User's Manual for the Cloud and Scavenging Module Version 1.2

B. C. Scott

April 1987

**Prepared for
the U.S. Environmental Protection Agency
under a Related Services Agreement
with the U.S. Department of Energy
Contract DE-AC06-76RLO 1830**

**Pacific Northwest Laboratory
Operated for the U.S. Department of Energy
by Battelle Memorial Institute**

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government of any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, or Battelle Memorial Institute.

PACIFIC NORTHWEST LABORATORY
operated by
BATTELLE MEMORIAL INSTITUTE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC06-76RLO 1830

Printed in the United States of America
Available from
National Technical Information Service
United States Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22161

NTIS Price Codes
Microfiche A01

Printed Copy

Pages	Price Codes
001-025	A02
026-050	A03
051-075	A04
076-100	A05
101-125	A06
126-150	A07
151-175	A08
176-200	A09
201-225	A010
226-250	A011
251-275	A012
276-300	A013

USER'S MANUAL FOR
THE CLDUD AND SCAVENGING MODULE
VERSION 1.2

B. C. Scott

April 1987

Prepared for the
U.S. Environmental Protection Agency
under a Related Services Agreement
with the U.S. Department of Energy
Contract DE-AC06-76RLO 1830

Pacific Northwest Laboratory
Richland, Washington 99352

EPA NOTICE

This report is based on work supported by the U.S. Environmental Protection Agency under a Related Services Agreement with the U.S. Department of Energy, Contract DE-AC06-76RLO 1830, Interagency Agreement EPA-DW89930059. Although the research described in this report has been funded wholly or in part by the U.S. Environmental Protection Agency (EPA), it has not been subject to EPA review and therefore does not necessarily reflect the views of EPA and no official endorsement should be inferred.

SUMMARY

This report presents a detailed description of a stable cloud model. This model was designed to characterize a steady-state scavenging environment using meteorological conditions provided by a regional scale meteorology code. The model described in this report explicitly simulates processes related to cloud microphysics.

The cloud model predicts the vertical profiles of condensed water that correspond to specified surface precipitation rates, surface temperatures, and cloud top heights. The cloud model also predicts profiles of various microphysical constituents, such as fall velocity, accretion rates, etc., that exist in conjunction with the condensed water profiles.

Descriptions are presented of routines written to solve mass conservation equations for air, cloud, and precipitation water. The solutions are appropriate for precipitating or nonprecipitating clouds within a mesoscale grid with horizontal spacing on the order of 50 km. Routines are also provided for generating tables of profiles for many discrete input conditions. Once tables are generated, additional routines may be used to interpolate between the tables and to rapidly provide values different from the discrete input levels.

The model was developed with funding provided by the Environmental Protection Agency and the National Acid Precipitation Assessment Program. The immediate application of this work is to provide a means of defining the cloud environment simulated within the scavenging module of the Regional Acid Deposition Model (RADM) Scavenging Module (RSM). Within the RSM, the stable-cloud model provides data on a steady-state, stratiform-like scavenging environment in which a time-dependent, aqueous phase chemical kinetic system will operate.

CONTENTS

SUMMARY	iii
1.0 INTRODUCTION	1
1.1 CLOUD MODEL/RADM/RSM INTERTIES	1
1.2 THE RSM CLOUD MODEL	3
2.0 GOVERNING EQUATIONS	7
2.1 DYNAMIC EQUATIONS	7
2.2 MICROPHYSICAL EQUATIONS	12
2.2.1 Condensation Rate	12
2.2.2 Depositional Growth	12
2.2.3 Ice Crystal Concentration	14
2.2.4 Evaporation	15
2.2.5 Precipitation Fall Speeds	15
2.2.6 Pseudo-Diffusion	16
2.2.7 Scavenging Coefficients	16
2.2.8 Autoconversion	17
2.3 NONPRECIPITATING, STRATIFORM CLOUDS	18
3.0 FINITE DIFFERENCE EXPRESSIONS	19
4.0 CLOUD GENERATION ROUTINES	21
4.1 SUBROUTINE WATER	21
4.2 SUBROUTINE WINDS	24
4.3 PROGRAM STMAIN	25
4.4 SUBROUTINE MAIN1	26
4.5 SUBROUTINE SIGMA	28
4.6 SUBROUTINE COEF	28

4.7	SUBROUTINE VDEP	28
4.8	SUBROUTINE XMULT	28
4.9	SUBROUTINE DEFALT	29
4.10	SUBROUTINE NEUTON	29
4.11	SUBROUTINE ZERO1	29
4.12	SUBROUTINE UNITS	30
5.0	EXECUTION OF THE CLOUD MODEL ROUTINES	31
5.1	DATA INPUT	31
5.2	DATA OUTPUT	34
6.0	DIAGNOSTICS FROM CLOUD GENERATION ROUTINES	35
6.1	DIAGNOSTICS FROM WATER	35
6.2	DIAGNOSTICS FROM WINDS	35
6.3	DIAGNOSTICS FROM STMAIN	35
6.4	DIAGNOSTICS FROM MAIN1	36
7.0	INTERPOLATION ROUTINES	37
7.1	SUBROUTINE TABLEH	37
7.2	SUBROUTINE HEIGHT	38
7.3	PROGRAM BEGIN	38
7.4	SUBROUTINE SELECT	39
7.5	SUBROUTINE INTERP	39
8.0	EXECUTION OF THE INTERPOLATION ROUTINES	43
8.1	EXECUTION USING RAW CLOUD DATA	43
8.2	EXECUTION USING DATA PRODUCED BY TABLEH	43
8.3	EXECUTION REGARDLESS OF STARTING POINT	44
9.0	LIST OF MAJOR ASSUMPTIONS	47

10.0	REFERENCES	51
11.0	APPENDIX A - SAMPLE OUTPUT	A.1
12.0	APPENDIX B - PROGRAM LISTINGS	B.1

1.0 INTRODUCTION

1.1 CLOUD MODEL/RADM/RSM INTERTIES

This report presents a detailed description of one of three major components to Version 1.0 of the Regional Acid Deposition Model (RADM) Scavenging Module (RSM). The RADM is being developed at the National Center for Atmospheric Research (NCAR), funded by the Environmental Protection Agency, and coordinated by the National Acid Precipitation Assessment Program Task force.

RADM is a clear-air regional air pollution code that does not explicitly take cloud-related processes into account. The model described in this report was designed to provide data on a steady-state, stratiform-like scavenging environment in which a time-dependent, aqueous phase, chemical kinetic system will operate. Similar models describing a precipitating convective storm environment^(a) and a fair-weather cumulus scavenging environment^(b) are also being used in the RSM. Output from these cloud models will be used by the RSM to quantitatively describe cloud-related processes in the RADM. The reader is referred to Berkowitz and Easter (1986) for documentation on the structure and theory of the RSM.

In the absence of the RSM, the RADM is simply an advanced, clear-air regional air pollution code that operates in a Eulerian frame of reference over an extensive 80- by 80-km mesh grid. In contrast, the RSM deals with all processes related to clouds and precipitation (Figure 1). The interaction between the two modules is illustrated in the following steps:

-
- (a) B. C. Scott of the Pacific Northwest Laboratory, Richland, Washington, is preparing a user's manual for the convective storm component of the RSM (1986).
  - (b) F. Vukovich of the Research Triangle Institute, Research Triangle Park, North Carolina, is preparing a user's manual for the fair-weather cumulus component of the RSM (1986).

1. The RADM computes transport and clear-air chemistry for a 1-h period. The synoptic scale environment in which these processes occur are defined by data provided by the Meso-alpha Scale (Regional) Limited-Area Model of Anthes (1985).
2. The RSM is activated at each grid column. A grid column is the vertical column over a particular RADM surface grid point. In activating the RSM, the RADM/MM4 system provides

FIGURE 1. Illustration of calling hierarchy between the stratiform scavenging model (described in this report); the Convective Storm model, (a) the Fair-Weather Cumulus model, (b) the Regional Acid Deposition model (RADM), the Mesoscale Meteorology model (MM4), and the RADM Scavenging Module (RSM) (Berkowitz and Easter 1986)

- (a) B. C. Scott of the Pacific Northwest Laboratory, Richland, Washington, is preparing a user's manual for the convective storm component of the RSM (1986) and a fair weather cumulus scavenging environment.
- (b) F. Vukovich of the Research Triangle Institute, Research Triangle Park, North Carolina, is preparing a user's manual for the fair-weather cumulus component of the RSM (1986).

- vertical profiles of temperature, moisture, and updraft velocities
  - precipitation amounts (convective and nonconvective) for the past hour
  - vertical profiles of clear-air pollutant species (gases only)
3. Fields of clouds and precipitation consistent with these profile are established by the RSM. It is the primary role of the cloud model defined in this report to provide a detailed description of the cloud microphysical environment that would be associated with the meteorological conditions specified by MM4 for a particular RADM grid at a fixed time. Using the environmental characteristics defined, the RSM calculates effects of precipitation scavenging, aqueous chemical transformations, and vertical transport by subgrid (convective) motions for 1 h.
  4. At the end of an hour, pollutants in the RSM exist as gases and aerosols. Pollutants also exist in cloud, rain, and snow media. The RSM combines the vertical profiles of the different forms of each pollutant (e.g., sulfate) to produce an equivalent clear-air profile. These profiles are returned to the RADM along with surface wet deposition amounts for pollutants that have been scavenged.
  5. The simulation of scavenging effects of clouds and precipitations for 1 h are represented by the change between the initial profiles passed to the RSM from the RADM and the equivalent clear-air profiles the RSM passes back to the RADM. The RADM combines this change with those changes computed in Step 1, then proceeds to the next 1-h simulation.

1.2 THE RSM CLOUD MODEL

The primary role of the cloud model described in this report is to predict the vertical profiles of condensed water that correspond to specified surface

precipitation rates, surface temperatures, and cloud top heights.^(a) A secondary role is to predict profiles of various microphysical constituents, such as fall velocity, accretion rates, etc., that exist in conjunction with the condensed water profiles.

In the cloud model, only two condensed water categories are used: cloud water and precipitation water. Crude vertical resolution in the vicinity of the freezing level prohibits categories, such as melting snow or freezing water.

Cloud water is assumed to move with the air velocity. Precipitation water moves horizontally with the air but falls vertically with respect to it. The fall speeds and collection rates of precipitation depend on the classification of precipitation as snow or rain which, in turn, depend on the local air temperature.

Precipitation embryos are assumed to be ice crystals that grow at the top of the cloud by vapor deposition until they reach dimensions of 400 μm . The precipitation embryos continue to grow by vapor deposition and accretion as the crystals fall. Below the freezing level, the ice particles melt, accelerate, continue to accrete cloud droplets, and fall to the surface as millimeter-sized rain drops.

A series of routines has been written to solve mass conservation equations for air, cloud, and precipitation water. The solutions are appropriate for precipitating or nonprecipitating clouds within a mesoscale grid with horizontal spacing on the order of 50 km. The vertical grid spacing is determined at the discretion of the user.

The model equations are coded in FORTRAN, and the entire model has been successfully executed on a VAX 11-780 computer using a VMS operating system. The cloud model routines execute relatively slowly. To enable rapid access of information at many discrete input conditions, routines are provided for

---

(a) The present version of the RSM provides an updraft profile consistent with the MM4 predicted horizontal winds. This updraft drives the microphysics described in this report. The cloud model capability for providing an updraft, which would result in a surface precipitation rate specified by the MM4, is not used in the RSM.

generating tables of profiles for a variety of initial conditions. Once the tables are generated and saved, a user can evoke routines designed to interpolate within the tables. Thus, values at locations different from the discrete input levels can be rapidly obtained. This table lookup and interpolation procedure is 5 to 10 times faster than generating profiles for each discrete set of input data.

The relation between the major routines is illustrated in Table 1. The routines in the Level 3 column are called by the immediately adjacent routine in Level 2. Routines in the Level 2 column are called by the adjacent Level 1 routine, while the Level 1 routines are called by the main programs.

TABLE 1. Relationship Between the Routines in the Cloud Model

Cloud Generation Routines			
<u>Main Program</u>	<u>Level 1</u>	<u>Level 2</u>	<u>Level 3</u>
STMAIN	DEFAULT		
	SIGMA		
	MAIN1	NEUTON	
		DEFAULT	
		UNITS	
		WINDS	COEF
			NEUTON
		WATER	COEF
			NEUTON
			ZERO1
			XMULT
			VDEP
Interpolation Routines			
<u>Main Program</u>	<u>Level 1</u>	<u>Level 2</u>	<u>Level 3</u>
TABLEH	HEIGHT		
BEGIN	SELECT	INTERP	

This document describes the basic conservation equations for condensed water and provides a description of the function of each routine in the model. Also provided is a list of assumptions used in the model and the procedures for executing the model and interpolation routines.

2.0 GOVERNING EQUATIONS

2.1 DYNAMIC EQUATIONS

Within the model domain, the water concentrations are determined by a balance between the liquid water produced in a region, the water passing in or out through the boundaries of the region, and the water removed from the region after capture by falling precipitation. Specifically, by denoting M_c and M_p as the mass of water per unit volume of space associated with cloud water and precipitation water, respectively, the local change of each within a precipitating cloud is given by

$$\frac{\partial M_c}{\partial t} = - \vec{v} \cdot M_c \vec{v} + D_c - C_p - A_p \quad (2.1)$$

$$\frac{\partial M_p}{\partial t} = - \vec{v} \cdot M_p \vec{v} + \frac{\partial [(V_p - V_d)M_p]}{\partial z} + D_p + C_p + A_p \quad (2.2)$$

In the above, $\vec{v}$ represents the wind velocity vector, V_p the fall speed of precipitation (taken as positive here), and V_d the diffusion velocity for precipitation. D represents the diffusion of moisture from the air to cloud or precipitation water; C represents the collection of cloud water by precipitation; and A is the conversion rate of cloud water to precipitation water through an autoconversion process.

In order to form a mass budget equation for the condensed water within a storm region (or cell), several assumptions are made. First, the precipitation and cloud water is assumed to be horizontally uniform. Next, at the lateral inflow boundaries of each cell, the cloud and precipitation water concentration are assumed to be zero. And finally, the storm is assumed to persist for many hours and to be in a quasi-steady state.

Mathematically, these assumptions enable Equations (2.1) and (2.2) to be expressed as

$$-Gx \bar{M}_c + (M_c w) \Big|_z - (M_c w) \Big|_{z+\Delta z} + (\bar{D}_c - \bar{C}_p - \bar{A}_p) \Delta z = 0 \quad (2.3)$$

and

$$\begin{aligned} -Gx \bar{M}_p + \left[M_p (V_p - V_d - w) \right] \Big|_{z+\Delta z} - \left[M_p (V_p - V_d - w) \right] \Big|_z \\ + (\bar{D}_p + \bar{C}_p + \bar{A}_p) \Delta z = 0 \end{aligned} \quad (2.4)$$

where

$$Gx = \delta_u \bar{u} \Big|_{x+\Delta x} - \delta_u \bar{u} \Big|_x + \delta_v \bar{v} \Big|_{y+\Delta y} - \delta_v \bar{v} \Big|_y \quad (2.5)$$

and

($\bar{}$) = the quantity averaged over the interval z to $z + \Delta z$,

($\bar{}$) = the quantity averaged over the volume $\Delta x, \Delta y, \Delta z$.

Here, Gx represents the mass blown out through the sides of a cell by the horizontal winds. The variables u and v are the horizontal wind components, and the values of δ_u and δ_v are given by

$$\begin{aligned} \delta_u &= \Delta z / \Delta x \text{ if airflow at an } x\text{-boundary is out of the storm} \\ &= 0 \quad \text{otherwise} \\ \delta_v &= \Delta z / \Delta y \text{ if airflow at a } y\text{-boundary is out of the storm} \\ &= 0 \quad \text{otherwise.} \end{aligned} \quad (2.6)$$

Equation (2.3) is linear and can be solved for M_c using a recursive algorithm, given vertical profiles of M_p and w . Equation (2.4) is, however, nonlinear because of the strong dependence of V_p and C_p on M_p and must be solved iteratively for M_p .

Equations (2.3) and (2.4) are the basic dynamic equations of the model; they are coupled in the sense that one cannot be solved without knowledge of the values obtained from the other. The solution sequence for these two equations is carried out in subroutine WATER.

Concentrations obtained from Equations (2.3) and (2.4) are strongly dependent on the updraft velocity. In the current version of the model, a parabolic updraft profile is assumed to be representative of storm updrafts in a steady-state, mesoscale system. The form used is

$$w = \frac{6\bar{W}}{H} \left(z - \frac{z^2}{H} \right) \quad (2.7)$$

where $\bar{W}$ is the mean updraft profile obtained by vertically averaging from the ground to cloud top height, H . The user can easily choose a different vertical wind profile by modifying the statements in subroutine WINDS.^(a)

Also important in determining concentrations of condensed water is the magnitude of the horizontal wind components. All three wind components are coupled through the anelastic continuity equation

$$\vec{V} \cdot \rho_a \vec{V} = 0 \quad (2.8)$$

where ρ_a is the air density and is taken to be a function of height alone. Integrating (2.8) over a cell volume of $\Delta x \Delta y \Delta z$ and letting $\Delta x = \Delta y$ yields a continuity equation

---

(a) The present version of the RSM provides an updraft profile consistent with the MM4 predicted horizontal winds. This updraft drives the microphysics described in this report. The cloud model capability for providing an updraft, which would result in a surface precipitation rate specified by the MM4, is not used in the RSM.

$$\begin{aligned} \frac{\Delta z}{\Delta x} \bar{\rho} \left[\bar{u} \Big|_{x+\Delta x} - \bar{u} \Big|_x + \bar{v} \Big|_{y+\Delta y} - \bar{v} \Big|_y \right] \\ + \left(\bar{\rho w} \right) \Big|_{z+\Delta z} - \left(\bar{\rho w} \right) \Big|_z = 0 \end{aligned} \quad (2.9)$$

where the overbar implies a vertical average over the thickness of the cell.

The solution of Equation (2.9) is accomplished in subroutine WINDS by using the updraft profile together with several assumptions about the vertical variation of the inflow u and v components. The u and v wind components must be specified at two boundaries. Currently, the u component at $x + \Delta x$ is taken as

$$\begin{aligned} \bar{u}(x + \Delta x) &= \frac{UG \Delta z}{\Delta x} \left(1 - \frac{2z}{H} \right) \quad \text{for } z < ZMIX \\ \bar{u}(x + \Delta x) &= 0 \quad \text{for } z > ZMIX \end{aligned} \quad (2.10)$$

where UG is a storm speed, H is the storm height, and ZMIX is a mixing layer thickness. UG is varied by WINDS to obtain the desired storm efficiency. ZMIX is held constant and is set in STMAIN. The inflow v component is taken as

$$\bar{v}(y) = k \bar{u}(x + \Delta x) \quad (2.11)$$

where k is SHIR in the coding. SHIR is varied by WINDS to adjust efficiency. A further relationship between u and v is

$$\frac{\partial v}{\partial y} = C \frac{\partial u}{\partial x} \quad (2.12)$$

where

$$C = CB - CL(CB + 1) \frac{z}{H} \quad (2.13)$$

and for the current version $CB = 0$ and $CL = 0.15$.

With the above restrictions for the two u and v components, the remaining u and v components are required by Equation (2.9) to equal

$$\bar{u}(x) = \bar{u}(x + \Delta x) + \frac{2 \left[(\rho w)|_{z+\Delta z} - (\rho w)|_z \right] \Delta x}{(1+C) \left[\rho(z+\Delta z) + \rho(z) \right] \Delta z}$$

$$\bar{v}(y + \Delta y) = \bar{v}(y) + C[\bar{u}(x + \Delta x) - \bar{u}(x)] \quad (2.14)$$

The following sign convention is used. For airflow into the storm, $\bar{u}(x)$ and $\bar{v}(y)$ are positive, while $\bar{u}(x + \Delta x)$ and $\bar{v}(y + \Delta y)$ are negative.

Basically, u and v are adjusted to obtain, as closely as possible, a particular storm efficiency, E , defined as

$$E = \frac{\text{surface precipitation flux}}{\text{inflow vapor flux}} \quad (2.15)$$

The u and v adjustment is accomplished by varying the values of C and UG .

Storm efficiency is taken to have a seasonal variation (here, seasons are represented by surface temperature) with maximum values of 0.7 in the winter (cyclonic storms) and minimum values of 0.3 in the summer (convective storms). Storm efficiency is assumed to vary like

$$E = 0.7 - \frac{0.4 (T_0 - 4)}{21} \quad (2.16)$$

where T_0 is the surface temperature and E is limited to maximum and minimum values of 0.7 and 0.3, respectively.

Other values assumed to have a seasonal variation are

	Winter	Summer
storm speed (UG)	14 m/s	9 m/s
mixing level height ($ZMIX$)	0.5 km above sfc	2.0 km above sfc
storm width (XX)	300 km	85 km

The variation is described by

$$\begin{aligned}UG &= 14 - \frac{5 (T_0 - 4)}{21} \\ZMIX &= 0.5 + \frac{1.5 (T_0 - 4)}{21} \\XX &= 300 - \frac{215 (T_0 - 4)}{21}\end{aligned}\tag{2.17}$$

where T_0 is the surface temperature. The minimum T_0 used in Equation (2.17) is 4°C, while the maximum used is 25°C.

2.2 MICROPHYSICAL EQUATIONS

2.2.1 Condensation Rate

The total condensation rate, $\bar{D}_T$, is computed from Kessler (1969)

$$\bar{D}_T = -w\rho_a \left(\frac{\partial r_s}{\partial z} \right)\tag{2.18}$$

Both air density, ρ_a , and the saturation mixing ratio, r_s , are determined from thermodynamic variables computed along a moist adiabat in subroutine SIGMA. $\bar{D}_T$ is computed for both water and ice and the results are stored in array SORCE (for water) and SORCEI (for ice).

2.2.2 Depositional Growth

Depositional growth of precipitation is considered only for temperatures less than 0°C in the model. Condensation onto liquid precipitation is neglected. The depositional growth rate for ice, $\bar{D}_p$, is computed by considering the growth, by vapor deposition, of individual ice crystals in a snowflake (Scott 1982)

$$\frac{dx}{dt} = 31.5g\bar{S}_i x^{0.38}\tag{2.19}$$

where x is the mass of an ice crystal, g is a thermodynamic ice growth function from Fletcher (1969), and S_i is the cloud supersaturation with respect to ice. Integrating Equation (2.19) over the interval Δt and assuming that the number of ice crystals per unit volume, N , remains constant during Δt yields

$$M_d = \left(M_{do}^{0.62} + 19.53 \overline{gS_i} N^{0.62} \Delta t \right)^{1.61} \quad (2.20)$$

where $M_d = Nx$, M_{do} is the flake mass concentration at the beginning of the interval Δt , and the overbars represent average values during Δt . Note that where precipitation enters a cell, its mass is given by M_{do} .

The depositional growth rate $\bar{\bar{D}}_p$ can then be approximated from

$$\frac{M_d - M_{do}}{\Delta t} \approx \frac{\delta M_d}{\delta t} = \bar{\bar{D}}_p \quad (2.21)$$

or

$$\bar{\bar{D}}_p = \left\{ \left[M_{do}^{0.62} + \frac{19.53 \overline{gS_i} N^{0.62} \Delta z}{(\bar{v}_p - \bar{w})} \right]^{1.61} - M_{do} \right\} \frac{(\bar{v}_p - \bar{w})}{\Delta z} \quad (2.22)$$

where Δt has been replaced by $\Delta z/(\bar{v}_p - \bar{w})$, and all the quantities with overbars are evaluated in the interval Δz .

The function $\overline{gS_i}$ is curve-fit to give

$$\begin{aligned} \overline{gS_i} &= 0.98 \times 10^{-8} (-T)^{1/2} && \text{for } T > -10^\circ\text{C} \\ &= 3.2 \times 10^{-8} && \text{for } -20^\circ\text{C} < T < -10^\circ\text{C} \\ &= (5.4 + 0.11T) \times 10^{-8} && \text{for } -40^\circ\text{C} < T < -20^\circ\text{C} \\ &= 1 \times 10^{-8} && \text{for } T < -40^\circ\text{C} \end{aligned} \quad (2.23)$$

where T is the temperature.

Often, the depositional growth rate determined from Equation (2.22) can exceed the permissible growth rate computed in Equation (2.18), especially when ice crystal concentrations are large. Should excessively large growth rates occur, the depositional growth rate is limited to the total condensational growth rate, $\bar{D}_T$ (computed with respect to ice). In all instances, the depositional growth rate is subtracted from the total condensation rate. If any excess results, the remaining condensed water contributes to condensed cloud water $\bar{D}_C$, i.e.,

$$\begin{aligned}\bar{D}_C &= \bar{D}_T - \bar{D}_p & \text{for } \bar{D}_p < \bar{D}_T \\ \bar{D}_C &= 0 & \text{for } \bar{D}_p > \bar{D}_T\end{aligned}\tag{2.24}$$

If $\bar{D}_p$ should exceed $\bar{D}_T$, then $\bar{D}_p$ is set equal to $\bar{D}_T$.

2.2.3 Ice Crystal Concentration

The ice crystal concentration, N , is determined two different ways. If ice crystal multiplication is neglected, then N is a constant depending on cloud top temperature. If ice crystal multiplication is considered, then N depends on the maximum cloud water concentration encountered below the -10°C level.

When ice crystal multiplication is neglected, the ice crystal concentrations are determined by

$$N = 400 \left(\frac{-T}{16} \right)^4 \quad (\text{m}^{-3})\tag{2.25}$$

where T is the cloud top temperature.

