4  Multiplicative coefficient for converting
*.                       reflectivity to precipitation rate.
*. C  ZR_PWR_COEFF  R*4  Power coefficient for converting reflectivity to
*.                       precipitation rate.
*.
*.  INTERNAL TABLES/WORK AREA:
*.  DBZ              R*4         Variable for reflectivity in DBZ
*.  MAX_BREFL        I*4         Maximum biased Reflectivity for
*.  MIN_BREFL        I*4         Minimum biased Reflectivity for
*.  POWER_COEFF      R*4         Inverse of ZR_PWR_COEFF
*.  ZE               R*4         Equivalent reflectivity
*.*****************************************************************
**A3133CA.inc
C** Local versions of adaptation parameters
      INTEGER        MIN_THRESH_RATE, MAX_THRESH_RATE, RATE_TLTEST
     +,IN_RNG_TILT,  OUT_RNG_TILT,    MAX_BISCAN_RNG
      REAL           MIN_AR_LO_ECHO,  MIN_DBZ_AR_WGTD, MAX_AR_PCT_REDUC
     +,ZR_PWR_COEFF, ZR_MLT_COEFF
      INTEGER*2 RATE_TABLE(0:256)
      COMMON /A3133CA/ MIN_THRESH_RATE, MAX_THRESH_RATE,RATE_TLTEST
     $,IN_RNG_TILT,    OUT_RNG_TILT,    MAX_BISCAN_RNG, MIN_AR_LO_ECHO
     $,MIN_DBZ_AR_WGTD,MAX_AR_PCT_REDUC,ZR_PWR_COEFF,   ZR_MLT_COEFF
     $,RATE_TABLE
C*****************************************************************
      REAL      MIN_DBZ, MAX_DBZ
C*****************************************************************
C     WRITE(6,*)' Entering A3133F__FILL_PRECIP_TABLE'
C** initialize
      POWER_COEFF=1./ZR_PWR_COEFF
C** convert the minimum resolvable rate to dBz
      DBZ=10.*ALOG10(ZR_MLT_COEFF*(.005**ZR_PWR_COEFF))
C** use the maximum of the lowest resolvable reflectivity and the minimum dBz
C** from the adaptation parameter
      IF (DBZ.LT.MIN_DBZ) DBZ=MIN_DBZ
C** compute the minimum biased dBz index for which precip rate must be computed
      MIN_BREFL=NINT(2.*(DBZ+32.))+2
C** check to make sure adaptation max reflectivity doesn't correspond to a rate
C** greater than what an I*2 word can handle
      DBZ=10.*ALOG10(ZR_MLT_COEFF*(3200.**ZR_PWR_COEFF))
      IF (DBZ.GT.MAX_DBZ) DBZ=MAX_DBZ
C** compute the maximum biased dBz index for which precip rate must be computed
      MAX_BREFL=2.*(DBZ+32.)+2
C** set all table values = 0 for dBz < MIN_BREFL
      DO 100 I=0,MIN_BREFL-1
  100   RATE_TABLE(I)=0
C** Do for all dBz values that need a rate computed
      DO 200 I=MIN_BREFL, MAX_BREFL
        DBZ=FLOAT(I-2)/2. - 32. ! convert biased reflectivity to dBz
        ZE=10.**(DBZ/10.) ! convert dBz to equivalent reflectivity (Ze)
C** compute precip rate from Ze and scale by the scaling factor
  200   RATE_TABLE(I)=NINT(((ZE/ZR_MLT_COEFF)**POWER_COEFF)*100.)
C** Do for all dBz greater than the maximum biased dBz and set equal to the
C** maximum precip rate at the maximum reflectivity
      DO 300 I=MAX_BREFL+1, 255
  300   RATE_TABLE(I)=RATE_TABLE(MAX_BREFL)
      RATE_TABLE(256)=-9999.
C** Print out table during testing
C     WRITE(4,40)(I,I=0,9)
C  40 FORMAT(' BDBZ: ',10I7)
C     DO 50 J=0,24
C     II=10*J
C  50 WRITE(4,60)II,(RATE_TABLE(II+I),I=0,9)
C  60 FORMAT(11I7)
C     WRITE(4,*)'        mm/hr * 100, precipitation rate for snow'
      RETURN
      END

      SUBROUTINE GETRADAR    !! modified to use C function to read
C unpack Level II radar data from NEXRAD radars
      COMMON/CLOCK/ IDATE, ITIME, KDATE, KTIME, MDATE, MTIME
      COMMON/RADIAL/ VAL(6), NUM(20), DBZ(460), VEL(920), WID(920),
     ,  IFLAG, NDBZ(460), NVEL(920), NWID(920)
        INTEGER*2 NDBZ, NVEL, NWID
C above: index 1 properly is at 0km, below, if wanting index 1=1km bin 2
C     COMMON/RADIAL/ VAL(6), NUM(20), DBZ(0:463), VEL(920), WID(916),
C     ,  IFLAG, NDBZ(0:463), NVEL(920), NWID(916)
C       INTEGER*2 NDBZ, NVEL, NWID
      COMMON /CBUF/ C24, C2432
        CHARACTER C1(2432), C24*24, C2432*2432
        INTEGER*2 I2NBUF(1216)          !! had an INT*1 IINBUF for Ed H.
        INTEGER   LABEL(608), LBL24(6)
        EQUIVALENCE (I2NBUF,IINBUF,LABEL,C1,C2432), (LBL24,C24)
      EXTERNAL RDNEXRAD !$pragma C( RDNEXRAD )
      NCLOCK(NSEC) = NSEC+(NSEC/60)*40+(NSEC/3600)*4000
C   IDATE,ITIME: for radial
C   KDATE,KTIME: for processing
C   MDATE,MTIME: for message
C   VAL(1): unambibuous range, km
C      (2): center azimuth angle, degrees
C      (3): center elevation angle, degrees
C      (4): Nyquist velocity, m/s
C      (5): atmospheric attenuation factor, dB/km
```

```
C       (6): threshold parameter, watts
C    NUM( 1): message size, 2-byte words
C       ( 2): message type: 1=digital radar data, codes to 14
C       ( 3): I.D. sequence number 0 to 7FFF
C       ( 4): number of message segments
C       ( 5): message segment number
C       ( 6): radial number within the elevation scan
C       ( 7): radial status: 0=start of new elevation
C                            1=intermediate radial
C                            2=end of elevation
C                            3=start of volume scan
C                            4=end of volume scan
C       ( 8): RDA elevation number within the volume scan
C       ( 9): range to first gate of reflectivity data, meters
C       (10): range to first gate of doppler data, meters
C       (11): reflectivity data gate size, meters
C       (12): doppler data gate size, meters
C       (13): number of reflectivity gates
C       (14): number of velocity gates
C       (15): sector number within cut
C       (16): reflectivity data pointer, byte #
C       (17): velocity data pointer, byte #
C       (18): spectrum width pointer, byte #
C       (19): doppler velocity resolution: 2 = 0.5 m/s, 4 = 1.0 m/s
C       (20): volume coverage pattern: 11, 21, 31, 32
C    Velocity bins 1,2,3,4 look at same volume as bin 1 of reflectivity,
C    which is centered at 0 km from radar.
C    DBZ(1-460): reflectivity values, dBz
C    VEL(1-920): velocity values, m/s
C    WID(1-920): spectral width values, m/s
      IFLAG = -1    !! record type not yet defined
C read in buffer, all 2432 bytes
   10 CONTINUE
C     READ(1, ERR=90) IINBUF !! instead use Ra's RDNEXRAD on Suns
C     READ(1,'(A)', IOSTAT=IER) C2432  !! on Sun use C function RDNEXRAD
      C24(1:1) = 'a'          !! old "ARCHIVE2" to "aRCHIVE2"
      CALL RDNEXRAD(IER)      !! IER=bytes put in C2423, else -1 if error
      IF (IER.LT.0) THEN      !! if any read error
        WRITE(6,'(33H GETRADAR: read error in rdnexrad,I8)') IER
        GOTO 90
      ENDIF
C check for possible header
      IF (C24(1:9).EQ.'ARCHIVE2.') THEN
        IFLAG = 1  !! header found
        CALL DATEJ(LBL24(4), KDATE)
        KTIME = NCLOCK(LBL24(5)/1000)
        RETURN
      ENDIF
      IFLAG = 2               !! should have a radial
      NUM(1) = I2NBUF(7)      !! expect #04B8 = 1208
      NUM(2) = I2NBUF(8)      !! expect 1 for regular data
      IF (I2NBUF(8).NE.1) GOTO 10    !! for another record; reject this one
      NUM(3) = I2NBUF(9)      !! I.D. sequence = 0 to #7FFF
      JDA = I2NBUF(10)        !! days since 1970/1/00 from INT*2 to INT*4
      CALL DATEJ(JDA, MDATE) !! transform into YrMoDa date
      MTIME = NCLOCK(LABEL(6)/1000)        !! generation time for messages
      NUM(4) = I2NBUF(13)     !! number of message segments
      NUM(5) = I2NBUF(14)     !! message segment number
      JDA =  I2NBUF(17)
      CALL DATEJ(JDA,IDATE)
      ITIME = NCLOCK(LABEL(8)/1000) ! collection time for radial
      VAL( 1)=FLOAT(I2NBUF(18))/10. ! unambiguous range (scaled: Value/10. = km)
      VAL( 2)=FLOAT(I2NBUF(19))/8.*(45./1024.) ! degrees, azimuth angle
      IF(VAL(2).LT.0.)VAL(2)=VAL(2)+360.
      NUM( 6)=I2NBUF(20) ! radial number with the elevation scan
      NUM( 7)=I2NBUF(21) ! radial status, 0 to 4: 0=start of new elev.
C 1 = intermediate radial        2 = end of elevation
C 3 = beginning of volume scan   4 = end of volume scan
      VAL( 3)=FLOAT(I2NBUF(22))/8.*(45./1024.) ! elev.1 deg. from horiz.
      NUM( 8)=I2NBUF(23) ! RDA elevation number within the volume scan
      NUM( 9)=I2NBUF(24) ! range to first reflectivity data, meters
      NUM(10)=I2NBUF(25) ! range to first doppler data, meters
      NUM(11)=I2NBUF(26) ! reflectivity data gate size, meters
      NUM(12)=I2NBUF(27) ! doppler data gate size, meters
      NUM(13)=I2NBUF(28) ! number of reflectivity gates
      NUM(14)=I2NBUF(29) ! number of doppler gates
      NUM(15)=I2NBUF(30) ! sector number within cut
      ! I2NBUF(31-32) R#4 system gain calibration constant (DB biased)
      NUM(16)=I2NBUF(33) ! pointer to reflectivity data, bytes
      NUM(17)=I2NBUF(34) ! pointer to doppler velocity data, bytes
      NUM(18)=I2NBUF(35) ! pointer to doppler spectrum width data, bytes
      NUM(19)=I2NBUF(36) ! doppler velocity resolution: in .25 m/s units
      NUM(20)=I2NBUF(37) ! volume coverage pattern: 11, 21, 31, 32
C I2NBUF(38-41) unused; I2NBUF(42-44) RDA playback pointers
      VAL( 4)=FLOAT(I2NBUF(45))/100. ! Nyquist velocity: value/100 = m/s
      VAL( 5)=FLOAT(I2NBUF(46))/1000.! atmospheric attenuation factor, dB/km
      VAL( 6)=FLOAT(I2NBUF(47))/10.  ! threshold parameter, value/10 = watts
C I2NBUF(48-64) unused
      IF (NUM(16).GE.100) THEN ! calculate start and end of reflectivity data
        IRS = 29+NUM(16)
        IRE = 488+NUM(16)
        IF (NUM(17).GE.100) THEN
          IDS = 29+NUM(17)
          IF (IDS.LT.IRE) IRE = IDS-1
        ENDIF
        DO 40 I=IRS,IRE ! load up to 460 values
          J = ICHAR(C1(I))
          NDBZ(I-IRS+1) = J
   40   DBZ(I-IRS+1) = (FLOAT(J-2)/2.)-32.
C note: IINBUF=0 means below signal to noise threshold (too weak)
        JR = IRE-IRS+2
        IF (JR.LT.460) THEN
          DO 41 I=JR,460
   41     DBZ(I) = -33.
        ENDIF
      ENDIF
      IF (NUM(17).GE.100) THEN
        IDS = 29+NUM(17)
        IDE = 948+NUM(17)
```

