QUATERNARY LIBRARY

A Numerical Investigation of Sulfate Production and Deposition in Midlatitude Continental Cumulus Clouds

by

Gregory R. Taylor

A dissertation submitted in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Washington

1987

Approved by_____

(Chairperson of Supervisory Committee)

Program Authorized
to Offer Degree_____

Date_____

Table of Contents

Chapter

Ch. 1 Introduction 1

Ch. 2 Cloud Model 8

2.2 Area averaged equations 9

2.3 Momentum terms 20

2.4 Energy equation 24

2.5 Microphysical parameterization 26

2.6 Sub-cloud forcing 28

2.7 Radar equation 31

Ch. 3 Comparison with Observations 33

3.1 CCOPE July 19 34

3.2 CCOPE August 6 55

Ch. 4 Model Sensitivity 75

4.1 Sub-cloud forcing 75

4.2 Entrainment formulation 81

4.3 Microphysical parameters 83

4.4 Summary 93

Ch. 5 Chemical Model 98

5.1 General comments 98

5.2 Calculation of pH 100

5.3 Sulfate source terms 101

5.4 Ammonium source terms 109

5.5 Hydrogen peroxide source terms 111

5.6 Fallout terms 113

Ch. 6 Chemical Sensitivities 114

QUATERNARY LIBRARY

6.1 General Comments 114
6.2 Relative importance of scavenging and oxidation 115
6.3 Linearity of sulfate deposition 122
6.4 Role of entrainment 124
6.5 Role of the ice phase 127
6.6 Clouds growing in a modified environment 129
6.7 Summary 130

Ch. 7 Summary 150

References 155

Appendix
Ap. A Flux-Corrected transport advection 165
Ap. B Outer region Bessel mode computation 168
Ap. C Temperature calculation 169
Ap. D Microphysical parameterization 171
Ap. E Supplemental figures 184
Ap. F Model Code 218
Subroutine Clouds 227
Subroutine Tspec 244
Function EW 253
Subroutine Forcer 254
Subroutine Chemset 266
Subroutine Adchem 272
Subroutine Dtset 275
Subroutine Parice 279
Subroutine Parkes 282
Subroutine Intep 286
Subroutine Sumer 287
Subroutine ETBFCT 289

LIST OF FIGURES

Figure
2.1 Cloud model schematic 9
2.2 H2 cumulus cloud penetration 17
2.3 Model flow diagram 19
2.4 Microphysics schematic 27
July 19 base run figures
3.1 July 19 Miles City Sounding 34
3.2 Inner region vertical velocity plot 42
3.3 Inner region total water plot 43
3.4 Radar contour plot 44
3.5 Outer region vertical velocity plot 45
3.6 Outer region total water plot 46
3.7 Observed radar contours plot 47
3.8 July 19 virtual temperature sounding 48
3.9 Composite H2 penetration Paluch plot 49
3.10 Saturation point mixing diagram 50
3.11 Model point Paluch plot - inner region 51
3.12 Model point Paluch plot - outer region 52
3.13 Composite model Paluch plot 53
3.14 Model point time development 54
August 6 base run figures
3.15 August 6 Miles City sounding 56
3.16 222610 H2 penetration 60
3.17 223820 H2 penetration 61
3.18 224120 H2 penetration 62
3.19 Inner region vertical velocity plot 63
3.20 Inner region total water plot 64
3.21 Outer region vertical velocity plot 65
3.22 Outer region total water plot 66

QUATERNARY LIBRARY

3.23 Composite H2 Paluch plot 67
3.24 Saturation Point mixing diagram 68
3.25 SP mixing diagram - 567 mb 69
3.26 SP mixing diagram - 694 mb 70
3.27 Model point Paluch plot - inner region 71
3.28 Model point Paluch plot - outer region 72
3.29 Composite model Paluch plot 73
3.30 Model point Paluch plot 74

Sensitivity study figures

4.1 A&K entrainment Paluch plot 84
4.2 Cotton entrainment Paluch plot 85
4.3 TKE entrainment: avl=600 m Paluch plot 86
4.4 TKE entrainment: avl=800 m Paluch plot 87
4.5 TKE entrainment: avl=1000 m Paluch plot 88
4.6 Radar reflectivity: $N_{TG} = 10\ m^{-3}$ 95
4.7 Radar reflectivity: $N_{TG} = 100\ m^{-3}$ 96
4.8 Radar reflectivity: $N_{TG} = 1000\ m^{-3}$ 97

Chemistry figures

5.1 SO_2 and SO_4^{-2} reaction schematic 105
5.2 NH_4^+ schematic 110
5.3 H_2O_2 schematic 111
6.1 SO_2 oxidation rate - Cont. Back. case 134
6.2 SO_2 oxidation rate - Polluted case 135
6.3 Cloud droplet pH - Cont. Back. case 136
6.4 Cloud droplet pH - Polluted case 137
6.5 Inner region SO_2 oxidation rate; Poll. base run 138
6.6 Outer region SO_2 oxidation rate; Poll. base run 139
6.7 Inner region SO_2 ox. rate; no entrainment 140
6.8 Outer region SO_2 ox. rate; no entrainment 141
6.9 SO_2 oxidation rate; 5.0 ppb H_2O_2 142
6.10 Non-precip. cloud vertical velocity field 143
6.11 Non-precip. cloud total water field 144
6.12 Non-precip. cloud pH field 145

6.13 Modified SO_2 profile 146
6.14 Modified SO_4^{-2} profile 147
6.15 Modified NH_4^+ profile 148
6.16 Modified H_2O_2 profile 149

Figures in Appendix E

E.1.1 July 19 inner region entrainment velocity 185
E.1.2 July 19 inner region TKE 185
E.1.3 July 19 inner region cloud droplet contour plot 186
E.1.4 July 19 inner region precipitation contour plot 186
E.1.5 July 19 inner region snow contour plot 187
E.1.6 July 19 inner region graupel contour plot 187
E.1.7 July 19 outer region entrainment velocity 188
E.1.8 July 19 outer region TKE 188
E.1.9 July 19 outer region cloud droplet contour plot 189
E.1.10 July 19 outer region precipitation contour plot 189
E.1.11 July 19 outer region snow contour plot 190
E.1.12 July 19 outer region graupel contour plot 190
E.1.13 August 6 inner region entrainment velocity 191
E.1.14 August 6 inner region cloud droplet contour plot 192
E.1.15 August 6 inner region precipitation contour plot 192
E.1.16 August 6 inner region snow contour plot 193
E.1.17 August 6 inner region graupel contour plot 193
E.1.18 August 6 outer region entrainment velocity 194
E.1.19 August 6 outer region TKE 195
E.1.20 August 6 outer region cloud droplet contour plot 195
E.1.21 August 6 outer region precipitation contour plot 196
E.1.22 August 6 outer region snow contour plot 196
E.1.23 August 6 outer region graupel contour plot 197
E.2.1 Inner region vertical velocity variation with *wmult* 198
E.2.2 Inner region vertical velocity for A&K entrainment 199
E.2.3 Inner region total water for A&K entrainment 199
E.2.4 Outer region vertical velocity for A&K entrainment 200
E.2.5 Outer region total water for A&K entrainment 200

QUATERNARY LIBRARY

E.2.6 Inner region vertical velocity for Cotton entrainment 201
E.2.7 Inner region total water for Cotton entrainment 201
E.2.8 Outer region vertical velocity for Cotton entrainment 202
E.2.9 Outer region total water for Cotton entrainment 202
E.2.10 Inner region vertical velocity for TKE, avl=600 m 203
E.2.11 Inner region total water for TKE, avl=600 m 203
E.2.12 Outer region vertical velocity for TKE, avl=600 m 204
E.2.13 Outer region total water for TKE, avl=600 m 204
E.2.14 Inner region vertical velocity for TKE, avl=800 m 205
E.2.15 Inner region total water for TKE, avl=800 m 205
E.2.16 Outer region vertical velocity for TKE, avl=800 m 206
E.2.17 Outer region total water for TKE, avl=800 m 206
E.2.18 Inner region vertical velocity for TKE, avl=1000 m 207
E.2.19 Inner region total water for TKE, avl=1000 m 207
E.2.20 Outer region vertical velocity for TKE, avl=1000 m 208
E.2.21 Outer region total water for TKE, avl=1000 m 208
E.2.22 Microphysical contour plots for $\beta = 0.4$ 209
E.2.23 Microphysical contour plots for $\beta = 0.5$ 210
E.2.24 Microphysical contour plots for $\beta = 0.6$ 211
E.2.25 Microphysical contour plots for $N_{TG} = 10$ 212
E.2.26 Microphysical contour plots for $N_{TG} = 100$ 213
E.2.27 Microphysical contour plots for $N_{TG} = 1000$ 214
E.3.1 NH_4^+ aerosol plot for Continental Background case 215
E.3.2 Cloud drop NH_4^+ plot for Continental Background case 215
E.3.3 Precipitation NH_4^+ plot for Continental Background case 216
E.3.4 Snow NH_4^+ plot for Continental Background case 216
E.3.5 Graupel NH_4^+ plot for Continental Background case 217

LIST OF TABLES

2.1 Deformation tensor averaging lengths 13
2.2 Deformation tensor parameters 14
2.3 Relative importance of energy terms 25
2.4 Microphysical terms and acronyms 29
2.5 Bulk water source terms 30
3.1 Primary observations of July 19 35
3.2 Model base run parameters 36
3.3 Observation, 2D, 1.5D model intercomparison 41
3.4 August 6 base run parameters 57
4.1 Sensitivity run default parameters 75
4.2 Sub-cloud forcing tests: constant cloudbase velocity 77
4.3 Sub-cloud forcing tests: inner region velocity increased 78
4.4 Sub-cloud forcing tests: variable *wmult* 80
4.5 Sensitivity to entrainment formulation 82
4.6 Sensitivity to β in snow field calculation 90
4.7 Sensitivity to number of graupel, N_{TG} 92
5.1 Chemistry rate coefficients 101
6.1 Continental Background and Polluted distributions 116
6.2 NH_4^+ mixing ratios for scavenging estimate 117
6.3 S and NH_4^+ deposition masses for NH_4HSO_4 aerosol 118
6.4 S and NH_4^+ deposition masses for $(NH_4)_2SO_4$ aerosol 120
6.5 Summary of scavenging and oxidation fractions 122
6.6 Cloud base pollutant concentrations 123
6.7 S and NH_4^+ deposition linearity 125
6.8 Entrainment test summary 126
6.9 Entrainment distribution fractions 127
6.10 Ice phase test summary 128
6.11 Non-precip. cloud initialization parameters 130
6.12 Modified environment summary 131

QUATERNARY LIBRARY

Acknowledgements

Whatever is written here will be woefully inadequate for expressing my gratitude to Marcia Baker for all the guidance, inspiration and intellectual challenges she has given to me in the past few years. Her students are extremely fortunate!

During this time, from my wife, Debi, I have demanded much and given little - her love has made much of this possible. My son, Christopher, has seen far too little of me (and vice versa), and I hope soon he will realize that the act of putting on shoes is a necessary, but not sufficent, condition for going to work.

I would also like to thank my office mates; in particular, Jorgen Jensen, Phil Austin and Graciela Raga have all provided good company, scientific insight and assistance.

It is also fitting to acknowledge the support of two Seattle Institutions: The Blue Moon and her crew, for providing the 'atmosphere' required at times for resolving particularly throny problems, and KXRX (FM 96.5) for helping me through many long nights in the office.

Hwilum ic þurhræse þæt me on bæce rideð,
won wægfatu, wide to þringe
lagustreama full, · · ·

At times I rush through the dark water vessels
that ride upon my back, scattering far
the sea-stream's cup,

from the Exeter Book Old English
riddle no. 1, *Storm*

University of Washington

Abstract

A Numerical Investigation of Sulfate Production and Deposition in Midlatitude Continental Cumulus Clouds

by Gregory R. Taylor

Chairperson of the Supervisory Committee: Prof. Marcia B. Baker
Geophysics Program

I have developed a 1.5 dimensional Eulerian cumulus cloud model which incorporates: (a) two in-cloud regions and the environment, (b) a bulk water parameterization including the ice phase, (c) an entrainment formulation based on the turbulent kinetic energy, (d) a sustained sub-cloud forcing, and (e) a parameterization of the scavenging and production of sulfate. This model formulation was chosen in order to provide a simple dynamic framework which permits simultaneous up- and downdrafts in-cloud. It is used to examine the sensitivity of sulfate oxidation and deposition to the physical processes important in the development of midlatitude continental cumulus clouds.

The model is compared against aircraft and radar data for two clouds observed during the Cooperative Convective Precipitation Experiment held in South-central Montana during the summer of 1981. The overall dynamic, thermodynamic and microphysical agreement is good.

The cloud model is then used to examine characteristics of the sulfate cycle. Specifically: I have studied (a) the relative importance of aerosol scavenging to in-cloud production of sulfate through oxidation of sulfur dioxide by ozone and hydrogen

peroxide, (b) the importance of the entrainment of environmental air in the formation and subsequent deposition of sulfate, (c) the role of ice phase microphysical processes in determining the chemical properties of precipitation, and (d) the chemical effects on clouds due to a modification of the environment by previous clouds.

The results indicate for a continental background or moderately polluted atmosphere that aerosol scavenging accounts for between 50 and 80% of the in-cloud sulfate ultimately deposited; that ozone oxidation of sulfur dioxide may be important in the upper regions of the cloud; that although entrainment of environmental air is important in the cloud dynamics and microphysics, it is relatively unimportant in the chemistry, and that neglect of the ice phase when considering chemistry in these clouds may lead to overestimates of about 200% in sulfate deposition.

CHAPTER 1

INTRODUCTION

A great deal of effort is currently directed toward understanding the cloud chemical processes and their effects upon mesoscale wet deposition. Since the lifetimes and spatial dimensions of individual clouds cannot, in general, be explicitly resolved on a mesoscale grid, the cloud processes must be parameterized.

Cloud models used in cumulus parameterizations tend to be simple steady state Lagrangian adiabatic parcel models with little or no microphysics (e.g. Kuo, 1974, or Kuo and Anthes, 1984). The cloud models incorporated into the regional deposition models are generally of the same sophistication (see, for example, NCAR/TN-256+STR, 1985) For the most part, these simple models are used because of limitations in computer resources. This constraint will continue to impose severe limitations upon cloud models used in mesoscale deposition simulations. However, within these limitations it is important to represent the physical and chemical processes which dominate the deposition process adequately. As discussed below, current models are inadequate on several counts (in particular, the treatment of entrainment and the ice phase), and thus may fail to capture important features in the deposition process. In this dissertation I describe the development and testing of a new 1.5D Eulerian cumulus cloud model which yields cloud behavior consistent with observations, and the results of an investigation of the processing of sulfate by midlatitide continental cumulus clouds under a set of realistic environmental conditions.

1.1. Historical perspective

Numerical modelling of cumulus clouds has been an active field for over 25 years. However, adequate representation of cloud properties remains a difficult problem. Dynamical processes span seven orders of magnitude and three dimensions; drop sizes may span four orders of magnitude, and all three phases of water often coexist within cloud. The problem quickly becomes intractable; one is forced to model explicitly only

QUATERNARY LIBRARY

the important properties for the problem at hand, and attempt to parameterize the remaining physical processes. Even this more modest goal may present a challenge when the model is used to investigate electrification or in-cloud chemistry. While a major emphasis in the past few years has been on three dimensional modelling (e.g.; Klemp and Wilhelmson (1978), or Tripoli and Cotton (1982)) or the use of techniques such as nested grids (e.g., Klaassen and Clark, 1985) to achieve reasonable spatial resolution, these models quickly become real time computations, which limits their widespread use, and understanding the dynamics of such models becomes a full time effort.

Chemistry has been also been included in higher dimension cloud models. Sarma (1986) examined the in-cloud conversion of SO_2 for a single case using a two dimensional Eulerian model, but did not consider different clouds or environmental parameters. Futhermore, although he simulated a midlatitude continental convective storm, the ice phase was ignored. Tremblay and Leighton (1984, 1986) have constructed a chemistry model which includes the sulfate and NO_x cycles, and used it in conjunction with a three dimensional dynamic cloud model to investigate redistribution and deposition from small cumulus. Also, Lee (1986) has coupled chemistry and explicit warm rain microphysics in a 1.5 dimensional model investigation of fair weather cumulus.

In addition, although some work has been done in the tropics (Gidel, 1983) concerning the effects of cumulus ensembles in the deposition and redistribution processes, and for passive tracers (Niewiadomski, 1986), little has been reported on chemical evolution for midlatitude continental cumulus ensembles.

Ultimately, the aim of modelling must be to understand the three dimensional time-dependent interaction of the cloud dynamics, in-cloud turbulence, explicit microphysics and other relevant processes such as chemistry or electrification within the large scale environment. In the interim, simple models continue to play a useful role.

The limitations on computational resources dictate that most modelling studies involving secondary cloud processes such as chemistry (secondary in the sense that they normally have little effect upon the overall dynamics and microphysics) will be undertaken with relatively simple models of the dynamics and microphysics. Lagrangian parcel models (e.g., Hales, 1982); variations involving mixed and adiabatic regions (e.g., Walcek and Taylor, 1986); or versions of the Eulerian axisymmetric model of Asai and Kasahara (1967) (e.g., Lee and Shannon, 1984) have been used.

Most of these simple models are not without problems, however. Warner (1970) discussed the inability of Lagrangian bulk entrainment models to predict cloudtop height and in-cloud liquid water simulataneously. Furthermore, investigations by Jensen (1985) indicate Lagrangian bulk entrainment models cannot, in general, simultaneously predict the observed in-cloud total water and a conserved thermodynamic tracer, such as θ_q (Paluch, 1979). Cotton (1975a) compared the total liquid water calculated in a one dimensional time dependent Eulerian model based on the model of Asai and Kasahara (1967) with observations of small maritime cumulus, and found reasonable agreement when the time of predicted model values lagged the aircraft penetration by five minutes. However, no comparison of thermodynamic tracers was done.

Most of these cloud models ignore several important observations; a) that ice phase processes can be extremely important in midlatitude continental precipitation formation (e.g., Dye et al., 1986), which modifies the in-cloud aqueous chemistry, and b) cumulus clouds involve time dependent dynamic and microphysical non-linear processes, which may not couple in an obvious manner to aqueous chemistry. Furthermore, c) some important recent observations (Jensen and Blyth (1986), Rogers and Gardiner (1986), Jensen (1985), Boatman and Auer (1983), Raymond and Wilkening (1982), Telford and Wagner (1980), and Paluch (1979), for example) have illuminated the role entrainment plays in continental cumulus cloud evolution.

1.2. Observational constraints

Any cloud model results must ultimately be compared with observations. Currently, these observations are of several types; a) cloud penetrations by instrumented aircraft, b) radar reflectivity measurements, and c) ground based visual observations. The observed cloud properties have been noted and discussed by Warner (1970), Simpson (1971), Telford (1975), Cotton (1975a,b), Paluch (1979), Jensen (1985) and many others. From these, several important constraints for cumulus cloud modelling emerge:

a) Microphysical evolution

Midlatitude continental cumulus clouds have high cloud droplet concentrations (300-1000 cm^{-3}) and narrow initial droplet spectra ($\overline{r} \approx 10\mu$, $\sigma \approx 2\mu$) with large portions of the cloud below 0 °C. These conditions are all important for the microphysical parameterization of these clouds. The high droplet concentration and narrow droplet spectra imply that the precipitation formation mechanism will likely be much different than in maritime cumulus. For clouds with temperatures below 0 °C, the ice phase processes should be considered.

Observations of isolated continental cumulonimbus (e.g., Cooper, 1978; Hobbs et al., 1980; or Dye et al., 1986) indicate the dominant precipitation formation mechanism in continental cumulus clouds is vapor deposition on ice crystals which subsequently grow into graupel particles. These in turn collect supercooled cloud droplets via riming and melt when falling through the 0 °C isotherm. Observations by Heymsfield et al. (1984) of cumulonimbus in Oklahoma also indicate that the ice phase is important in the precipitation process for clouds with moderate (i.e., droplet concentrations of $\approx$ 300 cm^{-3}) continental droplet specta. Furthermore, significant portions of deep cumulus may have temperatures below the homogeneous nucleation temperature of water ($\approx$–37 °C).

b) Vertical transport

Cumulus clouds are characterized by vigorous convection. Vertical velocities may exceed 10 ms^{-1} over a significant portion of the cloud lifetime for a large system. These clouds therefore represent an efficient means of vertical transport from the boundary layer. Large negative velocities are also observed and often exist simultaneously with updrafts. These downdrafts are usually driven by evaporative cooling and water loading, but recently there has been an indication that dynamic forcing (Klaassen and Clark, 1985) may also be important in some instances. In addition, shear may play an important role in organizing the convection and permitting the convection to persist by allowing falling precipitation to fall away from the updraft rather than down through it. The driving force in cumulus clouds is the buoyancy generated via latent heat release as water condenses or freezes. However, the pressure perturbation (from hydrostatic equilibrium) also modifies the velocity field, and as shown by Wilhemson

and Ogura (1972), and others, it may be of the same order of magnitude as the buoyancy.

c) Entrainment

While observed in-cloud liquid water values often show much small scale variablilty, no monotonic gradient is usually seen from cloud center to edge, and the transition at cloud edge is often abrupt. Also, the observed in-cloud liquid water values are generally less than expected for a non-precipitating parcel undergoing adiabatic ascent, although adiabatic cores have been reported (Jensen, 1985).