When ice crystal multiplication is considered, the crystal concentration at height z is assumed to depend linearly on the maximum cloud water concentration between height z and the height at the -10°C level, i.e.,

$$N = (4 \times 10^{-4}) M_C(\text{max}) \quad (\text{m}^{-3})\tag{2.26}$$

An upper limit of 30 crystals/L is assumed. Ice crystal multiplication is not allowed at temperatures colder than -10°C .

2.2.4 Evaporation

Below cloud base, the term $\bar{D}_p$ in Equation (2.4) is taken to represent the evaporation of precipitation. The expression used originates from Kessler (1969) and is written in the form

$$\bar{D}_p = 5.4 \times 10^{-4} \rho_a r_s (RH - 1) M_p^{0.65} \quad (2.27)$$

where RH is the relative humidity at height j .

The relative humidity below cloud base is assumed to vary like

$$RH = RH_0 \exp(kz^2) \quad (2.28)$$

where k is selected such that $RH = 1$ at cloud base.

2.2.5 Precipitation Fall Speeds

For snow, fall speeds in units of m/s are given by Scott (1982)

$$\begin{aligned} v_p &= 2.43 M_p^{0.2} & M_p < 2.44 \times 10^{-4} \text{ g/m}^3 \\ v_p &= 2.749 \left(\frac{M_p}{N_s} \right)^{0.0684} & M_p \geq 2.44 \times 10^{-4} \text{ g/m}^3 \end{aligned} \quad (2.29)$$

For rain, fall speeds in units of m/s are given by Scott (1982)

$$\begin{aligned} v_p &= 385.267 \left(\frac{M_p}{N_w} \right)^{0.25} & M_p < 0.11 \text{ g/m}^3 \\ v_p &= 38.8496 \left(\frac{M_p}{N_w} \right)^{0.125} & M_p \geq 0.11 \text{ g/m}^3 \end{aligned} \quad (2.30)$$

2.2.6 Pseudo-Diffusion

For rain and snow, a "pseudo-diffusivity" is used to parameterize the vertical dispersion of precipitation, which results from the variation of settling velocity with particle size. Because the model does not directly consider diffusivities, this diffusion is expressed in terms of a diffusion velocity

$$V_d = -D_s \frac{\partial}{\partial z} \left[\ln \left(\frac{M_p}{\rho_a} \right) \right] \quad (2.31)$$

where D_s is the diffusivity expressed as

$$D_s = \frac{\Delta z \left(\overline{V_p^2} - \bar{V}_p^2 \right)}{\bar{V}_p} \quad (2.32)$$

Here, $\overline{V_p^2}$ is the mass-weighted average of the square of the settling velocity, and $\bar{V}_p$ is the mass-weighted average of the settling velocity. The diffusion velocity is computed in subroutine VDEP.

2.2.7 Scavenging (Accretion) Coefficients

The collection of cloud water by precipitation, C_p , is defined in terms of a scavenging coefficient, ψ , as

$$C_p = M_c(j)\psi \quad (2.33)$$

The scavenging coefficients are functions of the precipitation water concentration and are taken from Scott (1982). For rain

$$\psi = 7.008 \times 10^{-4} N_w^{0.125} E_w \bar{M}_p^{0.875} \quad (2.34a)$$

and for snow

$$\psi = \frac{2.427 \times 10^{-2} E_s \bar{M}_p^{1.0684}}{N_s^{0.0687}} \quad (2.34b)$$

where units are per second when $\bar{M}_p$ is in g/m^3 .

Scavenging by ice crystals is not allowed to begin until the ice concentration exceeds a threshold value given by X_{M0} , where

$$X_{M0} = N(2.4 \times 10^{-6}) \quad (2.35)$$

Here N is the ice crystal concentration, and 2.4×10^{-6} represents the mass (grams) an individual ice crystal must have before accretion begins.

Collection efficiencies for rain, E_w , and snow, E_s , and Marshall-Palmer intercepts for rain, N_w , and snow, N_s , are input in program STMAIN.

2.2.8 Autoconversion

The autoconversion rate in $\text{g/m}^3/\text{s}$ is from Berry and Reinhardt (1974) and is given by

$$A_p = 5.5 \times 10^9 M_c^2 X_c^{5/3} \left[1 - \left(\frac{2.35 \times 10^{-9}}{X_c} \right)^{4/3} \right] \left[1 - \left(\frac{2.04 \times 10^{-9}}{X_c} \right)^{1/3} \right] \quad (2.36)$$

where X_c is the average cloud drop mass,

$$X_c = \frac{\overline{M}_c}{N_d} \quad (2.37)$$

and N_d is the concentration of cloud droplets per m^3 . The cloud droplet concentration is held constant in the current version at $500/cm^3$. Changes in the cloud droplet concentration can be achieved by adjusting the variable DROPS in program STMAIN.

Autoconversion is allowed to proceed within the cloud regardless of cloud temperature. In fact, if the cloud top is warmer than $0^\circ C$, it is the autoconversion that generates the precipitation embryos within the cloud. If the cloud temperature is colder than $0^\circ C$, large cloud droplets are assumed to freeze immediately after being converted into raindrops. Thus, there is no intermediate raindrop category above the freezing level.

2.3 NONPRECIPITATING, STRATIFORM CLOUDS^(a)

The term "nonprecipitating" means only that precipitation is not reaching the ground. The cloud itself may be producing copious quantities of precipitation only to have it evaporate near cloud base.

Because of the ambiguity of the term nonprecipitating, several assumptions must be made in order to compute profiles for nonprecipitating, stratiform clouds. First, the precipitation rate at cloud base is assumed to be 0.2 mm/h unless the mean updraft velocity is required to be greater than 10 cm/s to generate such a rate. For those situations, the cloud base precipitation rate is reduced successively by $3/4$ until the mean generating updraft falls below 10 cm/s . Next, the evaporating precipitation is assumed to fall to levels below the observed cloud base. And finally, the updraft profile is still assumed to be parabolic from the ground to cloud top height. All other dynamic and microphysical specifications remain the same as for the precipitating cloud.

(a) This component of the model is not presently implemented in the RAOM Scavenging Module.

3.0 FINITE DIFFERENCE EXPRESSIONS

To express the cloud water equation, (2.3), in the finite difference form used in subroutine WATER, set C_p equal to $\psi \bar{M}_c$, set $\bar{M}_c$ equal to $[M_c(j+1) + M_c(j)]/2$, and solve for $M_c(j+1)$

$$M_c(j+1) = \frac{\left\{ M_c(j) [w(j) - 0.5(\phi \Delta z + Gx)] + (D_c - A_p) \Delta z \right\}}{[w(j+1) + 0.5(\phi \Delta z + Gx)]} \quad (3.1)$$

Using similar substitutions, the precipitation water concentration, (2.4), is written in finite difference form as

$$\begin{aligned} & - \left[M_p (V_p - V_d - w) \right] \Big|_j + \left[M_p (V_p - V_d - w) \right] \Big|_{j+1} \\ & + (D_p + C_p + A_p) \Delta z - \bar{M}_p Gx = 0 \end{aligned} \quad (3.2)$$

where

$$\bar{M}_p = \frac{[M_p(j+1) + M_p(j)]}{2} \quad (3.3)$$

Equation (3.2) is also solved in routine WATER.

The continuity equation, (2.9), becomes, in finite difference form, equal to

$$\frac{u(x)}{\Delta x} = \frac{\rho(j+1)w(j+1) - \rho(j)w(j)}{\bar{\rho}(j)\Delta z} + \frac{u(x+\Delta x) - u(x)}{\Delta x} + \frac{v(y+\Delta y) - v(y)}{\Delta y} \quad (3.4)$$

where

$$\bar{\rho}(j) = \frac{[\rho(j+1) + \rho(j)]}{2} \quad (3.5)$$

and Δx has been set equal to Δy .

The solution of (3.4) is accomplished in subroutine WINDS by using the parabolic updraft profile together with several assumptions about the vertical variation of the inflow u and v components. These assumptions are discussed in Section 2.1.

The diffusion velocity is based on a pseudo-diffusivity resulting from a variation of settling velocity with particle size [see Equation (2.31)]. V_d is expressed as

$$V_d = - \frac{D_s}{\Delta z} \ln \left[\frac{\left(\frac{M_p}{\rho_a} \right)_{z+\Delta z/2}}{\left(\frac{M_p}{\rho_a} \right)_{z-\Delta z/2}} \right] \quad (3.6)$$

where D_s is given by (2.32). After substitution D_s becomes

$$D_s = \frac{\Delta z}{V_p(j)} 96.39 \left[\frac{M_p(j)}{N_w} \right]^{0.25}$$

for rain and

$$D_s = \frac{\Delta z}{V_p(j)} 0.95 \left[\frac{M_p(j)}{N_s} \right]^{0.135} \quad (3.7)$$

for snow.

The diffusion velocity is computed in routine VDEP.

4.0 CLOUD GENERATION ROUTINES

Table 1 shows the relationship between the cloud generation routines. The main program STMAIN calls the routines in the Level 1 column. The Level 1 routines call the Level 2 routines, and the Level 2 routines call the Level 3 routines. The bulk of the operations is done in routines MAIN1, WATER, and WINDS. Several of these routines write diagnostic messages indicating some unacceptable condition has occurred. The actual messages and their interpretations are presented in Section 6.

4.1 SUBROUTINE WATER (called by MAIN1)

Subroutine WATER provides vertical profiles of condensed water for particular surface or cloud base precipitation rates and cloud thicknesses. To achieve these results, the updraft velocity is adjusted iteratively in order to provide the condensation rates that lead to growth rates of precipitation sufficient to achieve the desired surface or cloud base precipitation rate.

Subroutine WATER passes computed surface or cloud base precipitation rates to MAIN1 where they are compared with the desired precipitation rate. MAIN1 then passes a new guess for updraft velocity to WINDS where new wind components are computed for use in WATER. WATER then recomputes profiles of condensed water and passes the new surface or cloud base precipitation rate back to MAIN1. The process is repeated until the computed surface precipitation rate agrees with the desired value.

If the call to WATER by MAIN1 is with new environmental conditions (IFLG = 0), then initial guesses of precipitation concentration in WATER are obtained by assuming that the precipitation rate varies linearly from the

surface to cloud top. For this interpolation, the precipitation rate, J , is converted to concentrations using Scott (1982)

$$\text{for rain, and} \quad M_p = 123.8 N_w^{0.11} J^{0.889} \quad (4.1)$$

$$M_p = 0.117 N_s^{0.0687} J^{0.936}$$

for snow.

If MAIN1 calls WATER with unchanged environmental conditions but with new guesses of updraft velocity ($IFLG > 0$), the previous values of precipitation and cloud water are used for initial guesses of cloud and precipitation concentrations.

Early in the subroutine, condensation rates are computed using Equation (2.18). The results are stored in arrays SORCE and SORCEI. Next, a portion of the depositional growth term is computed, and the results stored in arrays SUP and SDEP. A call to XMULT is required here if ice crystal multiplication is considered. A call to COEF determines the horizontal winds for use in the balance equations.

Cloud water concentrations are computed from Equation (3.1) by integrating from the ground to cloud top and by using the previous guess of M_p for the loss terms represented by the accretion, autoconversion, and deposition coefficients.

Precipitation water concentrations are computed from Equation (3.2) by integrating from the cloud top to the surface. Subroutine NEUTON is used to determine $M_p(j)$ at each height so that Equation (3.2) is balanced. Thus, within this integration from cloud top to the ground, there is a separate iteration to determine M_p at each height.

If, within this second iteration, convergence is not achieved after 11 tries, subroutine ZERO1 is called to search the interval of M_p values from 1×10^{-11} to 30 g/m^3 for a solution. If a solution exists, the iteration in

the precipitation water loop (DO 260) begins again with the initial guess near the solution. If a solution does not exist, a diagnostic is written (see Section 6) and program execution halts.

Because precipitation concentrations can change drastically near the freezing level resulting from the acceleration of melting snow, special care must be taken in computing concentrations in the vicinity. The important term to consider is the first term in Equation (2.4)

$$Gx(i) \bar{M}_p(i) \quad (4.2)$$

where the overbar represents the average value of M_p in cell i . By referring to Figure 2, the average of M_p can be represented as

$$\bar{M}_p = \frac{1}{2\Delta z} \left\{ [M_f + M_p(i)](H_f - z) + [M_p(i+1) + M_f](z + \Delta z - H_f) \right\} \quad (4.3)$$

The value of M_f is estimated by extrapolating from the next higher cell

$$M_f = M_p(i+1) + \frac{[M_p(i+2) - M_p(i+1)]}{\Delta z} (H_f - z_{i+1}) \quad (4.4)$$

In all other cells, $\bar{M}_p$ is computed as a linear average of the values at the top and bottom of the cell.

When updated values of precipitation water are determined, cloud water concentrations are recomputed. These updated values of cloud water are used to recompute M_p and so on until negligible changes in M_c and M_p occur for each loop.

When the convergence criteria is satisfied, WATER computes the surface precipitation rate and passes the result, XJ1, out to MAIN1 for comparison with the desired value.

FIGURE 2. Precipitation Water Concentration in the Vicinity of the Freezing Level

4.2 SUBROUTINE WINDS (called by MAIN1)

Subroutine WINDS takes the average vertical velocity supplied by MAIN1, computes updraft profiles, and adjusts the horizontal winds to satisfy Equation (2.9) and to meet (if possible) any storm efficiency requirements. The updraft profile is currently taken to be parabolic.^(a)

When the horizontal winds are computed, the routine proceeds to calculate the inflow vapor flux using values of vapor density (VAP) returned by subroutine COEF. A storm efficiency is calculated and compared with the desired efficiency. The variable SHIR is first adjusted by routine NEUTON to get closer agreement. If further adjustment is needed, the storm speed, UG, is varied before again fine tuning by adjusting SHIR.

Occasionally, it is impossible, using Equation (2.9), to achieve efficiencies as large as desired. Should this occur, the routine computes the largest obtainable efficiency (by adjusting SHIR and UG) and uses that value as a default value.

(a) The present version of the RSM provides an updraft profile consistent with the MM4 predicted horizontal winds. This updraft drives the microphysics described in this report. The cloud model capability for providing an updraft, which would result in a surface precipitation rate specified by the MM4, is not used in the RSM.

4.3 PROGRAM STMAIN (main program)

STMAIN is the main program for the cloud model. Various input conditions, constants, and flags are set here. The program initiates calls to routines, which determine profiles of temperature and moisture. The program also writes computed cloud profiles on a mass storage device.

STMAIN is configured to generate a table of profiles at various input precipitation rates, cloud top and cloud base heights, and surface temperatures. The starting values for computations are set near the beginning of STMAIN by specifying values for cloud top (H0), surface temperature (TG), cloud base (ZBASE1), and surface precipitation rate (XJ0). All heights are input as height above the surface. Increments or step sizes for successive calculations are determined by setting DH, DJ, and DT. The number of increments are determined by setting NT, NH, and NP.

STMAIN is designed to write data onto device 10 as if it were being stored in an array R(NVM,NP,NH,NT) where NVM is the number of vertical grids, NP is the number of precipitation increments (Do loop 100), NH is the number of height increments (Do loop 400), and NT is the number of temperature increments (Do loop 200). The number of vertical grids (NVM) is established by setting the variable, L, in SCAV1.FOR.

It is in routine STMAIN that the user determines whether to consider ice multiplication in the computations, and whether calculations are done for precipitating or nonprecipitating clouds.

Constants currently have the values

- surface pressure = 980 mb
- surface height = 0 m (height above surface)
- surface humidity = 90% if $T > 0^{\circ}\text{C}$,
- surface humidity = 99% if $T < 0^{\circ}\text{C}$,
= 50% if the cloud is nonprecipitating
- collection efficiency for water = 0.65
- collection efficiency for ice = 0.2
- Marshall-Palmer intercept N_w for water = $1 \times 10^7 \text{ m}^{-4}$
- Marshall-Palmer intercept N_s for ice = $5 \times 10^7 \text{ m}^{-4}$
- cloud drop concentration = $500/\text{cm}^3$.

Other constants are assumed to vary linearly with surface temperature (thus, a seasonal nature is implied) as follows:

	<u>Winter</u>	<u>Summer</u>
mixing level height (ZMIX)	0.5 km above sfc	2.0 km above sfc
storm speed (UG)	14.0 m/s	9.0 m/s
storm efficiency (E)	0.7	0.3
storm width (XX)	300.0 km	85.0 km

The linear variation is described by Equations (2.16) and (2.17).

For each cloud top height and surface temperature combination, the program determines the temperature at the cloud top and the height of the freezing level. The temperature lapse rate is assumed to be moist adiabatic and is determined in routine SIGMA.

Currently, STMAIN writes eight variables to mass storage. If more variables are desired, the two WRITE(10) statements must be modified in the program.

4.4 SUBROUTINE MAIN1 (called by STMAIN)

Subroutine MAIN1 adjusts the mean vertical velocity to give desired surface precipitation rates. Some microphysical constants are initialized in MAIN1 and much of the program output is performed here.

The microphysical constants set in MAIN1 are used to determine ice crystal concentrations. If ice crystal multiplication is ignored, the ice crystal concentrations are assumed to be determined by Equation (2.25). If ice crystal multiplication is considered, ice crystal concentration is determined in XMULT after a call by WATER.

Ice crystals are assumed to grow to a riming size of 400 μm , which is equivalent to $X_{MI} = 2.4 \times 10^{-6}$. The total mass concentration of the ice crystals when riming (accretion) begins is $N \cdot X_{MI} = X_{M0}$. To simulate aggregation

of ice crystals and to permit the onset of riming as soon as possible (i.e., to limit cloud glaciation), the total ice crystal concentration is not allowed to exceed 30 crystals/L.

To adjust the mean updraft velocity, W , MAIN1 compares the precipitation rate (computed with the current guess of W) to the desired precipitation rate. The difference of the two rates, DIFF1, is then passed to NEUTON where a new value for W is returned.

Several default options are tested when a new vertical velocity is obtained. If the mean updraft velocity exceeds 30 cm/s, the assumptions of a mesoscale model are assumed to be violated and the data from the last successful set of input conditions are substituted by DEFAULT as a solution for the current run. Similarly, DEFAULT is called if the desired precipitation rate cannot be achieved without reducing the mean updraft velocity to near zero. Finally, DEFAULT is called if a satisfactory W has not been found after 20 tries.

Once a proper W is determined, MAIN1 prints out all desired information before returning to STMAIN for new input conditions.

Current units and output are

- cloud water (g/g)
- precipitation water (g/g)
- updraft speed (cm/s)
- scavenging coefficient (1/s)
- deposition or evaporation rate (1/s)
- precipitation rate (mm/h)
- precipitation fall speed (cm/s)
- autoconversion rate (1/s)
- mid-point cell height (m)

Units can be changed by adjusting the conversion factors in subroutine UNITS (called at the end of MAIN1).

4.5 SUBROUTINE SIGMA (called by STMAIN)

SIGMA provides the pressure (in dynes/cm²) and temperature (in degrees K) at each height grid in the cloud model. The temperature is assumed to vary moist adiabatically from the surface to cloud top. The thermodynamic variables are determined in PRESSF, SAT, SSAT, RHOAIR, and RELAPSE. These five routines are described in documentation for PLUVIUS (Easter and Hales 1984). All heights used in the calculations are taken to represent height above the surface. SIGMA is called by routine STMAIN.

4.6 SUBROUTINE COEF (called by WINDS, WATER)

COEF has two different functions, depending on the call. If V=-1, the routine returns the average saturation vapor density in the interval z to $z + dz$. The V=-1 call originates in WINDS.

If V=-2, the routine computes wind components for Equations (2.5), (3.1), or (3.2). The V=-2 call originates in WATER.

4.7 SUBROUTINE VDEP (called by WATER)

VDEP computes the diffusion velocity for rain and snow when called by WATER. Equation (3.6) provides the finite difference form used by VDEP.

4.8 SUBROUTINE XMULT (called by WATER)

XMULT computes ice crystal concentrations when called by WATER. If the flag MULT equals 1, the ice crystal concentration depends on a multiplication factor. If MULT = 0, the ice crystal concentration depends on the cloud top temperature, as expressed by Equation (2.25).

Ice crystal multiplication is only allowed in the temperature interval between 0 and -10°C. If ice crystal multiplication is desired, the actual crystal concentration at height z is assumed to depend linearly on the maximum cloud water concentration between height z and the height at the -10°C level. An upper limit of 3D crystals/L is assumed.

4.9 SUBROUTINE DEFALT (called by MAIN1, STMAIN)

DEFALT is called by MAIN1 and STMAIN whenever the program is preparing to save calculated profiles. If the current input conditions result in unrealistic profiles (or convergence in some iteration scheme is not achieved), DEFALT restores the previous values to all output profiles.

If acceptable values are produced for the current input conditions, DEFALT saves all the output variables for a future default set.

4.10 SUBROUTINE NEUTON (called by WINDS, MAIN1, WATER)

NEUTON uses Newton's method to iterate on 'A' such that the function $F(A)=0$. Here, the function F describes the difference between computed and desired values. F is equal to

- the surface precipitation rate differences when A = mean updraft velocity
- the mass flux balance Equation (3.2) differences when A = precipitation concentration
- the storm efficiency differences when A = the variable SHIR in Equation (2.11).

'A' is adjusted such that

$$A = A_0 - \frac{F(A_0)}{\left(\frac{dF}{dA}\right)} \quad (4.5)$$

In all calls to NEUTON, DIFF1 represents the function, F, and convergence is achieved when A is adjusted so that $F = 0$.

4.11 SUBROUTINE ZERD1 (called by WATER)

Subroutine ZERD1 steps through the concentration interval ranging from an initial value determined by R0 to 30 g/m^3 (on a log scale) searching for a solution to Equation (3.2). For a particular value of concentration, AXIS(K),

the left-hand side of Equation (3.2) has a value of VALUE(K). If two successive values of VALUE have different signs, then a zero has been crossed near AXIS(K-1), and the solution flag (IFLG=2) is set. If no solution to Equation (3.2) is found, calls to ZERO1 will continue with larger guesses of concentration until the maximum value of 30 is exceeded.

4.12 SUBROUTINE UNITS (called by MAIN1)

Subroutine UNITS is the location where the units are converted from the internal working units to the units desired by the user. The current form provides output units described in Section 4.4. The units used internally are

- lengths in (m)
- times in (s)
- mass in (g)
- cloud water concentration (g/m^3)
- precipitation water concentration (g/m^3)
- updraft velocity (m/s)
- accretion rate (1/s)
- deposition-evaporation rate ($\text{g/m}^3/\text{s}$)
- precipitation rate (mm/h)
- precipitation fall speed (m/s)
- autoconversion rate ($\text{g/m}^3/\text{s}$)

5.0 EXECUTION OF THE CLOUD MODEL ROUTINES

5.1 DATA INPUT

Data input is done near the beginning of STMAIN. First, the user determines whether to consider ice multiplication and if computations are to be done for precipitating or nonprecipitating clouds. The statements in STMAIN are

```
C
C ... FLAGS:
C     MULT = 0 FOR NO ICE MULTIPLICATION
C     MULT = 1 FOR ICE MULTIPLICATION
C     NOPRE = 0 FOR PRECIPITATING CLOUD
C     NOPRE = 1 FOR NON PRECIPITATING CLOUD
C
C     MULT = 1 (for example)
C     NDPRE = 0 (for example)
```

The program expects to vary the initial values of surface temperature, cloud top height, and precipitation rate. The next series of input statements designate the number of initial starting conditions.

```
C ... NT = # OF SURFACE TEMPERATURE INTERVALS BEING CONSIDERED
C ... NH = # OF CLOUD HEIGHT INTERVALS BEING CONSIDERED
C ... NP = # OF PRECIPITATION RATE INTERVALS BEING CONSIDERED
C     (OR IF THE NONPRECIPITATING CLOUD CASE IS RUN)
C     = # OF CLOUD BASE INTERVALS BEING CONSIDERED
C
C     NT = 4 (for example)
C     NH = 4 (for example)
C     NP = 4 (for example)
```

If only one set of input data are to be considered at a time (tables of data are not generated), set NT, NH, and NP = 1.