```
        IF (NUM(18).GE.100) THEN
          IWS = 29+NUM(18)
          IF (IWS.LT.IDE) IDE = IWS-1
        ENDIF
        IF (NUM(19).EQ.2) THEN
          DO 42 I=IDS,IDE ! load up to 920 values
            J = ICHAR(C1(I))
            NVEL(I-IDS+1) = J
C m/s for 0.5 m/s resolution
   42     VEL(I-IDS+1) = (FLOAT(J-2)/2.)-63.5
        ENDIF
        IF(NUM(19).EQ.4)THEN
          DO 44 I=IDS,IDE ! load up to 920 values
            J = ICHAR(C1(I))
            NVEL(I-IDS+1) = J
C m/s for 1 m/s resolution
   44     VEL(I-IDS+1) = FLOAT(J-2)-127.
        ENDIF
        JD = IDE-IDS+2
        IF (JD.LT.920) THEN
          DO 45 I=JD,920
   45     VEL(I) = 0.
        ENDIF
      ENDIF
C note:   IINBUF=1 means range is ambiguous for doppler data
      IF (NUM(18).GE.100) THEN
        IWS = 29+NUM(18)
        IWE = 948+NUM(18)
        IF (JD.LT.920) IWE = IWS+JD-2
        DO 46 I=IWS,IWE ! load up to 920 values
          J = ICHAR(C1(I))
          NWID(I-IWS+1) = J
   46   WID(I-IWS+1) = (FLOAT(J-2)/2.)-63.5 ! m/s
        JW = IWE-IWS+2
        IF (JW.LT.920) THEN
          DO 47 I=JW,920
   47     WID(I) = -64.5
        ENDIF
      ENDIF
C bytes 2429 to 2432 (4 of them) are frame check sequence numbers
      RETURN
C error (EOF) found
   90 IFLAG=-9
      RETURN
      END

      SUBROUTINE GETPROF
C   Calculates vertical wind profile using sine wave fitting from VAD program
C   Uses data from radials closest to nominated range.  Note that for a 10 km
C   nominated range and clear air mode, all kft levels except the bottom two
C   will use data from the 4.5 degree tilt.
      COMMON/CLOCK/IDATE,ITIME,KDATE,KTIME,MDATE,MTIME
      COMMON/RADIAL/VAL(6),NUM(20),DBZ(0:459),VEL(-3:916),WID(-3:916)
     +,IFLAG,NDBZ(0:459),NVEL(-3:916),NWID(-3:916)
      INTEGER*2 NDBZ,NVEL,NWID
      COMMON/SITE/IDSITE,RH(3),MS(3) ! radar site elev. m, for DEN, ALB, CLE
      CHARACTER*3 MS
      COMMON/SITEDATA/TERRA(230),OCCUL(6),HYBRD(230)
      INTEGER*2 TERRA,OCCUL,HYBRD ! direct access radial records
      COMMON/STORE/LIST(108225) ! general storage identified by equivalence
      INTEGER*2 IV(15,360,20),IN(15,360),MINDSL(15,360),VAD_RNG
      EQUIVALENCE (LIST(1),IV(1,1,1)),(LIST(54001),IN(1,1))
     +,(LIST(56701),MINDSL(1,1)),(LIST(59401),VAD_RNG)
      COMMON/SPAN/LASTSCAN,INITSCAN,IENDSCAN ! times for volume scan
      COMMON/TESTS/NFIT,TH_RMS,TSMY
      COMMON/WINDV/K,DK(15),SK(15),NK(15),RMSK(15)
      NREC=0
      VAD_RNG=10
C     WRITE(6,*)' Type nominal range for wind eval.: 10, 20, 30,... km'
C     READ(5,*)VAD_RNG
      WRITE(4,1)VAD_RNG
    1 FORMAT(' This wind profile is calculated with VAD_RNG=',I3,' km')
      NFIT=3
C     WRITE(6,*)' How many times through fit tests?  3?'
C     READ(5,*)NFIT
      TH_RMS=100.
C     WRITE(6,*)' Suggest a value for TH-RMS'
C     WRITE(6,*)' Maximum value RMS can be and still be accepted, 100.'
C     READ(5,*)TH_RMS
      TSMY=10.
C     WRITE(6,*)' Suggest a value for TSMY'
C     WRITE(6,*)' Maximum value CF1 can be and still be accepted, 0-10.'
C     READ(5,*)TSMY
      DO 3 J=1,360
      DO 3 K=1,15
      MINDSL(K,J)=999
    3 IN(K,J)=0
      WRITE(6,*)' arrays zeroed   azim    elev'
C   Load arrays until end of volume scan, searching for kft elevations
C   at range closest to nominated range.
    8 IF(IFLAG.LT.0)GO TO 45 ! else infinite loop
      CALL GETRADAR ! get a single radial
      NREC=NREC+1
      IENDSCAN=ITIME ! keep track of latest time
      IF(NREC.EQ.1)WRITE(4,9)IDATE,ITIME,NREC
    9 FORMAT(2I7,' Record=',I4)
      IF(NUM(8).EQ.1.OR.NUM(8).EQ.3)GO TO 8 ! don't want reflectivity data
      IF(NUM(17).EQ.0)GO TO 8 ! no velocity data available
      IF(NUM(7).EQ.2)WRITE(6,23)IDATE,ITIME,NUM(8),VAL(3) ! end of elevation
   23 FORMAT(3I7,F8.4,' end of elevation')
      IF(MOD(INT(VAL(2)),60).EQ.0)THEN
        WRITE(6,24)IDATE,ITIME,VAL(2),VAL(3),NUM(19),NUM(20)
      ENDIF
   24 FORMAT(2I7,2F8.3,' res*4=',I4,' mode=',I4)
C   calculate coordinates, load wind array for a particular elevation
      NA=NINT(VAL(2)) ! azimuth angle, nearest integer: centered
      IF(NA.LT.1)NA=NA+360
      IF(NA.GT.360)NA=NA-360
      B=SIND(VAL(3)) ! sine of elevation angle
```

```
      COSE=COSD(VAL(3)) ! cosine of elevation angle
      A=COSE*COSE*5.8869E-5
      DV=1000/NUM(12) ! doppler bins per km
C   Calculate ranges to each kft level
      DO 30 K=1,15 ! height increments, kft msl
      H=FLOAT(K)*.3048 ! convert kft level to km
      C=RH(IDSITE)/1000.-H ! BEAMHT at kft levels, m.s.l., but in km units
      IF(C.GT.0.)GO TO 30 ! below radar level
      SQ=B*B-4.*A*C
      IF(SQ.GT.0.)THEN ! calculate range at which beam is at that altitude
        R=(-B+SQRT(SQ))/(2.*A) ! km; quadratic solution of BEAMHT equation
      ELSE
        R=-1.
      ENDIF
      IF(R.LT.0..OR.R.GT.125.)GO TO 30 ! km; too far out
      NDSL=IABS(NINT(R)-VAD_RNG) ! departure from nominal range
      IF(NDSL.LT.MINDSL(K,NA))THEN ! found a closer range
        IN(K,NA)=0 ! throw away any other data from farther away
        MINDSL(K,NA)=NDSL ! update minimum departure from nominal range
      ENDIF
      IF(NDSL.EQ.MINDSL(K,NA))THEN ! remember the new velocity values
        IRM=NINT((R-.5)*DV) ! rounded doppler range bin a half km short of target
        IRP=NINT((R+.5)*DV) ! rounded doppler range bin a half km beyond target
        DO 28 IR=IRM,IRP
        IF(NVEL(IR).LT.2)GO TO 28 ! no good velocity data
        N=IN(K,NA)+1 ! use next available bin
        IF(N.GT.20)GO TO 30 ! disregard data if array is full
C   save the horizontal speeds by altitude and azimuth; *10. to preserve tenths
        IV(K,NA,N)=NINT(10.*VEL(IR)/COSE)
        IN(K,NA)=N ! count of entries
   28   CONTINUE ! loop for any data within 0.5 km of nominal range
      ENDIF
   30 CONTINUE ! end of kft loop: K=1,15
      IF(NUM(7).GE.4)IFLAG=-8 ! end of volume scan
      GO TO 8 ! for next radial
C   Volume scan complete.  Calculate winds, throwing out poor data.
   45 CALL A317G2__VAD_PROC ! to use VAD sine curve fitting routine
C   write out table of winds
      WRITE(4,*)
     +' kft   wind Azimuths, Speeds m/s, Number of measurements, RMS'
      WRITE(6,*)
     +' kft   wind Azimuths, Speeds m/s, Number of measurements, RMS'
      DO 70 K=15,1,-1
      IF(NK(K).EQ.0)GO TO 70 ! skip levels with no measurements
      H=FLOAT(K) ! altitude, kft
      WRITE(4,65)H,DK(K),SK(K),NK(K),RMSK(K)
      WRITE(6,65)H,DK(K),SK(K),NK(K),RMSK(K)
   65 FORMAT(F5.0,' AZ:',F5.0,' SP:',F5.1,' N:',I6,' RMS:',F8.4)
   70 CONTINUE
      WRITE(4,*) ! nothing, so end of list may be detected
      WRITE(4,9)IDATE,ITIME,NREC
      RETURN
      END