Mixing has remained one of the puzzling problems in the study of cumulus clouds. The early work on entrainment was centered on descriptions of laboratory tank experiments of thermals and plumes. These exhibit similarity flow characteristics, and lead to bulk entrainment descriptions for clouds based on the same similarity flow conditions. Warner (1970) summarized the observational evidence which indicated such an entrainment hypothesis was unsatisfactory because cloud models incorporating bulk entrainment based on similarity flow arguments were incapable of simultaneously predicting observed cloud liquid water and cloud top height. Observations presented by Paluch (1979), and others since have indicated that a two-point mixing process, i.e., mixing of cloudbase air that has risen adiabatically with environmental air at one pressure level, is sometimes the dominant mixing process. However, recent observations by Rogers and Gardiner (1986), Lawson and Rodi (1986), Jensen and Blyth (1986), and others indicate that the time dependent mixing process is often more complicated than a simple two point mixing. Mixing appears to occur at all levels of the cloud with the mixed parcels undergoing buoyancy sorting in the stages immediately prior to dissipation, as suggested by Telford (1975), and Raymond and Blyth (1986).

1.3. Purpose of this work

The goal of this work is twofold:

I). To develop a cloud model which is computationally simple, yet includes the physical processes important for chemical evolution in midlatitude continental cumuli. These processes include:

QUATERNARY LIBRARY

a) a microphysical parameterization which incorporates the ice phase,

b) time dependent dynamics which permits simultaneous updrafts and downdrafts to develop in-cloud, and

c) a formulation of entrainment based upon a physical parameterization.

The model will be tested against observations and examined for response sensitivities due to variation in the adjustable physical parameters.

II). To begin an examination of the dependence of the chemical fields upon the physical processes which take place in-cloud and in the near environment. Specifically, I will address five points relevant to the acidification of, and sulfate deposition from, midlatitude continental cumulus clouds.

a) The relative importance of aerosol scavenging to in-cloud production of SO_4^{-2} through oxidation of SO_2. The fraction of SO_4^{-2} deposited due to sub-cloud scavenging is also examined.

b) The linearity of SO_4^{-2} deposition with respect to changes in the net sulfur flux through cloudbase.

c) The importance of the entrainment of environmental air in the formation and subsequent deposition of SO_4^{-2}.

d) The role of ice phase microphysical processes in determining the chemical properties of precipitation.

e) The chemical effects on clouds due to a modification of the environment by previous clouds.

The first two questions concern the response of midlatitude continental cumulus clouds to variations in the chemical fields used as input. The third and fourth questions are concerned with the interaction of entrainment and ice processes and chemistry. Although important in the dynamics and microphysics for this type of cloud, their role in acidification and deposition have been studied little. The last question deals with the cloud - environment interaction in a simple case, i.e., precipitating clouds growing in an environment already processed by smaller clouds. Such a question is pertinent to the problem of the parameterization of an ensemble of cumulus clouds on a mesoscale grid.

1.4. Organization

The cloud model developed for this investigation is discussed in Chapter 2. The base run model simulations for two clouds observed during the Cooperative Convective

Precipitation Experiment (CCOPE) are found in Chapter 3. I examine the sensitivities of the dynamic and microphysical model response to variations in the adjustable physical parameters in Chapter 4. The cloud chemistry model is developed in Chapter 5. Chapter 6 is a discussion of the dependence of the chemical fields on the physical processes listed above. A summary and suggestions for future research are given in chapter 7.

CHAPTER 2

CLOUD MODEL

2.1. General comments

The cloud model used is similar to that of Asai and Kasahara (1967) and Yau (1980) in that two concentric axisymmetrical averaged regions are incorporated in a time dependent Eulerian grid. This cumulus model departs from these models in that both regions are initially in-cloud, and interaction with the environment outside the outer cloud region is via entrainment and turbulent mixing into the outer region. This formulation was chosen as it permits simultaneous udpraft and downdraft regions to develop independently in a simple dynamic framework, and furthermore allows a central core to entrain environmental air at a lower rate than either the outer region or a single region time dependent cumulus model such as used by Cotton (1975a) or Cheng (1981). Figure (2.1) illustrates the model geometry.

Because the cloud model is used for studying both shallow and deep convection, the anelastic continuity equation

$$\frac{1}{r}\frac{\partial}{\partial r}(\rho_o ru)+\frac{\partial}{\partial z}(\rho_o w)=0 \quad , \tag{2.1}$$

where ρ_o is the base state density, is used. As shown by Dutton and Fichtl (1969), this approximation for the continuity equation permits gravity wave solutions and is satisfactory when considering deep convection. Note that a consequence of incorporating the anelastic approximation is that all processes are referenced to a constant base state density ρ_o, which allows the density to be factored in and out of the partial time derivative in the advection routines. This will be discussed in the section below on numerical advection.

In the following sections I will first write the generalized area averaged advection equations for the inner and outer regions, then discuss the terms common to all the physical parameters;, i.e., the vertical and lateral eddy flux terms, dynamic entrainment, and numerical advection. The source terms for momentum, energy, and the

water variables will then be derived. Appendix F is a listing of the model code.

2.2. Area averaged equations

Following Asai and Kasahara (1967), the cloud model equations are expressed in cylindrical coordinates denoted by (r,θ,z). r is the radial coordinate, θ the azimuthal coordinate and z the vertical coordinate. The equations for momentum, energy, moisture and continuity are averaged by integrating over the appropriate cross section and dividing by ρ_o multiplied by the cross sectional area for the regions shown in figure 2.1, to obtain the cloud model equations. Defining

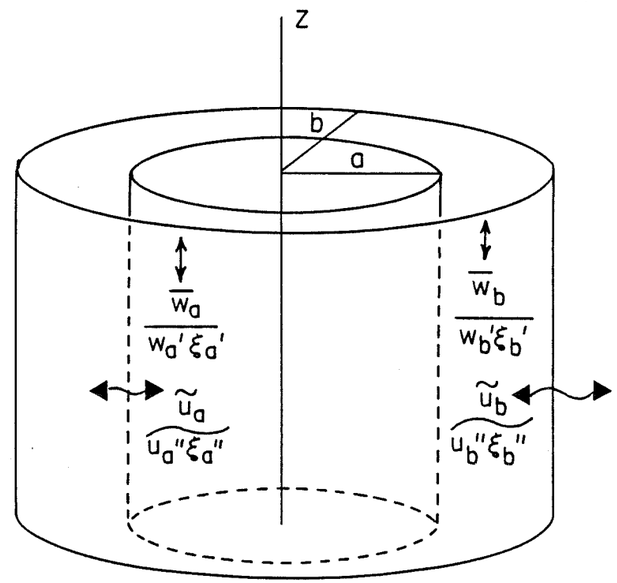


figure 2.1
Schematic of cloud model showing geometry of inner and outer regions.

$$\overline{\xi}_{a,b}=\frac{1}{\delta_{a,b}}\int_0^{2\pi}\int_{0,a}^{a,b}\xi_{a,b}\,r\,dr\,d\theta \tag{2.2}$$

with $\delta_a=\pi a^2$ and $\delta_b=\pi(b^2-a^2)$, and

$$\widetilde{\xi}_{a,b}=\frac{1}{2\pi}\int_0^{2\pi}\xi_{a,b}\,d\theta \tag{2.3}$$

evaluated at r=a,b, with

$$\xi'_{a,b}=\xi_{a,b}-\overline{\xi}_{a,b} \tag{2.4}$$

and,

$$\xi''_{a,b}=\xi_{a,b}-\widetilde{\xi}_{a,b}\quad , \tag{2.5}$$

the cloud model equations for the inner region with the radius a are of the form

$$\rho_o\frac{\partial\overline{\xi}_a}{\partial t}=-\frac{\partial}{\partial z}(\rho_o\overline{w}_a\overline{\xi}_a)+\widetilde{\xi}_a\frac{\partial}{\partial z}(\rho_o\overline{w}_a)$$

$$-\frac{\partial}{\partial z}(\rho_o\overline{w_a'\,\xi_a'})-\frac{2}{a}\rho_o(\widetilde{\xi_a''u_a''})+\overline{S}_{\xi a}\quad , \tag{2.6}$$

where $\overline{\xi}$ represents momentum, energy or any of the water variables, and $\overline{S}_{\xi a}$ is a generalized source term. The subscript a denotes the inner region. Mass continuity is given by;

$$\frac{2}{a}\rho_o\widetilde{u}_a+\frac{\partial}{\partial z}(\rho_o\overline{w}_a)=0\quad . \tag{2.7}$$

Similarly, for the outer region with radius b:

$$\rho_o\frac{\partial\overline{\xi}_b}{\partial t}=-\frac{\partial}{\partial z}(\rho_o\overline{w}_b\overline{\xi}_b)+[\widetilde{\xi}_b\frac{\partial}{\partial z}(\rho_o\overline{w}_b)+$$

$$\frac{a^2}{(b^2-a^2)}(\widetilde{\xi}_b-\widetilde{\xi}_a)\frac{\partial}{\partial z}(\rho_o\overline{w}_a)\,]-\frac{\partial}{\partial z}(\rho_o\overline{w_b'\,\xi_b})$$

$$-\frac{2b}{(b^2-a^2)}\rho_o[\,(\widetilde{\xi_b''u_b''})-\frac{a}{b}(\widetilde{\xi_a''u_a''})\,]+\overline{S}_{\xi b} \tag{2.8}$$

and mass continuity;

$$\frac{2b}{(b^2-a^2)}\rho_o(\widetilde{u}_b-\frac{a}{b}\widetilde{u}_a)+\frac{\partial}{\partial z}(\rho_o\overline{w}_b)=0\quad . \tag{2.9}$$

The terms on the right hand side of Eqs. 2.6 and 2.8 may be identified as: (1) vertical advection, (2) dynamic entrainment, (3) vertical eddy flux, (4) lateral eddy flux, and (5) relevant source terms. This set of equations differs from those of Asai and Kasahara (1967) and Yau (1980) by the inclusion of the vertical eddy flux terms, and a separate continuity equation for the outer region. Cotton (1975a) included the vertical eddy fluxes, but considered only one region in his calculations.

The properties of the eddy flux terms $\overline{\xi'\,w'}$ and $\widetilde{\xi''u''}$ and the dynamic entrainment parameter $\widetilde{\xi}_{a,b}$must be specified in order to close the above set of equations. Note also, that although the entrainment velocity $\widetilde{u}_a$ is not explicitly included in the dynamic entrainment term, it is evaluated from the continuity equation in each region and used for the lateral eddy flux terms.

2.2.1. Eddy fluxes

The lateral and vertical eddy fluxes are calculated using a Smagorinsky (1963) form of the eddy diffusivity in a manner similar to Cotton (1975a). The momentum terms are given by

$$\begin{aligned}\overline{w''w''}&=-2K_m\frac{\partial w}{\partial z}&&\text{vertical}\\ \widetilde{w''u''}&=-K_m(\frac{\partial w}{\partial r}+\frac{\partial u}{\partial z})&&\text{lateral}\end{aligned} \tag{2.10}$$

and the eddy mixing of scalars such as energy and moisture is given by

$$\begin{aligned}\overline{w''\xi''}&=-K_a\frac{\partial\xi}{\partial z}\\ \widetilde{\xi''u''}&=-K_a\frac{\partial\xi}{\partial r}\end{aligned} \tag{2.11}$$

where K_m is given by

$$K_m=\frac{(k_1\Lambda_b)^2}{\sqrt{2}}\,|\,D_{ij}\,| \tag{2.12}$$

and $K_a=\sigma_t K_m$. $|\,D_{ij}\,|$ is the magnitude of the deformation tensor of an anelastic medium evaluated at r=0 for $\overline{w''w''}_a$, r=a for $\widetilde{w''u''}$, at r=0.5(a+b) for $\overline{w''w''}_b$, and r=b

QUATERNARY LIBRARY

for $\widetilde{w'\psi'}$. Λ_b is the buoyancy modified (Lilly, 1962) averaging length, k_1 has been set to 0.28 and the Prandtl number $\sigma_t = 3.0$. The buoyancy modification to the mixing length is given by

$$\Lambda_b = \Lambda \cdot \max(1,(1-\sigma_t \mathrm{Ri})^{0.25}) \quad , \tag{2.13}$$

where Ri is a generalized gradient Richardson number;

$$\mathrm{Ri} = \frac{g(\frac{\partial}{\partial z}T_v - \Gamma)}{T_v \mid D_{ij} \mid^2} \quad . \tag{2.14}$$

T_v is the virtual temperature including water loading, defined in equations 2.32 and 2.33, and Γ is the adiabatic lapse rate. For unsaturated air,

$$\Gamma = \Gamma_d \equiv -\frac{g}{c_{pd}} \quad . \tag{2.15}$$

Saturated air uses a Γ weighted by the amounts of liquid water and ice present;

$$\Gamma = \frac{q_l\Gamma_l + q_i\Gamma_i}{q_l + q_i} \quad , \tag{2.16}$$

where q_l and q_i are the liquid water and ice mixing ratios, and Γ_l and Γ_i are the pseudoadiabatic gradients for water and ice, respectively;

$$\Gamma_l = \Gamma_d \left[\frac{1+5419.34\left(\frac{e_w(T)}{T\ p}\right)}{1+8388504.46\left(\frac{e_w(T)}{T^2\ p}\right)} \right] \quad , \tag{2.17}$$

and

$$\Gamma_i = \Gamma_d \left[\frac{1+6140.91\left(\frac{e_i(T)}{T\ p}\right)}{1+10771020.40\left(\frac{e_i(T)}{T^2\ p}\right)} \right] \quad , \tag{2.18}$$

for the saturated partial pressure of water vapor over liquid water, e_w, and ice, e_i. The temperature, T, is in Kelvins and the pressure, p, in Pascals.

This choice for the gradient Richardson number is an attempt to include the effects of the latent heat release due to condensation and freezing in the stability parameter. However, to avoid the necessity of computing the amount of freezing (or melting) water at each vertical level for this computation, all ice is assumed to have formed via sublimation. Such an approximation biases the calculated gradient Γ toward the ice pseudoadiabatic gradient when ice is present, but the error involved is no larger than that resulting from use of the pseudoadiabatic approximation of a system containing liquid water and ice (Saunders, 1957).

$\mid D_{ij} \mid_{r=0}$ is given by

$$\mid D_{ij} \mid_{(r=0)} = \left[\left(\frac{1}{\rho}\frac{\partial}{\partial z}\rho w\right)^2 + \left(\frac{\partial w}{\partial z}\right)^2 \right]^{0.5} \tag{2.19}$$

while $\mid D_{ij} \mid_r$ for r=a, .5(a+b), and b is

table 2.1
Averaging lengths for $\mid D_{ij} \mid$ evaluted at the midpoint of each region and the boundaries

r	Λ_b
0	$(\pi a^2 dz)^{1/3}$
a	$(2\pi a^3)^{1/3}$
$\frac{a+b}{2}$	$(\pi(b^2-a^2)dz)^{1/3}$
b	$(2\pi(b^2-a^2))^{1/2}$

QUATERNARY LIBRARY

$$| D_{ij} |_r =$$

$$\left[\left(\frac{u}{\rho} \frac{\Delta\rho}{\Delta r} + \frac{u}{r} + \frac{1}{\rho} \frac{\partial}{\partial z} \rho \hat{w} \right)^2 + \left(\frac{u}{r} \right)^2 + \left(\frac{\partial \hat{w}}{\partial z} \right)^2 + \frac{1}{2} \left(\frac{\Delta w}{\Delta r} + \frac{\partial \hat{u}}{\partial z} \right)^2 \right]^{0.5} \quad (2.20)$$

The ^ signifies a derivative evaluated over the same scale as Δr. Values for other parameters in Equations 2.19 and 2.20 are given in table 2.2.

2.2.2. Dynamic entrainment

The parameters $\widetilde{\xi}_b$ at the cloud outer boundary represent the material brought into the region to satisfy mass continuity via the anelastic approximation to the continuity equation. The non-physical geometric constraints of a fixed radius at the cloud boundary result in large amounts of entrainment to satisfy mass continuity. Using the Asai and Kasahara (1967) formulation for $\widetilde{\xi}_b$, i.e.,

table 2.2
$| D_{ij} |$ parameters evaluted at region midpoints and boundaries

	$\| D_{ij} \|_a$	$\| D_{ij} \|_{\frac{a+b}{2}}$	$\| D_{ij} \|_b$
u	$\tilde{u}_a$	$0.5(\tilde{u}_b+\tilde{u}_a)$	$\tilde{u}_b$
r	a	0.5(a+b)	b
Δr	a	b–a	b–a
ρ	$\rho_o\,(1+q_{tot\,a})$	$\rho_o\,(1+q_{tot\,b})$	$\rho_o\,(1+q_{tot\,b})$
$\Delta\rho$	$\rho_o\,(\,q_{tot\,b} - q_{tot\,a}\,)$	$\rho_o\,(\,\overline{q} - q_{tot\,a}\,)$	$\rho_o\,(\,\overline{q} - q_{tot\,b}$
w	w_a	w_b	w_b
Δw	$\overline{w}_b-\overline{w}_a$	$\overline{w}-\overline{w}_a$	$\overline{w}-\overline{w}_b$
Δz	a	0.5(a+b)	b

$$\widetilde{\xi}_b = \begin{cases} \overline{\xi} & \widetilde{u}_b \le 0 \\ \overline{\xi_a} & \widetilde{u}_b > 0 \end{cases} \quad (2.21)$$

results in rapid dilution of the cloud, and little growth.

Cotton (1975a) dealt with this problem in a fixed radius axisymmetric Eulerian cloud model by computing $\widetilde{\xi}_b$ from the lateral and upstream vertical flux. The lateral flux is given by

$$F_l = 2\pi\, R\, \delta z \;|\; \widetilde{u}_R \;| \quad (2.22)$$

while the upstream flux contribution is calculated from

$$F_{up} = \pi R^2 \;|\; \overline{w_{up}} \;| \quad (2.23)$$

for the boundary at radius R, and vertical grid spacing δz. The flux weighted value of $\widetilde{\xi}_b$ is then given by

$$\widetilde{\xi}_b = \begin{cases} F_l\xi_o + \dfrac{F_{up}\,\xi_{bup}}{F_l + F_{up}} & \widetilde{u}_b \le 0 \\ \dfrac{F_l\overline{\xi_b} + F_{up}\,\xi_{bup}}{F_l + F_{up}} & \widetilde{u}_b > 0 \end{cases} \quad (2.24)$$

where ξ_o is the environmental value, and ξ_{bup} is the upstream value of $\overline{\xi_b}$.

The rationale for such a parameterization lies in the assumption that the air in the immediate vicinity of the cloud boundary may be a mixture of cloud remnant and environmental air, rather than the unmixed environmental air. Entrainment through the cloud outer boundary will therefore be some mixture of cloud and environmental air. Although this is physically reasonable, the Cotton (1975a) flux- weighted scheme partitions the lateral and upstream contributions using the entrainment and vertical velocities. Since the entrainment velocity is calculated from $\frac{\partial}{\partial z}w$ using the continuity equation, the flux weighting depends on w and $\frac{\partial}{\partial z}w$, but remains a function of the fixed radius geometry.

Another approach is to base the entrainment of environmental air into the cloud on a simple physical model, and then formulate $\widetilde{\xi}_b$ to reflect this hypothesis. This

approach is used here.

Morton (1968), and Telford (1970) discussed the importance of turbulence in the entrainment process, and proposed that the turbulence intensity is a better predictor of entrainment rate than velocities derived from similarity flow arguments. As discussed in chapter 1, recent observations (e.g., Jensen et al., 1985) show that regions of nearly undiluted cloudbase air are characterized by low turbulence levels, while mixed regions may have high turbulence levels. Figure 2.2 contains the in-cloud liquid water and turbulence level tracings made during an aircraft penetration of a moderate cumulus on July 19, 1981 during CCOPE, and indicates the difference in turbulence levels between mixed and undiluted cloud regions.

While the idea of turbulence driven entrainment is not new, little has been done to parameterize entrainment in cumulus models in terms of the turbulence intensity. Lopez (1973) presented a Langrangian cloud model with an entrainment rate based on Telford's (1970) proposal. In this Eulerian cloud model I have incorporated the concept of entrainment as a function of the local turbulence using a method similar to Cotton's (1975a) flux weighted scheme. This entrainment formulation uses a weighted mean calculated from the local vertical velocity and a velocity derived from the local turbulent kinetic energy. The turbulent kinetic energy is calculated using an expression, derived by Lipps (1977), consistent with the Smagorinsky (1963) formulation of the eddy viscosity used above.

$$\mathrm{TKE} = 0.5 \left[\frac{k_1^{\,2}}{c_E} \right]^{0.67} \Delta^2 \mid D_{ij} \mid^{\,2} \qquad (2.25)$$

for a length scale Δ (avl in the model code) and deformation tensor D_{ij} (see equation 2.20). The constant k_1 is from equation 2.12, and the constant c_E from Deardorff (1973); $k_1 = 0.28$, $c_E = 0.7$.