The next input are values specifying cloud top (HD), surface temperature (TG), cloud base (ZBASE1), and surface precipitation rate (XJO). All heights are input as height above the surface.

```

C ***** INPUT HEIGHT AS HEIGHT ABOVE GROUND LEVEL *****
C
C ... TG = INITIAL SURFACE TEMPERATURE TO CONSIDER ( C)
C ... HO = INITIAL CLOUD TOP HEIGHT TO CONSIDER (METERS)
C ... XJO = INITIAL SURFACE PRECIP RATE TO CONSIDER (MM/HR)
C     CAN START AT 0 MM/HR IF GENERATING TABLES FOR PRECIPITATING
C     CLOUDS. IF NONPRECIP CLOUDS ARE BEING CONSIDERED, THIS
C     VARIABLE IS NOT USED.
C ... ZBASE1 = INITIAL CLOUD BASE HEIGHT TO CONSIDER (METERS)
C
C     HO = 8000. (for example)
C     TG = 5. (for example)
C     ZBASE1 = 333. (for example)
C     XJO = 0.0 (for example)
C
C

```

The increments used when varying cloud top, precipitation rate, and temperature are input next. If tables of data are not generated, the next variables will have no use and can be neglected.

```

C ... DH = INCREMENT TO USE WHEN VARYING CLOUD TOP AND BASE HEIGHT (M)
C ... DJ = INCREMENT TO USE WHEN VARYING SURFACE PRECIP RATE (MM/HR)
C ... DT = INCREMENT TO USE WHEN VARYING SURFACE TEMPERATURE (C)
C
C     DH = -2000. (for example)
C     DJ = 1.0 (for example)
C     DT = 10. (for example)
C

```

The next series of statements set values for various constants used in the model. If the user wishes to change any value, the change should be made here. The constants currently have the values:

- surface pressure = 980 mb
- surface height = 0 m (height above surface)

- surface humidity = 90% if $T > 0^{\circ}\text{C}$,
- surface humidity = 99% if $T < 0^{\circ}\text{C}$, = 50% if the cloud is nonprecipitating
- collection efficiency for water = 0.65
- collection efficiency for ice = 0.2
- Marshall-Palmer intercept N_w for water = $1 \times 10^7 \text{ m}^{-4}$
- Marshall-Palmer intercept N_s for ice = $5 \times 10^7 \text{ m}^{-4}$
- cloud drop concentration = $500/\text{cm}^3$

The corresponding statements in STMAIN are

```

C ... INPUT SURFACE PRESSURE (PG), HUMIDITY (RHG)
      PG = 980.
      RHG = 0.90
      IF(NOPRE .EQ. 1) RHG = 0.5
C
C ... CL IS USED TO ADJUST STORM EFFICIENCY
      CL = 0.15
C
C ... INPUT COLLECTION EFFICIENCY FOR WATER (EW), AND ICE (EI)
      EW = 0.65
      EI = 0.2
C
C ... INPUT MAR.-PALMER No FOR WATER (XNO), AND ICE (BPNO)
      XNO = 1.E+7
      BPNO = 5.E+7
      AXNO = XNO**0.222*EW
C
C ... INPUT CLOUD DROP NUMBER (#/M3)
      DROPS = 5.E8
C

```

Certain variables are assumed to depend on surface temperature. If the user wishes to use other relationships, the changes are made in STMAIN. The variable names are

- ZMIX = mixing level height (m)
- UG = storm speed (m/s)
- E = storm efficiency
- XX = storm width (km)

If the user wishes to change the number of vertical grids used in the model (currently equal to 20), one final bit of input data must be placed in the block SCAV1.FOR. The constant, L, which equals the number of vertical grids, must be set in the PARAMETER statement.

5.2 DATA OUTPUT

Unit 10 is the designated output unit. Output to mass storage is done in STMAIN with the WRITE(10) statements. Hard copy output (unit 6) is done in MAIN1. Current units and output are

- cloud water concentration (g/g)
- precipitation water concentration (g/g)
- updraft speed (cm/s)
- scavenging coefficient (1/s)
- deposition or evaporation rate (1/s)
- precipitation rate (mm/h)
- precipitation fall speed (cm/s)
- autoconversion rate (1/s)
- midpoint cell height (m)

STMAIN writes eight variables to mass storage. If more variables are desired, the two WRITE(10) statements must be modified in STMAIN. In addition, in the include block SCAV1.FOR, the constant LSAVE must be changed to the appropriate value, and the new variables must be added to the common block SET.

6.0 DIAGNOSTICS FROM CLOUD GENERATION ROUTINES

6.1 DIAGNOSTICS FROM WATER

- AT 3002, NO SOLUTION WATERNEW. No solution for precipitation water concentration exists for the current input conditions. Program execution halts.
- WARNING,KL TOO BIG,LOOP 2000,WATER, NO CONVERGENCE (a value = present guess of surface precipitation rate [mm/h]). No convergence. If the output value is different from the desired rate, then there is no problem. If the output value is equal to the desired rate, then the final answer is suspect. Execution continues.

6.2 DIAGNOSTICS FROM WINDS

- NO CONVERGENCE ON EFFICIENCY IN WINDS. No solution exists for current input conditions. Execution halts.
- DEFAULT E TAKEN TO BE (value). The desired storm efficiency cannot be obtained. The largest possible value is used as a default value.

6.3 DIAGNOSTICS FROM STMAIN

- CLOUD TOP IS LOWER THAN FREEZING LEVEL FOR H = ..., TG = ... Cloud top is required to be higher than the freezing level in the model; if not, the last acceptable data are used for the current input conditions
- CLOUD BASE GREATER THAN H, H = ... DEFAULT USED. Cloud base greater than cloud top. Last set of good data used for current input conditions
- CLOUD BASE LOWER THAN THE SURFACE, DEFAULT USED. The incremental step, DH, has reduced the cloud base level to below the surface for nonprecipitating clouds. Either reduce NP or DH in the initial conditions.

6.4 DIAGNOSTICS FROM MAIN1

- NO SOLUTION FOR $J = \dots$, $H = \dots$, $WBAR = \dots$. The desired precipitation rate can be achieved only if the mean updraft velocity is zero or greater than 30 cm/s. Both are unacceptable. The last set of good data are used for the current input conditions.
- NO CONVERGENCE IN MAIN. The program is unable to obtain an updraft profile that provides the desired precipitation rate. The last set of good data are used for the current input conditions.

7.0 INTERPOLATION ROUTINES

Several routines are provided for interpolation purposes (see Table 1). The first group includes TABLEH and HEIGHT. These routines remove tabulated profiles from a mass storage device and form a new table for interpolation purposes. The second group includes BEGIN, SELECT, and INTERP. These interpolate from the table formed by TABLEH to obtain vertical profiles at desired input conditions. To properly execute these programs, it is necessary to assure block TABLE.FOR is correctly constructed. The execution procedures are described in Section 8. Here the function of each subroutine is briefly described.

7.1 PROGRAM TABLEH

TABLEH unloads the cloud data from device 10 and forms a new table on device 8 with the user's desired grid spacing. The user's vertical grid is assumed to have linear height coordinates for its vertical scale.

To understand the unloading sequence in TABLEH, it is necessary to understand how the variables are stored by the main stable cloud routines. In these routines, each variable saved is written in a particular order to a mass storage device. Each variable is stored as if it were loaded into a multidimensional array R with dimensions R (NVM,NP,NH,NT). Here, NVM is the number of vertical levels used in the cloud model program, NP is the number of discrete surface precipitation rates considered, NH is the number of cloud top heights evaluated, and NT is the number of surface temperatures used. As in normal array storage, the left-most subscripts vary most rapidly.

TABLEH reads in a variable's vertical profile for a particular value of NP, NH, and NT and places these values in arrays ending with the letter 'E'. Program HEIGHT is then called to map the data onto the user's desired vertical grid spacing. These new data are stored in arrays ending with the letter 'X'. These newly interpolated X variables are written to unit 8 at the conclusion of TABLEH, thereby producing a new table of data loaded exactly as R, but with the user's vertical grid spacing.

Currently, TABLEH is configured to read and operate on eight variables; these are

- cloud water (g/g)
- precipitation water (g/g)
- updraft speed (cm/s)
- scavenging coefficient (1/s)
- deposition or evaporation rate (1/s)
- precipitation rate (mm/h)
- precipitation fall speed (cm/s)
- autoconversion rate (1/s).

TABLEH asks for input from the console. The number 0 should be typed for a stable, stratiform cloud.

7.2 SUBROUTINE HEIGHT (called by TABLEH)

HEIGHT interpolates data from the vertical grid spacing used in the cloud model to a desired vertical grid spacing of the user's model. A linear grid spacing must be used for both. The vertical coordinate is in terms of height rather than pressure or sigma. Present interpolation is linear.

The cloud variables are passed into HEIGHT in arrays ending in the letter 'E'. Interpolated variables are passed out of the routine in arrays ending in the letter 'X'. The variable JJ serves as the counter for storage position in the interpolated arrays.

7.3 PROGRAM BEGIN

BEGIN is a dummy program provided to illustrate the calling sequence for the interpolation routines entitled SELECT and INTERP. These latter two routines are necessary for interpolating cloud data from the table produced with the user's vertical grid spacing (see TABLEH). The call to SELECT must contain the variable LIM set equal to 1, and the variables Q(i) where

Q(1) = hourly surface precipitation amount (mm) or cloud base height
for nonprecipitating cloud

Q(2) = cloud top height (m)

Q(3) = surface temperature (C)

and the variable IIN, which is the number of the device containing the tabulated cloud data.

BEGIN asks for data to be input from the console. The number 0 should be typed for a stable, stratiform cloud, while appropriate values should be entered for rain amount (or cloud base height), cloud top, and temperature.

7.4 SUBROUTINE SELECT (called by BEGIN)

SELECT reads the tabulated convective cloud data on unit IIN into memory and places the interpolated cloud variables into common in block TABLE.FOR. Actual interpolation is done by a call to INTERP.

All interpolated variables are placed in arrays ending in 'I' as follows:

- RI = cloud water in updraft (g/g)
- RCI = precipitation water in updraft (g/g)
- WI = updraft speed in updraft (cm/s)
- ALPHI = scavenging coefficient in updraft (1/s)
- DEPOSI = deposition or evaporation rate in updraft (1/s)
- PRATEI = precipitation rate in updraft (mm/h)
- FALLI = precipitation fall speed in updraft (cm/s)
- AUTOI = autoconversion rate in updraft (1/s).

Unless disabled, SELECT will provide output on unit 6 (console) for debug purposes. To eliminate this printout, all statements, from and including WRITE(6,807) TIMEI, RATEI to and including 501 Z = Z+DZ*XINTER, can be deleted.

7.5 SUBROUTINE INTERP (called by SELECT)

INTERP interpolates from the table of values supplied by SELECT to find vertical profiles at given values of cloud top height, surface precipitation

rate, and surface temperature. A third order interpolation scheme is used to determine the profile value at the desired point [i.e., at a fixed height (for index I between 1 and NV]

$$\begin{aligned}
 R(J,H,T) = & R(J_0,H_0,T_0) + \frac{(H-H_0)}{(H_1-H_0)} [R(J_0,H_1,T_0) - R(J_0,H_0,T_0)] \\
 & + \frac{(J-J_0)}{(J_1-J_0)} [R(J_1,H_0,T_0) - R(J_0,H_0,T_0)] \\
 & + \frac{(T-T_0)}{(T_1-T_0)} [R(J_0,H_0,T_1) - R(J_0,H_0,T_0)] \\
 & + \frac{(J-J_0)(H-H_0)}{(J_1-H_0)(H_1-H_0)} [R(J_1,H_1,T_0) - R(J_1,H_0,T_0) - R(J_0,H_1,T_0) + R(J_0,H_0,T_0)] \\
 & + \frac{(T-T_0)(H-H_0)}{(T_1-T_0)(H_1-H_0)} [R(J_0,H_1,T_1) - R(J_0,H_0,T_1) - R(J_0,H_1,T_0) + R(J_0,H_0,T_0)] \\
 & + \frac{(T-T_0)(J-J_0)}{(T_1-T_0)(J_1-J_0)} [R(J_1,H_0,T_1) - R(J_0,H_0,T_1) - R(J_1,H_0,T_0) + R(J_0,H_0,T_0)] \\
 & + \frac{(T-T_0)(J-J_0)(H-H_0)}{(T_1-T_0)(J_1-J_0)(H_1-H_0)} [R(J_1,H_1,T_1) - R(J_1,H_0,T_1) - R(J_0,H_1,T_1) \\
 & - R(J_0,H_0,T_1) - R(J_1,H_1,T_0) + R(J_1,H_0,T_0) + R(J_0,H_1,T_0) - R(J_0,H_0,T_0)] \quad (7.1)
 \end{aligned}$$

where interval spacing is given by $(J_1 - J_0)$, $(H_1 - H_0)$ and $(T_1 - T_0)$, and J_0 , H_0 , T_0 are the indices at the beginning of the intervals that contain J , H , T .

The index, L , at any data point (J_0, H_0, T_0) in the array, R , is given by

$$L = 1 + NV \{J_0 - 1 + NP[H_0 - 1 + NH(T_0 - 1)]\} \quad (7.2)$$

where I is a height index. The indices of the other data points in Equation (7.1) can be expressed in terms of L. For example, at the same height, the index at J_1, H_0, T_0 is given by

$$\lambda = I + NV \{J_1 - 1 + NP[H_0 - 1 + NH(T_0 - 1)]\} \quad (7.3)$$

But $J_1 = J_0 + 1$ and

$$\lambda = L + NV \quad (7.4)$$

8.0 EXECUTION OF THE INTERPOLATION ROUTINES

Execution of the interpolation programs depends on what data are available to the user. If the user has available data with the desired vertical spacing and correct variable loading sequence, then program BEGIN (or SELECT) can be executed directly without any preliminary setup work. If, however, just the raw cloud data are available, then a new table must be produced using TABLEH before BEGIN can be executed.

8.1 EXECUTION USING RAW CLOUD DATA (TABLEH)

Program TABLEH assumes the data are stored on device 10. This unit number can be changed to a desired unit by changing the statement:

```
INUNIT = 11
```

to a value of one plus the desired unit number, i.e.,

```
INUNIT = number of device containing cloud data + 1.
```

The table generated by TABLEH is output on device 8. The output device can be changed simply by substituting for IOUT near the beginning of the routine.

TABLEH asks for input from the console. The number 0 should be typed for a stable, stratiform cloud

8.2 EXECUTION USING DATA PRODUCED BY TABLEH (SELECT-called from BEGIN)

Currently, subroutine SELECT expects to input the tabulated data on device 8. If the input data are on some other device, that device number should be set in the call to SELECT (the variable-IIN).

BEGIN asks for data to be input from the console. The number 0 should be typed for a stable, stratiform cloud, while appropriate values should be entered for rain amount (or cloud base if the cloud is nonprecipitating), cloud top, and temperature. To bypass or disable (see Section 7.3), SELECT will output data for debug purposes unless disabled. (See Section 7.4 for disabling process.)

8.3 EXECUTION REGARDLESS OF STARTING POINT

Before executing TABLEH or BEGIN (SELECT), it is necessary to ensure that various parameters have proper values in statements in block INCLUDE 'TABLE.FOR'. In particular, values must be set for the number of cloud top intervals (NH), the number of precipitation intervals (NP), the number of temperature intervals (NT), and the number of vertical grids (NVM) used by the cloud model during the generation of tables. Also, the number of vertical grids in the user's model must be specified (NV).

In TABLE.FOR, various increments and starting values must be defined and correspond with those used in the main cloud routines. These values are supplied through HMAX and arrays DV and VO in DATA statements. HMAX represents the upper boundary height. Slight differences in meaning of DV and VO occur, depending on whether or not the clouds in the tables are precipitating or nonprecipitating.

DV is the increment of initial conditions used in program STMAIN. For example, for a precipitating cloud, DV(1) represents increments of surface precipitation rate used as input (suggested to be 1 mm/h). For a nonprecipitating cloud, DV(1) is taken to represent differences in successive cloud base heights. DV(2) is the increment for cloud top height, which is currently set in STMAIN to decrease from some maximum value in 2000-m increments. Thus, DV(2) = -2000. Similarly, DV(3) is the increment for surface temperature (suggested to be 10°C).

VO is the starting value for each of the intervals defined by DV. For precipitating clouds, VO(1) represents the first precipitation rate value and is taken to be 0.0 mm/h. Successive precipitation rates are determined by DV(1) and are equal to 1.0, 2.0, ... If nonprecipitating clouds are considered, VO(1) represents the initial cloud base height to be used. VO(2) is the starting value for cloud top height (e.g., 10,000 m), and VO(3) is the starting value for surface temperature (e.g., -5 C). For DV(3) of 10, successive temperatures would be +5, +15, ...

As an example, consider nonprecipitating clouds where the user may wish to access tables generated for cold clouds with tops at 2, 3, and 4 km; cloud

bases at 1, 2, and 3 km; and surface temperatures of -5 and +5°C. In this case, the PARAMETER and DATA statements have the form

```
PARAMETER (.....NH = 3, NP = 3, NT = 2,...)
```

```
.
```

```
.
```

```
DATA DV /-1000,-1000,10,0,0,0/
```

```
DATA VO /3000,4000,-5,0,0,0/
```

Notice that the initial values and the intervals are given in the DATA statements, while the total number of intervals is given in the PARAMETER statement.

For another example, consider tables for warm, nonprecipitating clouds with cloud tops of 3.5, 4.75, and 6.0 km; cloud bases at 5, 3.75, 2.5, and 1.25 km; and surface temperatures at 5, 15, and 25°C. Here, the appropriate statements are

```
PARAMETER (.....NH = 3, NP = 4 ,NT = 3,...)
```

```
.
```

```
.
```

```
DATA DV /-1250,-1250,10,0,0,0/
```

```
DATA VO /5000,6000,5,0,0,0/
```

Finally, as an example for precipitating clouds, consider tables with cloud tops at 10, 8, 6, and 4 km; precipitation rates at 0, 1, 2, 3, and 4 mm/h; and surface temperatures ranging from -5 to +25°C in 10° intervals. The necessary statements are

```
PARAMETER (.....NH = 4, NP = 5, NT = 4,...)
```

```
.
```

```
.
```

```
DATA DV /1.,-2000,10,0,0,0/
```

```
DATA VO /0.,10000,-5,0,0,0/
```

Besides making the proper control statements to load the data on the desired device numbers and assuring that all desired variables are written by

the main cloud routine, no other changes or checks are required before executing the interpolation routines. Be sure to recompile if changes are made in the TABLE.FOR block.

9.0 LIST OF MAJOR ASSUMPTIONS

The solutions are appropriate for precipitating or nonprecipitating clouds within a mesoscale grid with horizontal spacing on the order of 50 km.

Only two condensed water categories are used: cloud water and precipitation water. Melting snow and freezing water categories are neglected.

Cloud water moves with the air's velocity in contrast to precipitation water, which falls with respect to the air.

The cloud and precipitation water are assumed to be horizontally uniform.

The air density is a function of height alone.

The temperature lapse rate is moist adiabatic from the ground to cloud top.

The relative humidity below cloud base varies according to a specified power law.

The horizontal dimensions of the cloud are the same in the x and y directions.

At the lateral inflow boundaries of the cloud, the cloud and precipitation water concentration are assumed to be zero.

The storm is assumed to persist for many hours and to be in a quasi-steady state.

A parabolic updraft profile is representative of storm updrafts in a steady state, mesoscale system.

The updraft profile is parabolic from the ground to cloud top for nonprecipitating, stratiform clouds.

Arbitrary definitions of u and v are used at two of the cloud boundaries.

If the mean updraft velocity is required to exceed 30 cm/s, mesoscale requirements are assumed to be violated, and the data from the last successful set of input conditions are substituted as a solution for the current input conditions.

Evaporating precipitation is assumed to fall to levels below the observed cloud base in nonprecipitating, stratiform clouds.

For nonprecipitating, stratiform clouds, the precipitation rate at cloud base is assumed to be 0.2 mm/h unless the mean updraft velocity is required to be greater than 10 cm/s to generate such a rate. For those situations, the cloud base precipitation rate is reduced successively by 3/4 until the mean generating updraft falls below 10 cm/s.

Variables assumed to vary linearly with surface temperature are

- mixing level height
- storm speed
- storm efficiency
- storm width

The fall speeds and collection rates of precipitation depend on the classification of precipitation as snow or rain which, in turn, depends on the local air temperature.

Precipitation embryos are assumed to be ice crystals, which grow at the top of the cloud by vapor deposition until they reach dimensions of 400 μm . Growth then continues by vapor deposition and accretion as the crystals fall.

Condensation onto liquid precipitation is neglected.

The number of ice crystals per unit volume remains constant at each height.

If ice crystal multiplication is ignored, the ice crystal concentrations are determined by cloud top temperature.

If ice crystal multiplication is included, the actual crystal concentration at height z depends linearly on the maximum cloud water concentration between height z and the height at the -10°C level.

The total ice crystal concentration is not allowed to exceed 30 crystals/L.

Ice crystal multiplication is only allowed in the temperature interval between 0 and -10°C .

Constants are assumed to have the values:

- surface pressure = 980 mb
- surface height = 0 meters (height above surface)
- surface humidity = 90% if $T > 0^{\circ}\text{C}$,
- surface humidity = 99% if $T < 0^{\circ}\text{C}$, = 50% if the cloud is nonprecipitating
- collection efficiency for water = 0.65
- collection efficiency for ice = 0.2
- Marshall-Palmer intercept N_w for water = $1 \times 10^7 \text{ m}^{-4}$
- Marshall-Palmer intercept N_s for ice = $5 \times 10^7 \text{ m}^{-4}$
- cloud drop concentration = $500/\text{cm}^3$

10.0 REFERENCES

- Anthes, R. A. 1985. Summary of Meteorological Data Sets from PSV/NCAR Meso- α Scale (Regional) Limited-Area Model NCAR Ms. 0901-85-02, National Center for Atmospheric Research, Boulder, Colorado.
- Berkowitz, C. M., and R. C. Easter. 1986. Interim User's Guide to the RADM Scavenging Module (RSM) (Version 1.0). Atmospheric Sciences Research Laboratory, U.S. Environmental Protection Agency, Research Triangle Park, North Carolina.
- Berry, E. X., and R. L. Reinhardt. 1974. "An Analysis of Cloud Drop Growth by Collection." J. of Atmos. Sci. 31:2118-2126.
- Easter, R.C., and J. M. Hales. 1984. Pluvius: A Generalized One-Dimensional Model of Reactive Pollutant Behavior, Including Dry Deposition, Precipitation Formation, and Wet Removal. 2nd ed. PNL-4046, Pacific Northwest Laboratory, Richland, Washington.
- Fletcher, N. H. 1969. The Physics of Rainclouds. Cambridge University Press, London.
- Kessler, E. 1969. "On the Distribution and Continuity of Water Substance in Atmospheric Circulation." Met. Monogr. 32:84.
- Scott, B. C. 1982. "Predictions of In-Cloud Conversion Rates of SO_2 to SO_4 Based Upon a Simple Chemical and Kinematic Storm Model." Atmos. Environ. (16)7:1735.

APPENDIX A

SAMPLE OUTPUT

APPENDIX A

SAMPLE OUTPUT

TG= 1.500E+01 TT=-3.610E+01 XN0= 1.000E+07
 MULT= 0 NOPRE= 0 CL= 0.150000E+00 EFF= 0.490476E+00 XNC= 0.103640E+02
 EI= 0.200000E+00 SHIR= 0.297071E+00 ZMIX= 0.128571E+04 UG= 0.569048E+01
 XJ= 0.100000E+01 EW= 0.650000E+00 BPN0= 0.500000E+08 H= 0.800000E+04
 W= 0.399393E-01 HF= 0.294036E+04 ZBASE= 0.333000E+03 RHG= 0.900000E+00
 STORM HEIGHT (M)= 8.000E+03
 CLOUD BASE HEIGHT (M)= 3.330E+02
 FREEZING LEVEL HEIGHT (M)= 2.940E+03
 MIXING LEVEL HEIGHT (M)= 1.286E+03
 SURFACE PRECIPITATION RATE (MM/HR)= 1.000E+00
 SURFACE TEMPERATURE (K)= 2.881E+02
 SURFACE HUMIDITY= 9.000E-01
 STORM EFFICIENCY= 4.905E-01
 WATER COLLECTION EFFICIENCY= 6.500E-01
 ICE COLLECTION EFFICIENCY= 2.000E-01

GRID	CLOUD WATER (G/M ³)	PRECIP WATER (G/M ³)	SCAV COEF (1/SEC)	DEPOS-EVAP COEF (1/S)	UPDRAFT SPEED (M/S)
1	0.000E+00	7.673E-02	3.859E-04	-1.346E-04	0.000E+00
3	9.791E-02	8.804E-02	3.988E-04	0.000E+00	2.257E-02
5	1.741E-01	7.888E-02	3.570E-04	0.000E+00	3.983E-02
7	2.335E-01	6.245E-02	3.241E-04	5.727E-06	5.178E-02
9	6.281E-02	2.308E-01	2.688E-04	7.029E-05	5.842E-02
11	7.890E-05	1.480E-01	1.638E-04	5.639E-05	5.974E-02
13	6.812E-06	8.452E-02	8.644E-05	3.945E-05	5.576E-02
15	1.776E-06	3.945E-02	3.597E-05	2.300E-05	4.647E-02
17	1.068E-06	1.286E-02	0.000E+00	9.849E-06	3.186E-02
19	8.286E-07	1.490E-03	0.000E+00	1.477E-06	1.195E-02

<u>GRID</u>	<u>HEIGHT (M)</u>	<u>PRECIP RATE (MM/HR)</u>	<u>FALL SPEED (M/S)</u>	<u>AUTO (1/S)</u>
1	2.105E+02	1.000E+00	3.620E+00	0.000E+00
3	1.053E+03	1.180E+00	3.747E+00	0.000E+00
5	1.895E+03	1.024E+00	3.645E+00	0.000E+00
7	2.737E+03	7.614E-01	3.439E+00	0.000E+00
9	3.579E+03	5.628E-01	7.357E-01	0.000E+00
11	4.421E+03	3.485E-01	7.137E-01	0.000E+00
13	5.263E+03	1.920E-01	6.868E-01	0.000E+00
15	6.105E+03	8.599E-02	6.520E-01	0.000E+00
17	6.947E+03	2.648E-02	6.038E-01	0.000E+00
19	7.789E+03	2.730E-03	5.211E-01	0.000E+00