      SUBROUTINE A317G2__VAD_PROC
*.  MODULE FUNCTION:
*.     This module is the main processing routine of the vad algorithm. It
*.     receives data via the calling sequence and the common areas. It calls
*.     the various routines to perform the various tasks required to generate
*.     the wind speed, wind direction.
*.
*.  PARAMETERS:          (*:  G = GLOBAL, C = COMMON, P = PASSED)
*.  G  NFIT I*4  Number of times through the loop which performs the
*.                   least squares fit and then removes low magnitude
*.                   outlier data points.
*.  G  TH_RMS    R*4  Maximum value RMS can be and still be accepted as a good
*.                   wind estimate.
*.  G  TSMY      R*4  ADAPTABLE PARAMETER: The maximum value CF1 can be and
*.                   still be accepted as a good wind estimate, in m/s.
*.                   Rng:[0,10].
*.
*.  INTERNAL TABLES/WORK AREA:
*.  CF1  R*4  Fourier coefficient (zeroth harmonic). Rng:[-100,+100]
*.  CF2  R*4  Fourier coefficient (real part of first harmonic).
*.            Rng:[-100,+100]
*.  CF3  R*4  Fourier coefficient (imaginary part of first harmonic.
*.            Rng:[-100,+100]
*.  DTR  R*4  Degrees to radians converstion factor (0.017453...).
*.  PI   R*4  Mathematical constant,(3.1415.....)
*.  SYM  L*4  A logical variable indicating that the current least squares
*.            fitted curve is symmetric about the zero velocity line.
*.            "TRUE = SYMMETRIC"
*.
*.  ASSUMPTIONS/RESTRICTIONS:
*.          Assume that the fit test procedure does not need
*.          to be performed the last time through the
*.          DO FOR ALL FIT TESTS loop since the velocity data
*.          is not used anymore after there.
*.  MISC:  This software uses the MKS units system.
*.          If not enough data exits to perform the vad least
*.          squares fitting the output field is filled with the
*.          missing data flag.
*.*****************************************************************
C     include 'a3cd70c7.inc'
C     include 'a3cd97.inc'
C     include 'a317vi.inc'
C     include 'a317vp.inc'
C     include 'a317vd.inc'
C     include 'a317va.inc'
C
      LOGICAL SYM
      PARAMETER (PI=3.14159265,DTR=0.01745329)
      COMMON/STORE/LIST(108225) ! general storage identified by equivalence
      INTEGER*2 IV(15,360,20),IN(15,360)
      EQUIVALENCE (LIST(1),IV(1,1,1)),(LIST(54001),IN(1,1))
      COMMON/TESTS/NFIT,TH_RMS,TSMY
      COMMON/WINDV/K,DK(15),SK(15),NK(15),RMSK(15)
C*****************************************************************
      DO 40 K=1,15 ! kft
```

```
      DO 10 I=1,NFIT ! for all fit tests
C* compute coefficients of least squares fitted curve, in m/s.
        CALL A317H2__VAD_LSF(K,NK(K),CF1,CF2,CF3)
C* if no data exists for vad analysis then CF1, CF2, and CF3 will
C* all be equal to missing, and NK(K) will equal 0.
C* then exit module, (i.e. go to 17)
        IF(NK(K).LT.15) GO TO 17 ! arbitrary 15 point requirement
C* compute horizontal wind direction, in degrees (& check arguments).
        IF (CF3.NE.0..OR.CF2.NE.0.) THEN
          DKR=PI-ATAN2(CF3,CF2)
        ELSE
          DKR=0.0
        ENDIF
C* check if wind direction computes to be negative, then convert to degrees.
        IF(DKR.LT.0.)DKR=DKR+2.*PI
        DK(K)=DKR/DTR
C* compute the square root of the mean squared deviations between the
C* velocity data points and the least squares fitted curve, in m/s.
        CALL A317I2__VAD_RMS(K,DK(K),CF1,CF2,CF3,RMSK(K))
C* if not the last time through this loop, perform fit test.
C* if the last time through this loop, skip fit test since it will
C* only perform un-needed calculations.
        IF(I.LT.NFIT) THEN
C* perform fit test to remove velocity data which lies more than
C* 1 rms away from the least squares fitted curve and toward
C* the zero velocity line, then go back up and perform least squares
C* fitting again.
          CALL A317J2__FIT_TEST(K,CF1,CF2,CF3,DK(K),RMSK(K))
        END IF
   10 CONTINUE
C* compute symmetry of the least squares fitted curve.
      CALL A317K2__SYM_CHK(CF1,CF2,CF3,TSMY,SYM)
C* only continue with calculations if the rms is less than the
C* threshold and the fit is symmetric.
C     WRITE(4,15)RMSK(K),TH_RMS
C  15 FORMAT(' Testing if RMSK(K)<TH_RMS:',F10.4,'<',F10.4)
      IF(RMSK(K).LT.TH_RMS.AND.SYM) THEN
C* if the atmosphere is void of precipitation within the vad
C* analysis region, the precip. fall speed is zero.
C* compute horizontal wind speed in m/s.
        SK(K)=SQRT(CF2**2+CF3**2)
      ELSE
        SK(K)=-99.
      END IF
   17 CONTINUE
   40 CONTINUE ! end of kft loop
      RETURN
      END

      SUBROUTINE A317H2__VAD_LSF(K,DNPT,CF1,CF2,CF3)
*.  MODULE FUNCTION:
*.     This module least squares fits a sine-wave curve to velocity
*.     data points. Data used to perform the fitting is in the form
*.     of Doppler velocity v.s. azimuth angle for a specific slant
*.     range.
*.
*.  PARAMETERS:            (*:  G = GLOBAL, C = COMMON, P = PASSED)
*.  P   CF1      R*4  Fourier coefficient (zeroth harmonic. Rng:[-100,+100]
*.  P   CF2      R*4  Fourier coefficient (real part of first harmonic).
*.                    Rng:[-100,+100]
*.  P   CF3      R*4  Fourier coefficient (imaginary part of first harmonic.
*.                    Rng:[-100,+100]
*.  P   DNPT     I*4  Number of data points used to perform the least
*.                    squares fitting, Dummy variable.
*.
*.  INTERNAL TABLES/WORK AREA:
*.    CCJ_Q4     C*8          Intermediate value representing the conjugate of
*.                            Q4 (used to compute the least squares fitted
*.                            harmonic coefficients.
*.    COS_AZ     R*4          The cosine of the azimuth angle for a particular
*.                            radial. Rng:[-1,1]
*.    INT_COEFF  C*8          Used to compute the fourier coefficient.
*.    Q0         C*8          Intermediate used to compute the least squares
*.                            fitted harmonic coefficients.
*.    Q1         C*8          Intermediate used to compute the least squares
*.                            fitted harmonic coefficients.
*.    Q2         C*8          Intermediate variable used to compute the least
*.                            squares fitted harmonic coefficients.
*.    Q3         C*8          Intermediate variable used to compute the least
*.                            squares fitted harmonic coefficients.
*.    Q4         C*8          Intermediate variable used to compute the least
*.                            squares fitted harmonic coefficients.
*.    Q5         C*8          Intermediate variable used to compute the least
*.                            squares fitted harmonic coefficients.
*.    QQ         C*8          Intermediate variable to reduce some calculations
*.                            and eliminate complex zero divide and floating
*.                            point overflows.
*.    QQ_INT     C*8          Intermediate variable to reduct some calculations
*.                            and eliminate complex zero divide and floating
*.                            point overflows.
*.    SIN_AZ     R*4          The sine of the azimuth angle for a particular
*.                            radial. Rng:[-1,1]
*.    SUM_Q0R    R*4          The variables
*.    SUM_Q3I    R*4          SUM_Q0R -> SUM_Q5R
*.    SUM_Q3R    R*4          are summation
*.    SUM_Q4I    R*4          variables used to
*.    SUM_Q4R    R*4          compute the real and
*.    SUM_Q5I    R*4          imaginary parts of the
*.    SUM_Q5R    R*4          variables Q0 -> Q6.
*.    TWO_N      I*4          Two times the number of data points used to
*.                            perform the least squares fitting. Rng:[0,2*400]
*.
*.  MISC:  This software uses the MKS units system.
*.            If not enough data exist to perform the least squares
*.            fitting, CF1 CF2 and CF3 are returned set to missing
*.*****************************************************************
      COMMON/STORE/LIST(108225) ! general storage identified by equivalence
      INTEGER*2 IV(15,360,20),IN(15,360)
      EQUIVALENCE (LIST(1),IV(1,1,1)),(LIST(54001),IN(1,1))
```