The length scale Δ is the characteristic size of entrainment events, and will, in general, be an unknown function of cloud and environmental parameters. Turbulence spectra obtained by Rodi (1981) suggest that the dominant eddy size was approximately 300-800 meters in the Montana cumuli he observed. Δ of this size corresponds to eddy sizes larger than the vertical resolution of the model, i.e., 100-200 meters, but smaller than the effective radial resolution ($\approx$ 1000 meters). This insures that the

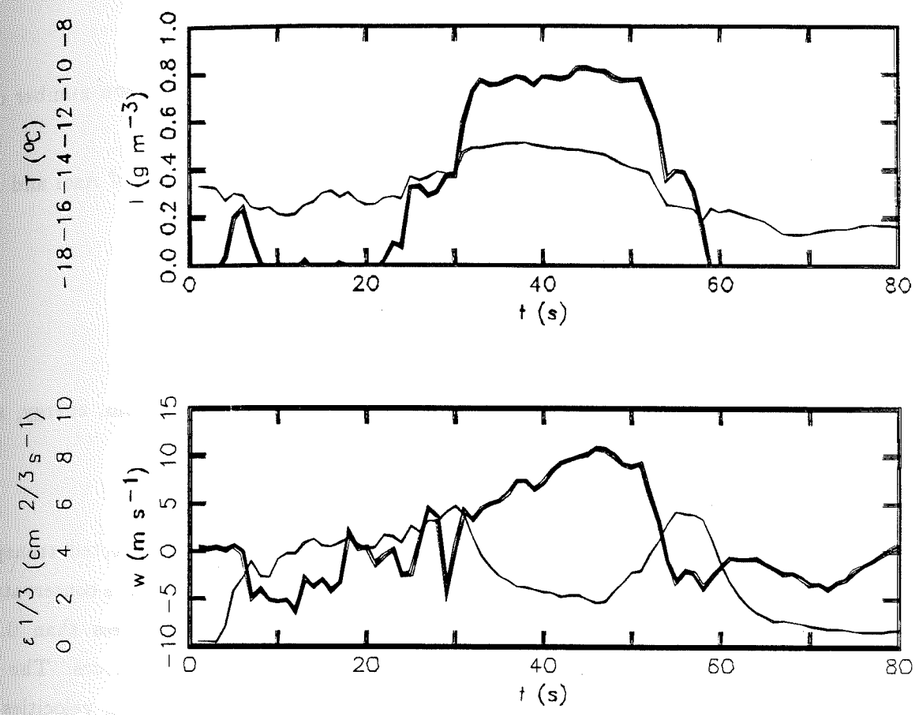


figure 2.2
H2 penetration of cloud "b" on July 19, 1981 during CCOPE.

turbulent kinetic energy calculated using equation 2.25 represents the subgrid turbulent kinetic energy when evaluated at the outer boundary. The sensitivity of the model to Δ is considered in chapter 4.

As can be seen in equation 2.25, this simple representation of the turbulent kinetic energy includes the assumption that the turbulent kinetic energy is in local equilibrium with the large scale dynamics, and as a consequence ignores the role of both advection and buoyancy generation in the local turbulent kinetic energy value. Buoyancy is included in a simple manner by using the buoyancy modified mixing length, i.e,

$$\Delta_{buoy} = \Delta \max(1, (1-\sigma_t Ri)^{0.25}) \quad , \tag{2.26}$$

where σ_t is the turbulent Prandtl number, and Ri is the Richardson number defined in the section above.

The properties for $\widetilde{\xi}_a$ are determined using the formulation of Asai and Kasahara (1967), i.e.,

$$\widetilde{\xi}_a = \begin{cases} \overline{\xi_b} & \widetilde{u}_a \leq 0 \\ \overline{\xi_a} & \widetilde{u}_a > 0 \end{cases} \tag{2.27}$$

$\widetilde{\xi}_a$ is calculated in this manner to insure continuity of momentum, energy, and moisture, in addition to mass, between the inner and outer regions.

2.2.3. Numerical advection

The area averaged equations are solved with the flux-corrected transport algorithm *ETBFCT* (Book, Boris and Zalesak, 1981). This algorithm is conservative, maintains positivity of advected quantities for Courant numbers less than 0.5, and is characterized by low diffusion and fourth-order-accurate phase errors. The numerical scheme is a modified Lax-Wendroff method in which the vertical velocities are computed at the half time step and used to advect momentum, energy and moisture over the full time step. This numerical algorithm has been shown (Boris and Book, 1976) to repress non-linear grid separation. Therefore all variables are defined at the same grid point, instead of a staggered grid for the velocities as is commonly used. Figure 2.3 shows the steps involved in the advection for one time step. Because the scheme is explicit, the inner and outer region equations are solved sequentially rather than simultaneously. The inner region is solved first in these computations.

The time step, δt, is permitted to vary according to the algorithm

$$\delta t = \frac{0.25\ \delta z}{\min(v_o, \ |\ v_{max}\ |)} \quad , \tag{2.28}$$

where δz is the grid spacing, $v_o = 10\ ms^{-1}$, and v_{max} is the maximum velocity in the model domain during the last time step; i.e.,

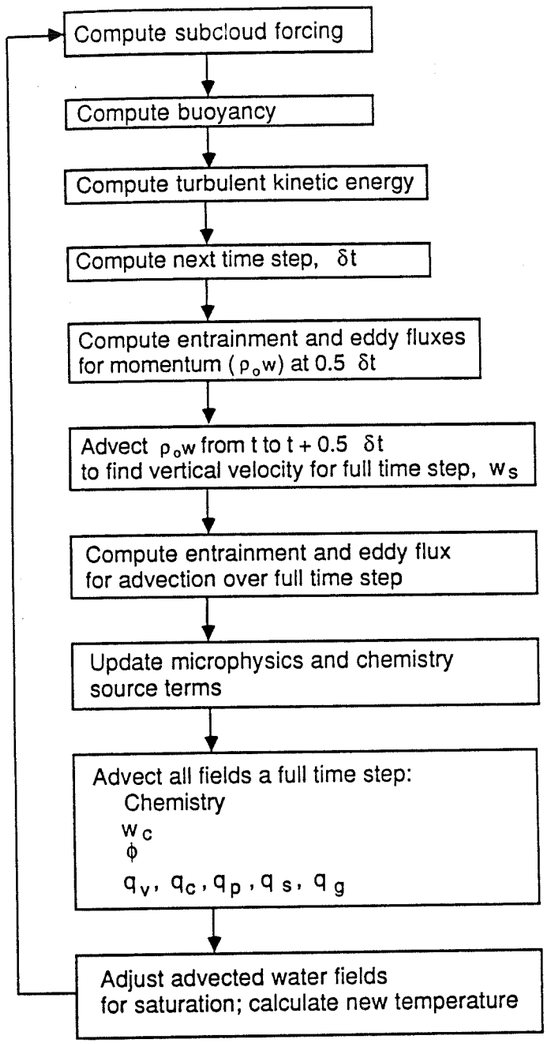


figure 2.3
flow diagram of the advection scheme for one time step

$$v_{max} = \max(\ |\ \overline{w}_a\ |\ ,\ |\ v_{fall}\ |\ ) \quad , \tag{2.29}$$

where v_{fall} represents the fall velocities calculated for precipitation, snow and graupel. v_o limits the time step to a maximum amount, e.g., $\delta t = 10$ s for $|\ v_{max}\ | \leq 10\ ms^{-1}$ and $\delta z = 200$ m, to limit the truncation error due to a variable time step, yet permit a smaller time step if required.

Since advection is via a flux transport algorithm, the quantities advected must be in units of density, (kg m^{-3}), rather than mixing ratio, (kg/kg). For this reason, the reference density, ρ_o, is taken through the time derivative in the advection equation, and all other terms retain ρ_o. The details of the advection algorithm are given in appendix A.

2.3. Momentum

The model is in essence one dimensional. Therefore only the vertical equation of motion is required. As the form for the equation in both the inner and outer regions are similar, consider only the inner region. The momentum equation is then given by:

$$\frac{\partial}{\partial t}\rho_o\overline{w}_a = -\frac{\partial}{\partial z}(\rho_o\overline{w}_a\overline{w}_a) + \widetilde{w}_a\frac{\partial}{\partial z}(\rho_o\overline{w}_a)$$

$$-\frac{\partial}{\partial z}(\rho_o\overline{w_a' w_a'}) - \frac{2}{a}\rho_o\widetilde{(w'_a u''_a)} + \rho_o g\overline{\beta}_a - \rho_o c_{pd}\theta_o\frac{\partial}{\partial z}\overline{\pi}_a \quad , \tag{2.30}$$

where w is the vertical velocity (positive upwards), u is the radial component (positive outwards), and ρ_o is the density of the base state defined from the hydrostatic equation

$$\frac{\partial p_o}{\partial z} = -g\rho_o \tag{2.31}$$

for the averaged pressure p_o.

2.3.1. Buoyancy

β is the buoyancy expressed in terms of virtual temperatures (T_v) and mixing ratios of water vapor (q_v), total liquid water (q_l) and ice (q_i) referenced to the density of dry air,

$$\beta = \frac{T_v - T_{vo}}{T_{vo}}\left(1 - \frac{q_l+q_i}{1+q_v}\right) - \frac{q_l+q_i}{1+q_v} \tag{2.32}$$

where the environmental value is denoted with a subscript o , and T_v is given by

$$T_v = T\left[\frac{1+1.608q_v}{1+q_v}\right] \quad . \tag{2.33}$$

2.3.2. Pressure perturbation

The last term on the right hand side of equation 2.30 represents the pressure perturbation from hydrostatic equilibrium. The pressure perturbation is included in the momentum equation because of work by Cheng (1981), Yau (1979), Schlesinger (1975), Wilhemson and Ogura (1972), Holton (1973) and others who have shown that the pressure perturbation term is of the same order of magnitude as the buoyancy term, and is dynamically important in deep convection. I have attempted to incorporate this effect in a simple manner, following the work of Wilhemson and Ogura (1972), and Yau (1979). A diagnostic equation for the pressure pertubation is derived from taking the divergence of the Navier-Stokes equation, i.e.,

$$\nabla\cdot(c_p\theta_o\rho_o\nabla\pi) = -\nabla\cdot(\rho_o u\cdot\nabla u) + \frac{\partial}{\partial z}g\rho_o\beta \tag{2.34}$$

where θ_o and ρ_o refer to the hydrostatic base state, and $g\rho_o\beta$ is the buoyancy. Expanding the equation and use of the continuity equation yields;

$$c_p\theta_o\rho_o\frac{\partial^2}{\partial^2 z}\pi + \left(\frac{\partial}{\partial z}c_p\theta_o\rho_o\right)\frac{\partial}{\partial z}\pi + c_p\theta_o\rho_o\nabla^2_H\pi = F(r,z) \quad , \tag{2.35}$$

where

$$F(r,z) = \frac{\partial}{\partial z}g\rho_o\beta - \frac{1}{r}\left[\frac{\partial^2}{\partial^2 r}r\rho_o(u^2+\overline{u'u'}) + 2\frac{\partial}{\partial z}\frac{\partial}{\partial r}r\rho_o(uw+\widetilde{w'u'}) + \frac{\partial^2}{\partial^2 z}r\rho_o(w^2+\overline{w'w'})\right] \tag{2.36}$$

To eliminate the radial derivatives in F(r,z), equation (2.35) is integrated over r. Writing $\pi \equiv \chi(z)\psi(r)$ and integrating from r_i to r_f;

$$c_p\theta_o\rho_o\frac{\partial^2}{\partial^2 z}\chi(z) + \left(\frac{\partial}{\partial z}c_p\theta_o\rho_o\right)\frac{\partial}{\partial z}\chi(z) - c_p\theta_o\rho_o\chi(z)k_{on}^2 = \frac{\int_{r_i}^{r_f} rF(r,z)dr}{\int_{r_i}^{r_f} r\psi(r)dr} \quad , \tag{2.37}$$

k_{on} in the last term on the left hand side of equation 2.37 is the first zero of the Bessel function J_n, and occurs in the equation because of the approximation (Holton, 1973) of the radial dependance $\psi(r)$ as a single mode of a Fourier-Bessel series. The inner region

QUATERNARY LIBRARY

uses $\psi(r) = J_o(k_{oo},r)$ similar to Holton (1973), while $\psi(r) = J_n(k_{on},r)$ in the outer region. n is chosen by requiring that the mode have a maximum near the center of the outer region is used, i.e.;

$$\frac{\partial}{\partial r}J_n(k_{on},r) = 0.0 \tag{2.38}$$

The derivative is evaluated at r = 0.5(a+b) for an inner region radius of r = a and an outer region radius r = b. A discussion of the technique used to determine the order of the Bessel mode used in the outer region may be found in Appendix B.

The integral of F(r,z) must be evaluated over each region independently. Furthermore, although integration of F(r,z) eliminates the necessity of knowing the radial dependence of the first derivatives in equation 2.36, a radial derivative of $u^2+\overline{u'u'}$ remains, so that some assumption must be made concerning the radial dependence of u and $\overline{u'u'}$. Two simple assumptions are used here. The radial dependence of u is calculated using the assumption that mass continuity is maintained throughout the region, i.e.,

$$u(r) = \widetilde{u}_a \frac{r}{a} \tag{2.39}$$

in the inner region, where $\widetilde{u}_a$ is the velocity at radius a, and

$$u(r) = \left[\frac{\widetilde{bu}_b - \widetilde{au}_a}{b^2-a^2}\right] r + \left[\frac{\frac{a}{b}(\widetilde{bu}_a - \widetilde{au}_b)}{b^2-a^2}\right]\frac{1}{r} \equiv \Upsilon r + \Omega\frac{1}{r} \quad , \tag{2.40}$$

for an entrainment velocity $\widetilde{u}_b$ at the outer cloud region radius b. The lateral eddy term $\overline{u'u'}$ is computed in an identical manner, i.e.,

$$\overline{u'u'}\,(r) = \overline{u'u'}_a \frac{r}{a} \quad , \tag{2.41}$$

for the inner region, and

$$\overline{u'u'}\,(r) = \left[\frac{b\overline{u'u'}_b - a\overline{u'u'}_a}{b^2-a^2}\right] r + \left[\frac{\frac{a}{b}(b\overline{u'u'}_a - a\overline{u'u'}_b)}{b^2-a^2}\right]\frac{1}{r} \quad , \tag{2.42}$$

for the outer region, where $\overline{u'u'}_a$ and $\overline{u'u'}_b$ are the values of $\overline{u'u'}$ evaluated at the inner region boundary ,r=a, and the outer cloud boundary, r=b. These may be evaluated

using an equation similar to equation 2.10, i.e.,

$$\overline{u'u'} = -2K_m\frac{\partial}{\partial r}u \quad , \tag{2.43}$$

which yields

$$\overline{u'u'}_a = -2K_{ma}\widetilde{u}_a\frac{1}{a} \tag{2.44}$$

for the inner region, and

$$\overline{u'u'}_a = -2K_{ma}\left[\Upsilon - \Omega\frac{1}{a^2}\right] \tag{2.45}$$

and

$$\overline{u'u'}_b = -2K_{mb}\left[\Upsilon - \Omega\frac{1}{b^2}\right] \tag{2.46}$$

for the outer region. Υ and Ω are defined in equation 2.40. $\overline{u'u'}_a$ is evaluated in each region separately because while the lateral velocity is continuous across the boundary at r=a, the radial derivative of $\widetilde{u}_a$ is not continuous, which implies a discontinuity in $\overline{u'u'}$ at r=a. As these terms only appear in the pressure perturbation equation no effort has been made to formulate a more consistent approximation for $\overline{u'u'}$.

Equation (2.37) is solved in each region independently using a simple marching technique (e.g., Richtmyer and Morton, 1967). The boundary conditions for each region are

$$\frac{\partial}{\partial z}\pi = \frac{g\rho_o\beta}{c_p}\theta_o\rho_o \tag{2.47}$$

at the lower boundary, and

$$\pi = 0 \tag{2.48}$$

at the upper boundary. The solution to equation 2.34 is then area averaged for use in the momentum equation (2.30);

$$\overline{\pi_a} = \frac{2\,\chi_a(z)}{a^2}\int_0^a r\psi_a(r)dr \tag{2.49}$$

for the inner region, and

$$\overline{\pi}_b = \frac{2\,\chi_b(z)}{b^2-a^2}\int_a^b r\psi_b(r)dr \qquad (2.50)$$

in the outer region.

2.4. Energy

The total energy (per unit mass) of a system containing water vapor, liquid and ice may be approximated by

$$\phi = H+gz+0.5w^2 \quad , \qquad (2.51)$$

for the enthalpy H of a system containing total water $q_{tot} = q_v+q_l+q_i$ referenced to 0 °C, and the density of dry air;

$$H = (c_{pd}+c_w q_{tot})T+L_v q_v-L_i q_i \quad , \qquad (2.52)$$

with the specific heats of dry air and water given by c_{pd} and c_w, and the latent heats of vaporization and freezing given by L_v and L_i, respectively. The moist static energy, H+gz, may be expressed in terms of the potential temperature, θ, by use of the hydrostatic assumption, i.e.,

$$H+gz \equiv H_\theta = (c_{pd}+c_w q_{tot})\theta+L_v q_v-L_i q_i \qquad (2.53)$$

Again considering the inner region, conservation of energy is given by;

$$\frac{\partial}{\partial t}\rho_o\overline{\phi}_a = -\frac{\partial}{\partial z}(\rho_o\overline{w}_a\overline{\phi}_a) + \widetilde{\phi}_a\frac{\partial}{\partial z}(\rho_o\overline{w}_a) - \frac{\partial}{\partial z}(\rho_o\overline{w_a'\,\phi_a'}) - \frac{2}{a}\rho_o(\widetilde{\phi_a''\,u_a''}) + \overline{S}_{\phi a} \quad , \qquad (2.54)$$

where $S_{\phi a}$ is the enthalpy flux due to falling precipitation given by

$$S_\phi = \frac{\partial}{\partial z}\Big[\rho_o\,(v_p q_p c_w T + [c_w(\min(T, 273.16)) + L_i](v_s q_s+v_g q_g)\Big] \qquad (2.55)$$

for the fall velocities v and mixing ratios q of rain, snow and graupel denoted by the subscripts p, s, and g, respectively. The temperature constraint is imposed on the specific heat of the ice terms to reflect an ice temperature maximum of 273.16 K.

The averaged energy, $\overline{\phi}$, contains the eddy terms representing the covariance of vertical velocity flucuations, and the moisture and temperature flucuations;

$$\overline{\phi} = \Big[(c_{pd}+c_w\overline{q}_{tot})\overline{\theta}+L_v\overline{q}_v-L_i\overline{q}_i\Big] + c_w\overline{q'_{tot}\theta'} + 0.5(\overline{w}^2+\overline{w'w'}) \quad . \qquad (2.56)$$

The relative importance of the terms in equation 2.56 may be estimated using the parameter values given in table 2.3. From this it may be seen that the covariance terms are of approximately the same order of magnitude as the mean velocity term. However, although $\overline{w'w'}$ is calculated for the eddy flux terms in the momentum equation using a mixing length approximation, computation of the moisture-temperature covariance term is not so straightforward. For this reason, and because the term is small compared to $\overline{\phi}$, it is not included in the expression used for the total energy, which is given by

$$\overline{\phi} = \overline{H}_\theta+0.5(\overline{w}^2+\overline{w'w'}) \qquad (2.57)$$

After each time step, the temperature is calculated at each grid point, and the water vapor mixing ratio, cloud droplet mixing ratio and cloud ice mixing ratio are

table 2.3
relative importance of terms in total energy expression

term	variable estimate	J kg^{-1}	% contribution
$c_{pd}\,\overline{\theta}$	$\overline{\theta} = 300$ K	301500	92.3
$c_w\,\overline{q}_{tot}\,\overline{\theta}$	$\overline{q}_{tot} = 0.010$ kg/kg	12654	3.9
$L_v\,\overline{q}_v$	$\overline{q}_v = 0.005$ kg/kg	12505	3.8
$c_w\,\overline{q'_{tot}\theta'}$	$q'_{tot} \approx 0.001$ kg/kg $\quad \theta' \approx 0.6$ K	2.5	$7.75(10^{-4})$
$0.5\,\overline{w}^2$	$\overline{w} = 5$ ms^{-1}	12.5	$3.83(10^{-3})$
$0.5\,\overline{w'w'}$	$w' \approx 3.5$ ms^{-1}	6.1	$1.88(10^{-3})$

QUATERNARY LIBRARY

adjusted for condensation or evaporation by solving Eq. 2.57 for the enthalpy, and inverting Eq. 2.53 using an iterative technique (discussed in Appendix C). At saturation, the vapor pressure, e_{sat}, at pressure p is approximated in a manner similar to Lord, et al., (1984), to include the effects of both liquid water and ice;

$$e_{sat}(T) = \frac{\lambda e_w(T) + \mu e_i(T)}{p - \lambda e_w(T) - \mu e_i(T)} \tag{2.58}$$

for vapor saturated over liquid, $e_w(T)$, and ice, $e_i(T)$, with

$$\lambda = \frac{q_l}{q_l + q_i} \tag{2.59}$$

$$\mu = \frac{q_i}{q_l + q_i} \quad .$$

2.5. Bulk water parameterization categories

Five bulk water categories are considered; vapor (q_v), cloud droplets (q_c), precipitatable water drops (q_p), cloud ice (q_s) and graupel (q_g). The water variables for the inner region are described by

$$\frac{\partial}{\partial t}\rho_o \overline{q}_{ka} = -\frac{\partial}{\partial z}(\rho_o \overline{w}_a \overline{q}_{ka}) + \widetilde{q_{ka}}\frac{\partial}{\partial z}(\rho_o \overline{w}_a) \tag{2.60}$$

$$-\frac{\partial}{\partial z}(\rho_o \overline{w_a'' \, q_{ka}''}) - \frac{2}{a}\rho_o(\widetilde{q_{ka}'' u_a''}) + \overline{S}_{ka}$$

for k = v, c, p, s, and g. S_k includes the microphysical transitions between the five water categories and the fallout terms for precipitation, snow and graupel.