APPENDIX B

PROGRAM LISTINGS

APPENDIX B

PROGRAM LISTINGS

STRATIFORM CLOUD LISTINGS

```
C ... COMMON LIST IN 'SCAV.FOR'
C
C ... L=NUMBER OF VERTICAL GRIDS, LP1=L+1, LSAVE=LP1*NUMBER OF
C   VARIABLES IN BLOCK /SET/
C
      PARAMETER (L=20,LP1=L+1,LSAVE=18*LP1)
      COMMON /SET/ R(LP1),RC(LP1),W(LP1),ALPH(LP1),DEPOS(LP1)
1 PRATE(LP1),FALL(LP1),AUTO(LP1)
2, POUT(LP1),COUT(LP1),FEN_UP(LP1),FUP_EN(LP1),FUP_DN(LP1)
3, FEN_DN(LP1),FDN_EN(LP1),FDN_UP(LP1),TEMP(LP1),TEMPO(LP1)
      COMMON /SET2/ AXNO,BPNO,EI,EW,Z10,TOP,ZBASE,HFU,HFD,NOPRE,
1 C012(LP1),DROPS
      COMMON /SET4/ MULT,AIR(3),Q(5),QI(5),AV(5),AVI(5)
      COMMON /SET5/ VAP(LP1),SORCE(LP1),SORCEI(LP1),SDEP(LP1),RS(LP1),
1 RCS(LP1),VAPI(LP1)
      COMMON /BK1/ G(LP1),UOUT(LP1),VIN(LP1),VO(LP1)
      COMMON /BK2/ DATA(12),RHO(L)
      COMMON /BK4/ PRESSD(L),RHOD(L)
      COMMON SAVE(LSAVE),SAVER(LSAVE)
```

```
-----

      SUBROUTINE WATER(RHG,IFLG,XJ1)
      INCLUDE 'SCAV.FOR'
C
C
C ... WATER COMPUTES PROFILES OF PRECIPITATION CONCENTRATION,
C   PRECIPITATION RATE, PARTICLE FALL SPEED, AND SCAVENGING
C   COEFFICIENTS.
```

```

C
C
C ... VARIABLES IN CALL LIST:
C     RHG=SURFACE RELATIVE HUMIDITY (DECIMAL NOT %)
C     IFLG= COUNTER FOR DETERMINING 1ST AND SUCCESSIVE TIMES
C           THROUGH WATER
C     XJ1=COMPUTED SURFACE PRECIPITATION RATE (MM/HR)
C
C
C ... FLAGS:
C     MULT= 0 FOR NO ICE MULTIPLICATION
C           = 1 FOR ICE MULTIPLICATION
C     NOPRE=0 FOR PRECIPITATING CLOUD
C           =1 FOR NON PRECIPITATING CLOUD
C     IFLG= 0 WHEN CLOUD AND PRECIP PROFILES NEED TO BE INITIALIZED
C           = 1 WHEN INITIALIZATION ALREADY DONE
C     JFLG= 0 WHEN CLOUD WATER PROFILE HAS BEEN CHANGED AND 'XMULT'
C           MUST RE-INITIALIZE SOME VARIABLES
C           = 1 WHEN 'XMULT' HAS DONE THE RE-INITIALIZATION.
C
C
C     DIMENSION XLIM(7),SUMC(L),SUMD(L),FLUX(L),PX(L),PXI(L)
C     DIMENSION SRC1(L),GX(L),SUP(L),RBAR(L),ICONV(L)
C     XLIM(3)=1.E-14
C     XLIM(4)=1.E-5
C
C
C ... OTHER DEFINITIONS:
C     AXNO=VALUE OF  $E \cdot N_0^{**0.22}$  FOR RAIN
C     BPNO=VALUE OF  $N_0$  FOR SNOW
C     EW=WATER COLLECTION EFFICIENCY.
C     EI=ICE COLLECTION EFFIC.
C     HF=FREEZING LEVEL HEIGHT (M)
C     R(I)=CONCENTRATION OF CLOUDWATER AT LEVEL I (G/M3)

```

```

C      RC(I)=CONCENTRATION OF PRECIP WATER AT LEVEL I (G/M3)
C      ZBASE=CLOUD BASE (M)
C      H=STORM HEIGHT (M)
C      XMO=ICE CRYSTAL MASS AT TOP OF RIMING LEVEL (G)
C      XJ=PRECIPITATION RATE (MM/HR)
C      L=NUMBER OF VERTICAL GRIDS (DEFINED IN SCAV.FOR)
C
      BBP=BPNO**0.0687/2.749
C ... USED IN COMPUTING EVAPORATION
      COE=-ALOG(RHG)/ZBASE/ZBASE
C
C ... UNLOAD DATA ARRAY
      H=DATA(1)
      HF=HFU ! FREEZING LEVEL HT
      XMO=DATA(2)
      P=DATA(8)
      XJ=DATA(7)
C
C ... COMPUTE GRID SPACING (DZ)
      LL=L-1
      DZ=H/FLOAT(LL)
C
C
C ... CAN SKIP TO 110 IF PROFILES INITIALIZED ON FIRST PASS
      IF(IFLG .GT. 0) GO TO 110
C
C ... RNO AND RNO2 ARE No**.125 AND No**.250, RESP.
      RNO=(AXNO/EW)**0.5625
      RNO2=RNO*RNO
C
C ... INITIALIZE CLOUD AND PRECIP PROFILES
      XJ1=XJ
C

```

```

C ... FOR NON-PRECIPITATING CLOUD SET BELOW CLOUD PRECIP CONCENTRATION
C   VERY SMALL.
      IF(NOPRE .EQ. 1) XJ1=1.E-3
C
C ... LOOP 100 COMPUTES FIRST GUESS OF PRECIPITATION PROFILE (RC) AND
C   DETERMINES GRIO INDEX OF CLOUD BASE (IBASE) FOR A NON-PRECIPITATING
C   CLOUD.
      Z=-DZ
      DO 100 I=1,L
      Z=Z+DZ
      R(I)=0.
C ... BELOW CLOUD BASE, CLOUD WATER CONCENTRATION (R) IS ZERO.
      IF(Z+DZ*.5 .GT. ZBASE) GO TO 103
      IBASE=I+1
C
C ... INITIALLY ASSUME PRECIP RATE VARIES LINEARLY FROM CLOUD BASE TO
C   CLOUD TOP AND IS CONSTANT BELOW CLOUD BASE.
C
103  IF(I .LE. L) R(I+1)=0.1
      RC(I)=XJ*(1.-(Z-ZBASE)/(H-ZBASE))
      IF(Z+DZ*.5 .LT. ZBASE) RC(I)=XJ1
      IF(RC(I) .LT. 0.) RC(I)=0.
C ... FROM PRECIP RATE DATA, COMPUTE PRECIP CONCENTRATION
      IF(Z .LT. HF) GO TO 102
      RC(I)=RC(I)**0.936*0.32*BBP
      GO TO 100
102  RC(I)=RC(I)**0.88889*1.238E-2*(AXNO/EW)**0.5
100  CONTINUE
      IF(NOPRE .EQ. 0) IBASE =1
      R(1)=0.
C
C
C   *****DEPOSITIONAL ICE GROWTH CALCULATIONS*****
C

```

```

C
C
C ... LOOP 350 COMPUTES PART OF THE DEPOSITIONAL GROWTH RATE.
C
      Z=-DZ
      DO 350 K=1,L
C
      Z=Z+DZ
C ... DEPOSITIONAL GROWTH ONLY IF ABOVE FREEZING LEVEL.
      SRC1(K)=0.
      IF(Z .GT. HF) GO TO 201
      SDEP(K)=0.
      SUMO(K)=0.
      GO TO 350
C
C ... HERE S IS PART OF THE DEPOSITIONAL GROWTH RATE (G/M2/S * M/S)
C   IT NEEDS TO BE DIVIDED BY THE PRECIP FALL SPEED TO GIVE A
C   GROWTH RATE.
C
C ... COMPUTE ABSOLUTE VALUE OF THE TEMPERATURE (T)
201  T=ABS(TEMP(K)-273.16)
      S=1.
      IF(T .LT. 10.) S=D.98*T**0.5
      IF(T .LT. 20. .AND. T .GE. 10.) S=3.2
      IF(T .LT. 40. .AND. T .GE. 20.) S=5.4-0.11*T
      SUP(K)=S*1.E-8
350  CONTINUE
C
C
C *****CONDENSATION CALCULATIONS*****
C
C

```

```

C ... LOOP 300 COMPUTES THE CONDENSATION RATE (SX--G/M3/S) AS A FUNCTION
C   OF Z AND STORES IT IN SORCE
      SORCE(1)=0.
      SORCEI(1)=0.
      DO 300 I=2,LL
        KK=I-1
        RBAR(KK)=(R(KK)+R(I))/2.
        SXI=ALOG(RHO(I+1)/RHO(KK))/4.
        SX=VAP(I)-VAP(KK)-(VAP(I)+VAP(KK))*SXI
        SXI=VAPI(I)-VAPI(KK)-(VAPI(I)+VAPI(KK))*SXI
        SX=-SX
        SXI=-SXI
        IF(SX .LT. 0.) SX=1.E-8
        IF(SXI .LT. 0.) SXI=SX
        PX(I)=SX
300    PXI(I)=SXI
C
110    Z=-DZ
        D=0.
        DO 310 I=1,LL
          Z=Z+DZ
          KK=I+1
C
C ... 'COEF' RETURNS THE VELOCITY TERMS FOR THE HORIZONTAL FLUX OF
C   PRECIPITATION & CLOUD WATER AT LEVEL Z
C
        V=-2.
        CALL COEF(I,DZ,Z,D,V,A1,GX(I))
        SORCE(I)=(PX(I)*W(I)+PX(KK)*W(KK))/2.
        SORCEI(I)=(PXI(I)*W(I)+PXI(KK)*W(KK))/2.
310    SRC1(I)=SORCE(I)
C
C
C   END OF ICE GROWTH CALCULATIONS

```



```

C
C
C *****CLOUD WATER CALCULATIONS*****
C
      BARM2=1
      P1=19.53*DZ
      IF(Z10 .GT. H) Z1D=H
      DO 2000 KL=1,20
C ... DEPOSITIONAL GROWTH RATES MUST BE DETERMINED ONLY 1ST TIME THROUGH
C      UNLESS ICE MULTIPLICATION IS OCCURRING
      IF(MULT .NE. 1 .AND. KL .GT. 1) GO TO 301
C
C
C ... LOOP 360 COMPUTES PART OF THE DEPOSITIONAL GROWTH RATE.
C
      XNC=P**0.621*P1
      Z=-DZ
      DO 360 I=1,L
      Z=Z+DZ
C
C ... DEPOSITIONAL GROWTH ONLY IF ABOVE FREEZING LEVEL.
      IF(Z .LT. HF) GO TO 360
      XNT=P
C
C ... NO MULTIPLICATION FOR TEMPERATURES COLDER THAN -10.
      IF(Z .GT. Z10) GO TO 303
C
C ... 'XMULT' DETERMINES AN ICE CONCENTRATION BASED UPON TOTAL CLOUD
C      WATER IF ICE MULTIPLICATION IS OCCURRING.
      CALL XMULT(P,DZ,H,Z,XNT)
C
303  SDEP(I)=XNC*SUP(I)
      XNC=XNT**0.621*P1
360  CONTINUE

```

```

C
301  CONTINUE
      IF(BARM2 .GT. 0.) BARS=BARM2
C
C ... COMPUTE CLOUD WATER CONCENTRATION (R)
C
C
C
C ... LOOP 210 PROCEEDS FROM TOP DOWNWARD TO CALCULATE CLOUD WATER (R)
C   AS A FUNCTION OF CURRENT PRECIP WATER CONTENT (RC)
C
      IF(KL .EQ. 1) GO TO 212
      Z=-DZ
      DO 210 I=1,LL
      Z=Z+DZ
      KK=I+1
      IF(Z+DZ/2. .LE. ZBASE) GO TO 210
C
      FR=0.
C ... COMPUTE AUTOCONVERSION COEFFICIENT
      AUTO(I)=0.
      IF(RBAR(I) .LE. 0.) GO TO 200
      XXC=RBAR(I)/DROPS
C
C ... THRESHOLD FOR AUTOCONVERSION (MEAN RADIUS=8.75 MICRONS)
      IF(XXC .LT. 2.348E-9) GO TO 200
      XBAR=RBAR(I)
      AUTO(I)=5.5E+9*XBAR*XBAR*XXC**1.667*(1.-(2.35E-9/XXC)**1.33)
1  *(1.-(2.04E-9/XXC)**0.333)*DZ
      IF(AUTO(I) .LT. 0.) AUTO(I)=0.
      FR=FR-AUTO(I)
C

```

```

200  SRC1(I)=SORCE(I)
      IF(RBAR(I) .LE. 0. .AND. Z+DZ/2. .GT. HF) SRC1(I)=SORCEI(I)
      IF(Z .LT. ZBASE) SRC1(I)=SRC1(I)/2.
C
      FR=FR+SRC1(I)-SUMD(I)
      FRR=GX(I)+SUMC(I)/2.
      R(KK)=(R(I)*(W(I)-FRR)+FR)/(W(KK)+FRR)
      IF(R(KK) .LT. 0.) R(KK)=0.
      RBAR(I)=(R(I)+R(KK))/2.
      IF(KL .GT. 2 ) RBAR(I)=(RCS(I)+RBAR(I))/2.
      IF(RBAR(I) .LT. 1.E-5) RBAR(I)=0.
      RCS(I)=RBAR(I)
210  CONTINUE
C
C  ***** PRECIP WATER CALCULATION *****
C
212  Z=H
      RC(L)=0.
      FLUX(L)=0.
      PRATE(L)=0.
      FALL(L)=0.
C
      BARM2=0.
      JKNT=0
      DO 2550 IA=2,L
        I=L+1-IA
        XLIM(4)=1.E-5      ! LIMITS FOR NEUTON
        Z=Z-DZ
        KK=I+1
        SUMC(I)=0.
        SUMD(I)=0.
        FLUXIN=FLUX(KK)*RC(KK)
        RSAV=RC(KK)
C

```

```

C
EVAP=0.
VP=0.
VD=0.
DMDT=0.
IVALUE=2
IF(FLUXIN .GT. 0.) GO TO 256 ! ZERO VALUES BELOW THIS LEVEL
SD=(SDEP(KK)+SDEP(I))/2.
C=0.
S6=0.
ISTART=2
JLIMIT=12 ! NUMBER OF GUESSES BEFORE ZERO SEARCH
C
IF(Z+DZ*0.5 .GT. ZBASE) GO TO 1999
IF(ABS(FLUXIN) .GT. 5.E-7) GO TO 209
RC(I)=0. ! EVAPORATE REMAINDER BELOW I
GO TO 256
C
C ... EVAPORATION BELOW CLOUD BASE
209 RH=RHG*EXP(COE*Z*Z)
RH=5.4E-4*VAP(I)*(1.-1./RH)
C
C
1999 KUP=1
IF(ISTART .EQ. 2) GO TO 1998
RO=10**EXPON(RC(KK)) ! EXPONENT OF STARTING GUESS
IF(Z .GT. ZBASE) RC(I)=RO ! SHOULD GROW FROM UPPER LEVEL
IF(Z .LE. ZBASE) RC(I)=RO/1000 ! ALLOW FOR EVAPORATION
1998 DO 3000 IVALUE=ISTART,2
3001 DO 260 J=1,20
JKNT=JKNT+1
C
RPBAR=(RSV+RC(I))/2.
C

```

```

C ... COMPUTE SCAVENGING COEF. (C--1/S) AND FALL SPEED (VP) FOR PRECIP
      IF(Z .LT. HF) GO TO 251
      IF(RC(I) .LT. 2.44E-4) GO TO 254
      VP=RC(I)**0.0684/BBP
      GO TO 252
254  VP=2.43*RC(I)**0.2
252  C=8.829E-3*EI/BBP*RPBAR**0.0684
      IF(RC(I) .LT. XM0) C=0.
C
C ... 'VDEP' ACCOUNTS FOR A PSEUDO-DIFFUSION ARISING FROM A VARIATION
C     IN FALL SPEED WITH PARTICLE SIZE.
      VD=0.
      IF(VP .GT. 0.) CALL VDEP(1,DZ,BPNO,VP,I,VD)
C
C ... INCLUDE CONTRIBUTION OF DEPOSITIONAL MASS TO TOTAL MASS.
251  VBAR=VP-(W(I)+W(KK))/2.
      IF(VBAR .LT. 0.1) VBAR=0.1
      DMDT=((RC(KK)**0.62111802+SD/VBAR)**1.61-RC(KK))*VBAR
C
C ... LIMIT DEPOSITIONAL GROWTH RATE TO WATER CONDENSATION RATE
C     IF CLOUD WATER EXISTS, OTHERWISE LIMIT TO ICE CONDENSATION
C     RATE.
C
      DM=SRC1(I)-DMDT
      IF(DM .LT. 0.) DMDT=SRC1(I)
      IF(Z .GE. HF) GO TO 253
C
C ... SEARCH FOR THE FREEZING LEVEL
      IF(Z+DZ .LT. HF) GO TO 2252
C ... FREEZING LEVEL IS IN INTERVAL Z TO Z+DZ
C

```

```

C ... USE AVERAGE OF ACCRETION COEFFICIENTS NEAR FREEZING HT
  C1=0.
  IF(RC(I) .GT. 0.) C1=7.008E-4*RNO*EW*RC(I)**(-0.125)
  C2=8.829E-3*EI/BBP*RC(KK)**0.0684
  S6=(C1*RC(I)+C2*RC(KK))/2.*DZ
  GO TO 2251

C
2252 DMDT=0.
  IF(RPBAR .EQ. 0.) GO TO 2251
  C=7.008E-4*RNO*EW/RPBAR**0.125
2251 IF(RC(I) .GT. 0.11) VP=RC(I)**0.125/RNO*38.8496
  IF(RC(I) .LE. 0.11) VP=385.267/RNO2*RC(I)**0.25
  VD=0.
  IF(VP .GT. 0.) CALL VDEP(2,DZ,RNO2,VP,I,VD)

C
  IF(S6 .GT. 0.) GO TO 255
253 S6=C*OZ*RPBAR
255 C=S6*RPBAR(I)

C
C ... EVAPORATE PRECIP IF BELOW CLOUD BASE
  IF(Z+DZ*.5 .LE. ZBASE) EVAP=RH*RPBAR**0.65*DZ
  IF(Z+DZ*.5 .LE. ZBASE) DMDT=0.

C ... HORIZONTAL PRECIPITATION FLUX TERM
  S5=RPBAR*GX(I)*2.

C ... ACCRETION PLUS AUTOCONVERSION TERM
  SY=C+AUTO(I)
  EMAX=FLUXIN-SY+S5 ! MAX POSSIBLE EVAPORATION (NEGATIVE)
  IF(EMAX .GT. 0.) EMAX=-100.
  IF(EVAP .LT. EMAX) EVAP=EMAX

C ... DEPOSITIONAL GROWTH TERM
  S3=OMDT+EVAP

C ... FALLOUT TERM
  FLUX(I)=W(I)-VP-VD

C

```

```

C ... 'NEUTON' ADJUSTS RC(I) SUCH THAT DIFF1=0
      DIFF1=-FLUXIN+SY+S3-S5+FLUX(I)*RC(I)
C
      IF(IVALUE .EQ. 2) GO TO 261 ! NOT IN ZERO SEARCH
      IF(KUP .GT. 1) GO TO 256 ! IN ZERO SEARCH, 2ND
C                                     OR GREATER GUESS
      IF(Z .GT. ZBASE) GO TO 256 ! KUP=1, Z > ZBASE
      IF(DIFF1 .LT. 0.) GO TO 262
      DIFF2=-FLUXIN+SY+S3-S5 ! CHECK FOR TOTAL EVAP
C                                     BELOW CLD BASE, KUP=1
      IF(DIFF2 .GT. 0.) GO TO 256
262  RC(I)=0. ! ALL EVAPORATES BELOW CLD BASE
      GO TO 2552
C
261  CONTINUE
      CALL NEUTON(DIFF1,RC(I),XLIM,J,IFLAG)
C
      SUMC(I)=S6
      SUMD(I)=OMDT
      IF(IFLAG .EQ. 1) GO TO 256
      IF(J .LT. JLIMIT) GO TO 260
      IF(ISTART .EQ. 1) STOP 'ISTART=1, NEAR 260'
      ISTART=1 ! CHECK TO SEE IF ANY SOLUTION EXISTS
      JLIMIT=19 ! ALLOWS FOR CONVERGENCE NEAR ZERO
      GO TO 1999
260  CONTINUE
C
C ... SAVE FALLSPEED (VP-M/S) AND PRECIP RATE (PRATE--G/M2/S)
256  IF(IVALUE .EQ. 2) GO TO 2552
      CALL ZERO1(3.,RC(I),DIFF1,VALUE,KUP,JFLG)
      GO TO(3001,3000,3002) JFLG
3002  WRITE(6,900) I
      STOP ' AT 3002, NO SOLUTION WATERNEW '
3000  CONTINUE

```

```

2552  CONTINUE
      FLUX(I)=W(I)-VP-VD
      ALPH(I)=SUMC(I)/DZ
      PRATE(I)=RC(I)*(VP-W(I))
      RSAV=RC(I)
      FALL(I)=VP
      DEPOS(I)=(SUMD(I)+EVAP)/DZ
      BARM2=BARM2+RC(I)
C
901   FORMAT(1H ,3(I3,1X),6(1X,1PE10.3))
900   FORMAT(1H ,I3,6(1X,1PE10.3))
2550  CONTINUE
C
C
C ... GO AT LEAST 3 TIMES BEFORE A CONVERGENCE CHECK IS MADE
      IF( KL .LT. 3) GO TO 2000
      IF(ABS(BARS) .LT. 1.E-9) GO TO 2002
      BARS=(BARS-BARM2)/BARS
      IF(ABS(BARS) .LT. 5.E-5) GO TO 2002
C
C ... IF HERE, NO CONVEGENCE. TO REDUCE UNSTABLE WAVES IN VERTICAL
C     PROFILE, USE, FOR THE NEW VALUES OF PRECIP CONCENTRATION, THE
C     AVERAGE OF THE RECENTLY COMPUTED PRECIP WATER AND THE PREVIOUS
C     AVERAGED VALUE OF PRECIP WATER.
C
      DO 405 I=1,L
      IF(KL .EQ. 3) GO TO 406
      R1=RS(I)
      RC(I)=(R1+RC(I))/2.
406   RS(I)=RC(I)
405   CONTINUE
2000  CONTINUE
C

```



```

        WRITE(6,801) PRATE(IBASE)*3.6
801  FORMAT(1H ,' WARNING,KL TOO BIG,LOOP 2000 WATER, NO CONVERGENCE
      1, PRATE=',1X,1PE10.3)
C
C
800  FORMAT(1H ,I5,5(1X,1PE13.6))
C
C ... XJ1 IS THE COMPUTED VALUE OF PRECIP RATE AT THE SURFACE OR
C     CLOUD BASE. COMPARE (IN 'MAIN') WITH THE DESIRED VALUE.
2002 XJ1=PRATE(IBASE)*3.6
      DATA(3)=XNT/1000.
      RETURN
      END

```

```

      FUNCTION EXPON(R)
C ... DETERMINES EXPONENT OF CONCENTRATION R
      EXPON=-10
      IF( R .LE. 0) RETURN
      I=-ALOG10(R)+1
      EXPON=-FLOAT(I)
      RETURN
      END

```

```

      SUBROUTINE WINDS(E,ZMIX,RHG,UG,XX,IFLG)
      INCLUDE 'SCAV.FOR'
      DIMENSION XLIM(7)
      DATA XLIM/0.,0.,1.E-8,1.E-4,0.,15.,-1./

```

```

C
C ... COMPUTES VERTICAL PROFILES OF U,V,W GIVEN A STORM EFFICIENCY AND
C     AN AVERAGE STORM UPDRAFT VELOCITY. THE VERTICAL VELOCITY PROFILE
C     IS PARABOLIC. IF THE DESIRED EFFICIENCY CANNOT BE ACHIEVED, THE
C     LARGEST POSSIBLE EFFICIENCY IS USED AS A DEFAULT VALUE.
C
C ... VARIABLES IN CALL LIST:
C     E=STORM EFFICIENCY (PRECIP RATE/VAPOR INFLOW)--INPUT
C     ZMIX=MIXING LEVEL HEIGHT (M)--INPUT
C     RHG=SURFACE RELATIVE HUMIDITY (DECIMAL-NOT %)--INPUT
C     UG=STORM SPEED (M/S)--COMPUTED
C     XX=STORM WIDTH (M)--INPUT
C     IFLG=COUNTER--INPUT
C
C ... FLAGS:
C     IFLG=COUNTER GIVING NUMBER OF PREVIOUS CALLS TO 'WINDS'
C     JFLG=0 WHEN CONVERGENCE HAS NOT OCCURRED FOR PROPER EFFICIENCY
C         =1 WHEN CONVERGENCE HAS OCCURRED FOR EFFICIENCY
C     NOPRE=0 FOR PRECIPITATING CLOUD
C         =1 FOR NON-PRECIPITATING CLOUD
C
C ... UNITS ARE:
C     LENGTHS--METERS
C     VELOCITY--METERS/SECOND
C     HUMIDITY--DECIMAL RATHER THAN %
C     DENSITY--GRAM/M3
C
C ... OTHER DEFINITIONS:
C     R1=RATIO OF GRID WIDTH TO LENGTH
C     HF=FREEZING LEVEL HEIGHT (M)
C     ZBASE=CLOUD BASE (M)
C     L=NUMBER OF VERTICAL GRIDS (DEFINED IN SCAV.FOR)
C
C     R1=1.0