```
      INTEGER DNPT,TWO_N
      COMPLEX Q0,Q5,Q4,Q3,Q2,Q1,CCJ_Q4,INT_COEFF,QQ,QQ_INT
C*****************************************************************
C* zero out variables used for summations.
      DNPT=0
      CF1=-9999.
      CF2=-9999.
      CF3=-9999.
C* the following variables are used to get the real and
C* imaginary parts of the variables Q3 through Q5.
      SUM_Q0R=0
      SUM_Q5R=0
      SUM_Q5I=0
      SUM_Q4R=0
      SUM_Q4I=0
      SUM_Q3R=0
      SUM_Q3I=0
C* perform summations for all good data points.
      DO 12 J=1,360 ! azimuths
      A=FLOAT(J)-.5
C* compute sine and cosine of azimuth angles; they are used several times.
      SIN_AZ=SIND(A)
      COS_AZ=COSD(A)
      SIN_2AZ=SIND(2.*A)
      COS_2AZ=COSD(2.*A)
      ILIM=IN(K,J) ! number of data entries
      IF(ILIM.EQ.0)GO TO 12
      DO 10 I=1,ILIM
      IF(IV(K,J,I).EQ.-9999)GO TO 10 ! found an eliminated outlier
C* increment number of good data points.
      DNPT=DNPT+1
C* perform summations used to construct complex variables Q3 -> Q5.
      SUM_Q0R=SUM_Q0R+FLOAT(IV(K,J,I))/10. ! /10. to get back to m/s
      SUM_Q5R=SUM_Q5R+COS_2AZ
      SUM_Q5I=SUM_Q5I+SIN_2AZ
      SUM_Q4R=SUM_Q4R+COS_AZ
      SUM_Q4I=SUM_Q4I+SIN_AZ
      SUM_Q3R=SUM_Q3R+FLOAT(IV(K,J,I))*COS_AZ/10. ! /10. to get back to m/s
      SUM_Q3I=SUM_Q3I+FLOAT(IV(K,J,I))*SIN_AZ/10. ! /10. to get back to m/s
   10 CONTINUE ! end of data point loop
   12 CONTINUE ! end of azimuth loop
C* if there is at least one good data point, complete calculations.
      IF(DNPT.GT.0) THEN
        TWO_N=2*DNPT
        Q0=CMPLX(SUM_Q0R/DNPT)
        Q5=CMPLX(SUM_Q5R/TWO_N,-SUM_Q5I/TWO_N)
        Q4=CMPLX(SUM_Q4R/TWO_N,SUM_Q4I/TWO_N)
        Q3=CMPLX(SUM_Q3R/DNPT,-SUM_Q3I/DNPT)
C* compute conjucate of Q4 since it is used several times
        CCJ_Q4=CONJG(Q4)
C* compute QQ (intermediate step) to save computations and to avoid
C* zero divide errors and floating point overflow errors
        QQ=Q4-1/(4*CCJ_Q4)
        IF (QQ.NE.0.0) THEN
          Q2=(CCJ_Q4-Q5/(2*CCJ_Q4))/QQ
          Q1=(Q0-Q3/(2*CCJ_Q4))/QQ
C* compute INT_COEFF since it is used several times.
C* compute QQ_INT here, to avoid a zero divisor error and subsequent pause
          QQ_INT=(1-(CABS(Q2))**2)
          IF (QQ_INT.NE.0.0) THEN
            INT_COEFF=(Q1-Q2*CONJG(Q1))/(1-(CABS(Q2))**2)
            CF3=IMAG(INT_COEFF)
            CF2=REAL(INT_COEFF)
            CF1=REAL(Q0)-2*REAL(INT_COEFF*Q4)
C* otherwise, Fourier coefficients are missing (-9999.)
          END IF
        END IF
      END IF
C     WRITE(4,90)K,DNPT,CF1,CF2,CF3
C  90 FORMAT(' K,DNPT,CF1,CF2,CF3=',I3,I6,3E12.4)
      RETURN
      END

      SUBROUTINE A317I2__VAD_RMS(K,DHWD,CF1,CF2,CF3,DRMS)
*.  MODULE FUNCTION:
*.     This module computes the square root of the mean squared
*.     deviations between the least squares fitted curve and the
*.     velocity data points. This is better known as the RMS.
*.
*.  PARAMETERS:          (*:  G = GLOBAL, C = COMMON, P = PASSED)
*.  P  CF1      R*4  Fourier coefficient (zeroth harmonic. Rng:[-100,+100]
*.  P  CF2      R*4  Fourier coefficient (real part of first harmonic).
*.                   Rng:[-100,+100]
*.  P  CF3      R*4  Fourier coefficient (imaginary part of first harmonic.
*.                   Rng:[-100,+100]
*.  P  DHWD     R*4  Wind direction, degrees. Dummy variable. Rng:[0,360]
*.  P  DRMS     R*4  The square root of the mean squared deviations
*.                   between the least square fitted curve and the
*.                   data points, in m/s. Dummy variable. Rng:[0,100]
*.
*.  INTERNAL TABLES/WORK AREA:
*.    DNPT    I*4          Number of data points used to perform the least
*.                         squares fitting, Rng:[0,400].
*.    SPEED   R*4          The amplitude of the vad least squares fitted curve,
*.                         in m/s. Rng:[0,100]
*.    SUM_DEV R*4          The summation of squared deviations between the
*.                         fitted curve and the data values, in m**2/s**2.
*.                         Rng:[0,4000000]
*.
*.  MISC:  This software uses the MKS units system.
*.            If not enough data exist to compute a RMS, the rms
*.            is set to the missing data flag.
*.*****************************************************************
      COMMON/STORE/LIST(108225) ! general storage identified by equivalence
      INTEGER*2 IV(15,360,20),IN(15,360)
      EQUIVALENCE (LIST(1),IV(1,1,1)),(LIST(54001),IN(1,1))
      INTEGER DNPT
C*****************************************************************
C* zero variables used for summations.
```

```
      SUM_DEV=0
      DNPT=0
C* compute SPEED since it is used several times.
      SPEED=SQRT(CF2**2+CF3**2)
C* perform summations of the squared deviations between the
C* least squares fitted curve and the velocity data values.
      DO 12 J=1,360 ! all azimuths
      A=FLOAT(J)-.5
      ILIM=IN(K,J) ! data point limit
      IF(ILIM.EQ.0)GO TO 12
      DO 10 I=1,ILIM
C* do only for valid velocities.
        IF(IV(K,J,I).GT.-9999) THEN
          VE=FLOAT(IV(K,J,I))/10. ! convert to m/s
          SUM_DEV=SUM_DEV+(-COSD(A-DHWD)*SPEED+CF1-VE)**2
          DNPT=DNPT+1
        END IF
   10 CONTINUE ! data point loop
   12 CONTINUE ! azimuth loop
C* compute DRMS if there is at least one good data point.
      IF(DNPT.GT.0) THEN
        DRMS=SQRT(SUM_DEV/DNPT)
      ELSE
C* if you are here, there is not enough data points to compute a rms.
        DRMS=-9999.
      END IF
      RETURN
      END

      SUBROUTINE A317J2__FIT_TEST(K,CF1,CF2,CF3,DHWD,DRMS)
*.  MODULE FUNCTION:
*.     This module performs the "fit test". The purpose of this module
*.     is to remove data which: Lies farther than 1 RMS away from the
*.     least squares fitted curve AND toward the zero velocity line.
*.     This procedure removes low magnitude velocity "outliers".
*.
*.  PARAMETERS:          (*:  G = GLOBAL, C = COMMON, P = PASSED)
*.  P  CF1      R*4  Fourier coefficient (zeroth harmonic. Rng:[-100,+100]
*.  P  CF2      R*4  Fourier coefficient (real part of first harmonic).
*.                   Rng:[-100,+100]
*.  P  CF3      R*4  Fourier coefficient (imaginary part of first harmonic.
*.                   Rng:[-100,+100]
*.  P  DHWD     R*4  Wind direction, degrees. Dummy variable. Rng:[0,360]
*.  P  DRMS     R*4  The square root of the mean squared deviations between
*.                   the least square fitted curve and the data points, in
*.                   m/s. Dummy variable. Rng:[0,100]
*.
*.  INTERNAL TABLES/WORK AREA:
*.    FIT   R*4          The velocity on the least squares fitted curve at a
*.                       particular azimuth angle in m/s. Rng:[-100,100]
*.    SPEED R*4          The amplitude of the vad least squares fitted curve,
*.                       in m/s. Rng:[0,100]
*.*****************************************************************
      COMMON/STORE/LIST(108225) ! general storage identified by equivalence
      INTEGER*2 IV(15,360,20),IN(15,360)
      EQUIVALENCE (LIST(1),IV(1,1,1)),(LIST(54001),IN(1,1))
C*****************************************************************
C* compute SPEED since it is used several times.
      SPEED=SQRT(CF2**2+CF3**2)
      DO 12 J=1,360 ! all azimuths
      A=FLOAT(J)-.5
      ILIM=IN(K,J) ! data point limit
      IF(ILIM.EQ.0)GO TO 12
      DO 10 I=1,ILIM
C* only process azimith if the measured velocity data exists.
        IF(IV(K,J,I).GT.-9999) THEN
          VE=FLOAT(IV(K,J,I))/10. ! convert to m/s
          FIT=-COSD(A-DHWD)*SPEED+CF1
C* check if vad fitted curve is above the zero velocity line.
          IF(FIT.GT.0.) THEN
C* if velocity point is farther than 1 rms away from the vad fitted
C* curve and toward the zero velocity line (i.e. less positive)
C* then set it to missing.
            IF(FIT-VE.GT.DRMS)IV(K,J,I)=-9999
          ELSE
C* if you are here, vad fitted curve is below the zero velocity line
C* next... if velocity point is more than 1 rms away from the vad
C* fitted curve and toward the zero velocity line (i.e.less
C* negative), then set it to missing.
            IF(VE-FIT.GT.DRMS)IV(K,J,I)=-9999
          END IF
        END IF
   10 CONTINUE
   12 CONTINUE
      RETURN
      END