2.5.1. Microphysics parameterization

The microphysical transitions include a Kessler (1969) form of autoconversion which includes a simple dependence upon the droplet concentration, the Cotton (1972) formulation of collection, and the Cheng (1981) version of the Stephens (1979) ice phase parameterization. Figure 2.3 shows the physical processes parameterized. A detailed discussion of the terms shown in figure 2.4 may be found in appendix D; however a synopsis of that appendix is presented below, while table 2.4 contains a list of

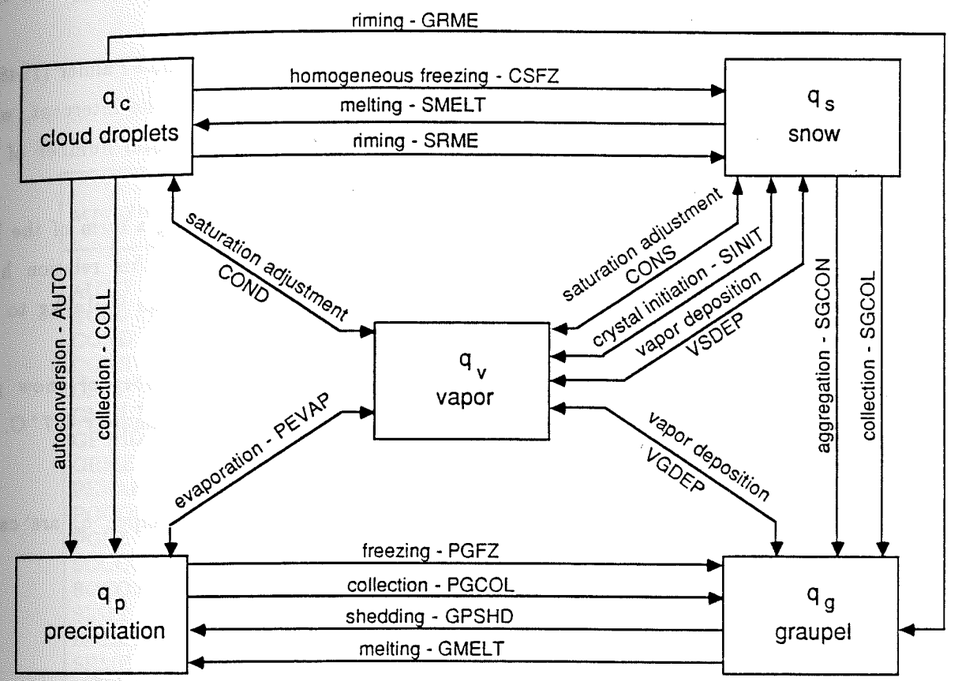


figure 2.4
Schematic of the microphysical parameterization used in the cloud model. See appendix D for a discussion of the terms.

the parameterized terms, and table 2.5 contains the bulk water category source terms.

The microphysics parameterization is an attempt to deal with both liquid and ice phases in a simple fashion. In order to keep the parameterization a reasonable size the Cheng (1981) version of Stephens (1979) ice phase microphysics is used rather that the more complicated South Dakota (e.g., Lin et al., 1983) microphysics scheme. The primary distinction between the parameterizations is the separation of the Stephens (1979) snow field into an ice field and snow field in the South Dakota parameterization.

The basic assumptions in the microphysical parameterization used are:

a). A monodisperse, time invariant cloud droplet population in which the total number of droplets is fixed.

b). Precipitation and graupel distributions follow a Marshall-Palmer (1948) distribution. The precipitation distribution is fixed by the intercept, while the graupel distribution is determined using a fixed total number of graupel particles.

c). The total number of snow particles is calculated using a form of the Fletcher (1969) relationship. As discussed in appendix D, this relation has been modified to permit snow concentrations greater than 0.01g/kg to exist in regions which have $T \geq -12\,^{\circ}C$.

d). Homogeneous nucleation occurs when the total number of snow particles exceeds 10^7. This occurs at a temperature of approximately $-37\,^{\circ}C$.

2.5.2. Fallout terms

The fallout terms for precipitation q_p, snow q_s, and graupel, q_g are calculated using

$$S_{fall,\,k} = -\frac{\partial}{\partial z}\rho_o v_k q_k \quad , \tag{2.61}$$

for k= p, s, or g. v_k is the mass weighted fall velocity calculated from equation D.5 for precipitation, D.9 for snow, and D.13 for graupel.

2.6. Subcloud forcing

The cloud model is initialized with a sustained velocity forcing at cloudbase. The initial values of cloudbase pressure p_{cb}, temperature T_{cb}, total cloud water q_{cb}, and the velocity at cloudbase w_{cb} are input into the model. Although, in general, actual values of these parameters exhibit some variation at cloudbase, in this model the cloudbase values of all parameters except velocity are held fixed during the forcing period.

In order to smooth the initial discontinuity at cloud base in the advected fields, a perturbation is added to the fields above cloud base at time t=0. The depth of this perturbation is adjustable, and set using the variable *radm* . The initial vertical velocity above cloud base is computed using

table 2.4
microphysical terms and acronyms used for the warm and cold cloud transitions

acronym	microphysical process	equation
AUTO	autoconversion of cloud droplets to precipitation	D.15
COLL	collection of cloud droplets by precipitation	D.17
COND	equilibrium adjustment to cloud droplets after advection	C.8
CONS	equilibrium adjustmen to snow after advection	C.9
CSFZ	homogeneous freezing of cloud droplets	D.27
GMELT	melting of graupel	D.44
GPSHD	precipitation shedding by graupel	D.37
GRME	riming of graupel	D.29
PEVAP	evaporation of precipitation	D.20
PGCOL	collection of precipitation by graupel	D.31
PGFZ	precipitation freezing via ice contact	D.28
SGCOL	collection of snow by graupel	D.32
SGCON	snow aggregation and excess riming to graupel	D.33
SINIT	snow crystal initiation	D.22
SMELT	melting of snow	D.43
SRME	riming of snow	D.24
VGDEP	vapor deposition on graupel	D.42
VSDEP	vapor deposition on snow	D.38

$$w_c(z,\ t{=}0) = w_{cb}\, e^{-\frac{z-z_{cb}}{b\ radm}^2} \quad , \tag{2.62}$$

where b is the outer cloud region radius, and z_{cb} is the cloud base height. The moisture is initially set to a fraction of the saturated value at the environmental temperature;

$$q_v(z,\ t{=}0) = \left[\frac{0.622\ e_w(\ \overline{T}(z)\ )}{p(z) - e_w(\ \overline{T}(z)\ )} \right] e^{-\frac{z-z_{cb}}{b\ radm}^2} \quad . \tag{2.63}$$

The energy is recomputed at each level using the perturbed values of velocity and moisture using equation 2.56. All model runs discussed in this dissertation use a value of radm between 0.1 and 0.3, which results in perturbed fields for about 2 to 3 levels above cloud base.

The subcloud velocity forcing is accomplished by modifying the advection equations for momentum (2.30), energy (2.54), and water vapor (2.60) in both the inner and outer cloud regions to yield a sustained source of energy and moisture through cloud-base for a time *ptime* in a manner similar to Ferrier (1987). During the perturbation time, the advection equations for the levels below cloudbase are set to

$$\frac{\partial}{\partial t}\rho_o\xi = 0 \quad , \tag{2.64}$$

by using $\overline{w}_a \frac{\partial}{\partial z}\rho_o\xi$ for the source term, and setting the dynamic entrainment and eddy fluxes to 0. The initial velocity at height z (below cloudbase) is set to $w_{cb}\ \frac{z}{z_{cb}}$.

table 2.5
Source terms for the bulk water categories

	source terms
q_v	PEVAP-SINIT-VSDEP-VGDEP-COND-CONS
q_c	SMELT-AUTO-COLL-SRME-CSFZ-GRME+COND
q_p	AUTO+COLL-PEVAP-PGFZ-PGCOL+GMELT+GPSHD
q_s	SINIT+VSDEP+SRME+CSFZ-SGCOL-SGCON-SMELT+CONS
q_g	PGFZ+VGDEP+GRME+PGCOL+SGCOL+SGCON-GMELT-GPSHD

For the duration of the forcing (ptime), the sub-cloud moisture is set to the cloud base value, and the total energy is computed at each sub-cloud level using equation 2.56. At the end of the forcing period the sub-cloud moisture and energy are set to the environmental values. The response of the model to this formulation of the subcloud forcing will be examined in chapter 4.

2.7. Radar reflectivity equations

Radar reflectivities from the model output precipitation and graupel fields are computed using equations from Smith, et al. (1975). The relations are:

$$Z = 3.631(10^9)\ (\ \rho_o q_p\ )^{1.75} \quad [\ mm^6 m^{-3}\ ] \tag{2.65}$$

for precipitation, and

$$Z = \frac{1.3861(10^{19})}{N_{TG}} \left[\frac{\rho_o q_g}{\rho_g} \right]^2 \quad [\ mm^6 m^{-3}\ ] \tag{2.66}$$

for graupel. Equation 2.65 is derived assuming a Marshall-Palmer distribution with $N_{or} = 8(10^6)\ m^{-4}$. Equation 2.66 also is derived assuming graupel follows a Marshall-Palmer distribution, but with the total number of graupel fixed (Stephens, (1979)), e.g., $N_{tg} = 100\ m^{-3}$. ρ_g is the graupel density. If $T \geq 273.16$, wet hail is assumed. The reflectivity is then given by

$$Z = 7.411(10^{18}) \left[\frac{1}{N_{TG}} \left(\frac{\rho_o q_g}{\rho_g} \right)^2 \right]^{0.95} .[\ mm^6 m^{-3}\ ] \tag{2.67}$$

2.8. Summary

The cloud model for this sensitivity study is an axisymmetric Eulerian model based on Asai and Kasahara (1967), but which contains two cloud regions coupled to the environment via entrainment. Vertical and lateral eddy fluxes are parameterized using the Lilly (1962) buoyancy modified Smagorinsky (1963) formulation. The dynamic entrainment terms are calculated using a turbulent kinetic energy entrainment formulation developed for this model. The flux corrected transport algorithm, *ETBFCT*, developed by Boris and Book (1976), is used to advect momentum, total

QUATERNARY LIBRARY

energy, and the five water variables, in each cloud domain region. A simple pressure perturbation estimate using a single mode of a Fourier-Bessel series for each region of the cloud model is incorporated into the momentum equation. The interactions involving water vapor, liquid and ice are approximated by bulk water parameterizations of warm and cold rain processes. The cloud model is initiated using a sustained velocity at cloudbase.

CHAPTER 3

COMPARISON WITH OBSERVATIONS

In this chapter I will compare cloud model runs with observations of clouds made on July 19, 1981 and August 6, 1981 during the Cooperative Convective Precipitation Experiment (CCOPE) in south-central Montana. The time evolution of the model dynamic and microphysical fields are examined, and compared with aircraft and radar measurements. In addition, the base model run of July 19 will be compared with results from a published two-dimensional model simulation. Sensitivities of the cloud model to variations in the adjustable parameters will also be discussed in the next chapter.

3.1. CCOPE July 19

3.1.1. Observation summary

The cloud studied from 1616 - 1700 MDT on July 19, 1981 was a small, isolated cumulonimbus. As the temporal development of the dynamic structure and microphysics has been extensively discussed in the literature; e.g. Dye, et al., 1986 or Helsdon and Farley, 1987, the primary points will be summarized here. Figure 3.1 is a representative sounding for the day, which was characterized by weak shear (1.3 (10^{-3}) s^{-1} from 800 to 400 mb) and a potential instability of 1 – 2 °C as shown by the cloudbase θ_e. The cloudtop was observed to be approximately 6.5 km MSL (mean sea level) by the UND Citation from the time of the study initialization until about 1621 MDT, when a turret on the northeast side of the cloud began to rapidly grow. This turret reached 10.5 km MSL at around 1630 MDT. By 1650 MDT, the cloud had decayed to an anvil with falling precipitation. During the cloud study, the University of Wyoming King Air (H2) made 10 passes at approximately 6 km MSL, and the NCAR/NOAA sailplane (H9) made a spiral ascent in the cloud updraft from cloudbase to about 7 km MSL. The cloud was also observed with the NCAR CP-2 radar (10 cm) and the University of Chicago/Illinois State Water Survey CHILL radar (10 cm). Table 3.1 lists the primary

QUATERNARY LIBRARY

observations used in this comparison.

3.1.2. Base Model Run

The base model run was initialized with the sounding shown in figure 3.1 and the parameters listed in table 3.2. Since table 3.2 contains the variable parameters for the

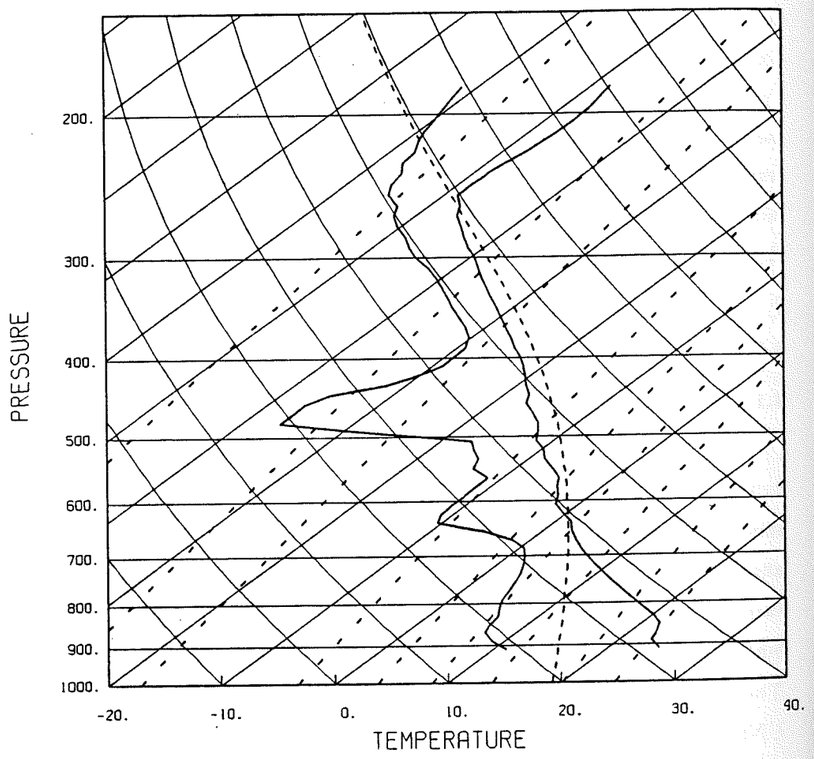


figure 3.1
Miles City 1440 MDT sounding on July 19, 1981. The light dashed line represents $\theta_e = 332.55$ K; corresonding to cloudbase at 635 mb, 0.95 °C, and 6.5 g/kg. Cloudbase is from Dye, et al., 1986.

table 3.1
Primary observations of July 19, 1981 cloud

cloud base	height	3.9 km MSL
	temperature	0.8–1.1 °C
	q_{total}	6.5 g/kg
	pressure	634-636 mb
	vertical velocity	1–5 ms^{-1}
cloudtop	$\approx$ 6.5 km MSL	1616 MDT 1st observation
	$\approx$ 10.5 km MSL	1630 MDT
radar	1st 5 dBZ return 1623 MDT	6.5-7.5 km MSL
H9 updraft observations	vertical velocity	6-7 ms^{-1} up to 5.5 km MSL
		10-11 ms^{-1} to 7 km MSL
	themodynamics	nearly adiabatic

cloud model, the names given in the table are those used in the model listing and are defined in the list of symbols. The choice of inner and outer radii (1000 and 2500 m, respectively) for the model simulation were made on the basis of aircraft penetrations and the CP-2 RHI radar scans. The model domain was dynamically forced with a velocity of 3.0 ms^{-1} at cloudbase in both the inner and outer regions. The cloudbase velocity in the inner region was increased to 6 ms^{-1} at 1500 seconds in order to reduce the entrainment near cloudbase and therefore create a more nearly adiabatic core. This velocity variation is consistent with that observed near cloudbase. Forcing for the entire cloud was terminated at 2700 seconds, at which time the subcloud region was returned to the observed temperature and moisture profiles. Sensitivities of the model output to variations in the subcloud forcing are discussed in chapter 4. The cloud droplet concentration was chosen as representative of the aircraft penetrations, and yields an autoconversion threshold of 5.75 g/kg according to the relation described in

QUATERNARY LIBRARY

appendix D. The cloudbase values of pressure, temperature and moisture are those reported by Dye, et al., (1986) as measured by the sailplane (H9). The model run was terminated at 5500 seconds.

Figures 3.2 through 3.6 show the temporal development of the dynamic and microphysical fields for both the inner and outer regions. The model domain begins at ground level ($\approx$ 800 m MSL). The vertical velocity (figures 3.2, 3.5) and the total liquid water and ice (figures 3.3, 3.6) contour plots show a cloud which grows to approximately 6.5 km (7.3 km MSL), retains a nearly constant cloudtop height for about 1000 seconds, and then undergoes rapid growth (starting at about 2400 seconds) to final cloudtop height of approximately 10.3 km (11.1 km MSL). The peak vertical velocity is about 12 ms^{-1}. Additional plots showing the separate hydrometeor fields, entrainment velocity and turbulent kinetic energy may be found in Appendix E.

table 3.2
July 19 model base run parameters

model domain	0-12000 m
dz	200 m
arad	1000 m
brad	2500 m
ptime	2700 s
utime	1500 s
w_{cb}	3.0 ms^{-1}
wmult	2.0
avl	800 m
N_{TG}	100 m^{-3}
β	0.50
N_{cc}	800 cm^{-3}
p_{cb}	635 mb
T_{cb}	0.95 ° C
q_{cb}	6.5 g/kg
radm	0.1
O_{NS}	0.01 m^{-3}

3.1.3. Precipitation Formation and Development

The total water plots (figures 3.3 and 3.6) and the inner region radar contour plot (figure 3.4) show a moderate intensity (45-55 dBz, $>$ 3 g/kg total liquid and ice), but brief ($\approx$ 10-15 minutes) shower originating from this cloud. The total precipitation at the lowest level in the model domain was 4.9 mm rain and .008 mm graupel. The precipitation water (figures E.1.4 and E.1.10 in Apendix E) and graupel (figures E.1.6 and E.1.12) indicate that precipitation formation was through the ice phase. This is consistent with the observations of Dye, et al. (1986) of no raindrops or drizzle drops within the cloud. Precipitation formation first occurs around 6 km in the model domain (6.8 km MSL), again in good agreement with the CP-2 observations of the first 5 dBz return between 6.5 and 7.5 km (MSL). No measurement was made of the actual time of formation of the observed cloud. However, a relative comparison of the model and observation is possible. Figure 3.7 shows the maximum reflectivity contours reported by Dye et al., (1986) plotted against time corresponding to the model times shown in figure 3.4. The times shown in figure 3.7 were established by setting the time of the first observed 5 dBz return to that predicted by the model. The overall agreement is generally good, with the model predicting a higher 55 dBz contour, and a shower of slightly shorter duration. Two 45 dBz contours are seen in the model domain. The lower contour is in good agreement with that reported by Dye, et al. (1986). Examination of the microphysical fields indicates that the upper contour is a result of the graupel density changing from 600 kg m^{-3} to 100 kg m^{-3} due to q_g falling below the threshold discussed in appendix D.

The rapid growth of the study cloud observed to begin at about 1621 MDT is nearly coincident with the first 5 dbZ returns from the cloud at around 1623 MDT (Dye, et al., 1986), and can be understood in terms of a combination of precipitation unloading of the updraft and glaciation. Figure 3.8 is a plot of the virtual temperature along a moist abiabat, (constant θ_q trajectory), i.e., $T_v = T(1+0.608q_v-q_l)$, and along a moist pseudoadiabat (constant θ_e); $T_v = T(1+0.608q_v)$, where q_v is the saturation vapor mixing ratio. The curve plotted for constant θ_q corresponds to that of a parcel undergoing adiabatic expansion with no fallout of precipitation. A constant θ_e more closely approximates the virtual temperature of a parcel in which all the condensed water has fallen out. Any entrainment will result in a cooler parcel, while freezing will

increase the buoyancy and move the thermodynamic path to the right in figure 3.8.

The observed high droplet concentration (600-800 cm^{-3}) suggests that adiabatic ascent in this cloud may be approximated initially by constant θ_q as there is no precipitation formation via drop coalescence. The subsequent formation of precipitation via ice phase processes yields both an increase in temperature (over moist adiabatic ascent) and a reduction of the water loading through fallout to produce a virtual temperature which approximately follows that shown in figure 3.8 for θ_e. (The actual path is slightly different due to the additional latent heat release; $L_s/L_v \approx 1.12$).

Both the observed general cloudtop in the field of clouds around the study cloud on July 19 and the cloudtop of the study cloud before the rapid growth phase are consistent with the maximum height obtainable (around 400 mb) by a nearly adiabatic wet parcel which started with the observed cloudbase properties. As can be seen in figure 3.5, this is also the cloudtop of the base run model cloud before the rapid growth phase. The model cloud droplet, precipitation water, snow and graupel fields (figures E.1.3-E.1.12 in appendix E) show that the rapid growth, which begins at approximately 2600 seconds, is coincident with the initial graupel formation (via snow aggregation and riming in the microphysical parameterization).

3.1.4. Entrainment

Several aircraft were used in a combined study of this cloud on July 19. The University of Wyoming King Air (H2) made 10 penetrations of the cloud between 5.3 and 6 km MSL (507 to 481 mb; 6 to 7 km in the model domain). The first 4 passes were through the main core of the cloud and at a constant height (6 km MSL; 481 mb). Figure 3.9 is a Paluch plot (Paluch, 1979) of the 1 Hz values from these 4 H2 penetrations (made over a 10 minute period; 1616 to 1626 MDT) using the Johnson-Williams probe liquid water which was corrected according to the procedure described by Jensen (1985). To reduce the interpretation problems of unsaturated regions in-cloud and at the boundaries, only those points in which the 50 Hz FSSP nstrobe measurement showed drops for the entire second (the 50 Hz nstrobe measurement) and the 10 Hz JW liquid water was positive for the entire second are plotted. The observed in-cloud points exhibit no definite tendency towards mixing from a single location in the environment and cloudbase, but rather seem to lie in two groups; a nearly adiabatic

group centered at about 6.2 g/kg and 329 K, and the remaining points which have undergone various degrees of mixing.

The dashed line in figure 3.9 represents the mixing line of cloudbase air and environmental air at 421 mb. This upper limit on the observed mixing height is consistent with the maximum penetration depth of an evaporatively cooled mixture of cloudbase air and environmental air from 421 mb as may be seen in figure 3.10, which shows a saturation point mixing diagram (Betts, 1982) of air from cloudbase and the environment at 421 mb. Any mixtures of cloudbase air with environmental air from above 421 mb would be positively buoyant at 481 mb (the penetration level), and so are unlikely to penetrate to that level in the cloud. The mixtures of air from 421 mb and cloudbase air with $f \approx .35$ (f is the fraction of cloudbase air, (1-f) the fraction of environmental air) are neutrally buoyant at 481 mb.