```

```

C
C ... XJ=PRECIPITATION RATE (G/M2/S)
C ... H=STORM HEIGHT (METERS)
      H=DATA(1)
      XJ=DATA(7)/3.6
C
C ... COMPUTE VERTICAL GRID THICKNESS (DZ) AND CONSTANTS FOR VELOCITY
C      AND HUMIDITY CALCULATIONS (A1, COE)
      LL=L-1
      DZ=H/FLOAT(LL)
      Z=-DZ
      A1=6.*DATA(11)/H
      COE=-ALOG(RHG)/ZBASE/ZBASE
C
C
C ... COMPUTE VERTICAL VELOCITY--PARABOLIC PROFILE
C
C ... LOOP 100 COMPUTES PROFILES OF VERTICAL VELOCITY (W), VAPOR DENSITY
C      (VAP) AND AIR DENSITY
      DO 100 I=1,L
        Z=Z+DZ
900    FORMAT(1H ,I3,4(1X,1PE10.3))
        W(I)=A1*(Z-Z*Z/H)
C      A1=6*DATA(11)/8000.
C      W(I)=0.015
C      IF(Z .LT. 8000.) W(I)=A1*(Z-Z*Z/8000)
        IF(IFLG .GT. 0) GO TO 100
C ... OPERATIONS BELOW ONLY NEED TO BE DONE 1ST TIME THROUGH.
        ZX=Z+DZ/2.
C ... HUMIDITY VARIES LIKE EXP(Z*Z)
        RH=1.
        IF(ZX .LT. ZBASE) RH=RHG*EXP(COE*ZX*ZX)

        CO12(I)=RHO(I-1)/RHO(I)

```

```

      IF(I .EQ. 1) CO12(I)=1.0
      V=-1.
C ... 'COEF' RETURNS AVERAGE SATURATION VAPOR DENSITY IN THE INTERVAL
C   Z TO Z+DZ. VAP IS THE ACTUAL VAPOR DENSITY IN THE INTERVAL OZ.
      CALL COEF(I,DZ,Z,C,V,RV,B)
      VAP(I)=RV*RH
      VAPI(I)=B
100  CONTINUE
C
C
C ... COMPUTE U,V COMPONENTS BASED ON CONTINUITY EQUATION.
C
C ... VARIABLES TO ADJUST FOR EFFICIENCY=ZUP,UG,CB,CL,SHIR
C
C ... WIND SHEAR CONSTANTS (CB,CL)
C ... CB=4.*X(2)*XX/H/UG-1.--A POSSIBLE VARIATION TO TRY LATER.
      CB=0.
      CL=DATA(9)
C
C ... SEE IF MAX POSSIBLE EFFICIENCY > REQUESTED EFFICIENCY
C
      ZUP=ZMIX
      UG0=UG*2.
C
C ... LOOP 300 TRY TO REDUCE INFLOW (INCREASE EFFICIENCY) BY DECREASING
C   SPEED (UG OR UG0)
      DO 300 K=1,3
      SHIR=0.05
C ... FOR A NON PRECIPITATING CLOUD, EFFICIENCY DOES'NT MATTER--USE DEFAULT
C   OF SHIR=0.4
      IF(NOPRE .EQ. 1) SHIR=0.4
      UG0=UG0/2.
      A=UG0/XX*DZ
C

```

```

C ... LOOP 350 ADJUSTS SHIR TO GET PROPER EFFICIENCY
      DO 350 J=1,20
        SUM5=0.
        SUM6=0.
        Z=-0Z
C
C ... LOOP 200 COMPUTES HORIZONTAL WIND SPEEDS AND INFLOW VAPOR FLUX
C
C ... DEFINITIONS--HORIZONTAL WINDS (M/S)
C      G=WIND IN X-DIRECTION AT X ( POSITIVE INTO STORM)
C      UOUT= WIND IN X-DIRECTION AT X+DX ( NEGATIVE INTO STORM)
C      VIN=WIND IN Y-DIRECTION AT Y (POSITIVE INTO STORM)
C      VOUT=WIND IN Y-DIRECTION AT Y+DY (NEGATIVE INTO STORM)
C ... ALL HORIZONTAL WIND SPEEDS HAVE BEEN MULTIPLIED BY DZ/XX
C
      DO 200 I=1,LL
        Z=Z+DZ
        UOUT(I)=2.*A*(Z+DZ/2-H/2)/H
        IF(Z+DZ/2. .GT. ZUP) UOUT(I)=0.
        CZ=CB-CL*(1.+CB)*(Z+DZ/2.)/H
C ... CZ NE -1 REQUIRES CL < 1.
        G(I)=UOUT(I)+2.*(W(I+1)-C012(I+1)*W(I))/(1+CZ)/(1+C012(I+1))
        VIN(I)=UOUT(I)*SHIR*R1
        VO(I)=VIN(I)+CZ*(UOUT(I)-G(I))
        F=0.
        IF(G(I) .GT. 0.) F=F+G(I)
        IF(UOUT(I) .LT. 0.) F=F-UOUT(I)
        IF(VIN(I) .GT. 0.) F=F+VIN(I)
        IF(VO(I) .LT. 0.) F=F-VO(I)
C ... SUM5=INFLOW VAPOR FLUX, F=INFLOW WINDS, SUM6=OUTFLOW WINDS
        SUM5=SUM5+F*VAP(I)
        IF(UOUT(I) .GT. 0.) SUM6=SUM6+UOUT(I)
        IF(VO(I) .GT. 0.) SUM6=SUM6+VO(I)
        IF(G(I) .LT. 0.) SUM6=SUM6-G(I)

```

```

        IF(VIN(I) .LT. 0.) SUM6=SUM6-VIN(I)
200  CONTINUE
C
C ... EFF1=COMPUTED EFFICIENCY, E=DESIRED EFFICIENCY
      EFF1=XJ/SUM5
C ... 'NEUTON' TRIES TO ADJUST EFFICIENCY SO THAT DIFF1=0
      DIFF1=E-EFF1
C ... SKIP ITERATION IF CLOUD IS NON-PRECIPITATING
      IF(NOPRE .EQ. 1) E=EFF1
      IF(NOPRE .EQ. 1) GO TO 301
C ... EXPERIENCE SUGGESTS ABOUT 4 TRIES TO GET CLOSE TO PROPER UPDRAFT
C    SPEED. DON'T ADJUST E UNTIL THEN.
      IF(IFLG .LT. 4) GO TO 301
C ... IF DIFF1 > 0, MUST INCREASE COMPUTED EFFICIENCY
      IF(J .EQ. 1 .AND. DIFF1 .GT. 0.) GO TO 300
C
C ... ITERATE ON SHIR TO GET PROPER EFFICIENCY
C
802  FORMAT(1H ,I3,5(1X,E13.6))
      CALL NEUTON(DIFF1,SHIR,XLIM,J,JFLG)
      IF(JFLG .EQ. 1) GO TO 301
350  CONTINUE
C
      STOP 'NO CONVERGENCE ON EFFICIENCY IN WINDS'
300  CONTINUE
C
C ... IF HERE, IT IS NOT POSSIBLE TO HAVE EFFICIENCY AS LARGE AS
C    DESIRED
      E=EFF1
      PRINT 801,E
801  FORMAT(1H , ' DEFAULT E TAKEN TO BE ',E10.3)
C
C    PRINT 800,J,UGO,EFF1,SHIR
301  CONTINUE

```

```

800  FORMAT(1H0,' J, UGO, EFF1, SHIR, ',I3,3(1X,1PE10.3))
      DATA(4)=UGO
      DATA(10)=SHIR
      DATA(5)=E
      RETURN
      END

```

```

C ... PROGRAM STMAIN
C
      INCLUDE 'SCAV.FOR'
      COMMON /SET3/ BM1,CL,SHIR
      COMMON /BK3/ZLZ(L),PRESS(L)
C
C ... MAIN PROGRAM FOR CLOUD/SCAVENGING MODULE
C ... PRIMARILY SETUP AND INITIALIZATION OPERATIONS ARE DONE HERE
C
C ... UNITS ARE:
C      LENGHTS--METERS
C      PRESSURE---MB.
C      VELOCITY--METER/SECOND
C      TEMPERATURE--DEG C
C      RAINRATE--MM/HR
C      HUMIDITY--DECIMAL
C
C ... FLAGS:
C      MULT=0 FOR NO ICE MULTIPLICATION
C      MULT=1 FOR ICE MULTIPLICATION
C      NOPRE=0 FOR PRECIPITATING CLOUD
C      NOPRE=1 FOR NON PRECIPITATING CLOUD

```

```

C
    MULT=0
    NOPRE=0
C
C ... NT=# OF SURFACE TEMPERATURE INTERVALS BEING CONSIDERED
C ... NH=# OF CLOUD HEIGHT INTERVALS BEING CONSIDERED
C ... NP=# OF PRECIPITATION RATE INTERVALS BEING CONSIDERED
C      (OR IF THE NON-PRECIPITATING CLOUD CASE IS RUN)
C      =# OF CLOUD BASE INTERVALS BEING CONSIDERED
C
    NT=1
    NH=1
    NP=1
C
C
C ***** INPUT HEIGHT AS HEIGHT ABOVE GROUND LEVEL *****
C
C ... TG=INITIAL SURFACE TEMPERATURE TO CONSIDER ( C )
C ... HO=INITIAL CLOUD TOP HEIGHT TO CONSIDER (METERS)
C ... XJO=INITIAL SURFACE PRECIP RATE TO CONSIDER (MM/HR)
C      CAN START AT 0 MM/HR IF GENERATING TABLES FOR PRECIPITATING
C      CLOUDS. IF NON-PRECIP CLOUDS ARE BEING CONSIDERED, THIS
C      VARIABLE IS NOT USED.
C ... ZBASE1=INITIAL CLOUD BASE HEIGHT TO CONSIDER (METERS)
C
    HO=8000.
    TG=15.
    ZBASE1=333.
    XJO=1.0
C
C ... DH=INCREMENT TO USE WHEN VARYING CLOUD TOP AND BASE HEIGHT (M)
C ... DJ=INCREMENT TO USE WHEN VARYING SURFACE PRECIP RATE (MM/HR)
C ... DT=INCREMENT TO USE WHEN VARYING SURFACE TEMPERATURE (C)
C

```



```

        DH=-1000.
        DJ=1.0
        DT=10.

C
C ... INPUT SURFACE PRESSURE (PG), HUMIDITY (RHG)
        PG=980.
        RHG=0.9
        IF(NOPRE .EQ. 1) RHG=0.5

C
C ... CL IS USED TO ADJUST STORM EFFICIENCY
        CL=0.15

C
C ... INPUT COLLECTION EFFICIENCY FOR WATER (EW), AND ICE (EI)
        EW=0.65
        EI=0.2

C
C ... INPUT MAR.-PALMER No FOR WATER (XNO), AND ICE (BPNO)
        XNO=1.E+7
        BPNO=5.E+7
        AXNO=XNO**0.222*EW

C
C ... INPUT CLOUD DROP NUMBER (#/M3)
        DROPS=5.E8

C
C ... WRITE OUTPUT ON TAPE 10
        REWIND 10

C
C ... LOOP 200 VARIES SURFACE TEMPERATURE
        DO 200 I=1,NT
C ... DETERMINE INFLOW WIND (UG), STORM EFFIC. (E), MIXING LEVEL HT (ZMIX)
C ... AND STORM WIDTH (XX). ASSUME SEASONAL (TEMPERATURE) VARIATION FOR
C ... UG,E,ZMIX,XX
C
        TTG=TG

```

```

      IF(TG .LT. 4) TTG=4.
      IF(TG .GT. 25) TTG=25.
      FACT=(TTG-4.)/21.
      UG=14.-5.*FACT
      E=0.7-0.4*FACT
      ZMIX=(0.5+1.5*FACT)*1000.
      XX=(300.-215.*FACT)*1000.
C
      RH=RHG
      IF(TG .LT. 0. .AND. NOPRE .EQ. 0) RH=0.99
C
      H=H0
      ZBASE0=ZBASE1
C
C ... LOOP 400 VARIES STORM HEIGHT
      DO 400 K=1,NH
C ... ZERO STORAGE ARRAYS
      DO 500 M=1,LSAVE
      R(M)=0.
500   SAVE(M)=0.
      IF(NOPRE .EQ. 1) ZBASE0=ZBASE1
C
C ... DETERMINE TEMPERATURE AT FREEZING LEVEL (TM) AND CLOUD TOP (TT)
C   ALSO FIND HEIGHT OF FREEZING LEVEL (HF)
C
C ... LAPSE IS MOIST ADIABATIC. IF SURFACE TEMP IS BELOW
C   FREEZING, FIND TEMP AT H/10.
      HF=H
      Z10=H
C ... COMPUTE TEMPERATURE PROFILE ALONG MOIST ADIABAT
C
      CALL SIGMA(H,TG,PG)
C

```

```

      IF(TEMP(1) .LT. 273.16) HF=-1.
      DO 550 M=1,L
      IF(Z10 .EQ. H .AND. TEMP(M) .GT. 263.16) GO TO 502
C ... DETERMINE HEIGHT OF -10 LEVEL FOR ICE MULTIPLICATION (Z10)
      Z10=ZLZ(M)/100.  !HEIGHT IN METERS
      GO TO 501
C
C ... FIND FREEZING LEVEL HEIGHT
502  IF(TEMP(M) .GT. 273.16) GO TO 550
      IF(TEMP(1) .LE. 273.16) GO TO 550
      IF(TEMP(M-1) .LE. 273.16) GO TO 550
C
C
      HF=(273.16-TEMP(M-1))*(ZLZ(M)-ZLZ(M-1))/(TEMP(M)-TEMP(M-1))
1    +ZLZ(M-1)
      HF=HF/100.  !FREEZING LEVEL HEIGHT ABOVE GROUND-METERS
C
550  CONTINUE
501  TT=TEMP(L)-273.16  ! CLOUD TOP TEMP (C)
      HFU=HF
C
C ... SEE IF CLOUD TOP IS BELOW FREEZING
      IF(HF .LT. H) GO TO 402
      WRITE(6,803) H,TG
803  FORMAT(1H0,' CLOUD TOP IS LOWER THAN FREEZING LEVEL FOR H= ',
11PE10.3,' TG=',1PE10.3)
      IF(NOPRE .EQ. 1) GO TO 402
      CALL DEFALT(1)
C
      DO 403 J=1,NP
403  WRITE(10) R,RC,W,ALPH,DEPOS,PRATE,FALL,AUTO
      GO TO 400
C
C

```

```

C ... LOOP 100 VARIES SURFACE RAINRATE (XJ) OR CLOUD BASE IF
C   NOPRE=1. PUT INITIAL XJ HERE.
402  XJ=XJ0
      TEMP(L+1)=TEMP(L)
      TOP=TT
      WBAR=0.015
      IF(NOPRE .EQ. 1) XJ=0.2
C
C ... LIMIT CLOUD TOP TEMP TO > -50 DEG C
      IF(TT .LT. -50.) TT=-50.
      TOP=TT
      TO=-HF*TT/(H-HF)
      TL=-TT/(H-HF)
C
      DO 100 J=1,NP
      IF(WBAR .LT. 0.015) WBAR=0.015
      E=0.7-0.4*FACT
      IF(ZBASE0 .LT. H) GO TO 105
      WRITE(6,804) H
804  FORMAT(1H0,' CLOUD BASE .GT. H, H=',1PE10.3,' DEFAULT USED')
      CALL DEFALT(1)
      GO TO 102
C
105  IF(ZBASE0 .GE. 0.) GO TO 103
      WRITE(6,805)
805  FORMAT(1H0,' CLOUD BASE LOWER THAN THE SURFACE, DEFAULT USED ')
      CALL DEFALT (1)
      GO TO 102
C ... MOVE ZBASE OFF OF GRID POINT FOR EASE IN WATER
103  ZBASE=ZBASE0
C
C ... NUMBER OF GRID INTERVALS IS GIVEN BY L. L DEFINED IN SCAV.FOR
C

```

```

      XK=L-1
      XI=ZBASE*XK/H+0.5
      II=XI
      XK=II
      IF(XI .EQ. XK) ZBASE=ZBASE-1.
C
      WRITE(6,801) TG,TT,XNO
801  FORMAT(1H0,' TG=',1PE10.3,' TT=',1PE10.3,' XNO=',1PE10.3)
      WRITE(6,802) MULT,NOPRE
802  FORMAT(1H0,' MULT=',I3,' NOPRE=',I3)
      XJSAVE=XJ
      NOSAVE=NOPRE
      IF(NOPRE .EQ. 1) GO TO 101
      IF(XJ .GT. 0.) GO TO 101
C
C ... FIRST TIME THROUGH, ASSUME NON-PRECIP CLOUD
      NOPRE=1
      XJ=0.1
C ... THE BULK OF CALCULATIONS ARE DONE IN 'MAIN' OR 'WATER'
C
101  CALL MAIN1(E,H,XJ,RH,XX,ZMIX,WBAR,UG)
      NOPRE=NOSAVE
      XJ=XJSAVE

C
C
C ... OUTPUT INFO IS STORED IN COMMON BLOCK /SET/
102  WRITE(10) R,RC,W,ALPH,OEPOS,PRATE,FALL,AUTO
C
C
      IF(NOPRE .EQ. 1) GO TO 104
      XJ=XJ+DJ
      GO TO 100
C

```

```

104  ZBASE0=ZBASE0+DH
100  CONTINUE
C
400  H=H+OH
C
200  TG=TG+DT
      END

```

```

      SUBROUTINE MAIN1(E,H,XJO,RHG,XX,ZMIX,WBAR,UG)
      INCLUDE 'SCAV.FOR'
      COMMON /SET3/ BM1,CL,SHIR
      DIMENSION XLIM(7)
      DATA XLIM/0.,0.,1.E-8,1.E-4,0.,15.,-1./
      IOUT=6
C
C ... THIS ROUTINE ADJUSTS VERTICAL VELOCITY TO GIVE PROPER SURFACE
C    PRECIPITATION RATE. U,V,W COMPONENTS ARE COMPUTED HERE.
C ... SOME MICROPHYSICAL CONSTANTS ARE INITIALIZED. MUCH OF HARDCOPY
C    OUTPUT IS DONE HERE.
C
C
C ... VARIABLES IN CALL LIST:
C    E=STORM EFFICIENCY (PRECIP RATE/VAPOR INFLOW)--INPUT
C    H=STORM HEIGHT (M)--INPUT
C    XJO=PRECIP RATE (MM/HR)--INPUT
C    RHG=SURFACE RELATIVE HUMIDITY (DECIMAL NOT %)--INPUT
C    XX=STORM WIDTH (M)--INPUT
C    ZMIX=MIXING LEVEL HEIGHT (M)--INPUT
C    WBAR=AVERAGE VERTICAL VELOCITY (M/S)--COMPUTED

```

```

C      UG=HORIZONTAL STORM SPEED (M/S)--COMPUTED
C
C ... FLAGS:
C      IFLAG=0 WHEN NO CONVERGENCE OCCURS FOR VERTICAL VELOCITY
C      =1 WHEN VERTICAL VELOCITY CONVERGES
C      IFLG = COUNTER TO DETERMINE 1ST AND SUCCESSIVE TIMES THROUGH
C      'WINDS' AND 'WATER'
C
C ... TOTAL CRYSTAL NUMBER (XNC-#/M3) DEPENDS ON CLOUD TOP TEMP
C      call pnl$second(START)
C      TTOP=-TOP
C      IF(TTOP .LE. 0.) TTOP=0.
C      XNC=6.104E-3*TTOP**4.
C
C ... LIMIT THE TOTAL CRYSTAL NUMBER TO 30000/M3 (=30/L) TO
C      SIMULATE AGGREGATION AND PERMIT EARLIER RIMING
C      IF(XNC .GT. 3.E+4) XNC=3.E+4
C
C ... COMPUTE MASS OF 400 MICRON CRYSTAL WHICH HAS REACHED RIMING
C      SIZE (XMI-GRAMS), AND MASS OF ALL 400 MICRON CRYSTALS (XMO-GRAMS)
C      XMI=2.4E-6
C      XMO=XMI*XNC
C
C ... LOOP 1390 REDUCES CLOUD BASE PRECIP RATE FOR NON-PRECIPITATING
C      CLOUDS IF WBAR IS COMPUTED TO BE GREATER THAN 10 CM/S.
C      XJ=XJ0
C      WBAR0=WBAR
C      DO 1390 LOOP=1,5
C      WBAR=WBAR0
C      IFLG=0
C
C ... LOOP 1400 ITERATES TO FIND VERTICAL VELOCITY CORRESPONDING TO
C      INPUT SURFACE PRECIPITATION RATE. HERE WBAR IS VERTICAL VELOCITY

```

```

C      AVERAGED FROM 0 TO H.
      DATA(11)=WBAR
      DATA(1)=H
      DATA(2)=XMO
      DATA(7)=XJ
      DATA(8)=XNC
      DATA(9)=CL
      DO 1400 J=1,20
C
C ... 'WINDS' USES W PROFILE AND EFFICIENCY REQUIREMENTS TO COMPUTE
C      PROFILES OF HORIZONTAL WINDS U,V
      CALL WINDS(E,ZMIX,RHG,UG,XX,IFLG)
C
C ... 'WATER' COMPUTES PROFILES OF CONDENSED WATER
      CALL WATER(RHG,IFLG,XJ1)
900   FORMAT(1H ,I3,4(1X,1PE10.3))
      call pn1$second(end)
      cputime=end-start
C      print 927, cputime
927   format(1h , 'cputime in seconds:',1pe13.6)
      IFLG=IFLG+1
C
C ... HERE XJ1 IS SURFACE PRECIP COMPUTED WITH CURRENT GUESS OF W, AND
C      XJ IS THE DESIRED VALUE.
C ... 'NEUTON' TRIES TO ADJUST VERTICAL VELOCITY SO THAT DIFF1=0
      DIFF1=XJ-XJ1
C      PRINT 801,J,DIFF1,DATA(11)
      CALL NEUTON(DIFF1,DATA(11),XLIM,J,IFLAG)
      IF(IFLAG .EQ. 1) GO TO 1300
C      PRINT 801,J,DIFF1,DATA(11)
C

```



```

C ... STOP FOR VERTICAL VELOCITY TOO BIG
      IF(NOPRE .EQ. 1 .AND. DATA(11) .GT. 0.1) GO TO 1390
      IF(DATA(11) .GT. 4.) GO TO 1401
      IF(DATA(11) .GT. .0005 .OR. XJ1 .LT. XJ) GO TO 1400

C
C ... IF HERE, TO OBTAIN DESIRED PRECIP RATE, W NEEDS TO BE APPROX. =0
C      OR VERTICAL VELOCITY TOO BIG
1401  WRITE(IOUT,809) XJ,H,DATA(11)
809   FORMAT(1H0,' NO SOLUTION FOR J=',E13.6,' H=',E13.6,'---
      1 WBAR=',E13.6)
801   FORMAT(1H ,1X,I3,6(1X,1PE10.3))
802   FORMAT(1H0)
800   FORMAT(1H ,3(1X,I3),5(1X,1PE10.3))
      WBAR=.015
      CALL DEFALT(1)
      RETURN

C
1400  CONTINUE

C
C ... IF J=21, CONVERGENCE HAS'NT BEEN ACHIEVED
      WRITE(IOUT,801)J,DIFF1,XJ1
      WRITE(IOUT,814)
814   FORMAT(1H0,' NO CONVERGENCE IN MAIN')
      IF(ABS(DIFF1) .LT. 5.E-3) GO TO 1300
      CALL DEFALT(1)
      WRITE(IOUT,803)
803   FORMAT(1H0,' NO CONVERGENCE IN MAIN, DEFAULT USED')
      RETURN

C
1390  XJ=0.75*XJ
      GO TO 1401

C
C ... ALL OUTPUT BELDW HERE
C

```

```

C ... SAVE CURRENT VALUES
1300  WBAR=DATA(11)
      SHIR=DATA(10)
      WRITE(IOUT,805) CL,DATA(5),DATA(3),EI
      WRITE(IOUT,806) SHIR,ZMIX,DATA(4),XJ
      WRITE(IOUT,807) EW,BPNO,H
      WRITE(IOUT,808) DATA(11),HFU,ZBASE,RHG
C
      WRITE(IOUT,820) H,ZBASE,HFU,ZMIX,XJ,TEMP(1),RHG,DATA(5),EW,EI
820   FORMAT(1H0,' STORM HEIGHT (M)=' ,1PE10.3,'      CLOUD BASE
1HEIGHT (M)=' ,1PE10.3,/,' FREEZING LEVEL HEIGHT (M)=' ,1PE10.3
2,'      MIXING LEVEL HEIGHT (M)=' ,1PE10.3,/,' SURFACE PRECIP
3ITATION RATE (MM/HR)=' ,1PE10.3,/,' SURFACE TEMPERATURE (K)=
4' ,1PE10.3,/,' SURFACE HUMIDITY=' ,1PE10.3,'      STORM EFFICIE
5NCY=' ,1PE10.3,/,' WATER COLLECTION EFFICIENCY=' ,1PE10.3,/,'
6 ICE COLLECTION EFFICIENCY=' ,1PE10.3)
808   FORMAT(1HD,' W=' ,E13.6,' HF=' ,E13.6,' ZBASE=' ,E13.6,' RHG=' ,E13.6)
805   FORMAT(1H0,' CL=' ,E13.6,' EFF=' ,E13.6,' XNC=' ,E13.6,' EI=' ,E13.6)
806   FORMAT(1H0,' SHIR=' ,E13.6,' ZMIX=' ,E13.6,'UG=' ,E13.6,' XJ=' ,E13.6)
807   FORMAT(1H0,' EW=' ,E13.6,' BPNO=' ,E13.6,' H=' ,E13.6)
C ... CONVERT FROM G/M3 TO MOLES-WATER/M3
      DZ=L-1
      DZ=H/DZ
      DO 601 I=1,L
      AUTO(I)=AUTO(I)/DZ
      PRATE(I)=PRATE(I)*3.6
601   CONTINUE
      WRITE(IOUT,802)
      WRITE(IOUT,813)
813   FORMAT(1H0,2X,'GRID   CLOUD       PRECIP       SCAV       DEPOS-EVAP
1  UPDRAFT')
      WRITE(IOUT,810)