      SUBROUTINE A317K2__SYM_CHK(CF1,CF2,CF3,TSMY,SYM)
*.  MODULE FUNCTION:
*.     This module determines if the least squares fitted curve is
*.     "symmetric" around the zero velocity line, within the limits
*.     set by TSMY.
*.
*.  PARAMETERS:    (*:  G = GLOBAL, C = COMMON, P = PASSED)
*.  P  CF1    R*4  Fourier coefficient (zeroth harmonic. Rng:[-100,+100]
*.  P  CF2    R*4  Fourier coefficient (real part of first harmonic).
*.                 Rng:[-100,+100]
*.  P  CF3    R*4  Fourier coefficient (imaginary part of first harmonic.
*.                 Rng:[-100,+100]
*.  P  SYM    L*4  A logical variable indicating that the current least squares
*.                 fitted curve is symmetric about the zero velocity line.
*.                 "TRUE = SYMMETRIC"
*.  P  TSMY   R*4  ADAPTABLE PARAMETER: The maximum value CF1 can be and still
*.                 be accepted as a good wind estimate, in m/s. Rng:[0,10].
*.
*.  INTERNAL TABLES/WORK AREA:
*.  SPEED  R*4  The amplitude of the vad least squares fitted curve, in m/s.
*.              Rng:[0,100]
*.*****************************************************************
```

```
      LOGICAL SYM
C*****************************************************************
C* compute SPEED (vad curve amplitude) here for convienence.
      SPEED=SQRT(CF2**2+CF3**2)
C* the fit is symmetric if: 1) CF1 < TSMY (i.e. mean of the vad
C* fitted curve is sufficiently small), and 2) |CF1| - SPEED < 0
C* (i.e. mean of the fitted curve is smaller than the amplitude
C* of the curve).
      IF(ABS(CF1).LT.TSMY.AND.(ABS(CF1)-SPEED).LE.0) THEN
        SYM=.TRUE.
      ELSE
        SYM=.FALSE.
      END IF
C     WRITE(4,90)CF1,TSMY,ABS(CF1),SPEED
C  90 FORMAT(' from A317K2: CF1<TSMY,ABS(CF1)<=SPEED -->SYM=.TRUE.'/
C    +F10.5,'<',2F10.5,'<=',F10.5)
      RETURN
      END

      SUBROUTINE GETARI
C   Loads four arrays with elevation scan reflectivities.  THe first two tilts
C   have two full scans, the second for a better velocity.  Those have elevation
C   numbers 1, 2, 3, 4 for tilts 0.5 and 1.5 degrees.  Reflectivities are
C   loaded from scans 1, 3, 5, and 6.
      COMMON/ANGLES/AZT(4,370),ELT(4),SECELT(4),JAT(4)
      COMMON/CLOCK/IDATE,ITIME,KDATE,KTIME,MDATE,MTIME
      COMMON/RADIAL/VAL(6),NUM(20),DBZ(0:459),VEL(-3:916),WID(-3:916)
     +,IFLAG,NDBZ(0:459),NVEL(-3:916),NWID(-3:916)
      INTEGER*2 NDBZ,NVEL,NWID
      COMMON/SITE/IDSITE,RH(3),MS(3) ! radar site elev. m, for DEN, ALB, CLE
      CHARACTER*3 MS
      COMMON/SPAN/LASTSCAN,INITSCAN,IENDSCAN ! times for volume scan
      COMMON/STORE/ND05(230,370),ND15(230,370),ND25(115,370)
     +,ND35(10,370) ! upper tilts do not need as much range (108,225 words)
      INTEGER*2 ND05,ND15,ND25,ND35
      DIMENSION SUM(4),TL(4)
      DO 3 J=1,370 ! azimuths
      DO 3 I=1,230 ! ranges
      IF(I.LE. 10)ND35(I,J)=0 ! biased dBz array for 3.5 degrees
      IF(I.LE.115)ND25(I,J)=0 ! biased dBz array for 2.5 degrees
      ND15(I,J)=0 ! biased dBz array for 1.5 degrees
    3 ND05(I,J)=0 ! biased dBz array for 0.5 degrees
      WRITE(6,*)' loading biased dBz'
      WRITE(6,*)' array zeroed   azim    elev'
      DO 5 L=1,4
      JAT(L)=0 ! array indices
      SUM(L)=0. ! sums and total entries for determining average tilt angle
    5 TL(L)=0.
C   Read Level II data (recycle point)
   10 IF(IFLAG.LT.0)GO TO 90 ! else infinite loop
      CALL GETRADAR ! a single radial
      IF(NUM(8).GT.6)GO TO 40 ! do not want higher tilts or velocities
      IF(NUM(8).EQ.2.OR.NUM(8).EQ.4)GO TO 10 ! do not want velocities
      IF(NUM(7).EQ.2)WRITE(6,12)IDATE,ITIME,NUM(8),VAL(3) ! end of elevation
   12 FORMAT(3I7,F8.4,' end of elevation')
      IF(MOD(INT(VAL(2)),60).EQ.0)THEN
        WRITE(6,14)IDATE,ITIME,VAL(2),VAL(3),NUM(20)
      ENDIF
   14 FORMAT(2I7,2F8.3,' mode=',I4)
C   Load the reflectivities
      L=NUM(8)/2 +1 ! calculate tilt number (1,2,3,4) from (1,3,5,6)
      JAT(L)=JAT(L)+1 ! # dBz radials
      IF(JAT(L).GT.370)JAT(L)=370 ! don't exceed the array limit
      GO TO (15,17,19,21) L
   15 DO 16 I=1,229 ! 230 not present
   16 ND05(I,JAT(L))=NDBZ(I) ! 0.5 deg. tilt
      GO TO 23
   17 DO 18 I=1,229 ! 230 not present
   18 ND15(I,JAT(L))=NDBZ(I) ! 1.5 deg. tilt
      GO TO 23
   19 DO 20 I=1,115 ! present array limit to clear mountains at Denver
   20 ND25(I,JAT(L))=NDBZ(I) ! 2.5 deg. tilt
      GO TO 23
   21 DO 22 I=1,10 ! don't need the farther ranges
   22 ND35(I,JAT(L))=NDBZ(I) ! 3.5 deg. tilt
   23 AZT(L,JAT(L))=VAL(2) ! record exact azimuth used
      SUM(L)=SUM(L)+VAL(3)
      TL(L)=TL(L)+1.
      GO TO 10 ! for next radial
C   All data are now loaded for that elevation scan.
   40 DO 50 L=1,4
      ELT(L)=SUM(L)/TL(L) ! average tilt angle
   50 SECELT(L)=1./COSD(ELT(L)) ! secant of elevation angle
   90 WRITE(6,92)(JAT(L),L=1,4)
   92 FORMAT(' Done with loading of first four tilts:',4I4,' radials')
      RETURN
      END

      SUBROUTINE OCCULT
C   Applies occultation correction to PPI files of reflectivity data
C   Replaces A3133P__PARTIAL_OCCULT_CORREC
      COMMON/ANGLES/AZT(4,370),ELT(4),SECELT(4),JAT(4)
      COMMON/SITE/IDSITE,RH(3),MS(3) ! radar site elev. m, for DEN, ALB, CLE
      CHARACTER*3 MS
      COMMON/SITEDATA/TERRA(230),OCCUL(6),HYBRD(230)
      INTEGER*2 TERRA,OCCUL,HYBRD ! direct access radial records
      COMMON/STORE/ND05(230,370),ND15(230,370),ND25(115,370)
     +,ND35(10,370) ! upper tilts do not need as much range (108,225 words)
      INTEGER*2 ND05,ND15,ND25,ND35
      INTEGER OCINDX,OCBEGBIN,OCENDBIN
C   correct lowest tilt
      DO 20 J=1,JAT(1) ! loop for all available azimuths, including duplicates
      OCINDX=AZT(1,J)*5. + 1.05 ! increment and truncate
      IF(OCINDX.GT.1800)OCINDX=1
C     REC=1800*(ELINDX-1)+OCINDX ! if upper tilts are ever need correction
      READ(7,REC=OCINDX)OCCUL ! six range values
C   Perform only if there are some partially occulted bins in this radial
C   given by the ending range of occultation code 0
      IF(OCCUL(1).LT.230)THEN ! this radial is partially occulted
```

```
      DO 15 K=1,4 ! cycle through occult codes
      OCBEGBIN=OCCUL(K)+1
      OCENDBIN=OCCUL(K+1)
      IADD=K*2 ! amount to add to biased dBz
      DO 10 I=OCBEGBIN,OCENDBIN ! radius loop
      IF(ND05(I,J).GT.1)ND05(I,J)=ND05(I,J)+IADD
      IF(ND05(I,J).GT.255)ND05(I,J)=255
   10 CONTINUE ! end of radius loop
   15 CONTINUE ! end of occult code loop
      ENDIF ! end of occultation need test
   20 CONTINUE ! end of azimuth loop
C   Check on need to correct upper tilts
      IF(IDSITE.EQ.1)THEN ! Denver only
        DO 70 J=1,JAT(1) ! loop for all available azimuths, including duplicates
        OCINDX=AZT(2,J)*5. + 1.05 ! increment and truncate
        IF(OCINDX.EQ.1287)THEN ! Mt. Evans only
          DO 60 I=98,230
          IF(ND15(I,J).GT.1)ND15(I,J)=ND15(I,J)+4
          IF(ND15(I,J).GT.255)ND15(I,J)=255
   60     CONTINUE ! end of radius loop
        ENDIF ! end of Mt. Evans check
   70   CONTINUE ! end of azimuth loop
      ENDIF ! end of site check
      WRITE(6,*)' End of occult corrections'
      RETURN
      END