The model output may be compared to the H2 observations keeping in mind that the model output corresponds to a time series because of the limited horizontal resolution. In addition, the limited sample volume (in space and time) of the aircraft measurements makes any direct correlation with model computations difficult. However, in this case H2 made repeated penetrations at the same altitude, which should provide some indication of the temporal evolution of convective elements at that height. From an analysis of radar and the MAPS tracks, Dye et al., (1986) reported that H2 flew through the main storm updraft during the penetrations used for comparison here. This lends support to the use of both model regions for comparison with the observed points.

Figures 3.11, 3.12 and 3.13 are a sequence of Paluch-plots showing the observed points from figure 3.9 and model points at 60 second intervals from the inner and outer regions which are saturated and have a total ice concentration $< .01$ g/kg. The time development of the air at 481 mb in these regions is shown in figure 3.14.

As seen in figures 3.11 and 3.14, the inner region first shows air which has characteristics from the levels below the penetration level and becomes increasingly closer to cloud base air, with most of the points clustering in the nearly adiabatic region at $\theta_q \approx 329$, $q_{tot} \approx 6.2$ g/kg. The outer region points, on the other hand, show evidence of mixing from about 540 mb to 420 mb (figures 3.12 and 3.14). This mixing from the

QUATERNARY LIBRARY

region above the penetration level (481 mb) is also seen in the vertical velocity of the outer region (figure 3.5), which shows a downdraft of ≤ 3 ms^{-1} between about 6000 m and 4800 m during the mature phase before the onset of rapid growth (from about 1300 to 2600 seconds).

Detailed thermodynamic comparisons between model output and aircraft observations after about 2800 seconds are not possible because, while H2 made 6 more penetrations, the limited horizontal resolution of the model yields too rapid of glaciation of the cloud as all graupel formed in the upper levels of the cloud falls through the lower levels. The study cloud did grow in weak shear, but this was sufficent to permit some graupel fallout from the upper levels into clear air rather than into the lower cloud regions (see Helsdon and Farley, (1987), fig.3).

3.1.5. Two dimensional model comparison

The same cloud was simulated using the South Dakota 2D slab-symmetric model by Helsdon and Farley, (1987). Initial pressure, temperature and moisture for the two dimensional domain was the Miles City sounding shown in figure 3.1. Convection on the 19.2 $\times$ 19.2 km grid was initiated using a random temperature and moisture perturbations in the lowest three kilometers coupled with a 4.8 km diameter warm 'bubble' (Δ T $\leq$1.5 °C) between 400 and 2000 m. The grid resolution was 200 m. Table 3.3 lists model output from the 2D model, the model discussed in this section, and some observations noted above.

The cloud base and cloud top reported for the 2D model run differ significantly from that reported by Dye, et al., (1986) and computed using the model presented here. Although there is some uncertainty in the observed cloudtop (estimated as $\pm$ 300 m ; Dye, personal communication), the cloudbase properties were relatively well determined. From the discussion of the 2D model results, it appears no attempt was made to modify the subcloud region in order to reflect the observed cloudbase properties for their simulation. The microphysical and dynamical characteristics for both model runs are in general agreement with the observed properties; however the maximum vertical velocity in the 2D simulation is much higher than that observed.

It would appear that the major drawback of the model presented here, relative to the 2D model, is the inability to account for updraft unloading in a sheared

table 3.3
Comparison with observations and 2D model

	Observed	2D	1.5D	
Cloud base height	3.9	2.8	3.9	km MSL
Max. cloud top	10.5	11.6	11.0	km MSL
Liquid cloud height	6.5	$\approx$ 6	$\approx$ 7	km MSL
First graupel	5.8	5.6	6.5	km MSL
max. vertical velocity	10-15	26	12	ms^{-1}
Cloud base updraft	1-5	1-7	3-6	ms^{-1}
First radar echo	7	7	6.8	km

environment. Even in this weak shear case, sufficent unloading occured to permit a liquid water updraft to coexist with falling graupel for several minutes in the 2D model; e.g., see figure 3 in Helsdon and Farley.

QUATERNARY LIBRARY

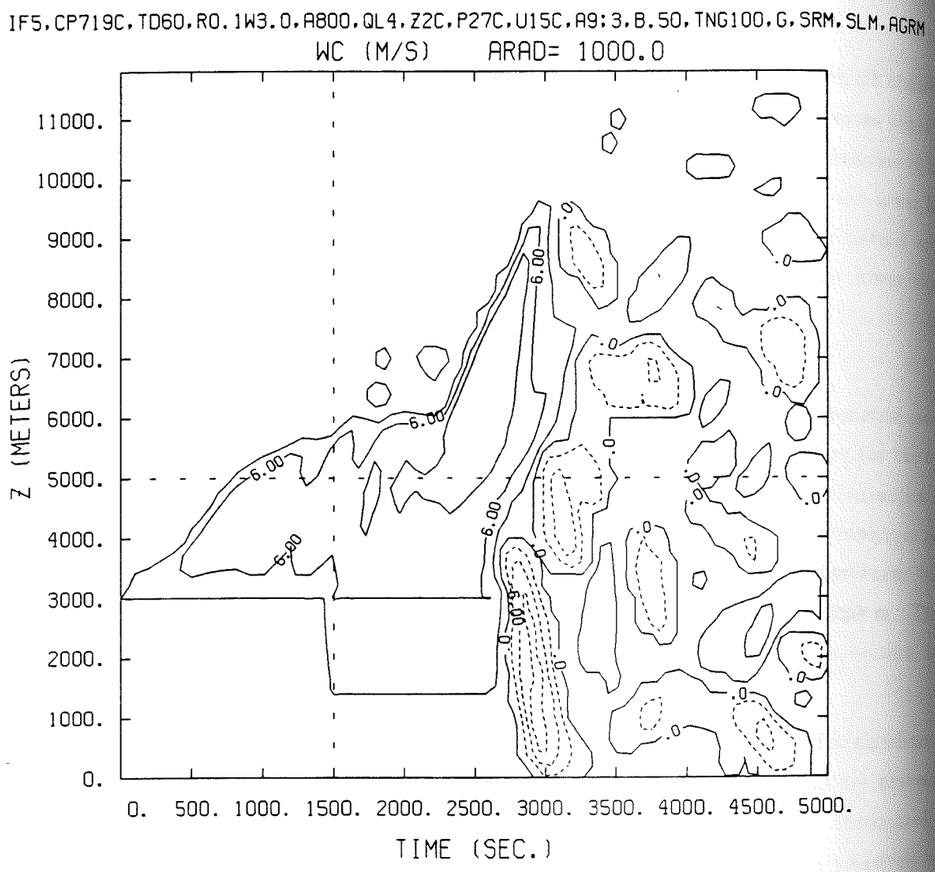


Figure 3.2 Contour plot of vertical velocity in the inner cloud region for the July 19 base run. Contour interval is 3 ms^{-1}; negative velocities are shown as dotted contours.

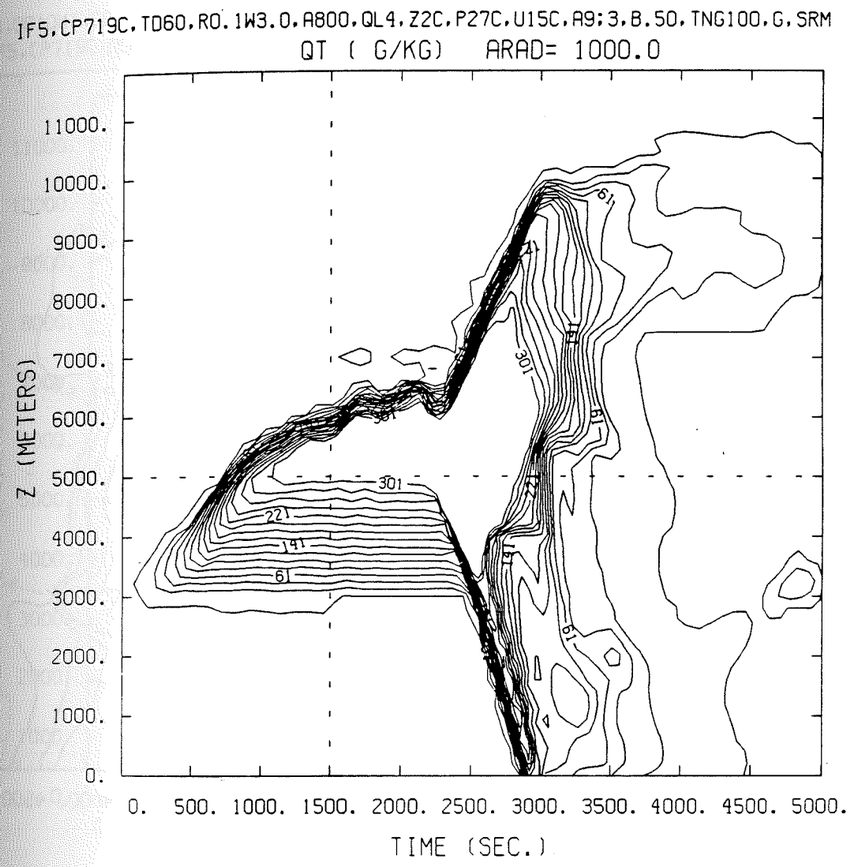


Figure 3.3 Contour plot of the total liquid water and ice mixing ratio in the inner cloud region. Contours are from .01 to 3.01 g/kg; the contour interval is 0.2 g/kg. Note that the contour labels are scaled by 100.

QUATERNARY LIBRARY

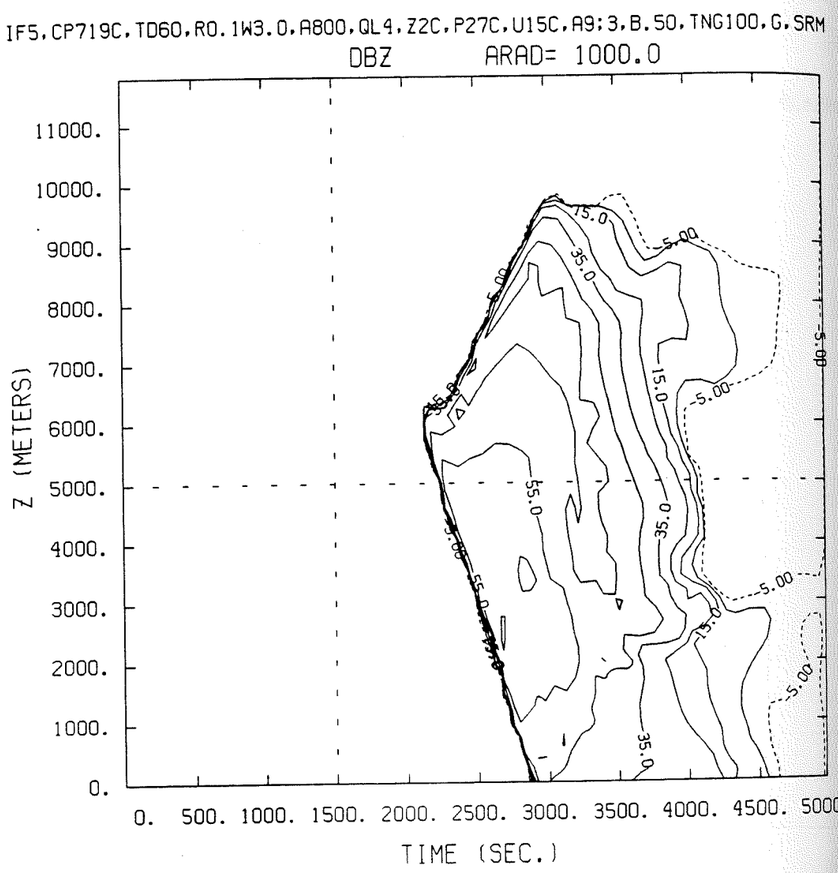


Figure 3.4 Contour plot of the radar reflectivity (computed from equations 2.66-2.68) for the inner region. Contours are from -5 to 65 dBZ; the contour interval is 10 dBZ.

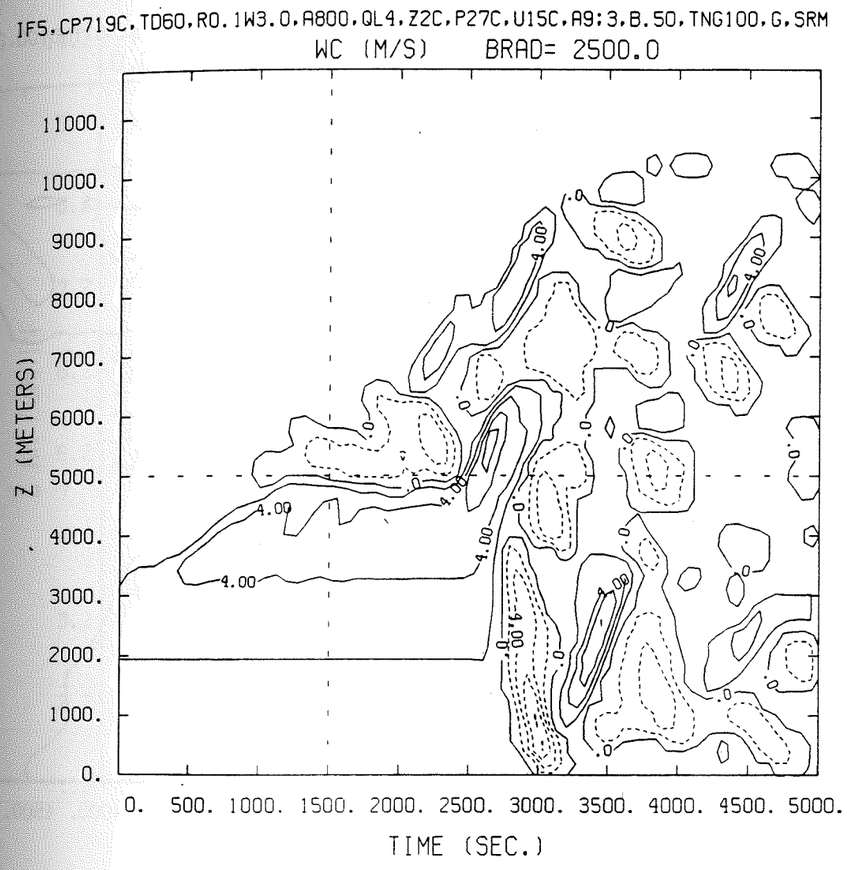


Figure 3.5 Contour plot of vertical velocity in the outer cloud region for the July 19 base run. Contour interval is 2 ms^{-1}; negative velocities are shown as dotted contours.

QUATERNARY LIBRARY

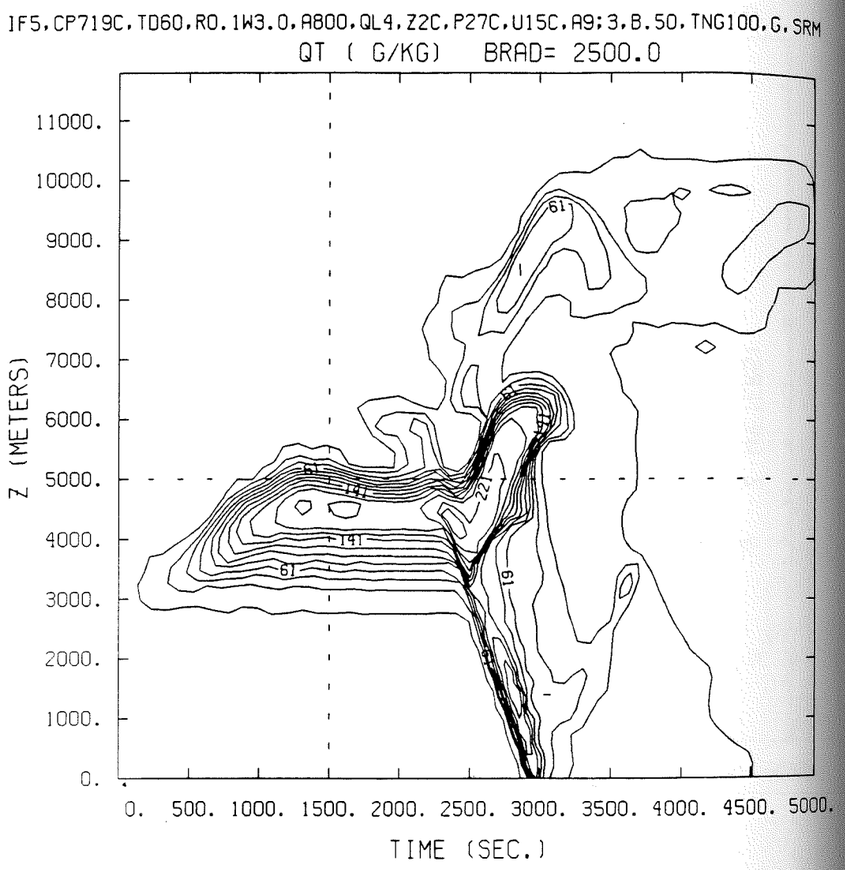


Figure 3.6 Contour plot of the total liquid water and ice mixing ratio in the outer cloud region. Contours are from .01 to 3.01 g/kg; the contour interval is 0.2 g/kg. Note that the contour labels are scaled by 100.

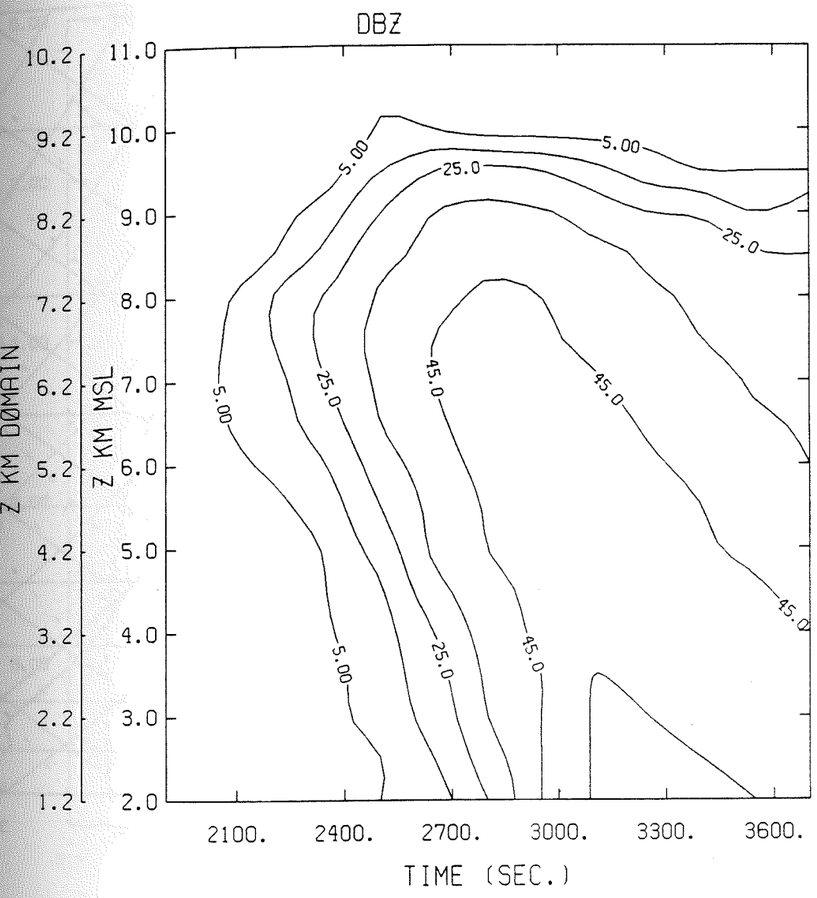


Figure 3.7 Contour plot of the observed radar reflectivity from Dye et al., 1986. The radar reflectivities presented in that paper have been plotted on the July 19 base run time scale. The height is plotted with respect to mean sea level (MSL) and the model domain. Reflectivities are plotted from 5 to 55 dBZ; the contour interval is 10 dBZ.

QUATERNARY LIBRARY

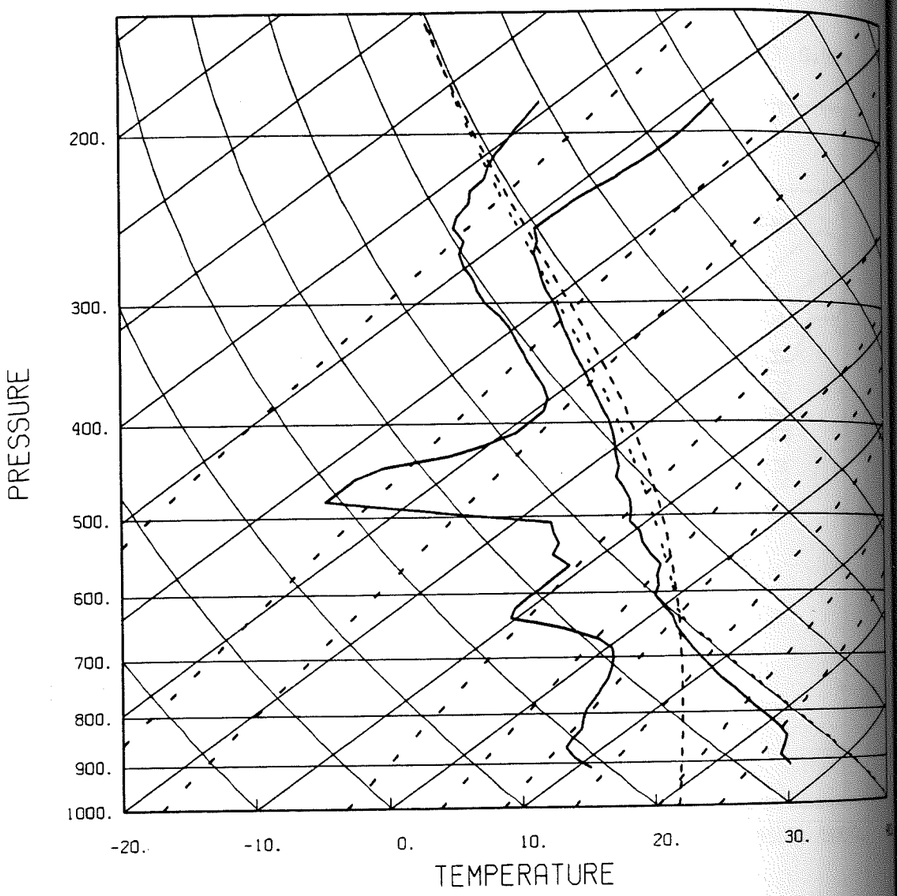


Figure 3.8 Virtual temperature plot of the July 19 sounding plotted in figure 3.1. The virtual temperature ($T_v = T\,(1 + 0.608 q_v - q_l)$) and dew point temperature are plotted. The light dotted line is the virtual temperature for constant θ_q, the dashed line is the virtual temperature for constant θ_e.