```

```

810  FORMAT(1H ,7X,'WATER      WATER      COEF      COEF
      1SPEED')
      WRITE(IOUT,821)
821  FORMAT(1H ,7X,'(G/M3)      (G/M3)      (1/SEC)      (G/M3/S)
      1      (M/S)',/)
      DO 600 I=1,L,2
600  PRINT 801,I,R(I),RC(I),ALPH(I),DEPOS(I),W(I)
      WRITE(IOUT,802)
      Z=-OZ/2.
C ... Z IS THE HEIGHT AT CENTER OF GRID
      WRITE(IOUT,811)
811  FORMAT(1H0,2X,'GRID  HEIGHT      PRECIP      FALL      AUTO')
      WRITE(IOUT,812)
812  FORMAT(1H ,'      RATE      SPEED')
      WRITE(IOUT,822)
822  FORMAT(1H ,9X,'(M)      (MM/HR)      (M/S)      (G/M3/S)')
      WRITE(IOUT,804)
804  FORMAT(1H )
      DO 1502 I=1,L,2
      Z=I
      Z=Z-0.5
      Z=DZ*Z
C ... CONVERT PRECIP RATE FROM G/M2/S TO MM/HR
1502  WRITE(IOUT,801) I,Z,PRATE(I),FALL(I),AUTO(I)
      WRITE(IOUT,802)
C
C ... SAVE CURRENT VALUES
      CALL UNITS ! CONVERT UNITS
      CALL DEFALT(0)
      RETURN
      END

```

```

-----

C .....

      SUBROUTINE COEF(I,DZ,Z,C,V,A,B)
      INCLUDE 'SCAV.FOR'

C
C ... IF V=-1, 'COEF' RETURNS:
C     AVERAGE SATURATION VAPOR DENSITY IN THE INTERVAL Z TO Z+DZ
C     IN UPDRAFT REGION
C ... IF V=0, 'COEF' RETURNS:
C     AVERAGE SATURATION VAPOR DENSITY IN THE INTERVAL Z TO Z+DZ
C     IN RAINSHAFT REGION
C ... IF V=-2, 'COEF' RETURNS THE TERMS IN THE DENOMINATOR OF THE
C     MASS FLUX EXPRESSIONS, I.E.: M(J)=SOURCE/DENONINATOR
C
C ... TERMS IN THE CALL LIST:
C     I=HEIGHT INDEX
C     DZ=GRID INCREMENT WHEN V=-1
C     Z=HEIGHT WHEN V=-1
C     C=SCAVENGING COEF WHEN V=-2
C     V=A FLAG (SEE ABOVE)
C     A=COMPUTED SATURATION VAPOR DENSITY OVER WATER IF V=0,-1 (G/M3)
C     B=COMPUTED DENOMINATOR OF FLUX EXPRESSION IF V=-2 (M/S)
C     R=COMPUTED SATURATION VAPOR DENSITY OVER ICE IF V=0,-1 (G/M3)
C
      IF(V .EQ. -2) GO TO 102
C ... COMPUTE SATURATION VAPOR DENSITY
C
      IF(V .EQ. -1.) CALL VAPOR(TEMP,I,A,B) ! UPDRAFT WATERVAPOR CONCEN.
      IF(V .EQ. 0.) CALL VAPOR(TEMPD,I,A,B) ! RAINSHAFT WATERVAPOR
      A=A*18.E+6 !AVERAGE BETWEEN I & I+1 IN G/M3

```

```

C
      B=B*18.E+6      !AVERAGE BETWEEN I & I+1 IN G/M3
      RETURN
C
C ... COMPUTE DENOMINATOR IN FLUX EQUATION
102      V=0.
          GG=C
          IF(V0(I) .GT. 0.) GG=V0(I)+GG
          IF(UOUT(I) .GT. 0.) GG=GG+UOUT(I)
          IF(G(I) .LT. 0.) GG=GG-G(I)
          IF(VIN(I) .LT. 0.) GG=GG-VIN(I)
          B=GG/2.
          RETURN
          END
C .....
C .....

```

```

      SUBROUTINE VAPOR(TEMP,I,A,B)
      DIMENSION TEMP(1)
C ... FUNCTION SAT,SSAT RETURNS SATURATION CONCENTRATION OF WATER
C ... VAPOR (G-MOLES/CC) AT TEMPK.
C
      A=(SAT(TEMP(I))+SAT(TEMP(I+1)))/2. ! SATURATION OVER WATER
      B=(SSAT(TEMP(I))+SSAT(TEMP(I+1)))/2. ! SATURATION OVER ICE
      RETURN
      END

```

```

C .....

```

```

SUBROUTINE DEFALT(I)
  INCLUDE 'SCAV.FOR'

C
C ... THIS ROUTINE RESTORES PREVIOUS (OR DEFAULT) VALUES TO VARIABLES
C   R, RC, W,..... IF CURRENT INPUT CONDITIONS RESULT IN UN-
C   REALISTIC PROFILES.
C
C ... WHEN I=0, UPDRAFT VALUES PLACED IN ARRAY 'SAVE'
C ... WHEN I=1, UPDRAFT VALUES RESTORED TO VARIABLES R,RC,...
C ... WHEN I=2, RAINSHAFT VALUES PLACED IN ARRAY 'SAVER'
C ... WHEN I=3, RAINSHAFT VALUES RESTORED TO VARAIBLES R,
C   RC, W,.... I.E., PLACED IN COMMON BLOCK /SET/.
C
C
      IF(I .EQ. 0) GO TO 50
      GO TO (200, 400, 500) I

C
C ... SAVE CURRENT VALUES
50   DO 100 J=1,LSAVE
100  SAVE(J)=R(J)
      RETURN

C
C ... RESTORE PREVIOUS VALUES
200  DO 300 J=1,LSAVE
300  R(J)=SAVE(J)
      RETURN

C
C ... SAVE CURRENT VALUES, RAINSHAFT PORTION, I=2
400  DO 450 J=1,LSAVE
450  SAVER(J)=R(J)
      RETURN

C
C ... RESTORE PREVIOUS VALUES, RAINSHAFT PORTION, I=3
500  DO 550 J=1,LSAVE

```

```

550  R(J)=SAVER(J)
      RETURN
      END

```

```

C .....

```

```

      SUBROUTINE NEUTON(DIFF1,A,XLIM,KK,IFLAG)

```

```

C

```

```

      DIMENSION XLIM(1)

```

```

C.....

```

```

C

```

```

C ... CONVECTIVE/STRATIFORM CLOUD MODULE

```

```

C

```

```

C ... RELEASED NOV, 1986

```

```

C

```

```

C ... FOR INFORMATION, CONTACT:

```

```

C   B.C. SCOTT OR C.M. BERKOWITZ

```

```

C   BATTELLE NORTHWEST LABORATORIES

```

```

C   P.O. BOX 999, RICHLAND, WA. 99352

```

```

C

```

```

C.....

```

```

C

```

```

C

```

```

C

```

```

C ... NEUTON'S METHOD ITERATES ON 'A' SUCH THAT THE FUNCTION F(A)=0;

```

```

C   I.E.,  $A=A_0 - F(A_0)/(DF/DA)$ . HERE THE FUNCTION F DESCRIBES:

```

```

C

```

```

C   1. THE SURFACE PRECIP RATE WHEN A= VERTICAL VELOCITY

```

```

C

```

```

C   2. MASS FLUX BALANCE WHEN A=PRECIP PARTICLE CONCENTRATION

```

```

C

```

```

C   3. STORM EFFICIENCY WHEN A= THE VARIABLE SHIR

```

```

C

```

```

C

```

```

C ... VARIABLES IN CALL LIST:
C     DIFF1=THE VALUE OF THE FUNCTION F AT A0
C     A=VALUE OF A IN ABOVE FORMULA
C     XLIM=ARRAY OF NUMBERS SUCH THAT
C         XLIM(1)=VALUE OF A0 IN FORMULA
C         XLIM(2)=PREVIOUS VALUE OF DIFF1 (=F(A0))
C         XLIM(3)=LARGEST ACCEPTABLE VALUE OF A-A0 FOR CONVERGENCE
C         XLIM(4)=CONVERGENCE CRITERIA FOR F=0
C         XLIM(5)=LOWER LIMIT FOR A
C         XLIM(6)=UPPER LIMIT FOR A
C         XLIM(7)= PREVIOUS DIRECTION OF ZERO SEARCH
C     KK=COUNTER FOR NUMBER OF PASSES THROUGH 'NEUTON'
C     IFLAG=CONVERGENCE FLAG
C
C ... FLAGS:
C     IFLAG=0 FOR NO CONVERGENCE
C           =1 FOR CONVERGENCE
C     IIN= -1 GUESS DECREASES SIZE OF A
C          = +1 GUESS INCREASES SIZE OF A
C
C .....

C ... BEGIN NEUTON'S METHOD
C
C     AOLD=A  ! SAVE CURRENT A
C     A2=XLIM(1)
C     DIFF2=XLIM(2)  ! OLD VALUE OF DIFFERENCE
C     ALOW=XLIM(5)
C     AHIGH=XLIM(6)
C     IIN=XLIM(7)
C     YLIM=XLIM(3)
C     IF(KK .GT. 30) YLIM=1.E-8
C     IF(KK .GT. 1) GO TO 204
C

```



```

C ... MUST DEFINE FIRST TWO GUESSES A2, AND 'A' TO START ALGORITHM
    ALOW=0 ! LOWER LIMIT OF A
    AHIGH=1.E+5 ! UPPER LIMIT OF A
    IIN=-1 ! FIRST GUESS DECREASES SIZE OF A
    A=.95*A
    GO TO 203

C
C ... DEFINE DELTA A (D) AND DELTA F FOR DERIVATIVE DF/DA (XM)
204 SIGN=DIFF1*DIFF2 ! SIGN NEGATIVE IF ZERO IS CROSSED
    IF(ABS(DIFF2) .LT. 1.E-15) GO TO 202
    IF(KK.EQ.2.AND.DIFF1/DIFF2.GT.1) IIN=-IIN
202 IF(SIGN .LE. 0.) IIN=-IIN ! CHANGE DIRECTION OF ZERO SEARCH
C
C ... COMPUTE NEW GUESS OF A
    D=AOLD-A2 ! CURRENT VALUE MINUS OLD VALUE
    IF(ABS(D) .LT. YLIM) GO TO 205
    XM=(DIFF1-DIFF2)/D
C
C ... SET LIMITS IF DERIVATIVE GETS TOO SMALL
    IF(ABS(XM) .LT. 1.E-9) XM=-100.
C
C ... COMPUTE NEW VALUE FOR 'A' BASED ON ABOVE FORMULA
    A=AOLD-DIFF1/XM
C
C ... KEEP 'A' FROM CHANGING TOO MUCH AND KEEP 'A' > 0
    A3=AOLD
    IF(AOLD .LT. XLIM(3)) A3=1.E-12
    IF(A/A3 .GT. 5) A=5.*AOLD
    IF(A .LE. 0.) A=(A+DIFF1/XM)*0.5
    IF(A .LE. 0.) A=-A
C
C ... CHECK CONVERGENCE CRITERIA
    IF(ABS(DIFF1) .GT. XLIM(4)) GO TO 203
    IF(ABS((A-AOLD)/A) .LT. 1.E-4) GO TO 205

```

```

203  DIFF2=DIFF1
      IFLAG=0
      IF(KK .GT. 1) GO TO 206
      GO TO 208
205  IFLAG=1
      RETURN

C
C ... CHECK REASONABLENESS OF NEW GUESS
206  IF(IIN .LT. 0) GO TO 207  ! GUESS DECREASING IF TRUE
      ALLOW=AOLD  ! GUESS OF A MUST BE > THIS LOWER LIMIT
      IF(A .LT. ALLOW) A=(ALLOW+AHIGH)/2.
      IF(A .LT. AHIGH) GO TO 208
      A=(AOLD+AHIGH)/2.  ! KEEP A BELOW UPPER LIMIT
      GO TO 208

C
207  AHIGH=AOLD  ! GUESS OF A MUST BE < THE UPPER LIMIT
      IF(A .GT. AHIGH) A=(ALLOW+AHIGH)/2.
      IF(A .GT. ALLOW) GO TO 208
      A=(AOLD+ALLOW)/2.  ! KEEP A ABOVE LOWER LIMIT

C
C ... SAVE OLD VALUES OF A AND F
208  XLIM(1)=AOLD
      XLIM(2)=DIFF2
      XLIM(5)=ALLOW
      XLIM(6)=AHIGH
      XLIM(7)=IIN
      RETURN
      END

```

C-----

```

      SUBROUTINE FLUX(FEN_CD,FCD_EN,FTOCELL)
C
C ... COMPUTES WINDS INTO AND OUT OF CONVECTIVE UPDRAFT AND
C   DOWNDRAFT CELLS.
C
      INCLUDE 'SCAV.FOR'
      DIMENSION FEN_CD(L),FCD_EN(L),FTOCELL(L)
      SCALE=DATA(12) ! FROM WINDUP
C
      DO 100 I=1,L
        GG=0 ! OUTFLOW (POSITIVE)
        F=0 ! INFLOW (POSITIVE)
C
        IF(G(I) .GT. 0.) F=F+G(I)
        IF(VIN(I) .GT. 0.) F=F+VIN(I)
        IF(VO(I) .LT. 0.) F=F-VO(I)
C
C ... FLUX FROM ENVIRONMENT TO CLOUD (CM*CM/S)
        FEN_CD(I)=F*SCALE ! POSITIVE
C
        IF(G(I) .LT. 0.) GG=GG-G(I)
        IF(VIN(I) .LT. 0.) GG=GG-VIN(I)
        IF(VO(I) .GT. 0.) GG=GG+VO(I)
C
C ... FLUX FROM CLOUD TO ENVIRONMENT (CM*CM/S)
        FCD_EN(I)=GG*SCALE ! POSITIVE
C
C ... FLUX FROM CURRENT CELL TO ADJACENT CELL (CM*CM/S)
        FTOCELL(I)=0.
        IF(UOUT(I) .GT. 0.) FTOCELL(I)=UOUT(I)*SCALE
100  CONTINUE
      RETURN
      END

```

C.....

SUBROUTINE UNITS

C

C ... CONVERT TO RADM UNITS

C

INCLUDE 'SCAV.FOR'

C

00 100 I=1,L

K=I+1

IF(K .GT. L) K=I

RBAR=(R(I)+R(K))/2.

RPBAR=(RC(I)+RC(K))/2.

IF(RBAR .GT. 0.) AUTO(I)=AUTO(I)/RBAR ! AUTOCONVERSION (1/S)

IF(RPBAR .GT. 0.) DEPOS(I)=DEPOS(I)/RPBAR ! DEPOS-EVAP (1/S)

C ... DENSITY IN UNITS OF MOLE/CM3

R(I)=3.44828E-8*R(I)/RHO(I) ! CLOUD WATER (G-WATER/G-AIR)

RC(I)=3.44828E-8*RC(I)/RHO(I) ! PRECIP WATER (G-WATER/G-AIR)

W(I)=W(I)*100. ! UPDRAFT (CM/S)

100 FALL(I)=FALL(I)*100. ! FALL SPEED (CM/S)

C

C ... P-DUMP (POUT) CHANGED IN INTERPOLATION ROUTINE

RETURN

END

C.....

```

        SUBROUTINE ZERO1(RMAX,RC,DIFF1,VAL,K,IFLG)
        DIMENSION VALUE(85),AXIS(85)

C
C .... IFLG=1 KEEP TRYING FOR SOLUTION
C      IFLG=2 SOLUTION FOUND
C      IFLG=3 NO SOLUTION EXISTS
C
        VALUE(K)=DIFF1
        VAL=VALUE(1)
        AXIS(K)=RC
        IF(K .GT. 1) GO TO 100
        X=1.
        Y=10.
        R0=RC
100      X=X+0.1*Y
        IF(X .LE. Y) GO TO 200
        X=X-0.1*Y
        Y=Y*10.
        X=X+0.1*Y
200      RC=X*R0
        VALUE(K)=DIFF1
        AXIS(K)=RC
C
        IFLG=1
        K=K+1
        IF(K .EQ. 2) RETURN
        SIGNX=VALUE(K-2)*VALUE(K-1)
        IF(SIGNX .GE. 0.) GO TO 300
        RC=AXIS(K-1)
        IF(K .GE. 3) RC=2.*AXIS(K-1)-AXIS(K-2)
        IFLG=2 ! SOLUTION
        RETURN
C

```

```

300      IF(RC .LT. RMAX) RETURN  ! KEEP TRYING
      IFLG=3  ! NO SOLUTION
      RETURN
      END

```

```

C.....

```

```

C

```

```

C

```

```

      SUBROUTINE SIGMA(HTOP,T,SFCPRES)

```

```

C

```

```

      INCLUDE 'SCAV.FOR'

```

```

      COMMON /BK3/ Z(L),PRESS(L)

```

```

C

```

```

C ... P,T,RHO FOR STABLE CLOUD

```

```

c establish vertical grid: defines z(i) in centimeters.

```

```

      N=L

```

```

      nml=n-1      !L IS # OF LAYERS IN CLOUD MODEL

```

```

      dz = hTOP*100./float(nml) !centimeters

```

```

      call grdset(N,DZ,Z)

```

```

      temp0 = T+273.15 !kelvin, sfc. temp.

```

```

      DO I = 1, N

```

```

        TEMP(I) = TEMPO - 7.E-5*(Z(I)-Z(1))

```

```

      END DO

```

```

c initial guess at pressure profile:

```

```

      press0 = sfcpres*1.e3 !dynes/cm2, sfc. pressure

```

```

      glevl = 0.      !cm, ground level

```

```

      DO 100 JJ=1,3

```

```

        CALL PRESSF( 1,PRESS0,PRESS,TEMP,RHO,Z,N )

```

```

C modify temp. to be along moist adiabat.
  TEMPX = TEMPO          !kelvin, sfc. temp.
  DO I = 2, N
    TEMPX = TEMPX + (Z(I)-Z(I-1))*RLAPSE( TEMPX,RHO(I-1), 1 )
    TEMP(I) = TEMPX
950   FORMAT(1H ,2(1X,I3),4(1X,1PE10.3))
    END DO
100   CONTINUE
    RETURN
    END

```

C-----

```

      SUBROUTINE GRDSET(N,DZ,Z)
C
      DIMENSION Z(1)
C  ESTABLISH UNIFORMLY SPACED GRID
C
      Z(1) = 0.      !HEIGHT ABOVE GROUND (CM)
      DO I = 2, N
        Z(I) = Z(I-1) + DZ
      END DO

      RETURN
      END

```

C-----

```

SUBROUTINE PRESSF( INIT,PRESSO,PRESS,C,CS,Z,N )
C
C      DIMENSION PRESS(1),C(1),CS(1),Z(1)
C
C      CALCULATES PRESSURE BY INTEGRATING THE HYDROSTATIC EQUATION
C      *** PRESSF SHOULD ALWAYS BE ENABLED (IPRESS=1) IF WAGRID IS NON-ZERO
C
C      WHEN WAGRID(1) IS NOT ZERO, THEN PRESS(1) - THE PRESSURE AT THE
C      BOTTOM OF THE GRID, MUST ALSO BE CALCULATED. IN THIS VERSION,
C      THIS IS DONE BY INTEGRATING HYDROSTATIC EQN. OVER GRID HEIGHT
C      CHANGE (GLEVL - GLEVLOLD)
C
C      NOTE:  PRESSURE UNITS = DYNES/CM**2
C      AIR DENSITY UNITS = MOLES/CM**3
C
C      SET PRESSURE AT BOTTOM GRID
C
C      IF (INIT.EQ.1) THEN
C          PRESS(1) = PRESSO
C      ELSE
C          DZDUM = GLEVL - GLEVLOLD
C          PRESS(1) = PRESS(1)*EXP(-3.416E-4*DZDUM/C(1))
C      END IF
C      GLEVLOLD = GLEVL
C      CS(1) = RHOAIR(PRESS(1), C(1))
C
C      INTEGRATE HYDROSTATIC EQUATION
C       $D(\ln(P))/DZ = -G/(R*T)$ 
C

```



```

DO 1000 I = 2,N
    DZDUM = Z(I)-Z(I-1)
    TEMPBB = .5*(C(I-1) + C(I))
    PRESS(I) = PRESS(I-1)*EXP(-3.416E-4*DZDUM/TEMPBB)
    CS(I) = RHOAIR( PRESS(I), C(I))
1000 CONTINUE
    RETURN
    END

```

C-----

C-----

```

    FUNCTION SAT (TEMPK)
C
C    ROUTINE FOR CALCULATING THE SATURATION CONCENTRATION OF
C    WATER VAPOR (GRAM-MOLES/CC) AT TEMPK DEGREES KELVIN
C    (ESAT IS SATURATION VAPOR PRESSURE IN MB)
C
    DATA A1,A2,A3,A4,A5,A6,A7/-21.2929759,-7629.01437,-450.173938,
110702.3730,-271.34792805,2479.85588,-296.2647743/
C
C    ESAT=A1+A2/(A3+TEMPK+A4/(A5+TEMPK+A6/(A7+TEMPK)))
    IF(TEMPK .LT. 220.) ESAT=10.** (9.2-2353./TEMPK)
    SAT = (ESAT*1.E3)/(8.3144E7*TEMPK)
    RETURN
    END

```

C.....

FUNCTION SSAT (TK)

C

C SSAT= THE SATURATION CONCENTRATION OF WATER VAPOR (GRAM MOLE/CC)

C OVER ICE AT TK DEGREES KELVIN

C (ESAT IS THE SATURATION VAPOR PRESSURE IN MB)

C

TC=TK-273.16

ESAT=6.109177956 +TC*(5.034698970E-1 + TC*(1.886013408E-2

+ +TC*(4.176223716E-4+TC*(5.824720280E-6

+ +TC*(4.838803174E-8 + TC*(1.838826904E-10))))))

SSAT=(ESAT*1.E3)/(8.3144E7*TK)

RETURN

END

C.....