      SUBROUTINE NOSPIKES
C   Scans the saved tilts for isolated and outlier bins and corrects them.
*.    In order to preserve the values for the next radial's testing,
*.  the correction of the bin is delayed by saving the bin's corrected value
*.  in the "corrections" array.
C   Replaces subroutines A3133S, A3133O, A3133G, A3133N
      COMMON/ANGLES/AZT(4,370),ELT(4),SECELT(4),JAT(4)
      COMMON/STORE/ND05(230,370),ND15(230,370),ND25(115,370)
     +,ND35(10,370) ! upper tilts do not need as much range (108,225 words)
      INTEGER*2 ND05,ND15,ND25,ND35
**A3133CA.inc
C** Local versions of adaptation parameters
      INTEGER        MIN_THRESH_RATE, MAX_THRESH_RATE, RATE_TLTEST
     +,IN_RNG_TILT,  OUT_RNG_TILT,    MAX_BISCAN_RNG
      REAL           MIN_AR_LO_ECHO,  MIN_DBZ_AR_WGTD, MAX_AR_PCT_REDUC
     +,ZR_PWR_COEFF, ZR_MLT_COEFF
      INTEGER*2 RATE_TABLE(0:256)
      COMMON /A3133CA/ MIN_THRESH_RATE, MAX_THRESH_RATE,RATE_TLTEST
     $,IN_RNG_TILT,    OUT_RNG_TILT,    MAX_BISCAN_RNG, MIN_AR_LO_ECHO
     $,MIN_DBZ_AR_WGTD,MAX_AR_PCT_REDUC,ZR_PWR_COEFF,   ZR_MLT_COEFF
     $,RATE_TABLE
C*****************************************************************
      INTEGER BIN
      INTEGER*2 NCOR(230,3) ! temporary correction array
      DIMENSION ILIM(4) ! range limits
      DATA ILIM/229,229,114,9/
      NUM_ISO_BINS=0
      NUM_REPLCD_BINS=0
      NUM_INTERP_BINS=0
      NUM_OUTL_BINS=0
      DO 80 L=1,4 ! correct all four tilts
      IF(JAT(L).EQ.0)GO TO 80 ! no data for tilt L
      DO 60 J=1,JAT(L) ! all saved azimuths
C** Initialize for this radial
      JJ=MIN0(J,3)
      DO 45 BIN=2,ILIM(L) ! for nearly all bins
C** IF the bin's BDBZ is above threshhold (45), it may be an isolated bin
      GO TO (11,12,13,14) L ! start by saving the actual bin value
   11 NDBZ=ND05(BIN,J)
      NCOR(BIN,JJ)=ND05(BIN,J)
      GO TO 15
   12 NDBZ=ND15(BIN,J)
      NCOR(BIN,JJ)=ND15(BIN,J)
      GO TO 15
   13 NDBZ=ND25(BIN,J)
      NCOR(BIN,JJ)=ND25(BIN,J)
      GO TO 15
   14 NDBZ=ND35(BIN,J)
      NCOR(BIN,JJ)=ND35(BIN,J)
   15 IF (NDBZ.GT.45) THEN ! BDBZ for -10. dBz
C*** Check the 8 surrounding points to see if this bin is isolated
*.    This module performs the isolated bin correction on the radial
*.  corresponding to the azimuth index AZ.  If a bin's BDBZ is above
*.  the threshold of 46 (-10. dBz), its 8 nearest neighbors are checked
*.  against the same threshold.  If the number of neighbors above the
*.  threshold is not greater than 1, the bin is labeled as
*.  isolated and its value will be replaced by 0.
        KOUNT=0
        DO 22 I=1,3 ! range search
        DO 21 K=1,3 ! azimuth search
        IF(K.EQ.2.AND.I.EQ.2)GO TO 21 ! do not include center point
        II=BIN+I-2 ! range index
        KK=J+K-2 ! azimuth index
        IF(KK.LT.1)KK=JAT(L)
        IF(KK.GT.JAT(L))KK=1
        GO TO (16,17,18,19) L ! check neighbors
   16   MDBZ=ND05(II,KK)
        GO TO 20
   17   MDBZ=ND15(II,KK)
        GO TO 20
   18   MDBZ=ND25(II,KK)
        GO TO 20
   19   MDBZ=ND35(II,KK)
   20   IF(MDBZ.GT.45)KOUNT=KOUNT+1
        IF(KOUNT.GT.1)GO TO 25 ! not isolated; skip the rest
   21   CONTINUE ! azimuth loop, K
   22   CONTINUE ! range loop, I
        NUM_ISO_BINS=NUM_ISO_BINS+1
        NCOR(BIN,JJ)=0 ! suppress the isolated bin
      END IF ! isolated bin checking
C** IF the bin's BDBZ is above threshhold (172), it is an outlier bin
   25 IF (NDBZ.GT.172) THEN ! BDBZ for 53. dBz
```

```
C*** Check the 8 surrounding points to determine the treatment style.
*.    This module performs the outlier bin correction on the radial
*.  corresponding to the azimuth index AZ.  If a bin's value is above the
*.  threshold 172 (=53. dBz), it is identified as an outlier bin.  The 8
*.  nearest neighbor bins are then checked against the same threshold and
*.  if the number of neighbors above the threshold is greater than 0, the bin
*.  is replaced by the value 0.  If the number above is not greater than 0,
*.  the 8 nearest neighbors are used to interpolate a replacement value based
*.  on an average of the precipitation rates.
      WRITE(6,*)' Outlier found:',NDBZ,' BIN,J=',BIN,J
      KOUNT=0
      DO 32 I=1,3 ! range search
      DO 31 K=1,3 ! azimuth search
      IF(K.EQ.2.AND.I.EQ.2)GO TO 31 ! do not include center point
      II=BIN+I-2 ! range index
      KK=J+K-2 ! azimuth index
      IF(KK.LT.1)KK=JAT(1)
      IF(KK.GT.JAT(1))KK=1
      GO TO (26,27,28,29) L ! check neighbors
   26 MDBZ=ND05(II,KK)
      GO TO 30
   27 MDBZ=ND15(II,KK)
      GO TO 30
   28 MDBZ=ND25(II,KK)
      GO TO 30
   29 MDBZ=ND35(II,KK)
   30 IF(MDBZ.GT.172)KOUNT=KOUNT+1
   31 CONTINUE ! azimuth loop, K
   32 CONTINUE ! range loop, I
      IF(KOUNT.GT.0)THEN
        NUM_REPLCD_BINS=NUM_REPLCD_BINS+1
        NCOR(BIN,JJ)=0 ! suppress the outlier entirely
      ELSE
        NUM_INTERP_BINS=NUM_INTERP_BINS+1
        SUM=0.
        DO 42 I=1,3 ! range search
        DO 41 K=1,3 ! azimuth search
        IF(K.EQ.2.AND.I.EQ.2)GO TO 41 ! do not include center point
        II=BIN+I-2 ! range index
        KK=J+K-2 ! azimuth index
        IF(KK.LT.1)KK=JAT(1)
        IF(KK.GT.JAT(1))KK=1
        GO TO (36,37,38,39) L ! sum neighbors
   36   MDBZ=ND05(II,KK)
        GO TO 40
   37   MDBZ=ND15(II,KK)
        GO TO 40
   38   MDBZ=ND25(II,KK)
        GO TO 40
   39   MDBZ=ND35(II,KK)
   40   SUM=SUM+RATE_TABLE(MDBZ) ! sum precip rates
   41   CONTINUE ! azimuth loop, K
   42   CONTINUE ! range loop, I
        AVE=SUM/800. ! average of 8 neighbors; 100: back to mm/hr
        DBZ=10.*ALOG10(ZR_MLT_COEFF*(AVE**ZR_PWR_COEFF))
        NCOR(BIN,JJ)=NINT(2.*(DBZ+32.))+2 ! back to biased dBz
      ENDIF
      NUM_OUTL_BINS=NUM_OUTL_BINS+1
    END IF ! outlier checking
   45 CONTINUE ! range bin loop
C   Make changes in reflectivity array
    IF(J.GE.3) THEN ! can update previous azimuth's radial
      DO 50 I=2,ILIM(L)
      GO TO (46,47,48,49) L ! replace reflectivity with correction
   46 ND05(I,J-1)=NCOR(I,2) ! update
      GO TO 50
   47 ND15(I,J-1)=NCOR(I,2) ! update
      GO TO 50
   48 ND25(I,J-1)=NCOR(I,2) ! update
      GO TO 50
   49 ND35(I,J-1)=NCOR(I,2) ! update
   50 NCOR(I,2)=NCOR(I,3) ! save corrections for present radial
    ENDIF
    IF(J.EQ.JAT(1))THEN ! last time through loop
      DO 55 I=2,ILIM(L)
      GO TO (51,52,53,54) L ! final replacements of reflectivity
   51 ND05(I,JAT(1))=NCOR(I,3) ! last radial
      ND05(I,1)=NCOR(I,1) ! first radial
      GO TO 55
   52 ND15(I,JAT(1))=NCOR(I,3) ! last radial
      ND15(I,1)=NCOR(I,1) ! first radial
      GO TO 55
   53 ND25(I,JAT(1))=NCOR(I,3) ! last radial
      ND25(I,1)=NCOR(I,1) ! first radial
      GO TO 55
   54 ND35(I,JAT(1))=NCOR(I,3) ! last radial
      ND35(I,1)=NCOR(I,1) ! first radial
   55 CONTINUE ! range loop, I
    ENDIF
   60 CONTINUE ! azimuth loop, J
    WRITE(6,62)ELT(L),NUM_ISO_BINS,NUM_OUTL_BINS,NUM_REPLCD_BINS
   +,NUM_INTERP_BINS
   62 FORMAT(' Blemishes fixed to',F4.1,' deg: ISO=',I6,' OUTL=',I6
   +,' (REPL=',I6,' INTP=',I6,')')
   80 CONTINUE ! tilt loop, L
    RETURN
    END

    SUBROUTINE PRECIP
C   Calculates precipitation and integrates into 1-hr, 3-hr, and storm totals
    COMMON/ANGLES/AZT(4,370),ELT(4),SECELT(4),JAT(4)
    COMMON/CLOCK/IDATE,ITIME,KDATE,KTIME,MDATE,MTIME
    COMMON/SNOWWATR/PPI2(230),P1HA(230),P1HV(230),P3HA(230),P3HV(230)
   +,PSTA(230),PSTV(230)
    INTEGER*2 PPI2,P1HA,P1HV,P3HA,P3HV,PSTA,PSTV
    COMMON/SITEDATA/TERRA(230),OCCUL(6),HYBRD(230)
    INTEGER*2 TERRA,OCCUL,HYBRD ! direct access radial records
    COMMON/SPAN/LASTSCAN,INITSCAN,IENDSCAN ! times for volume scan
    COMMON/STORE/ND05(230,370),ND15(230,370),ND25(115,370)
```