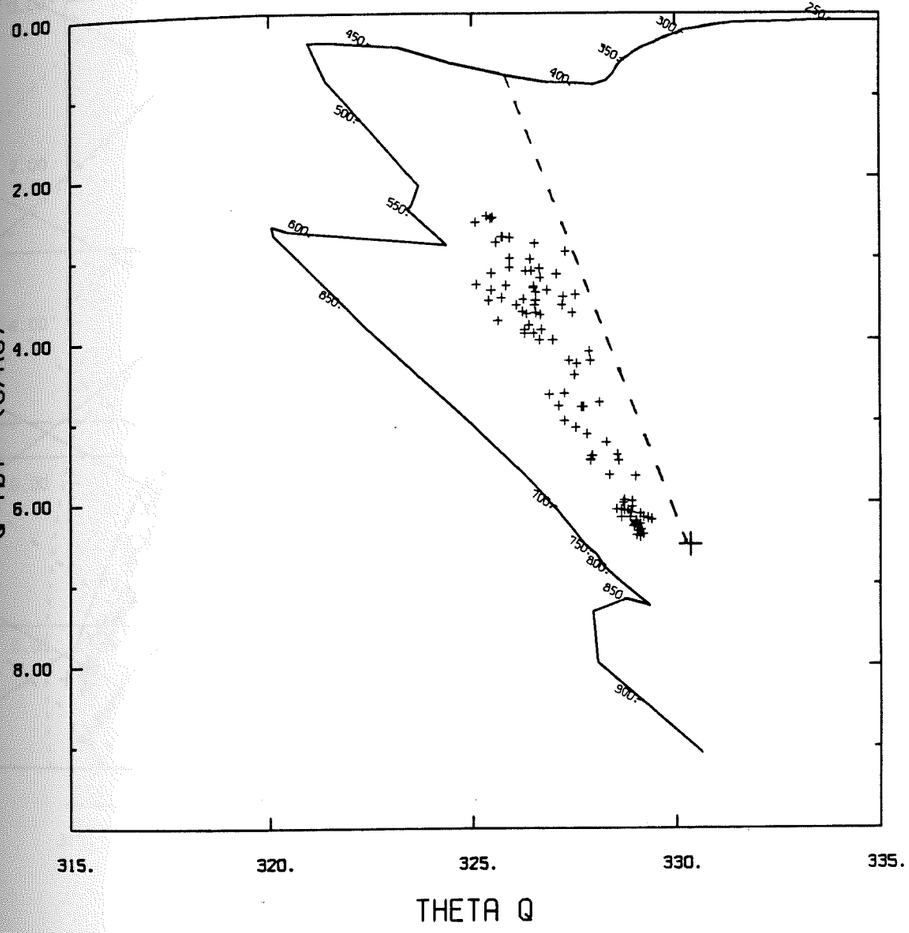


Figure 3.9 Composite Paluch plot of four H2 penetrations of the July 19 cloud. Each + represents one second of data. Cloud base is indicated by the large +. The sounding from figure 3.1 is also plotted with pressures (in mb) noted.

QUATERNARY LIBRARY

L'18/9

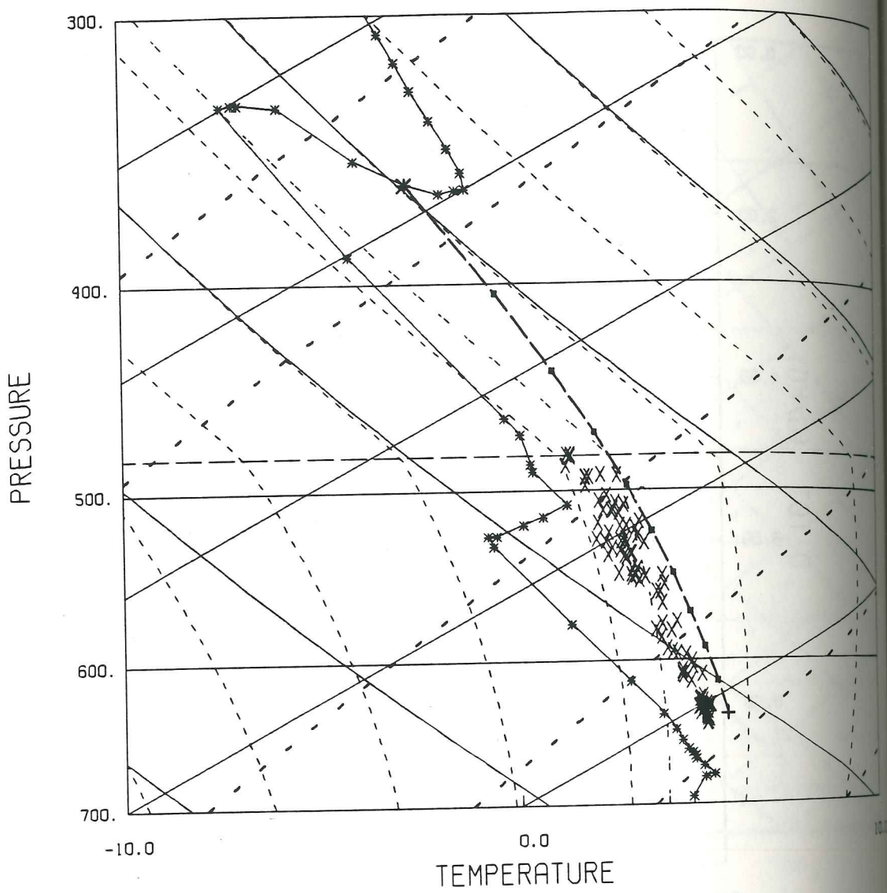


Figure 3.10 Saturation point mixing diagram for the penetration level at 481 mb. The light dashed lines which kink at the penetration level are density isopleths. The sounding (in saturation point coordinates) is given by the solid line. The heavy dashed line shows all possible mixtures of air from cloud base (+ at 635 mb) and the environment at 421 mb. The observed points from figure 3.9 are plotted (x).

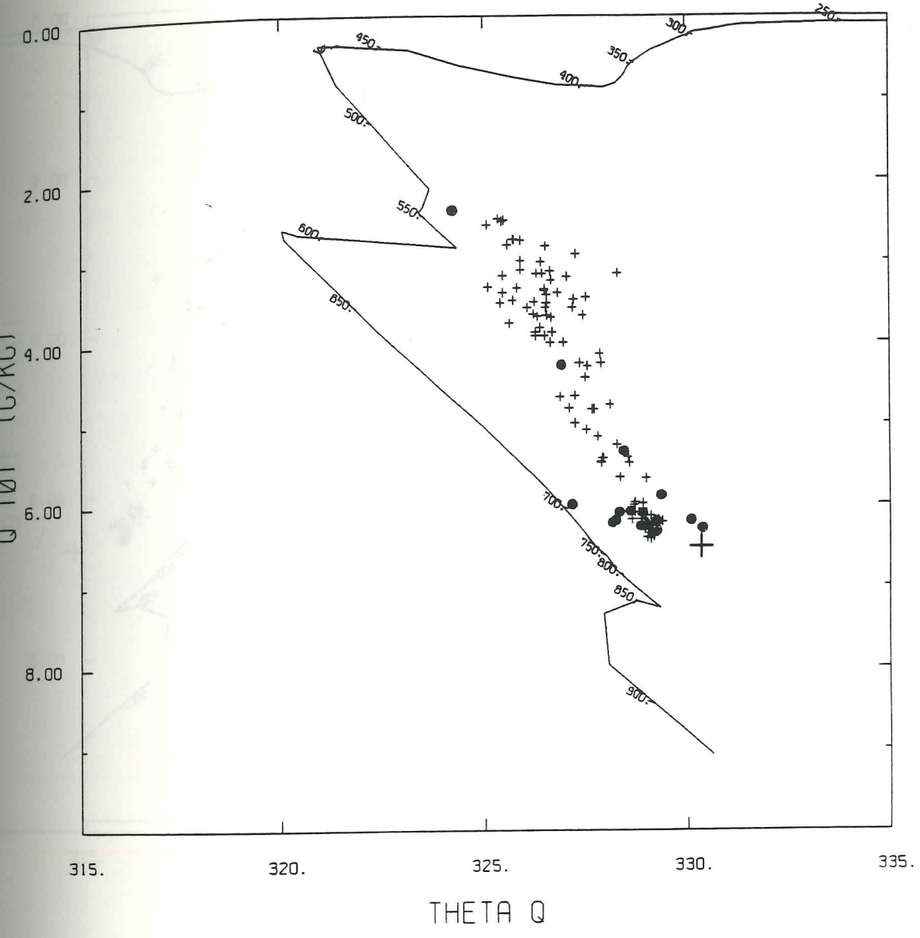


Figure 3.11 Paluch plot of model computed points for the inner region (dots) at 90 second intervals. Other points as in figure 3.9.

QUATERNARY LIBRARY

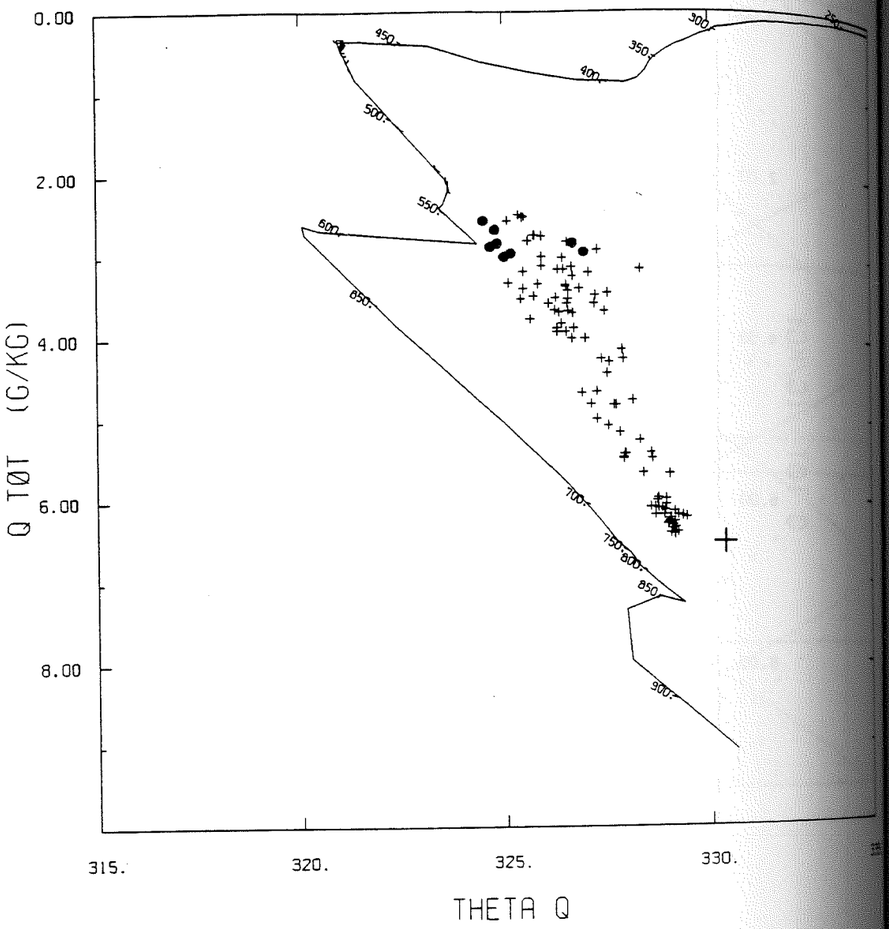


Figure 3.12 Paluch plot of model computed points for the outer region (dots) at 90 second intervals. Other points as in figure 3.9.

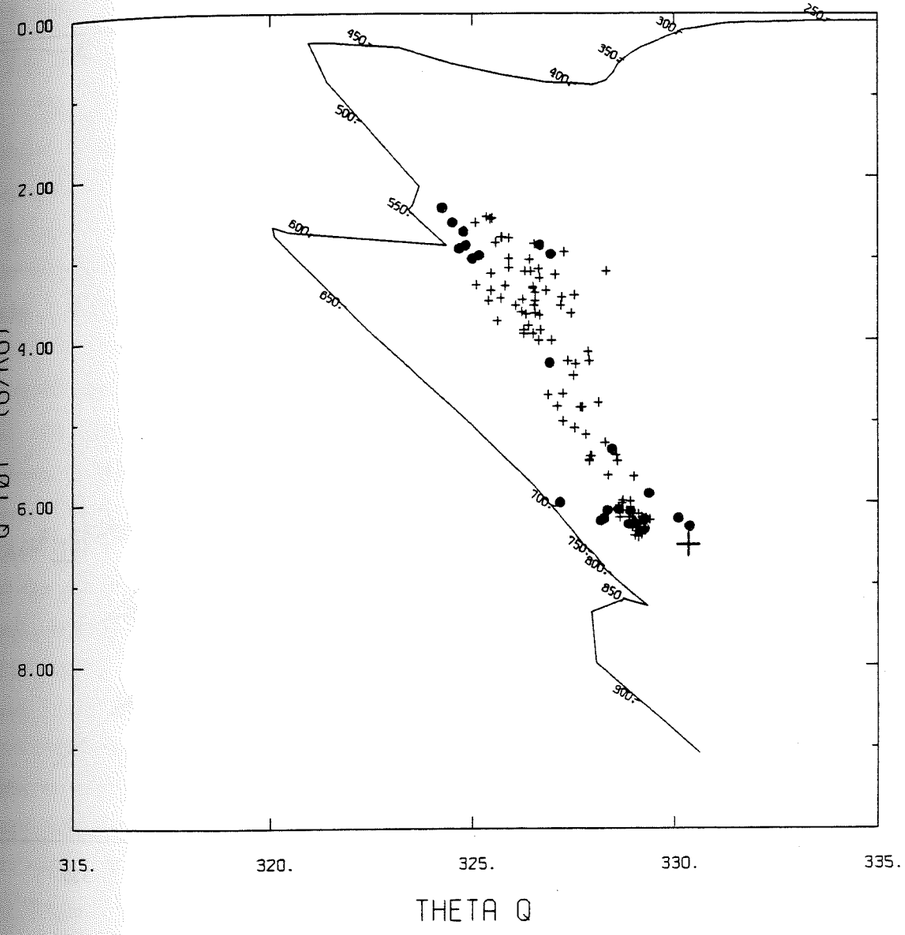


Figure 3.13 Composite Paluch plot of model computed points for the both regions (dots) at 90 second intervals. Other points as in figure 3.9.

QUATERNARY LIBRARY

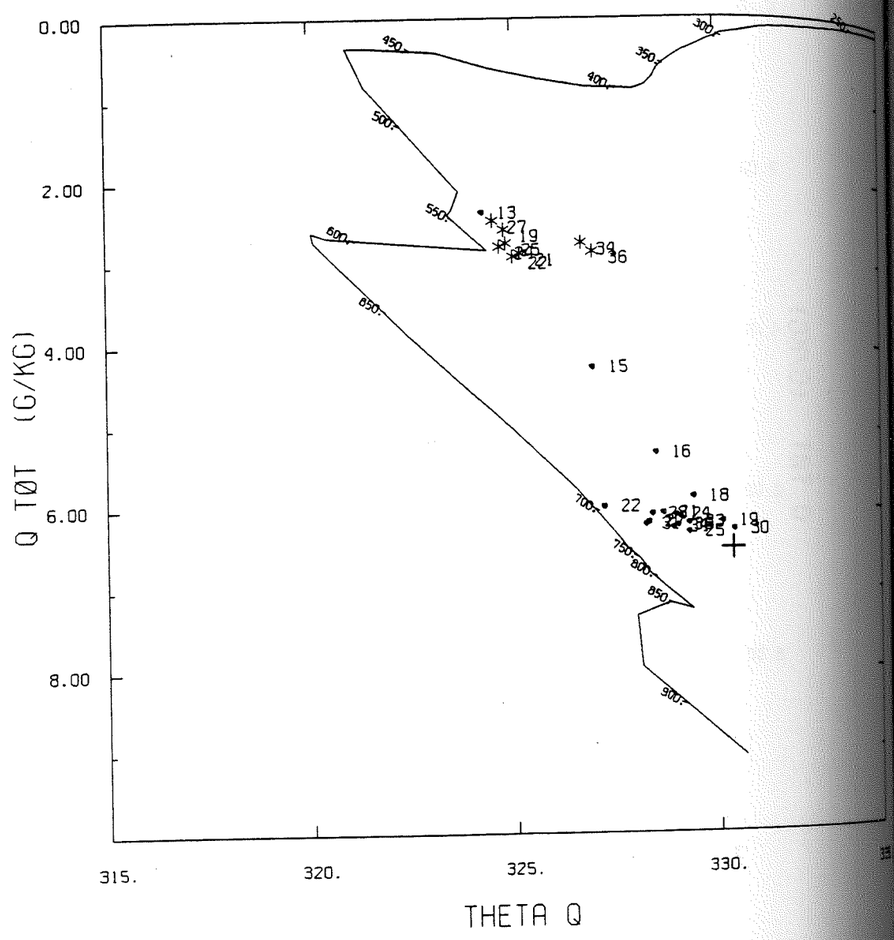


Figure 3.14 Composite Paluch plot of model computed points for the both regions showing the time development at the penetration level. The points are shown with the model simulation time (in minutes). The inner region points are shown as dots, the outer region points by *.

3.2. CCOPE August 6

3.2.1. Observation summary

The cloud studied from 1633 - 1659 MDT on August 6, 1981 was a cumulus congestus which developed at the end of a small line of turrets. Figure 3.15 is a representative sounding for the day, which was characterized by weak shear ($\leq 1\,(10^{-3})$ from 800 to 400 mb) and a potential instability of $1 - 2\ ^{\circ}$C as shown by the cloudbase θ_e. The daily operation log forecast cloudtops at about 6.1 km MSL (485 mb). Cloud measurements were made by the University of Wyoming King Air (H2) flying between 504 mb and 512 mb, while the NCAR Queen Air (H6) flew repeated legs beneath cloudbase. The cloud was also observed with the NCAR CP-2 radar (10 cm) .

The time series plots for the first three H2 penetrations at 509 - 512 mb are shown in figures 3.16 to 3.18. The liquid water presented is derived from the J-W measurement following the correction procedure outlined in Jensen (1985). All penetrations show a cloud colder than the environment. The vertical velocity indicates much of the cloud was descending (w $\approx$ -3 to -8 ms^{-1}), but at least two segments (15 to 22 seconds in figure 3.16 and 50 to 60 seconds in figure 3.18) show updrafts of $\leq +7$ ms^{-1}. The droplet concentration is around 400 to 500 cm^{-3} and shows sharp cloud boundaries and nearly uniform character across much of the cloud interior. This pattern is also reflected in the liquid water measurements. The moderately high turbulence levels suggest no adiabatic core exists at the penetration level. Ice concentrations were highly variable across both penetrations and ranged from 0.0 to about 30 l^{-1}. It is interesting to note that the first and second penetrations presented show high ice concentrations in the main cloud body near the interface with a relatively strong downdraft.

H6 flew seven legs beneath cloudbase over a 25 minute period and recorded peak vertical velocities of about 7 ms^{-1} (Fankhauser, et al. 1983). Precipitation was reported by leg 3 (1643 MDT) and became heavy by leg 5 (1651 MDT).

3.2.2. Base Model Run

The base model run was initialized with the sounding shown in figure 3.15 and the parameters listed in table 3.4. The inner and outer radii used in the model simulation were chosen as representative dimensions from the H2 penetrations and the subcloud legs by H6. The subcloud velocity of 3.5 ms^{-1} was near the average value

QUATERNARY LIBRARY

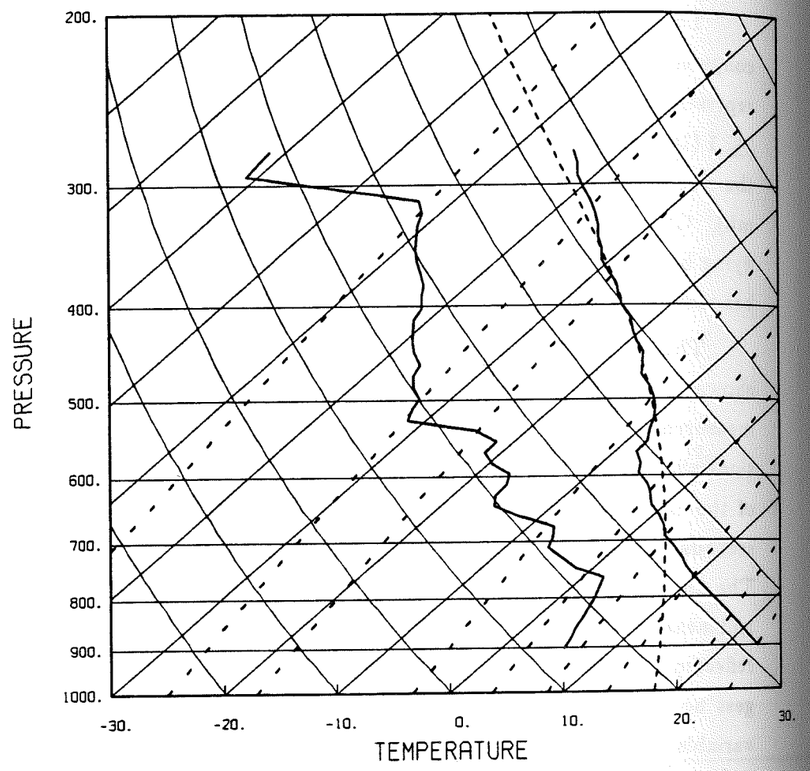


figure 3.15
Miles City 1610 MDT sounding on August 6, 1981. The light dashed line represents $\theta_e = 328.08$ K; corresponding to cloudbase of 708 mb, 4.00 °C, 7.26 g/kg from Jensen, 1985.

reported by H6. At 500 seconds the inner region velocity was increased to 6.3 ms^{-1}. The subcloud forcing for both regions was maintained for 3200 seconds, at which time the subcloud temperature and moisture were returned to the observed environmental values. The cloud droplet concentration was set at 500 cm^{-3}, which yields an autoconversion cutoff of 3.5 g/kg from the relation described in appendix D. Although the

perturbation depth scaling parameter above cloudbase (radm) was set higher than for the July 19 simulation, the actual perturbation depth is nearly identical as a result of the smaller outer radius for this case (See section 2.6). The model run was terminated at 4500 seconds.