C-----

FUNCTION RHOAIR(P, TEMPK)

C

C RHOAIR = AIR DENSITY (MOLES/CC)

C P = PRESSURE (DYNES/CM**2)

C TEMPK = TEMPERATURE (K)

C

RHOAIR = P/(8.3144E7*TEMPK)

RETURN

END

```

C-----
      FUNCTION RLAPSE ( TEMPK, RHOA, JFOG )
C
C      RLAPSE = LAPSE RATE (1/CM)
C      ADIABATIC IF JFOG=0
C      PSEUDOADIABATIC OTHERWISE
C      TEMPK = TEMP (K)
C      RHOA = AIR DENSITY (MOLES/CC)
C
C      THERMODYNAMIC UNITS ARE CGS (CONSTANTS USE MOLES)
C      THE SAT FUNCTION MUST RETURN THE SATURATION VAPOR
C      DENSITY (MOLES/CC) AT TEMP TEMPK

      DATA GAMMAD/-9.76E-5/! DRY ADIABATIC LAPSE RATE (DEG K/CM)
      DATA CP/2.91E8/! AIR HEAT CAPACITY, CONSTANT PRESSURE (ERG/MOLE

      DATA R/8.314E7/! GAS CONSTATN (ERG/MOLE/DEG K)
      DATA HLC/4.5E11/! LATENT HEAT OF CONDENSATION (ERG/MOLE WATER)

      RLAPSE = GAMMAD
      IF (JFOG.NE.0) THEN
        DUMMY1 = HLC*SAT(TEMPK)/(RHOA*R*TEMPK)
        RLAPSE = GAMMAD*(1. + DUMMY1)/(1. + DUMMY1*HLC/(CP*TEMPK))
      END IF

      RETURN
      END

C-----

      SUBROUTINE VDEP(IFLG,DZ,RNO,V,I,VD)
      INCLUDE 'SCAV.FOR'
C

```

```

C
C ... 'VDEP' COMPUTES THE DIFFUSION VELOCITY FOR RAIN AND SNOW. THE
C     DIFFUSION VELOCITY IS BASED ON A PSEUDO-DIFFUSIVITY RESULTING
C     FROM A VARIATION OF SETTLING VELOCITY WITH PARTICLE SIZE.
C
C ... THE PSEUDO-DIFFUSIVITY IS GIVEN BY  $V\text{-SQUARED BAR MINUS } V\text{-BAR SQUARED}$ 
C     WHERE V IS THE MEDIAN VOLUME FALL SPEED.
C
C ... VARIABLES IN CALL LIST:
C     IFLG=FLAG FOR ICE OR WATER PARTICLES
C     DZ=GRID INTERVAL (M)
C     RNO=No FOR SNOW AND  $No^{**} 0.25$  FOR RAIN
C     V=MEDIAN VOLUME PRECIPITATION FALL SPEED (M/S)
C     I=GRID INDEX
C     VD=COMPUTED DIFFUSION VELOCITY (M/S)
C
C ... FLAGS:
C     IFLG=1 WHEN COMPUTING FOR ICE
C           =2 WHEN COMPUTING FOR WATER
C
C
C     IF(IFLG .EQ. 1) GO TO 100
C
C ... HERE DIFFUSIVITY, D, AND V ARE FOR WATER
C     RC IS THE PRECIPITATION CONCENTRATION (G/M3)
C      $D=RC(I)^{**}0.25/RNO*96.4*DZ/V$ 
C     GO TO 101
C
C ... HERE DIFFUSIVITY, D, AND V ARE FOR SNOW
100   $D=(RC(I)/RNO)^{**}0.1373*D.95*0Z/V$ 
101  VD=0.
C
C ... BOUNDARY CONDITIONS. L=INDEX OF UPPER GRID
C

```

```

      IF(I .EQ. 1) GO TO 200
C ... UPPER BOUNDARY-SET DIFFUSIVITY AND DEPOSITION VELOCITY=0
      IF( I .EQ. L) RETURN
C
C ... INTERIOR GRID VALUES
      VD=C012(I)*(RC(I+1)+RC(I))/(RC(I)+RC(I-1))
      GO TO 202
C
C ... LOWER BOUNDARY
200  VD=C012(I)/2.*(RC(I+1)+RC(I))/RC(I)
C
202  VD=-D/DZ*ALOG(VD)
      VD=-VD  ! CORRECT FOR SIGN ERROR IN WATER
      RETURN
      END

```

C-----

```

      SUBROUTINE XMULT(P,DZ,H,ZZ,XNT)
      INCLUDE 'SCAV.FOR'
C
C ... XMULT COMPUTES AN ICE CRYSTAL CONCENTRATION.  IF MULT=1, THE
C      ICE CRYSTAL CONCENTRATION DEPENDS UPON A MULTIPLICATION
C      FACTOR.  IF MULT=0, THE ICE CRYSTAL CONCENTRATION
C      DEPENDS UPON THE CLOUD TOP TEMPERATURE.
C
C ... XMULT IS CALLED ONLY WHEN TEMPERATURE > -10 C.
C
C ... VARIABLES IN CALL LIST:
C      P=A CLOUD TOP CRYSTAL CONCENTRATION FROM 'MAIN'
C      DZ=GRID SPACING (M)
C      Z10=HEIGHT OF THE -10 DEGREE LEVEL (M)
C      H=CLOUD TOP HEIGHT (M)

```

```

C      ZZ=CURRENT HEIGHT (M)
C      XNT=COMPUTED ICE CRYSTAL CONCENTRATION
C
C ... FLAGS:
C      MULT=0 FOR NO ICE MULTIPLICATION
C      =1 FOR ICE MULTIPLICATION
C
C
C
C ... HERE XNT IS THE CLOUD TOP CRYSTAL CONCENTRATION (#/M3)
      XNT=P
      IF(MULT .EQ. 0) GO TO 101
C
C
C ... ICE MULTIPLICATION HERE
C ... IF CRYSTAL CONCENTRATION IS > 30/LITER DON'T CONSIDER MULT-
C      TIPLICATION.
200  IF(XNT .GT. 3.E+4) GO TO 101
C
C ... FIND CRYSTAL CONC., XNT
      Z=H+DZ/2.
      XBAR=0.
C
C ... LOOP 400 FINDS MAX CLOUD WATER BETWEEN THE -10 LEVEL (Z10) AND
C      THE CURRENT LEVEL (ZZ)
      DO 400 K=1,L
      I=L+1-K
      Z=Z-0Z
      IF(Z .LT. ZZ) GO TO 401
      IF(Z .GT. Z10) GO TO 400
      IF(R(I) .GT. XBAR) XBAR=R(I)

```

```

400  CONTINUE
C
C ... HERE CRYSTAL CONCENTRATION DEPENDS LINEARLY ON MAX WATER
C    CONCENTRATION BETWEEN CURRENT Z AND Z10; I.E., XBAR=4/LITER WHEN
C    WATER=0.1 G/M3.
401  XBAR=4.E+4*XBAR
C ... PUT UPPER LIMIT OF XBAR=30/LITER
      IF(XBAR .GT. 3.E+4) XBAR=3.E+4
C
      IF(XBAR .GT. XNT .AND. XBAR .LE. 3.E+4) XNT=XBAR
101  CONTINUE
      RETURN
      END

```

```

C ... BLOCK 'TABLE.FOR'
C
C
C ... RELEASED AUGUST, 1986
C
C ... FOR INFORMATION, CONTACT:
C    B.C. SCOTT OR C.M. BERKOWITZ
C    BATTELLE NORTHWEST LABORATORIES
C    P.O. BOX 999, RICHLAND, WA. 99352
C
C
C ... LX=THE # OF VERTICAL USER GRIDS * # OF HEIGHT INTERVALS
C    * # OF PRECIP INTERVALS * # OF TEMP INTERVALS.
C    = (NV) * (NH) * (NP) * (NT)
C

```

```

C ... NVM= NUMBER OF VERTICAL GRIDS FROM THE CLOUD MOOEL
C ... NVM1= ONE PLUS THE # OF VERTICAL GRIDS IN CLOUD MODEL
C   NV= NUMBER OF DESIRED VERTICAL GRIDS IN THE USER'S MODEL
C ... NH= NUMBER OF HEIGHT INTERVALS
C ... NP= NUMBER OF PRECIP INTERVALS
C ... NT= NUMBER OF TEMP INTERVALS
C           OR FREEZING LEVEL HEIGHT INTERVALS
C ... NVAR= NUMBER OF VARIABLES BEING SAVED FOR RADM MODEL
C
  PARAMETER (NV=20, NVM=20, NH=4, NP=3, NT=3, NVAR=13)
  PARAMETER (LX=NV*NH*NP*NT, NVM1=NVM+1, NC=NH*NP*NT, LY=2*LX+1)
  PARAMETER (NSPACE=NVAR*NV)
  COMMON TIMEX(NC),RATEX(NC)
  COMMON RX(LY),RCX(LY),WX(LY),POUTX(LY)
  COMMON ALPHX(LY),OEPOSX(LY),PRATEX(LY),FALLX(LY),AUTOX(LY)
  COMMON FEN_CDX(LY),FCD_ENX(LY),FTOCELX(LY),TEMPX(LY)
  COMMON RI(NV),RCI(NV),WI(NV),POUTI(NV)
  COMMON ALPHI(NV),DEPOSI(NV),PRATEI(NV),FALLI(NV),AUTOI(NV)
  COMMON FEN_CDI(NV),FCD_ENI(NV),FTOCELI(NV),TEMPI(NV)
  COMMON RR(NV),RCR(NV),WR(NV),POUTR(NV)
  COMMON ALPHR(NV),OEPOSR(NV),PRATER(NV),FALLR(NV),AUTOR(NV)
  COMMON FEN_CDR(NV),FCD_ENR(NV),FTOCELR(NV),TEMPR(NV)
  COMMON RE(NVM),RCE(NVM),WE(NVM),POUTE(NVM)
  COMMON ALPHE(NVM),OEPOSE(NVM),PRATEE(NVM),FALLE(NVM),AUTOE(NVM)
  COMMON FEN_COE(NVM),FCD_ENE(NVM),FTOCELE(NVM),TEMPE(NVM)
C
  DIMENSION J(7),DV(6),VO(6)
  DATA J /NV,NP,NH,NT,0,0,0/
  DATA DV /4.,-2000,1750.,0.,0.,0./
  DATA VO /4.,12000.,1000.,0.,0.,0./
  DATA HMAX /12000./

```

C


```

C ... J=DIMENSION OF EACH VARIABLE; J(1)=NV VERTICAL GRIDS,
C      J(2)=NP PRECIP INCREMENTS, ETC
C
C ... DV IS THE TABULATED INCREMENT FOR EACH OF THE VARIABLES
C      DV(1)=INCREMENT FOR VALUES OF PRECIP RATE (1.0 MM/HR)
C      OR CLOUD BASE HEIGHT (METERS)
C      DV(2)=INCREMENT FOR VALUES OF STORM HEIGHT (-2000 M)
C      DV(3)=INCREMENT FOR VALUES OF TEMPERATURE (10 DEGREES)
C      OR FREEZING LEVEL HEIGHT INCREMENT (M)
C      ETC. FOR OTHER INCREMENTS
C
C ... VO IS THE STARTING VALUE OF EACH VARIABLE; E.G.,
C      VO(1)= 0.0 MM/HR, VO(2)= 10,000 M, VO(3)= -5 DEG
C      OR V(3)= 1000 METERS
C ... ALL HEIGHTS ARE HEIGHTS ABOVE GROUND.
C
C ... HMAX IS THE MAXIMUM DESIRED HEIGHT OF THE USER'S GRID.
C      IF USER'S GRID IS A HEIGHT GRID

```

```

C      CALLED PROGRAM 'BEGIN' (DUMMY MAIN PROGRAM)
C
C
C ... INTERPOLATION ROUTINES FOR CONVECTIVE CLOUD MODULE
C
C ... RELEASED AUGUST, 1986
C

```

```

C ... FOR INFORMATION, CONTACT:
C   B.C. SCOTT OR C.M. BERKOWITZ
C   BATTELLE NORTHWEST LABORATORIES
C   P.O. BOX 999, RICHLAND, WA. 99352
C
C.....
C
C ... THIS CODE WAS DEVELOPED ON A VAX 11-780 USING VMS OPERATING
C   SYSTEM. THE ONLY CHANGES NEED TO RUN IT ON ANDTHER SYSTEM
C   ARE THE FOLLOWING:
C
C   1. THE INCLUDE STATEMENT FOUND AT THE START OF EACH PROGRAM UNIT
C   2. DEVICE NUMBERS FOR INPUT-OUTPUT
C
C.....

C   THIS PROGRAM INTERPOLATES FROM TABLES TO OBTAIN DESIRED
C   VERTICAL PROFILES AT SOME INTERMEDIATE INPUT CONDITION.
C
C   DIMENSION Q(6)
C   IIN=8 ! INPUT DEVICE NUMBER
C
C
C   WRITE(6,806)
806  FORMAT(1H0,' TYPE 0 FOR STABLE CLOUD, 1 FOR CONVECTIVE CLOUD')
C   ACCEPT *, ICLLOUD
C
C   LIM=ICLOUD+1
C
C
C ... INTERPOLATE FROM TABLE TO GET VERTICAL PROFILES AT SOME INTER-
C   MEDIATE INPUT CONDITION. DESIRED INPUT CONDITIONS ARE STORED
C   IN ARRAY Q

```

```

C
C
C ... Q(1)=DESIRED SURFACE PRECP HOURLY AMOUNT (MM)
C      OR CLOUD BASE HEIGHT (METERS)
C ... Q(2)=DESIRED CLOUD TOP HEIGHT (METERS)
C ... Q(3)=DESIRED SURFACE TEMPERATURE (DEGREES C)
C *****
C
C      GIVES 5 TRIES, REMOVE DO 100 WHENEVER DESIRED
C
C *****
      DO 100 L=1,5
      WRITE(6,805)
805  FORMAT(1H0,' DESIRED RAIN AMOUNT (MM) OR CLD BASE HT (M),
      1 CLOUD TOP (M), SFC TEMP (C)',/, ' OR FRZING LVL HT (M)')
      ACCEPT *,Q(1),Q(2),Q(3)
      IF(Q(1) .LT. 5 .OR . LIM .EQ. 2) WRITE(6,801) Q(1)
      IF(Q(1) .GT. 100.) WRITE(6,804) Q(1)
      WRITE(6,802) Q(2)
      WRITE(6,803) Q(3)
801  FORMAT(1HD,' SURFACE PRECIPITATION AMOUNT=',1PE10.3)
802  FORMAT(1H , ' CLOUD TOP HEIGHT=',1PE10.3)
803  FORMAT(1H , ' SURFACE TEMP OR FREEZING LVL HT=',1PE10.3)
804  FORMAT(1H0,' CLOUD BASE HEIGHT=',1PE10.3)
C
C ... DO THE INTERPOLATION AND OUTPUT THE INTERPOLATED VALUES
C      IN ARRAYS ENDING IN 'I' AND 'R'.
C
      CALL SELECT(LIM,Q,IIN)
100  CONTINUE
      END

```

```

-----

      SUBROUTINE SELECT(LIM,Q,IIN)
C
C ... THIS ROUTINE INTERPOLATES FROM PREVIOUSLY GENERATED TABLES TO
C   GET DATA FOR CLOUD PARAMETERS.
C
C ... RELEASED AUGUST, 1986
C
C ... FOR INFORMATION, CONTACT:
C   B.C. SCOTT OR C.M. BERKOWITZ
C   BATTELLE NORTHWEST LABORATORIES
C   P.O. BOX 999, RICHLAND, WA. 99352
C
C
C ... VARIABLES IN CALL LIST:
C   LIM = FLAG SET TO 1 FOR STABLE CLOUDS ,2 FOR CONVECTIVE CLOUDS
C   Q = ARRAY OF INPUT CONDITIONS, SEE BELOW
C   IIN= INPUT DEVICE NUMBER
C
      INCLUDE 'TABLE.FOR'
      DIMENSION Q(6)
      DATA IREAD /0/
C
      IF(IREAD .GT. 0.) GO TO 100
C ... INPUT TABLE INTO COMMON BLOCK
C
      READ(IIN) DZ,TIMEX,RATEX,RX,RCX,WX,POUTX,ALPHX,DEPOSX,PRATEX,
      1 FALLX,AUTOX,FEN_CDX,FCD_ENX,FTOCELX,TEMPX
      IREAD=1
C

```

```

100  ISTAB=LIM-1
      ICLOUD=-1  ! START INTERPOLATION SECTION
      DO 6000 NNN=1,LIM
      ICLOUD=ICLOUD+1
      IF(NNN .GT. 1) GO TO 602

C
C
C ... INTERPOLATE FROM TABLE TO GET VERTICAL PROFILES AT SOME INTER-
C    MEDIATE INPUT CONDITION. DESIRED INPUT CONDITIONS ARE STORED
C    IN ARRAY Q
C
C
C ... Q(1)=DESIRED SURFACE PRECP HOURLY AMOUNT (MM)
C    OR CLOUD BASE HEIGHT (METERS)
C ... Q(2)=DESIRED CLOUD TOP HEIGHT (METERS)
C ... Q(3)=DESIRED SURFACE TEMPERATURE (DEGREES C)
C    OR FREEZING LEVEL HEIGHT (M)
C
C
C
      N=-1
      DO 601 I=1,7
      IF(J(I) .GT. 1) N=N+1
601  CONTINUE
C
602  CALL INTERP(Q,N,ICLOUD,ISTAB,TIMEI,RATEI)
C
C
      IF(ICLOUD .EQ. 0) GO TO 650
C
C *****
C
C ..... ALL WRITE STATEMENTS CAN BE ELEMENATED AFTER DEGUGGING
C
C *****

```

```

C
      WRITE(6,807) TIMEI,RATEI
807  FORMAT(1H0,' STORM DURATION (MIN) =',1PE10.3,' STORM RATE (MM
      1/HR) =',1PE10.3)
650  CONTINUE
C ... INTERPOLATED VALUES ARE STORED IN:
C      RI,RR=INTERPOLATED PROFILES OF CLOUD WATER
C      RCI,RCR=INTERPOLATED PROFILES OF PRECIP
C      WI,WR=INTERPOLATED PROFILES OF VERTICAL VELOCITY
C      ETC....
C
C      '..R(j) ' MEANS RAINSHAFT
C      '..I(j) ' MEANS UPDRAFT
C
      IF(LIM .EQ. 1) GO TO 651
      IF(NNN .EQ. 1) WRITE(6,808) ! CONVECTIVE CD HEADINGS
808  FORMAT(1H0,' RAINSHAFT CONVECTIVE CLOUD')
      IF(NNN .EQ. 2) WRITE(6,809)
809  FORMAT(1H0,' UPDRAFT CONVECTIVE CLOUD')
      WRITE(6,812)
      GO TO 652
C
651  WRITE(6,810) ! STABLE CLOUD HEADINGS
810  FORMAT(1H0,' STABLE CLOUD')
811  FORMAT(1H0)
      WRITE(6,812)
C
652  Z=0
812  FORMAT(1H0,'          Z(M)          R(G/G)      RC(G/G)      W(CM/S)
      1  VP(CM/S) FTOCEL(CM2/S)')
813  FORMAT(1H0),'          T(A)      SCAV(1/S)  PRATE(MM/H)  P-DUMP
      1  FEN_CD      FCD_EN')

```

```

814  FORMAT(1H ,38X, '(G/CM2/S)  (CM2/S)  (CM2/S)')
      INTER=1
      XINTER=FLOAT(INTER)
      DO 500 I=1,NV,INTER
        IF(ISTAB .EQ. 0) PRINT 800,I,Z,RI(I),RCI(I),WI(I),FALLI(I)
        IF(ISTAB .EQ. 1) PRINT 800,I,Z,RI(I),RCI(I),WI(I),FALLI(I)
      1 ,FTOCELI(I)
500   Z=Z+DZ*XINTER
      C
        WRITE(6,811)
        WRITE(6,813)
        WRITE(6,814)
        Z=0.
        DO 501 I=1,NV,INTER
          IF(ISTAB .EQ. 0) PRINT 800,I,Z,ALPHI(I),PRATEI(I)
          IF(ISTAB .EQ. 1) PRINT 900,I,TEMPI(I),ALPHI(I),PRATEI(I),POUTI(I)
        1 , FEN_COI(I),FCO_ENI(I)
900   FORMAT(1H ,I3,6(1X,1PE10.3))
501   Z=Z+OZ*XINTER
        IF(LIM .EQ. 1) GO TO 6000
        IF(NNN .EQ. 2) GO TO 6000
        DO 675 I=1,NSPACE  ! SAVE RAINSHAFT FLUXES
675   RR(I)=RI(I)
6000  CONTINUE
800   FORMAT(1H ,I3,6(1X,1PE10.3))
      RETURN
      END

```

```

SUBROUTINE INTERP(Q,N,ICLOUD,ISTAB,TIMEI,RATEI)
C
C ... 'INTERP' INTERPOLATES FROM A TABLE OF VALUES TO FIND VERTICAL
C   PROFILES OF R AT GIVEN VALUES OF CLOUD TOP HEIGHT, SURFACE
C   PRECIP RATE, AND SURFACE TEMPERATURE.  THE INTERPOLATED VALUES
C   ARE PLACED IN ARRAYS RI, RCI, WI, TI
C
C
C ... RELEASED AUGUST, 1986
C
C ... FOR INFORMATION, CONTACT:
C   B.C. SCOTT OR C.M. BERKOWITZ
C   BATTELLE NORTHWEST LABORATORIES
C   P.O. BOX 999, RICHLAND, WA. 99352
C
C   INCLUDE 'TABLE.FOR'
C   DIMENSION V(6),DJ(6),M(7),L(128),Q(6)
C   DIMENSION IA(128)
C   INTEGER PROD
C
C
C ... VARIABLES IN CALL LIST:
C   Q= ARRAY OF POINTS WHERE INTERPOLATION IS DESIRED
C   N=NUMBER OF LEVELS FOR INTERPOLATION, I.E.; PRECIP,
C     STORM HEIGHT, SURFACE TEMPERATURE,.....
C   ICLOUD= SEE 'FLAGS' BELOW
C   ISTAB= SEE 'FLAGS' BELOW
C   TIMEI=STORM DURATION (MIN)-COMPUTED
C   RATEI=STORM RATE (MM/H)-COMPUTED
C
C ... FLAGS:
C   ICLOUD, ISTAB = 0  WHEN INTERPOLATION IS DONE FOR STABLE CLOUD OR FOR
C     RAINSHAFT PORTION OF CUMULUS CLOUD
C   ICLOUD, ISTAB = 1 WHEN INTERPOLATION IS FOR UPDRAFT PORTION OF

```



```

C          THE CUMULUS CLOUD
C
C
C
C ... RX=TABLE OF VALUES, RI= INTERPOLATED VALUES
C ... RX=RX(Z,XJ,H,T,...)=RX(J1,J2,J3,J4,...) WHERE:
C     J1=J(1)= # OF VERTICAL GRIDS,   J3=J(3)= # OF CLOUD TOP INTERVALS,
C     J2=J(2)= # OF PRECIP INTERVALS, J4=J(4)= # OF TEMPERATURE
C     INTERVALS, ETC.
C     THUS, J=DIMENSION OF EACH VARIABLE; J(1)=NV VERTICAL GRIDS,
C     J(2)=NP PRECIP INCREMENTS, ETC.
C
C ... N=NUMBER OF VARIABLES BEING CONSIDERED
C     VARIABLES ARE (IN ORDER):
C         PRECIPITATION RATE
C         STORM HEIGHT
C         SURFACE TEMPERATURE
C         OTHERS AS DESIRED
C ..... MAX N=6 .....
C
C ... 3RD ORDER INTERPLATION IS USED, THUS RX AT SOME
C     INTERPOLATED POINT IS (AT A FIXED HEIGHT):
C
C      $RI = R_o + DR/D(XJ) * (XJ - XJ_o) + DR/DH * (H - H_o) + DR/DT * (T - T_o) + \dots$ 
C         + HIGHER ORDER TERMS
C
C ... V(I)=VALUE OF ith VARIABLE AT THE BEGINNING OF THE PROPER INTERVAL,
C     V(1)=XJo, V(2)=Ho, V(3)=To ETC.
C
C ... V0 IS THE STARTING VALUE OF EACH VARIABLE; E.G.,
C     V0(1)= 0.0 MM/HR, V0(2)=10,000 M, V0(3)= -5 DEG
C
C ... DV IS THE TABULATED INCREMENT FOR EACH OF THE VARIABLES
C     DV(1)=INCREMENT FOR VALUES OF PRECIP RATE (1.0 MM/HR)
C     DV(2)=INCREMENT FOR VALUES OF STORM HEIGHT (-2000 M)

```

```

C      DV(3)=INCREMENT FOR VALUES OF TEMPERATURE (10 DEGREES)
C      ETC FOR OTHER INCREMENTS
C
C ... Q(I) IS THE DESIRED VALUE OF THE iTH VARIABLE WHICH IS PRESUMABLY
C      SOMEWHERE BETWEEN THE TABULATED VALUES. Q(1)=DESIRED PRECIP RATE,
C      Q(2)=DESIRED STORM HEIGHT, ETC.
C
C      NN=N+1
C
C ... MAKE SURE DESIRED VALUES ARE WITHIN LOWER TABLE LIMITS
      DO 550 K=1,N
        QSAVE=Q(K)
        IF(QSAVE .EQ. 0) QSAVE=1.E-8
        IF(DV(K) .GT. D.) GO TO 551
C ... HERE Q MUST BE < V0 TO START
        IF(Q(K) .LT. V0(K)) GO TO 550
        IF(V0(K) .GT. 0.) Q(K)=0.9999999*V0(K)
        IF(V0(K) .LT. 0.) Q(K)=1.00000001*V0(K)
        DELTQ=(QSAVE-Q(K))/QSAVE
        IF(ABS(DELTQ) .GT. 4.E-7 ) WRITE(6,800) K,Q(K)
        GO TO 550
C
C ... HERE Q MUST BE > V0 TO START
551 IF(Q(K) .GT. V0(K)) GO TO 550
        IF(V0(K) .GT. 0.) Q(K)=1.0000001*V0(K)
        IF(V0(K) .LT. 0.) Q(K)=0.9999999*V0(K)
        DELTQ=(QSAVE-Q(K))/QSAVE
        IF(ABS(DELTQ) .GT. 4.E-7 ) WRITE(6,800)K,Q(K)

```

```

550  CONTINUE
C
      DO 200 K=1,N
101  V(K)=V0(K)
      QSAVE=Q(K)
      IUP=J(K+1)
      DO 100 I=1,IUP
        IF(DV(K) .GT. 0. .AND. V(K) .GT. Q(K)) GO TO 200
        IF(DV(K) .LT. 0. .AND. V(K) .LT. Q(K)) GO TO 200
        M(K+1)=I
        DJ(K)=Q(K)-V(K)
100  V(K)=V(K)+DV(K)
C
C ... OUT OF TABLE LIMITS.  SELECT A DEFAULT VALUE
      IF(DV(K) .GT. 0.) Q(K)=0.9999999*(V(K)-DV(K))
      IF(DV(K) .LT. 0.) Q(K)=1.0000001*(V(K)-DV(K))
      DELTQ=(QSAVE-Q(K))/QSAVE
      IF(ABS(DELTQ) .GT. 4.E-7 ) WRITE(6,800) K,Q(K)
800  FORMAT(1H0,' OUT OF TABLE LIMITS FOR VARIABLE K=',I2,/,
1    ' DEFAULT OF Q(K) IS TAKEN TO BE ',2(1X,1PE10.3))
      GO TO 101
C
200  DJ(K)=DJ(K)/DV(K)
C
C ... FIND LOCATION OF STARTING INDEX ( L(1) )
C
      PROD=0.
      DO 600 I=1,N
        K=NN-I
600  PROD=(PROD+(M(K+1)-1))*J(K)
901  FORMAT(1H ,6(1X,I3))
      IUP=J(1)
C ... INTERPOLATE FOR EACH HEIGHT LEVEL (I)
      DZ1=HMAX/FLOAT(IUP-1)

```

```

C
C ... INDICIES FOR TIME & DURATION IN CONVECTIVE CLOUD
  IA(1)=PROD/J(1)+1
  ISUM=1
  DO 402 K=2,4
    IA(K)=IA(1)+ISUM
402  ISUM=ISUM*J(K)
C
  IA(5)=IA(2)+IA(3)-IA(1)
  IA(6)=IA(3)+IA(4)-IA(1)
  IA(7)=IA(2)+IA(4)-IA(1)
  IA(8)=IA(2)+IA(3)+IA(4)-2*IA(1)
  Z=0.
  RATEI=RATEX(IA(1))
  IF(ICLOUD .EQ. 1) PROD=LX+PROD
  DO 300 I=1,IUP
    RI(I)=0.
    RCI(I)=0.
    WI(I)=0.
    ALPHI(I)=0.
    DEPOSI(I)=0.
    PRATEI(I)=0.
    FALLI(I)=0.
    AUTOI(I)=0.
    POUTI(I)=0.
    IF(ISTAB .EQ. 0) GO TO 450
    FEN_CDI(I)=0.
    FCD_ENI(I)=0.
    FTOCELI(I)=0.
    TEMPI(I)=0.
450  IF(Z .GT. Q(2)) GO TO 300
    L(1)=PROD+I
    ISUM=J(1)
    DO 400 K=2,4