```
     +,ND35(10,370) ! upper tilts do not need as much range (108,225 words)
      INTEGER*2 ND05,ND15,ND25,ND35
      COMMON/UPWIND/AAD(0:4),RAD(0:4),TAD(0:4),LEVEL
**A3133CA.inc
C** Local versions of adaptation parameters
      INTEGER          MIN_THRESH_RATE, MAX_THRESH_RATE, RATE_TLTEST
     +,IN_RNG_TILT,    OUT_RNG_TILT,     MAX_BISCAN_RNG
      REAL             MIN_AR_LO_ECHO,   MIN_DBZ_AR_WGTD, MAX_AR_PCT_REDUC
     +,ZR_PWR_COEFF, ZR_MLT_COEFF
      INTEGER*2 RATE_TABLE(0:256)
      COMMON /A3133CA/ MIN_THRESH_RATE, MAX_THRESH_RATE,RATE_TLTEST
     $,IN_RNG_TILT,    OUT_RNG_TILT,     MAX_BISCAN_RNG, MIN_AR_LO_ECHO
     $,MIN_DBZ_AR_WGTD,MAX_AR_PCT_REDUC,ZR_PWR_COEFF,    ZR_MLT_COEFF
     $,RATE_TABLE
      DIMENSION NFILL(110),NPPI(115),JAN(4)
      DATA NFILL,NPPI/110*0,115*0/
C   Major functions (no longer needed because of use of RATE_TABLE):
C     DBZ(N)=(FLOAT(N-2)/2.)-32. ! convert from biased dBz (on tape) to dBz
C     ZE(D)=10.**(D/10.) ! convert dBz to equivalent reflectivity (Ze)
C     RATE(Z)=(Z/399.)**(1./2.21) ! convert Ze to precip rate, mm/hr
C   Identify times
      CALL HHMMSS(LASTSCAN,IH,IM,IS) ! ending time of last volume scan
      T0=FLOAT(IH)+FLOAT(IM)/60.+FLOAT(IS)/3600.
      CALL HHMMSS(INITSCAN,IH,IM,IS) ! starting time of this volume scan
      T1=FLOAT(IH)+FLOAT(IM)/60.+FLOAT(IS)/3600.
      CALL HHMMSS(IENDSCAN,IH,IM,IS) ! ending time of this volume scan
      T2=FLOAT(IH)+FLOAT(IM)/60.+FLOAT(IS)/3600.
      DT=T1-T0 ! time gap from end of last scan to start of this one
      IF(DT.LT.0.)DT=DT+24.
      IF(DT.LT..1)THEN ! less than 6 minute gap between volume scans
        DT=T2-T0 ! use time span between volume scan endings
      ELSE
        DT=T2-T1 ! use duration of this volume scan only
      ENDIF
      IF(DT.LT.0.)DT=DT+24.
      WRITE(4,2)DT
      WRITE(6,2)DT
    2 FORMAT(' DT=',F8.5,' hr')
C   Work first with vertically falling snow
      WRITE(6,*)' Working with vertical snowfall'
C   See if any former times are in precip total file; go on if empty
      DO 10 I=1,3 ! search for all three precip total files
      LU=11+I
      READ(LU,REC=361,ERR=6)JSTART,JSTOP,JDATE,SUMT,SPANT
C   Update times
      JSTOP=IENDSCAN
      JDATE=IDATE
      SUMT=SUMT+DT
      CALL HHMMSS(JSTART,IH,IM,IS)
      T0=FLOAT(IH)+FLOAT(IM)/60.+FLOAT(IS)/3600.
      SPANT=T2-T0 ! span of several volume scans
      IF(SPANT.LT.0.)SPANT=SPANT+24.
      GO TO 10
C   Initiate times
    6 JSTART=INITSCAN
      JSTOP=IENDSCAN
      JDATE=IDATE
      SUMT=DT
      SPANT=DT
      DO 7 J=1,360
    7 WRITE(LU,REC=J)NPPI ! zero the precip file
C   Update precip totals files with times
   10 WRITE(LU,REC=361)JSTART,JSTOP,JDATE,SUMT,SPANT,NFILL
C   Convert reflectivities to precipitation; vertical snowfall
      DO 40 J=1,360 ! azimuth loop
      IF(MOD(J,60).EQ.0)WRITE(6,*)' Precip totals; azim=',J
      READ( 8,REC=J)HYBRD
      READ(12,REC=J)P1HV ! precip totals
      READ(13,REC=J)P3HV
      READ(14,REC=J)PSTV
      A=J ! nominated azimuth
C   Search for angle index for all four tilts
      DO 22 L=1,4
      BMIN=400.
      DO 20 JJ=1,JAT(L) ! search for nearest azimuth for appropriate tilt
        B=ABS(AMOD(A-AZT(L,JJ)+540.,360.)-180.)
        IF(B.LT.BMIN)THEN
          BMIN=B
          JAN(L)=JJ ! index for nearest azimuth to that desired
        ENDIF
   20 CONTINUE ! loop through all recorded azimuths
   22 CONTINUE ! loop through four tilts
C   calculate precipitation contribution, IP, add to totals
      DO 30 I=1,230 ! radius loop
      IP=0
      L=HYBRD(I)
      GO TO (25,26,27,28) L ! one of the four tilts
C   Use 0.5 degree beam (already corrected for occultations)
   25 IF(ND05(I,JAN(L)).GT.1)
     +  IP=NINT(FLOAT(RATE_TABLE(ND05(I,JAN(L))))*DT) ! mm * 100
      GO TO 29
C   Use 1.5 degree beam (no need for occultation corrections for DEN,ALB,CLE)
   26 IF(ND15(I,JAN(L)).GT.1)
     +  IP=NINT(FLOAT(RATE_TABLE(ND15(I,JAN(L))))*DT) ! mm * 100
      GO TO 29
C   Use 2.5 degree beam (protect against I>115)
   27 IF(I.GT.115)GO TO 29
      IF(ND25(I,JAN(L)).GT.1)
     +  IP=NINT(FLOAT(RATE_TABLE(ND25(I,JAN(L))))*DT) ! mm * 100
      GO TO 29
C   Use 3.5 degree beam (protect against I>10)
   28 IF(I.GT.10)GO TO 29
      IF(ND35(I,JAN(L)).GT.1)
     +  IP=NINT(FLOAT(RATE_TABLE(ND35(I,JAN(L))))*DT) ! mm * 100
   29 P1HV(I)=P1HV(I)+IP
      P3HV(I)=P3HV(I)+IP
      PSTV(I)=PSTV(I)+IP
   30 CONTINUE ! end of radius loop, I
C   Update precip files
```

```
      WRITE(12,REC=J)P1HV
      WRITE(13,REC=J)P3HV
      WRITE(14,REC=J)PSTV
   40 CONTINUE ! end of azimuth loop, J
C   Now switch to advected snow
      WRITE(6,*)' Working with advected snowfall'
C   See if any former times are in precip total file; go on if empty
      DO 50 I=1,3 ! search for all three precip total files
      LU=8+I
      READ(LU,REC=361,ERR=46)JSTART,JSTOP,JDATE,SUMT,SPANT
C   Update times
C   Update times
      JSTOP=IENDSCAN
      JDATE=IDATE
      SUMT=SUMT+DT
      CALL HHMMSS(JSTART,IH,IM,IS)
      T0=FLOAT(IH)+FLOAT(IM)/60.+FLOAT(IS)/3600.
      SPANT=T2-T0 ! span of several volume scans
      IF(SPANT.LT.0.)SPANT=SPANT+24.
      GO TO 50
C   Initiate times
   46 JSTART=INITSCAN
      JSTOP=IENDSCAN
      JDATE=IDATE
      SUMT=DT
      SPANT=DT
      DO 47 J=1,360
   47 WRITE(LU,REC=J)NPPI ! zero the precip file
C   Update precip totals files with times
   50 WRITE(LU,REC=361)JSTART,JSTOP,JDATE,SUMT,SPANT,NFILL
C   Convert reflectivities to precipitation; advected snowfall
      DO 80 J=1,360 ! azimuth loop
      IF(MOD(J,20).EQ.0)WRITE(6,*)' Precip totals; azim=',J
      READ( 9,REC=J)P1HA ! precip totals
      READ(10,REC=J)P3HA
      READ(11,REC=J)PSTA
      AAD(0)=J ! nominated azimuth
      READ(3,REC=J)TERRA ! get terrain data
      DO 70 I=1,230 ! radius loop
      RAD(0)=I ! nominated range
      TAD(0)=TERRA(I) ! elevation (meters) of ground location
      CALL ADVECT ! to move that location upwind to 0.5 and 1.5 degree beams
C   LEVEL=1,2,3,4 for those tilts; 5=underground, 6=out of view
      IF(LEVEL.GT.4)GO TO 70 ! upwind is underground or out of view.
      L=LEVEL
      BMIN=400.
      DO 60 JJ=1,JAT(L) ! search for nearest azimuth for appropriate tilt
      B=ABS(AMOD(AAD(L)-AZT(L,JJ)+540.,360.)-180.)
      IF(B.LT.BMIN)THEN
        BMIN=B
        JAZ=JJ ! index for nearest azimuth to that desired for the tilt
      ENDIF
   60 CONTINUE ! loop through all recorded azimuths
      IR=NINT(RAD(L)) ! nearest integer advected range for the tilt
C   calculate precipitation contribution, add to totals
      IP=0
      GO TO (65,66,67,68,70,70) LEVEL
C   Use 0.5 degree beam (already corrected for occultations)
   65 IF(ND05(IR,JAZ).GT.1)
     +  IP=NINT(FLOAT(RATE_TABLE(ND05(IR,JAZ)))*DT) ! mm * 100
      GO TO 69
C   Use 1.5 degree beam (no need for occúltation corrections for DEN,ALB,CLE)
   66 IF(ND15(IR,JAZ).GT.1)
     +  IP=NINT(FLOAT(RATE_TABLE(ND15(IR,JAZ)))*DT) ! mm * 100
      GO TO 69
C   Use 2.5 degree beam
   67 IF(IR.GT.115)GO TO 70 ! protect against range limit
      IF(ND25(IR,JAZ).GT.1)
     +  IP=NINT(FLOAT(RATE_TABLE(ND25(IR,JAZ)))*DT) ! mm * 100
      GO TO 69
C   Use 3.5 degree beam
   68 IF(IR.GT.10)GO TO 70 ! protect against range limit
      IF(ND35(IR,JAZ).GT.1)
     +  IP=NINT(FLOAT(RATE_TABLE(ND35(IR,JAZ)))*DT) ! mm * 100
   69 P1HA(I)=P1HA(I)+IP
      P3HA(I)=P3HA(I)+IP
      PSTA(I)=PSTA(I)+IP
   70 CONTINUE ! end of radius loop, I
C   Update precip files
      WRITE( 9,REC=J)P1HA
      WRITE(10,REC=J)P3HA
      WRITE(11,REC=J)PSTA
   80 CONTINUE ! end of azimuth loop, J
      RETURN
      END