Figures 3.19 through 3.22 show the temporal development of the dynamic and microphysical fields for both the inner and outer regions (1000 and 1500 m, respectively). The model domain begins at 900 mb ($\approx$ 1080 m MSL).

The vertical velocity (figures 3.19 and 3.21) and total cloud water (figures 3.20 and 3.22) show a cloud which grows to about 5000 m (6100 m MSL, 485 mb) by 2700 seconds, with vertical velocities at the penetration level from -4 to +3 ms^{-1}. Snow (figures E.1.16 and E.1.22 in appendix E) is found at concentrations greater than 0.01g/kg by 2800 seconds, with subsequent rapid cloud glaciation and precipitation

table 3.4
August 6 model base run parameters

model domain	0-9000 m
dz	200 m
arad	1000 m
brad	1500 m
ptime	3200 s
utime	500 s
w_{cb}	3.5 ms^{-1}
wmult	1.8
avl	250 m
N_{TG}	100 m^{-3}
β	0.50
N_{cc}	500 cm^{-3}
p_{cb}	708 mb
T_{cb}	4.0 °C
q_{cb}	7.26 g/kg
radm	0.3
O_{NS}	0.01 m^{-3}

QUATERNARY LIBRARY

development, in agreement with observations. During the glaciation period the vertical velocity increases to $\leq +8$ ms^{-1}. The precipitation water fields (figures E.1.15 and E.1.21) show a small amount of water (as drizzle drops) co-existing with snow and graupel from 2700 to 3000 seconds near cloudtop. The maximum cloudtop of 5500 m (6600 m MSL, 455 mb) was achieved after the cloud glaciated. The cloud yielded a short rain shower with .91 mm of rain and .001 mm of graupel reaching the lowest level in the model domain during the model run.

3.2.3. Entrainment

Figure 3.23 shows a Paluch plot of the data from the three H2 penetrations at 511 mb shown in figures 3.16 to 3.18. The points plotted are one second averages. Any second in which; a) the liquid water measured by the Johnson-Williams probe was either negative or 0.0, b) the ice concentration measured by the 2DC probe was ≥ 2 l^{-1}, c) or the 50 Hz droplet detection (nstrobe) from the FSSP indicated no droplets at some point during the second or in either the preceding or following second was rejected. The plot shows evidence for mixing from several levels, but no unmixed parcels are indicated. This is consistent with the relatively high turbulence levels observed in the time series of the aircraft penetrations.

Plotting the data points from figure 3.23 on a saturation point mixing diagram shows in figure 3.24 that all the observed parcels are negatively buoyant at the penetration level, but the three points near $\theta_q \approx 324.7$ and $q_{tot} \approx 6.45$ in figure 3.23 (seconds 58, 59 and 60 from figure 3.18) are near neutral buoyancy. Assuming a single mixing event, these three points are mixtures of cloudbase air and environmental air from around 570 mb. As may be seen in figure 3.25, these parcels are positively buoyant at 567 mb and would continue to rise. The other data points in figure 3.23 with high θ_q and q_{tot}; $\theta_q \approx 323.7$ and $q_{tot} \approx 6.4$ (seconds 20 and 21 in figure 3.16), are consistent with mixing between cloudbase air and environmental air from around 695 mb. Figure 3.26 is a saturation point mixing diagram showing these points have near neutral buoyancy at the mixing level. These points are positively buoyant at 567 mb.

Examination of the observed points with Saturation Points near 690 mb (those which have undergone the least mixing) indicates that the remaining observed data points at 511 mb are consistent with mixing between environmental air near cloudtop

and cloud air previously mixed at 695 mb and 550 mb. Multiple mixing events result in a large number of possible mixing fractions of the various components, and in consequence diminish the value of the Paluch analysis as a means to investigate entrainment. Clouds with well defined nearly adiabatic cores; e.g., the July 19 case discussed at the beginning of this chapter, offer a better chance of observing two point mixing. However, as may be seen in figure 3.9, even clouds with nearly adiabatic cores may have complicated Paluch diagrams.

The model output may be compared with the data points presented in figure 3.23 keeping in mind that the data points exhibit the cloud properties over a seven minute period. Figures 3.27 to 3.29 show the values at the penetration level (511 mb) predicted by the cloud model initialized with the base run parameters from table 3.3. The model points plotted are those points with $q_l \geq 0.01$, a total ice concentration ≤ 0.01 g/kg and $q_p \leq 0.01$ g/kg. A point is output every 60 seconds. The time development is shown in figure 3.30. As may be seen in figure 3.30, the cloud model output shows reasonable agreement with the H2 data during the later stages of cloud evolution, but prior to glaciation.

3.3. Summary

The two case studies in this chapter indicate that the model discussed in chapter 2 is able to match the general dynamic and microphysical characteristics of midlatitude continental cumulus congestus and small cumulonimbus. The aircraft observations provide detailed information on a much finer horizontal resolution than contained within the model, but repeated penetrations at the same level within a cloud offer a means of comparing the aircraft measurements with model predictions. Plotting the time series from the cloud model at the penetration level for these two clouds on conserved variable diagrams demonstrated that the computed cloud properties were in general agreement with those obtained by multiple aircraft penetrations.

The sensitivity of these results to variation in model parameters is discussed in the next chapter.

QUATERNARY LIBRARY

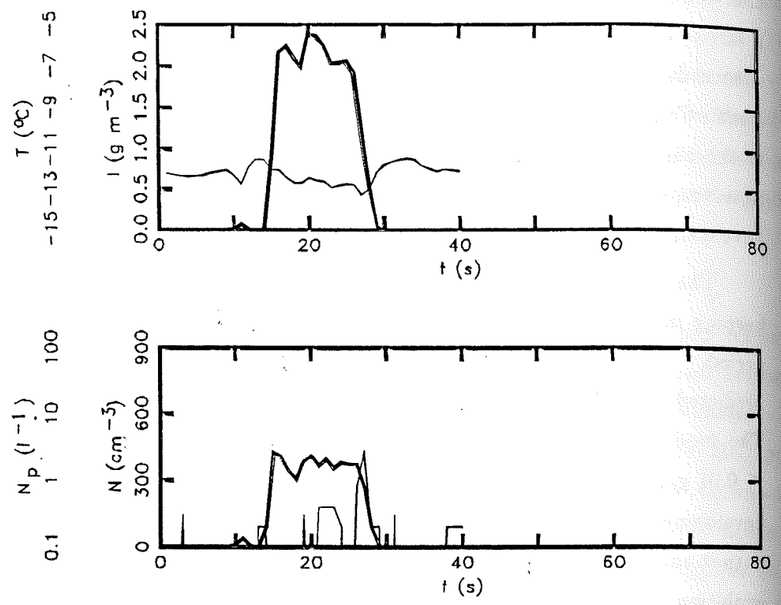


Figure 3.16 Time series of Temperature (T), corrected liquid water derived from the J-W probe, the number of ice particles (N_p), the droplet number concentration (N), the turbulence dissipation rate (ϵ), and the vertical velocity (w) for the H2 penetration starting at 22.26.10 UT on August 6. The penetration level was 508 mb. The aircraft speed was approximately 80 ms^{-1}.

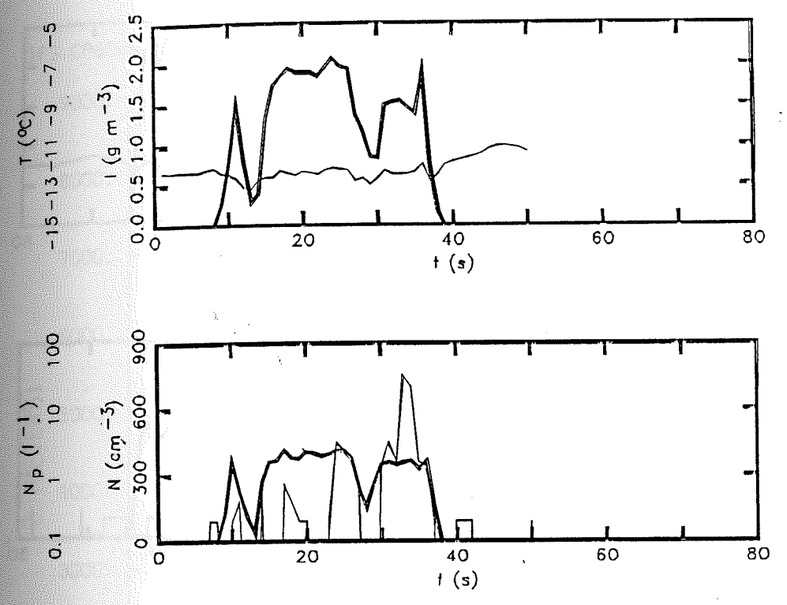


Figure 3.17 Time series of Temperature (T), corrected liquid water derived from the J-W probe, the number of ice particles (N_p), the droplet number concentration (N), the turbulence dissipation rate (ϵ), and the vertical velocity (w) for the H2 penetration starting at 22.38.20 UT on August 6. The penetration level was 512 mb. The aircraft speed was approximately 80 ms^{-1}.

QUATERNARY LIBRARY

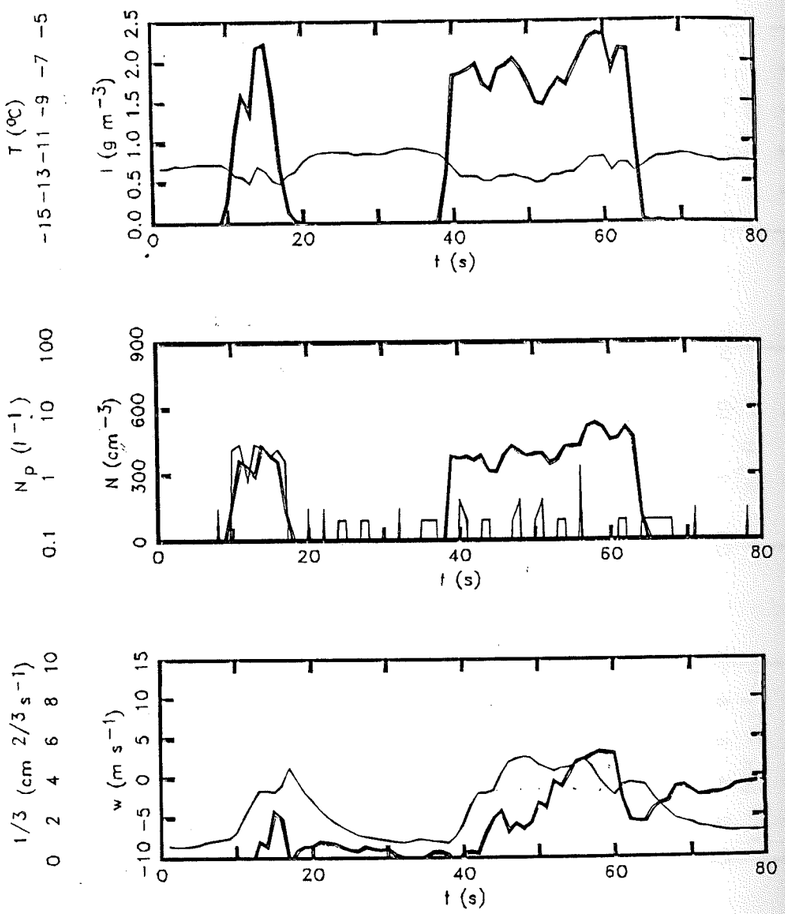


Figure 3.18 Time series of Temperature (T), corrected liquid water derived from the J-W probe, the number of ice particles (N_p), the droplet number concentration (N), the turbulence dissipation rate (ϵ), and the vertical velocity (w) for the H2 penetration starting at 22.41.20 UT on August 6. The penetration level was 508 mb. The aircraft speed was approximately 80 ms^{-1}.

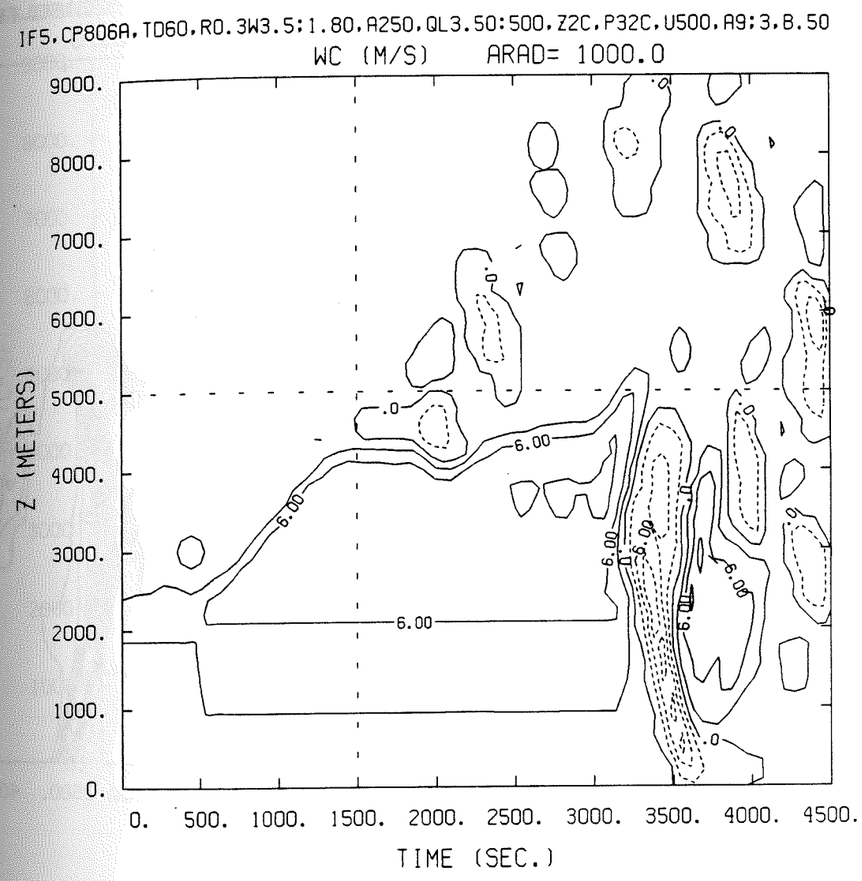


Figure 3.19 Contour plot of vertical velocity in the inner cloud region for the August 6 base run. Contour interval is 3 ms^{-1}; negative velocities are shown as dotted contours.

QUATERNARY LIBRARY

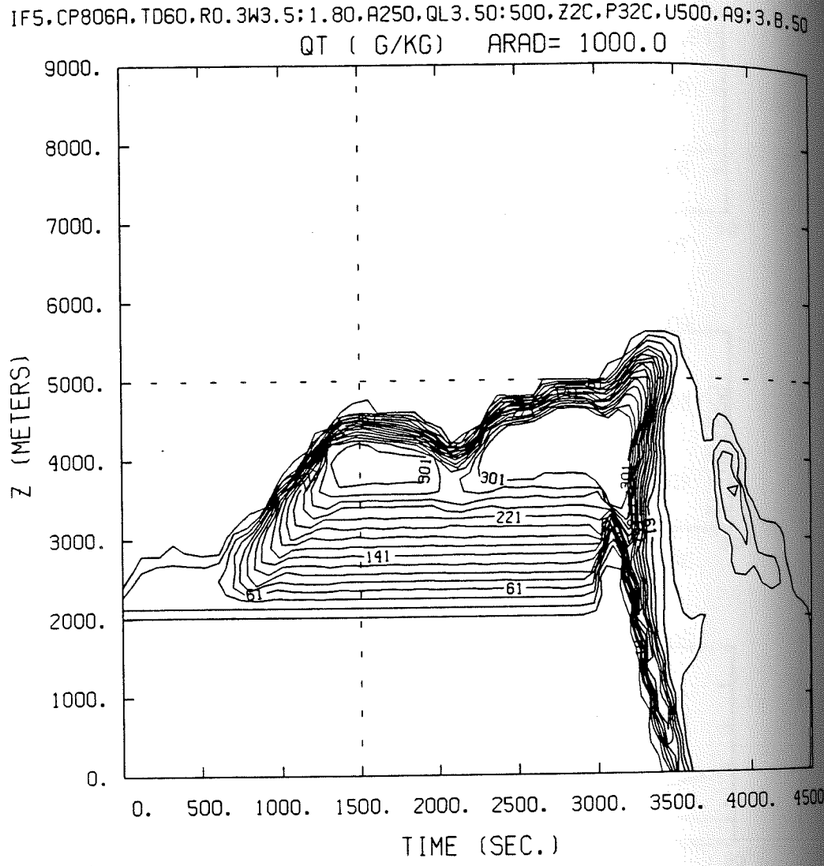


Figure 3.20 Contour plot of the total liquid water and ice mixing ratio for the inner region. Contours are from 0.01 to 3.01 g/kg at 0.2 g/kg intervals. The labels are scaled by 100.

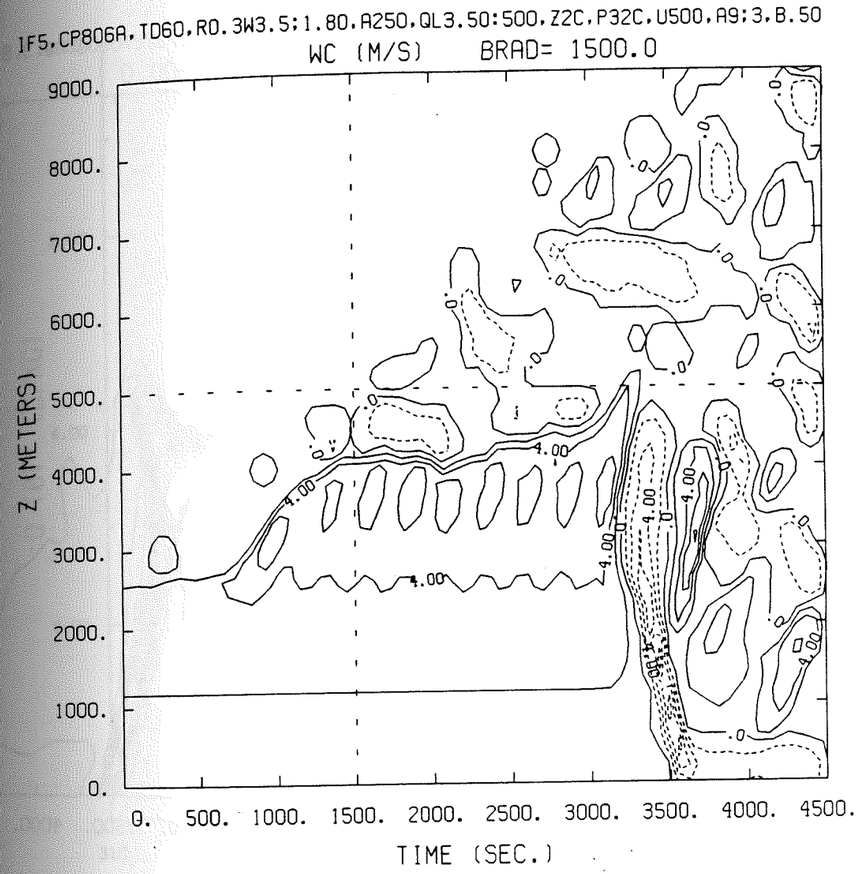


Figure 3.21 Contour plot of vertical velocity in the outer cloud region for August 6 base run. Contour interval is 2 ms^{-1}; negative velocities are shown as dotted contours.