```

```

        L(K)=L(1)+ISUM
400    ISUM=ISUM*J(K)
C
C ... INDICIES FOR SECOND AND THIRD ORDER TERMS
        L(5)=L(2)+L(3)-L(1)
        L(6)=L(3)+L(4)-L(1)
        L(7)=L(2)+L(4)-L(1)
        L(8)=L(2)+L(3)+L(4)-2*L(1)
C
C
C ... COMPUTE INTERPOLATED VALUE. PUT AS MANY VARIABLES AS DESIRED.
C
        R01=RX(L(1))
        R02=RCX(L(1))
        R03=WX(L(1))
        R04=ALPHX(L(1))
        R05=DEPOSX(L(1))
        R06=PRATEX(L(1))
        R07=FALLX(L(1))
        R08=AUTOX(L(1))
        R12=POUTX(L(1))
        IF(ISTAB .EQ. 0) GO TO 451
        R09=FEN_CDX(L(1))
        R10=FCD_ENX(L(1))
        R11=FTOCELX(L(1))
451    RI(I)=R01
        RCI(I)=R02
        WI(I)=R03
        ALPHI(I)=R04
        DEPOSI(I)=R05
        PRATEI(I)=R06
        FALLI(I)=R07
        AUTOI(I)=R08
        POUTI(I)=R12

```

```

      IF(ISTAB .EQ. 0) GO TO 452
      FEN_CDI(I)=R09
      FCD_ENI(I)=R10
      FTOCELI(I)=R11
C ... COMPUTE INTERPOLATED TEMP PROFILE IN CONVECTIVE CLD
C
      R13=TEMPX(L(1))
      TEMPI(I)=R13
      DO 501 K=1,N
501  CALL ORDER1(TEMPI(I),TEMPX,L,K,R13,DJ(K))
C
      DJH=DJ(1)*DJ(2)
      DTH=DJ(3)*DJ(2)
      DTJ=DJ(3)*DJ(1)
      DTJH=DTJ*DJ(2)
C
      CALL ORDER21(TEMPI(I),TEMPX,L,DJH)
      CALL ORDER22(TEMPI(I),TEMPX,L,DTH)
      CALL ORDER23(TEMPI(I),TEMPX,L,DTJ)
      CALL ORDER3(TEMPI(I),TEMPX,L,DTJH)
C
      TC=R13  ! TEMP OF COLD SOUNDING AT I
      TW=TEMPX(L(4))  ! TEMP OF WARM SOUNDING AT I
C
452  DO 500 K=1,N
      CALL SETUP(TW,TC,TEMPI(I),ISTAB,DJ,K,DJX)  ! FILTER AT FREEZ LVL
      CALL ORDER1(RI(I),RX,L,K,R01,DJX)
      CALL ORDER1(RCI(I),RCX,L,K,R02,DJX)
      CALL ORDER1(WI(I),WX,L,K,R03,DJ(K))
      CALL ORDER1(ALPHI(I),ALPHX,L,K,R04,DJX)
      CALL ORDER1(DEPOSI(I),DEPOSX,L,K,R05,DJX)
      CALL ORDER1(PRATEI(I),PRATEX,L,K,R06,DJX)
      CALL ORDER1(FALLI(I),FALLX,L,K,R07,DJX)
      IF(ICLOUD.EQ.1) CALL ORDER1(POUTI(I),POUTX,L,K,R12,DJX)

```

```

      IF(ISTAB .EQ. 0) GO TO 453
      CALL ORDER1(FEN_CDI(I),FEN_CDX,L,K,R09,DJ(K))
      CALL ORDER1(FCD_ENI(I),FCD_ENX,L,K,R10,DJ(K))
      CALL ORDER1(FTOCELI(I),FTOCELX,L,K,R11,DJ(K))
453  IF(I .GT. 1 .OR. ICLOUD .EQ. 0) GO TO 500
      CALL ORDER1(RATEI,RATEX,IA,K,RATEX(IA(1)),DJX)
500  CALL ORDER1(AUTOI(I),AUTOX,L,K,R08,DJX)
C
      DJH=DJ(1)*DJ(2)
      DTH=DJX*DJ(2)
      DTJ=DJX*DJ(1)
      DTJH=DTJ*DJ(2)
C
C ... SECOND ORDER TERMS
C
      CALL ORDER21(RI(I),RX,L,DJH)
      CALL ORDER21(RCI(I),RCX,L,DJH)
      CALL ORDER21(WI(I),WX,L,DJH)
      CALL ORDER21(ALPHI(I),ALPHX,L,DJH)
      CALL ORDER21(DEPOSI(I),DEPOX,L,DJH)
      CALL ORDER21(PRATEI(I),PRATEX,L,DJH)
      IF(ICLOUD.EQ.1) CALL ORDER21(POUTI(I),PDUTX,L,DJH)
      IF(ISTAB .EQ. 0) GO TO 454
      CALL ORDER21(FEN_CDI(I),FEN_CDX,L,DJH)
      CALL ORDER21(FCD_ENI(I),FCD_ENX,L,DJH)
      CALL ORDER21(FTOCELI(I),FTOCELX,L,DJH)
454  IF(I .GT. 1 .OR. ICLOUD .EQ. 0) GO TO 601
      CALL ORDER21(RATEI,RATEX,IA,DJH)
601  CALL ORDER21(FALLI(I),FALLX,L,DJH)
      CALL ORDER21(AUTOI(I),AUTOX,L,DJH)
C
      CALL ORDER22(RI(I),RX,L,DTH)
      CALL ORDER22(RCI(I),RCX,L,DTH)
      CALL ORDER22(WI(I),WX,L,DTH)

```

```

CALL ORDER22(ALPHI(I),ALPHX,L,DTH)
CALL ORDER22(DEPOSI(I),DEPOSX,L,DTH)
CALL ORDER22(PRATEI(I),PRATEX,L,DTH)
IF(ICLOUD.EQ.1) CALL ORDER22(POUTI(I),POUTX,L,DTH)
IF(ISTAB .EQ. 0) GO TO 455
CALL ORDER22(FEN_CDI(I),FEN_CDX,L,DTH)
CALL ORDER22(FCD_ENI(I),FCD_ENX,L,DTH)
CALL ORDER22(FTOCELI(I),FTOCELX,L,DTH)
455 IF(I .GT. 1 .OR. ICLOUD .EQ. 0) GO TO 602
CALL ORDER22(RATEI,RATEX,IA,DTH)
602 CALL ORDER22(FALLI(I),FALLX,L,DTH)
CALL ORDER22(AUTOI(I),AUTOX,L,DTH)

```

C

```

CALL ORDER23(RI(I),RX,L,DTJ)
CALL ORDER23(RCI(I),RCX,L,DTJ)
CALL ORDER23(WI(I),WX,L,DTJ)
CALL ORDER23(ALPHI(I),ALPHX,L,DTJ)
CALL ORDER23(DEPOSI(I),DEPOSX,L,DTJ)
CALL ORDER23(PRATEI(I),PRATEX,L,DTJ)
IF(ICLOUD.EQ.1) CALL ORDER23(POUTI(I),POUTX,L,DTJ)
IF(ISTAB .EQ. 0) GO TO 456
CALL ORDER23(FEN_CDI(I),FEN_CDX,L,DTJ)
CALL ORDER23(FCD_ENI(I),FCD_ENX,L,DTJ)
CALL ORDER23(FTOCELI(I),FTOCELX,L,DTJ)
456 IF(I .GT. 1 .OR. ICLOUD .EQ. 0) GO TO 603
CALL ORDER23(RATEI,RATEX,IA,DTJ)
603 CALL ORDER23(FALLI(I),FALLX,L,DTJ)
CALL ORDER23(AUTOI(I),AUTOX,L,DTJ)

```

C

C ... THIRD ORDER TERMS

C

```

CALL ORDER3(RI(I),RX,L,DTJH)
CALL ORDER3(RCI(I),RCX,L,DTJH)
CALL ORDER3(WI(I),WX,L,DTJH)

```



```

      CALL ORDER3(ALPHI(I),ALPHX,L,DTJH)
      CALL ORDER3(DEPOSI(I),DEPOX,L,DTJH)
      CALL ORDER3(PRATEI(I),PRATEX,L,DTJH)
      CALL ORDER3(FALLI(I),FALLX,L,DTJH)
      CALL ORDER3(AUTOI(I),AUTOX,L,DTJH)
      IF(ICLOUD.EQ.1) CALL ORDER3(POUTI(I),POUTX,L,DTJH)
      IF(ISTAB .EQ. 0) GO TO 457
      CALL ORDER3(FEN_CDI(I),FEN_CDX,L,DTJH)
      CALL ORDER3(FCD_ENI(I),FCD_ENX,L,DTJH)
      CALL ORDER3(FTOCELI(I),FTOCELX,L,DTJH)
457  IF(I .GT. 1 .OR. ICLOUD .EQ. 0) GO TO 301
      CALL ORDER3(RATEI,RATEX,IA,DTJH)
301  IF(FEN_CDI(I) .LT. 0.) FEN_CDI(I)=0. ! REMOVE INTERPOLATION ERROR
      IF(FCD_ENI(I) .LT. 0.) FCD_ENI(I)=0.
      IF(ABS(PRATEI(I)) .LT. 1.E-6) PRATEI(I)=0. ! REMOVE TRUNCATION ERROR
      IF(ABS(FALLI(I)) .LT. 1.E-6) FALLI(I)=0.
      POUTI(I)=POUTI(I)*1.E-4 ! CONVERT TO G/CM2/S
300  Z=Z+DZ1
      IF(ICLOUD .GT. 0) TIMEI=Q(1)/RATEI*60.
      RETURN
      END

```

C .....

SUBROUTINE SETUP(TWARM,TCOLD,TEMP,ISTAB,DJ,K,DJX)

```

C
C ... ROUTINE SELECTS THE MOST APPROPRIATE INTERPOLATION
C DATA IN THE VICINITY OF THE FREEZING LEVEL
C
C DIMENSION DJ(1)
C

```

```

C ... TWARM=TEMP AT HEIGHT I FOR WARM SOUNDING
C ... TCOLD=TEMP AT HEIGHT I FOR COLD SOUNDING
C ... TEMP=TEMP AT HEIGHT I FOR CURRENT SOUNDING
C
    DJX=DJ(K)
    IF(K .NE. 3) RETURN ! ONLY TINKERING WITH TEMP INTERP.
    IF(ISTAB .EQ. 0) RETURN ! IF TRUE, STABLE CLOUD RUN
    IF(TEMP .EQ. TWARM) RETURN
    IF(TEMP .EQ. TCOLD) RETURN ! DON'T TINKER ON DATA LINE
C
C ... ADJUSTMENTS FOR CONVECTIVE CLOUD PROFILE
    IF(TWARM .LE. 273.16) RETURN ! ALL SOUNDINGS BELOW FREEZING
    IF(TCOLD .GE. 273.16) RETURN ! ALL SOUNDINGS ABOVE FREEZING
    DJX=0.0 ! USE COLD LEVEL VALUES IN INTERP
    IF(TEMP .GT. 273.16) DJX=1.0 ! USE WARM LEVEL VALUES
    RETURN
    END

```

```

SUBROUTINE ORDER1(RI,RX,L,K,R01,DJ)
    DIMENSION RX(1),L(1)
    RI=RI+(RX(L(K+1))-R01)*DJ
    RETURN
    END

```

```

SUBROUTINE ORDER21(RI,RX,L,DJH)
DIMENSION RX(1),L(1)
RI=RI+DJH*(RX(L(5))-RX(L(2))-RX(L(3))+RX(L(1)))
RETURN
END

```

```

SUBROUTINE ORDER22(RI,RX,L,DTH)
DIMENSION RX(1),L(1)
RI=RI+DTH*(RX(L(6))-RX(L(4))-RX(L(3))+RX(L(1)))
RETURN
END

```

```

SUBROUTINE ORDER23(RI,RX,L,DTJ)
DIMENSION RX(1),L(1)
RI=RI+DTJ*(RX(L(7))-RX(L(4))-RX(L(2))+RX(L(1)))
RETURN
END

```

```

SUBROUTINE ORDER3(RI,RX,L,DTJH)
DIMENSION RX(1),L(1)
RI=RI+DTJH*(RX(L(8))-RX(L(7))-RX(L(6))+RX(L(4))
1      -RX(L(5))+RX(L(2))+RX(L(3))-RX(L(1)))
RETURN
END

```

```

C
C   CALLED 'TABLEH' IN DIRECTORY
C
C
C ... INTERPOLATION ROUTINES FOR CONVECTIVE CLOUD MODULE
C
C ... RELEASED AUGUST, 1986
C
C ... FOR INFORMATION, CONTACT:
C   B.C. SCOTT OR C.M. BERKOWITZ
C   BATTELLE NORTHWEST LABORATORIES
C   P.O. BOX 999, RICHLAND, WA. 99352
C
C .....
C
C ... THIS CODE WAS DEVELOPED ON A VAX 11-78D USING VMS OPERATING
C   SYSTEM. THE ONLY CHANGES NEED TO RUN IT ON ANOTHER SYSTEM
C   ARE THE FOLLOWING:
C
C   1. THE INCLUDE STATEMENT FOUND AT THE START OF EACH PROGRAM UNIT
C   2. DEVICE NUMBERS FOR INPUT-OUTPUT
C
C .....
C
C ... THIS PROGRAM FORMS A TABLE OF VALUES WHICH HAVE BEEN GENERATED BY
C   CONVECTIVE CLOUD ROUTINE. TABLE IS OUTPUT ON DEVICE 8.
C
C   INCLUDE 'TABLE.FOR'
C   COMMON R(NVM1),RC(NVM1),W(NVM1)
C   COMMON ALPH(NVM1),DEPOS(NVM1),PRATE(NVM1)
C   COMMON FALL(NVM1),AUTO(NVM1),POUT(NVM1)

```

```

COMMON FEN_CD(NVM1),FCD_EN(NVM1),FTOCEL(NVM1),TEMP(NVM1)
DIMENSION Q(6)
C
C ... FLAGS:
C     ICLCUD, ISTAR = 0  WHEN INTERPOLATION IS DONE FOR STABLE CLOUD OR FOR
C     RAINSHAFT PORTION OF CUMULUS CLOUD
C     ICLCUD, ISTAR = 1  WHEN INTERPOLATION IS FOR UPDRAFT PORTION OF
C     THE CUMULUS CLOUD
C
C
C ... FORM A TABLE FOR EACH OF THE DESIRED VALUES.  LOADING OF THE
C     VALUES IS RX(NV,NP,NH,NT)
C
C
C     INUNIT=11  ! ONE PLUS INPUT DEVICE NUMBER OF RAINSHAFT DATA
C     IOUT=8    ! OUTPUT DEVICE NUMBER FOR NEW TABLE
C
C     WRITE(6,B06)
8D6  FORMAT(1H0,' TYPE 0 FOR STABLE CLOUD, 1 FOR CONVECTIVE CLOUD')
C     ACCEPT *, ICLCUD
C     ISTAR=ICLCUD
C
C     IUNIT = INUNIT
C
C ... STABLE CLOUD AND RAINSHAFT PART OF CONVECTIVE CLOUD WRITTEN ON
C     UNIT 10. UPDRAFT PORTION OF CONVECTIVE CLOUD, RAINRATE, STORM
C     DURATION WRITTEN ON UNIT 9.
C
C     LIM=ICLCUD+1
C
C     ICLCUD=-1
C     JJ=1
C     DO 600 NNN=1,LIM
C     ICLCUD=ICLCUD+1

```

```

IUNIT=IUNIT-1
KK=0
DO 400 M=1,NT
H=V0(2)
DO 100 K=1,NH
DO 200 L=1,NP
KK=KK+1
IF(ISTAB .EQ. 0) READ (IUNIT) R,RC,W,ALPH,DEPOS,PRATE,FALL,AUTO
IF(ISTAB .EQ. 1) READ (IUNIT) R,RC,W,ALPH,DEPOS,PRATE,FALL,AUTO
1 ,FEN_CD,FCD_EN,FTOCEL,TEMP
IF(ICLOUD .EQ. 1) READ (IUNIT) POUT,TIMEX(KK),RATEX(KK)
901 FORMAT(1H ,4(13,1X),2(1X,1PE10.3))
DO 300 I=1,NVM
RE(I)=R(I)
RCE(I)=RC(I)
WE(I)=W(I)
ALPHE(I)=ALPH(I)
DEPOSE(I)=DEPOS(I)
PRATEE(I)=PRATE(I)
FALLE(I)=FALL(I)
AUTOE(I)=AUTO(I)
POUTE(I)=POUT(I)
IF(ISTAB .EQ. 0) GO TO 300
FEN_CDE(I)=FEN_CD(I)
FCD_ENE(I)=FCD_EN(I)
FTOCELE(I)=FTOCEL(I)
TEMPE(I)=TEMP(I)
300 CONTINUE
C
C ... INTERPOLATE TO PUT ALL DATA ON USER VERTICAL GRID SPACING. TABLES
C ARE GENERATED FOR CLOUD WATER (RX), PRECIP WATER (RCX), AND
C UPDRAFT VELDCITY (WX). DZ IS THE USER GRID SPACING.
CALL HEIGHT(H,DZ,JJ,ICLOUD,ISTAB)
200 CONTINUE

```

```

100  H=H+DV(2)
400  CONTINUE
600  CONTINUE
C
      WRITE(IOUT) DZ,TIMEX,RATEX,RX,RCX,WX,POUTX,ALPHX,DEPOSX,
1  PRATEX,FALLX,AUTOX,FEN_CD_X,FCD_ENX,FTOCELX,TEMPX
      WRITE (6,850)
850  FORMAT(1H ,' DONE')
      END

```

```

-----

      SUBROUTINE HEIGHT(HMAXA,DZU,JJ,ICLOUD,ISTAB)
C      THIS ROUTINE INTERPOLATES DATA FROM AN AVAILABLE HEIGHT GRID
C      PRODUCED IN PROGRAM 'WETUP'. THE DESIRED HEIGHT SPACING IS
C      DETERMINED BELOW. CURRENTLY A LINEAR GRID SPACING IS USED IN EACH.
C
C
C ... RELEASED AUGUST, 1986
C
C ... FOR INFORMATION, CONTACT:
C      B.C. SCOTT OR C.M. BERKOWITZ
C      BATTELLE NORTHWEST LABORATORIES
C      P.O. BOX 999, RICHLAND, WA. 99352
C
C      INCLUDE 'TABLE.FOR'
C
C ... VARIABLES IN CALL LIST:
C      HMAXA=AVAILABLE MAX HEIGHT (FROM CLOUD MODEL)
C      DZU=DESIRED GRID SPACING IN USER MODEL
C      JJ=COUNTER FOR STORAGE IN INTERPOLATION TABLE

```

```

C      ISTAB=0 FOR A STABLE CLOUD, =1 FOR CDNVECTIVE CLOUD
C
C ... OTHER VARIABLES
C      HMAX=MAX DESIRED HEIGHT (FOR USER'S MODEL)
C      DZC=AVAILABLE GRID SPACING IN CLOUD MODEL
C      DZU=DESIRED GRID SPACING IN USER MODEL
C      NV=NUMBER OF DESIRED GRIDS IN USER'S MODEL
C      NVM=NUMBER OF AVAILABLE GRIDS IN THE CLOUD MODEL
C
C      DZU=HMAX/FLOAT(NV-1)
C      DZC=HMAXA/FLOAT(NVM-1)
C
C      MAXC=NVM ! MAX CLOUD MODEL GRID INDEX
C      MAXU=NV ! MAX USER MODEL GRID INDEX
C
C      RX(JJ)=RE(1) ! SET VALUES AT USER GRID BASE
C      RCX(JJ)=RCE(1)
C      WX(JJ)=WE(1)
C      ALPHX(JJ)=ALPHE(1)
C      DEPOSX(JJ)=DEPOSE(1)
C      PRATEX(JJ)=PRATEE(1)
C      FALLX(JJ)=FALLE(1)
C      AUTOX(JJ)=AUTOE(1)
C      IF(1CLOUD.EQ.1) POUTX(JJ)=POUTE(1)
C      IF(ISTAB .EQ. 0) GO TO 150
C      FEN_CDX(JJ)=FEN_CDE(1)
C      FCD_ENX(JJ)=FCD_ENE(1)
C      FTOCELX(JJ)=FTOCELE(1)
C      TEMPX(JJ)=TEMPE(1)
C
C
150 I=1 ! USER GRID INDEX
L=2 ! CLOUD MODEL GRID INDEX
ZC=0. ! STARTING HEIGHT, CLOUD GRID
ZU=DZU !STARTING HEIGHT, USER GRID

```


C

```
105  JJ=JJ+1  ! ARRAY INDEX
      RX(JJ)=0.
      RCX(JJ)=0.
      WX(JJ)=0.
      ALPHX(JJ)=0.
      DEPOSX(JJ)=0.
      PRATEX(JJ)=0.
      FALLX(JJ)=0.
      AUTOX(JJ)=0.
      IF(ICLOUD .EQ. 1) POUTX(JJ)=0.
      IF(ISTAB .EQ. 0) GO TO 104
      FEN_CD(X(JJ)=0.
      FCD_ENX(JJ)=0.
      FTOCELX(JJ)=0.
```

C

```
104  IF(L .LE. MAXC) GO TO 201  ! IF TRUE, STILL CLD DATA AVAIL.
      I=I+1
      IF(I .GT. MAXU) GO TO 103  ! DONE, IF TRUE
      IF(ISTAB .EQ. 0) GO TO 105
      TEMPX(JJ)=TEMPX(JJ-1)-9.E-3*DZU
      TEMPX(JJ+1)=TEMPX(JJ)-9.E-3*DZU  ! ASSUME STABLE LAPSE
      GO TO 105  ! ZERO VALUES ABOVE CLOUD TOP
```

C

```
201  IF(ZU .GT. (ZC+DZC)) GO TO 101  ! INCREMENT CLD GRID AND HT
      I=I+1  ! INCREMENT USER GRID INDEX
      IF(I .GT. MAXU) GO TO 103  ! DONE IF TRUE
      DELTA=(ZU-ZC)/DZC
      RX(JJ)=RE(L-1)+(RE(L)-RE(L-1))*DELTA
      RCX(JJ)=RCE(L-1)+(RCE(L)-RCE(L-1))*DELTA
      WX(JJ)=WE(L-1)+(WE(L)-WE(L-1))*DELTA
      ALPHX(JJ)=ALPHE(L-1)+(ALPHE(L)-ALPHE(L-1))*DELTA
      DEPOSX(JJ)=DEPOSE(L-1)+(DEPOSE(L)-DEPOSE(L-1))*DELTA
      PRATEX(JJ)=PRATEE(L-1)+(PRATEE(L)-PRATEE(L-1))*DELTA
```

```

FALLX(JJ)=FALLE(L-1)+(FALLE(L)-FALLE(L-1))*DELTA
AUTOX(JJ)=AUTOE(L-1)+(AUTOE(L)-AUTOE(L-1))*DELTA
IF(ICLOUD.EQ.1) POUTX(JJ)=POUTE(L-1)+(POUTE(L)-POUTE(L-1
1 ))*DELTA
IF(ISTAB .EQ. 0) GO TO 100
FEN_CD X(JJ)=FEN_CDE(L-1)+(FEN_CDE(L)-FEN_CDE(L-1))*DELTA
FCD_EN X(JJ)=FCD_ENE(L-1)+(FCD_ENE(L)-FCD_ENE(L-1))*DELTA
FTOCELX(JJ)=FTOCELE(L-1)+(FTOCELE(L)-FTOCELE(L-1))*DELTA
TEMPX(JJ)=TEMPE(L-1)+(TEMPE(L)-TEMPE(L-1))*DELTA
100  ZU=ZU+DZU
      JJ=JJ+1
      GO TO 201
C
101  L=L+1
      ZC=ZC+DZC
      GO TO 104
C
103  RETURN
      END
>

```

DISTRIBUTION

No. of
Copies

OFFSITE

30 DOE Technical Information Center

J. Chang
National Center for Atmospheric
Research
PO Box 3000
Boulder, CO 80307

J. F. Clarke (MD-80)
U.S. Environmental Protection Agency
Atmospheric Sciences
Research Laboratory
Research Triangle Park, NC 27711

R. E. Dennis (MD-80)
U.S. Environmental Protection Agency
Atmospheric Sciences
Research Laboratory
Research Triangle Park, NC 27711

J. L. Durham (MD-57)
U.S. Environmental Protection Agency
Atmospheric Sciences
Research Laboratory
Research Triangle Park, NC 27711

R. G. Lamb (MD-80)
U.S. Environmental Protection Agency
Atmospheric Sciences
Research Laboratory
Research Triangle Park, NC 27711

F. G. Vukovich
Research Triangle Institute
PO Box 12194
Research Triangle Park, NC 27709

C. J. Walcek
National Center for Atmospheric
Research
PO Box 3000
Boulder, CO 80307

No. of
Copies

ONSITE

2 DOE Richland Operations Office

R. A. Rochette
J. J. Sutey

34 Pacific Northwest Laboratories

C. M. Berkowitz (5)
E. G. Chapman
S. F. Crowder (10)
D. W. Dragnich
R. C. Easter
J. M. Hales
P. C. Hays
D. J. Luecken
N. S. Laulainen
W. T. Pennell
B. C. Scott
J. A. Stottlemyre
M. E. Strong
R. E. Wildung
Publishing Coordination (2)
Technical Report Files (5)