      SUBROUTINE ADVECT
C   Estimates range bins upwind from ground sites.  In this version the
C   advection will start at the level of the radar dish.  Calm air is assumed
C   below that level because there are no wind data there, as in valleys below
C   the radar.
C   LEVEL=1,2,3,4 for those tilts; 5=underground, 6=out of view
      COMMON/ANGLES/AZT(4,370),ELT(4),SECELT(4),JAT(4)
      COMMON/CLOCK/IDATE,ITIME,KDATE,KTIME,MDATE,MTIME
      COMMON/RADIAL/VAL(6),NUM(20),DBZ(0:459),VEL(-3:916),WID(-3:916)
     +,IFLAG,NDBZ(0:459),NVEL(-3:916),NWID(-3:916)
      INTEGER*2 NDBZ,NVEL,NWID
      COMMON/SITE/IDSITE,RH(3),MS(3) ! radar site elev. m, for DEN, ALB, CLE
      CHARACTER*3 MS
      COMMON/SITEDATA/TERRA(230),OCCUL(6),HYBRD(230)
      INTEGER*2 TERRA,OCCUL,HYBRD ! direct access radial records
      COMMON/SPAN/LASTSCAN,INITSCAN,IENDSCAN ! times for volume scan
      COMMON/STORE/ND05(230,370),ND15(230,370),ND25(115,370)
     +,ND35(10,370) ! upper tilts do not need as much range (108,225 words)
      INTEGER*2 ND05,ND15,ND25,ND35
      COMMON/UPWIND/AAD(0:4),RAD(0:4),TAD(0:4),LEVEL
      COMMON/WINDV/K,DK(15),SK(15),NK(15),RMSK(15)
      MINZ=1.+RH(IDSITE)/304.8 ! first kft level above radar
C   adjust for slant range
```

```
      RI=RAD(0)*SECELT(1) ! range within beam right above ground site
      IR=NINT(RI)  ! nearest km along beam
C   calculate cartesian coordinates of gauge, (km,km,m)
      XG=RI*SIND(AAD(0))
      YG=RI*COSD(AAD(0))
      ZG=TAD(0)-RH(IDSITE) ! height of gauge above radar, m; can be negative
C   initialize floating coordinates
      X=XG
      Y=YG
      Z=ZG
      R=RI
      IF(ZG.LT.0.)THEN ! use lowest level winds for all of depth below radar
C       D=0. ! zero wind speed to protect against bad or missing data
C       IF(SK(MINZ).GE.0)D=-SK(MINZ)*ZG/1000. ! horizontal movement, km, in -ZG seconds
C       DX=D*SIND(DK(MINZ)) ! km
C       DY=D*COSD(DK(MINZ)) ! km
C   new radar coordinates:
C       X=X+DX
C       Y=Y+DY
        Z=0 ! by definition; now at radar level
C       R=SQRT(X*X+Y*Y)
C       IR=NINT(R)
      ENDIF
C   Cycle through four tilts if necessary
      DO 60 L=1,4
      LEVEL=L
C   advected ground location now at or above radar but below 0.5 degree beam
      B=BEAMHT(R,ELT(L),0.)*1000.-Z ! remaining beam clearance, m
      IF(B.LT.0.)GO TO 60 ! gauge above radar beam; don't advect yet
   10 ZFM5=(RH(IDSITE)+Z)/.3048-500. ! 500 ft below integrated elevation
      K=(ZFM5+1.)/1000.+1. ! index: kft level for wind integration
      !       ^ protection against truncation when ratio is just < integer
      H=(FLOAT(K)*1000.-ZFM5)*.3048 ! height to next 500 ft level, in meters
      DZ=AMIN1(B,H) ! next integration increment, m
C   fall time through that height increment is DZ sec at 1 m/s
      IF(K.GT.15)K=15 ! do not exceed array limits
      D=0. ! zero wind speed to protect against bad or missing data
      IF(SK(K).GE.0)D=SK(K)*DZ/1000. ! horizontal movement, km, in DZ seconds
      DX=D*SIND(DK(K)) ! km
      DY=D*COSD(DK(K)) ! km
C   potentially new radar coordinates:
      X2=X+DX
      Y2=Y+DY
      Z2=Z+DZ
      R2=SQRT(X2*X2+Y2*Y2) ! range to radar, km
      B2=BEAMHT(R2,ELT(L),0.)*1000.-Z2 ! remaining beam clearance, m
      IF(ABS(B2).LT.1.)THEN ! close enough to beam (within 1 meter)
        X=X2
        Y=Y2
        Z=Z2
        R=R2
        IR=NINT(R)
        B=B2
        GO TO 15 ! for final processing
      ENDIF
      IF(B2.GE.0.)THEN ! still have farther to go to reach beam
        X=X2
        Y=Y2
        Z=Z2
        R=AMIN1(R2,232.) ! stop at 232 km; beyond radar view limit
        IR=NINT(R)
        B=B2
        IF(R.GE.232.)GO TO 15 ! stop integration; too far out
        GO TO 10 ! for next integration step
      ENDIF
C   else we must have (B2.LT.0.), indicating we have crossed over the beam
      F=B/(B-B2) ! the beam is at a fractional distance F; interpolate
      X=X+F*DX
      Y=Y+F*DY
      Z=Z+F*DZ
      R=SQRT(X*X+Y*Y) ! range to radar, km
      IR=NINT(R)
      B=BEAMHT(R,ELT(L),0.)*1000.-Z ! remaining beam clearance, m
C   wrap up the calculations at the final position, update arrays
   15 A=0.
      IF(X.NE.0.OR.Y.NE.0.)A=ATAN2(X,Y)*57.29577951
      IF(A.LT.0.)A=A+360.
      IA=NINT(A)
      IF(IA.LT.1)IA=IA+360
      IF(IA.GT.360)IA=IA-360
      AAD(L)=A ! new radar azimuth for drifting point
      RAD(L)=R ! new radar range for drifting point, km
      TAD(L)=RH(IDSITE)+Z ! new elevation for drifting point, m
C   Check if we need to advect to a higher tilt
      IF(R.GE.230.)THEN ! beyond radar limit.  no data available.
        LEVEL=6
        RETURN
      ENDIF
      READ(3,REC=IA)TERRA
      T=TERRA(IR) ! terrain at advected location
      IF(T.GT.TAD(L))THEN ! advected point is underground.  no snow from there.
        LEVEL=5
        RETURN
      ENDIF
      READ(8,REC=IA)HYBRD
      IF(HYBRD(IR).LE.L)RETURN ! can use the present tilt
   60 CONTINUE ! loop for four tilts; should not need the next line
      RETURN ! no more advection calculations needed
      END

      SUBROUTINE HHMMSS(IT,IH,IM,IS)
C   converts clock time into hours, minutes, seconds
      IS=MOD(IT,100)
      I=IT/100
      IM=MOD(I,100)
      IH=I/100
      RETURN
      END
```

```
      FUNCTION NCLOCK(NSEC)
C   converts total seconds into clock time
      K=MOD(NSEC,60)
      M=NSEC/60
      J=MOD(M,60)
      I=M/60
      NCLOCK=ICLOCK(I,J,K)
      RETURN
      END

      FUNCTION ICLOCK(IH,IM,IS)
C   converts hours, minutes, seconds to clock time
      ICLOCK=(IH*100+IM)*100+IS
      RETURN
      END

      SUBROUTINE DATEJ(JDA,IDATE)
C   recovers date from julian day
      DIMENSION ND(12),NY(4)
C   month     J  F  M  A  M  J  J  A  S  O  N  D
      DATA ND/31,28,31,30,31,30,31,31,30,31,30,31/
      DATA NY/365,365,366,365/
      IY=70
      K=1
      J=JDA
    1 JJ=J-NY(K)
      IF(JJ.LE.0)GO TO 6
      J=JJ
      K=K+1
      IY=IY+1
      IF(K.GT.4)K=1
      GO TO 1
    6 JY=IY
    7 K=365
      ND(2)=28
      IF(MOD(JY,4).NE.0)GO TO 8
      ND(2)=29
      K=366
    8 IF(J.GE.1)GO TO 10
      JY=JY-1
      K=365
      ND(2)=28
      IF(MOD(JY,4).NE.0)GO TO 9
      ND(2)=29
      K=366
    9 J=J+K
      GO TO 8
   10 IF(J.LE.K)GO TO 11
      J=J-K
      JY=JY+1
      GO TO 7
   11 DO 12 M=1,12
      JM=J-ND(M)
      IF(JM.LT.1)GO TO 13
   12 J=JM
   13 IDATE=(JY*100+M)*100+J
      RETURN
      END

      FUNCTION BEAMHT(R,E,A)
C   calculates altitude, km msl, of radar beam from range, R, elevation, E,
C   and site elevation, A, km msl.
      BEAMHT=R*SIND(E)+5.8869E-5*R*R*COSD(E)*COSD(E)+A
      RETURN
      END

      FUNCTION SIND(A)
      SIND=SIN(A*.017453293)
      RETURN
      END

      FUNCTION COSD(A)
      COSD=COS(A*.017453293)
      RETURN
      END

      FUNCTION TAND(A)
      TAND=TAN(A*.017453293)
      RETURN
      END
```

Mission

The mission of the Bureau of Reclamation is to manage, develop, and protect water and related resources in an environmentally and economically sound manner in the interest of the American Public.