QUATERNARY LIBRARY

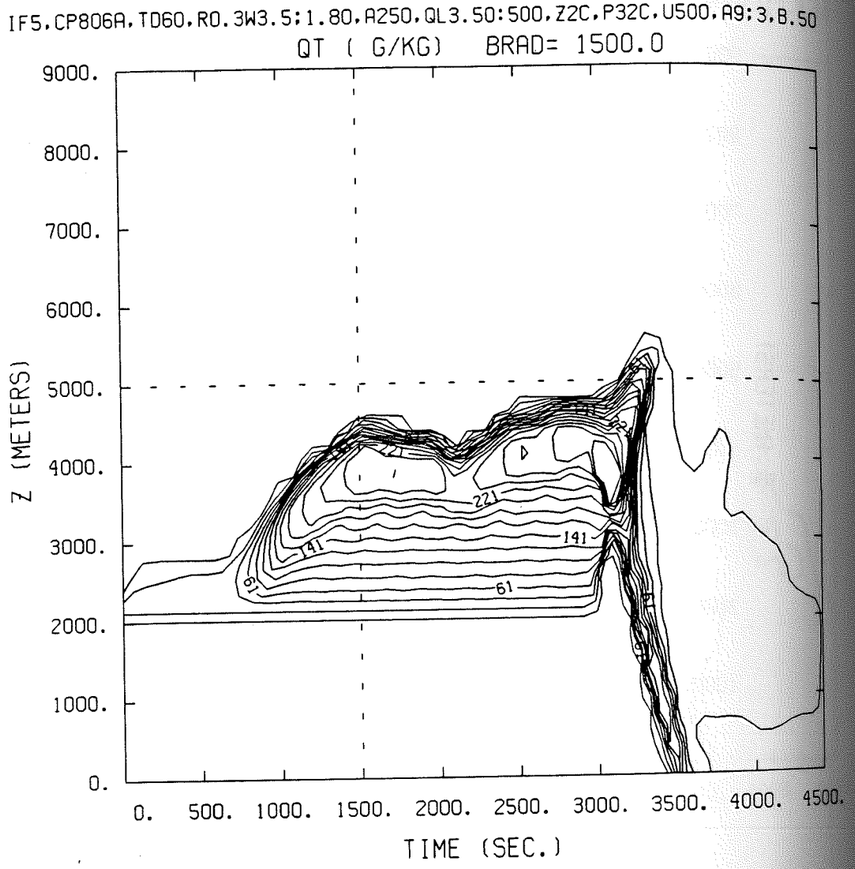


Figure 3.22 Contour plot of the total liquid water and ice mixing ratio for the outer region. Contours are from 0.01 to 3.01 g/kg at 0.2 g/kg intervals. The labels are scaled by 100.

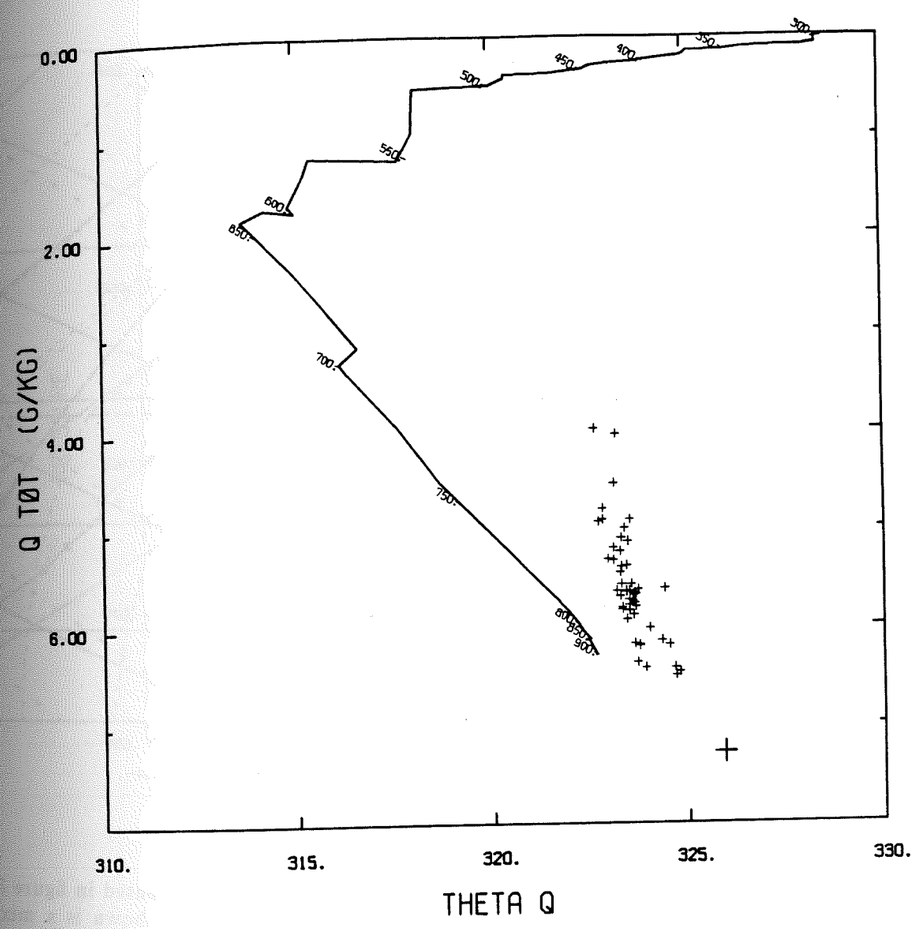


Figure 3.23 Composite Paluch plot of the three penetrations shown in figures 3.16-3.18. The August 6 sounding from figure 3.15 is also plotted. Each + represents one second of data. Cloud base is indicated by the large +.

QUATERNARY LIBRARY

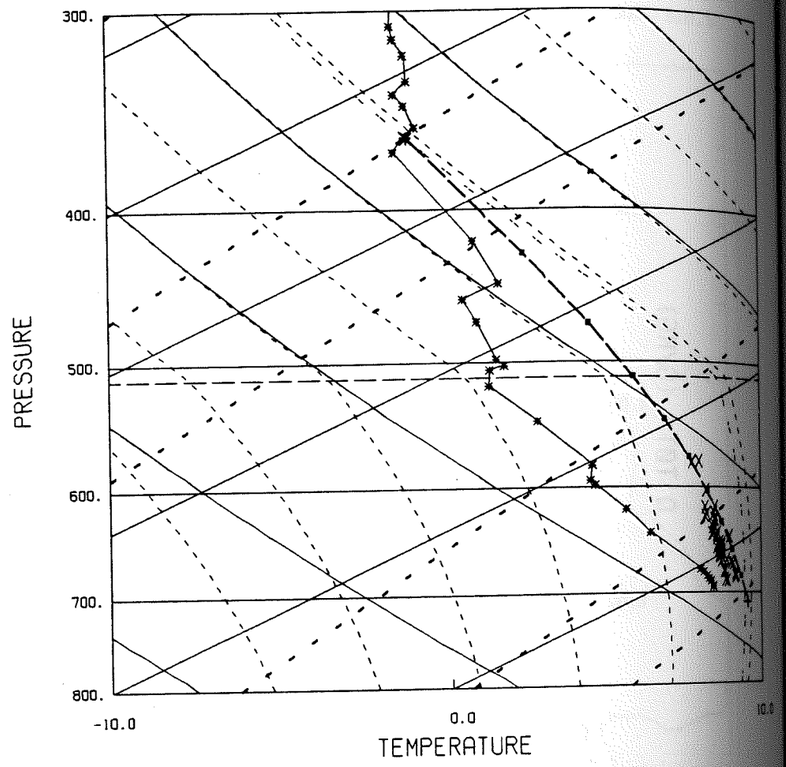


Figure 3.24 Saturation point diagram of data presented in figure 3.23 (x). The solid line is the sounding from figure 3.15. Also shown is a mixing line between cloud base (708 mb) and the environment at 511 mb. The density isopleths are computed for 511 mb.

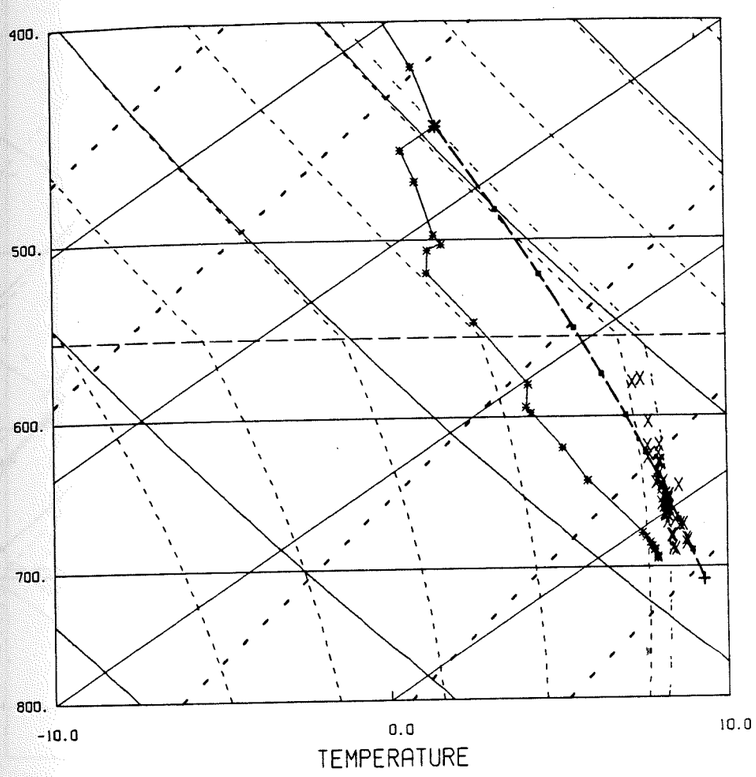


Figure 3.25 Saturation point diagram of data shown in figure 3.23. The mixing line is between cloud base and the environment at 567 mb. The density isopleths are computed for 567 mb.

QUATERNARY LIBRARY

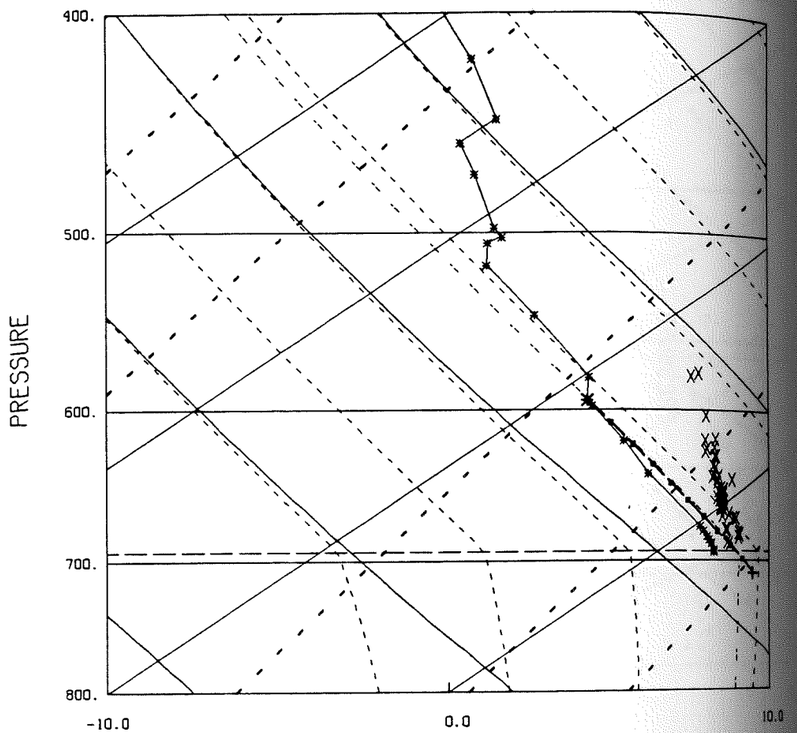


Figure 3.26 Saturation point diagram of data shown in figure 3.23. The mixing line is between cloud base and the environment at 694 mb. The density isopleths are computed for 694 mb.

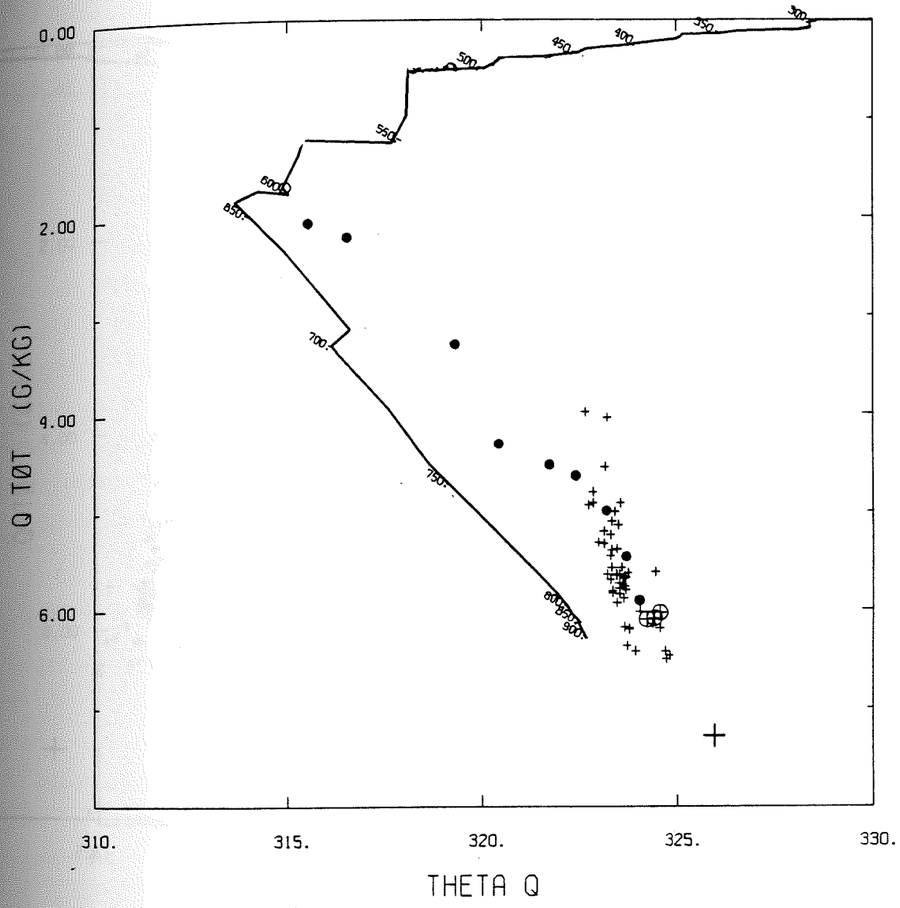


Figure 3.27 Paluch plot of data from figure 3.23 and the model points (dots) at 60 second intervals for the inner region.

QUATERNARY LIBRARY

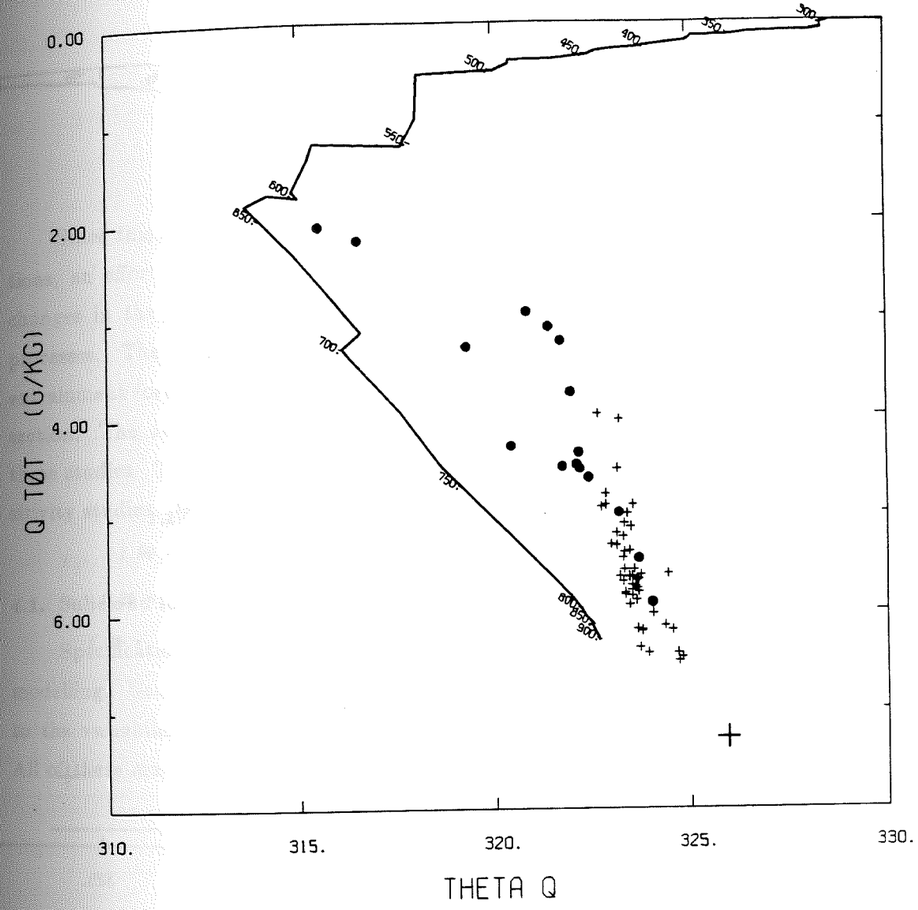


Figure 3.29 Paluch plot of data from figure 3.23 and the model points (dots) at 60 second intervals for both regions.

QUATERNARY LIBRARY

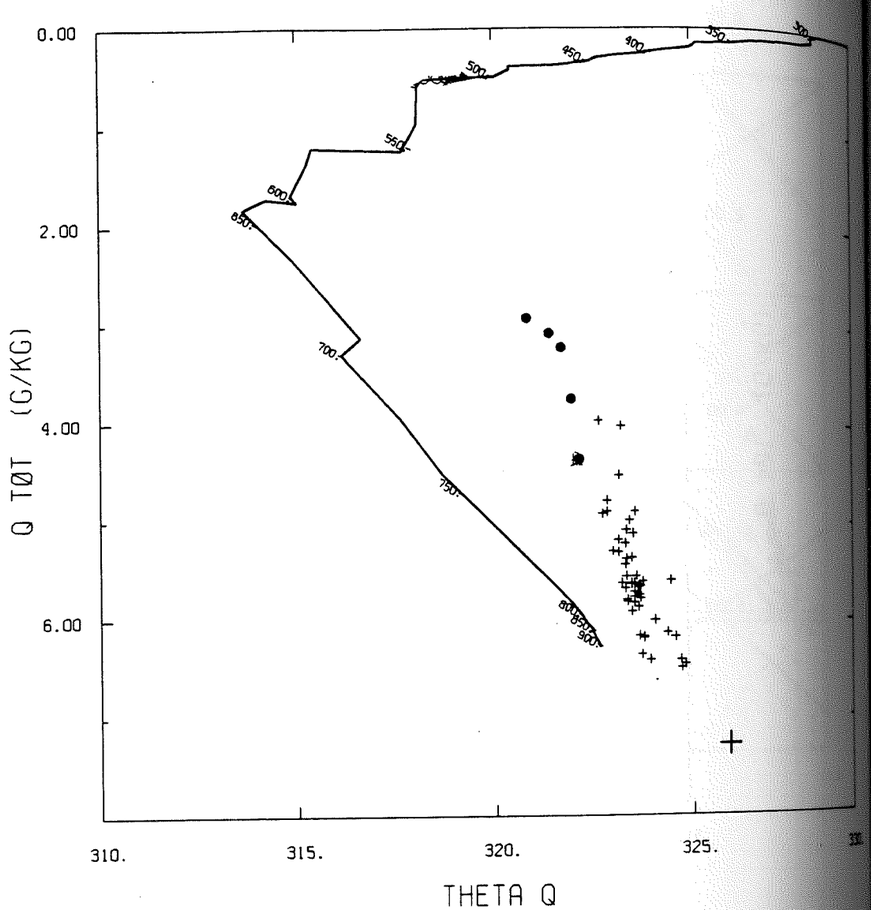


Figure 3.28 Paluch plot of data from figure 3.23 and the model points (dots) at 60 second intervals for the outer region.

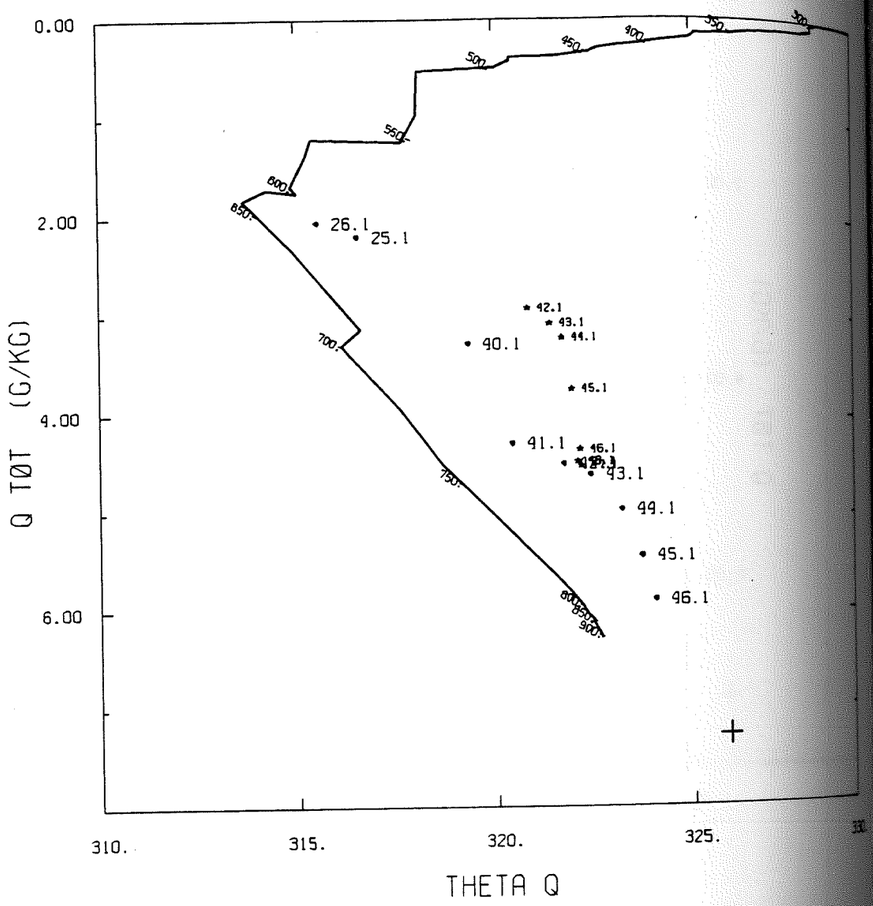


Figure 3.30 Paluch plot of data from figure 3.23 and the model points for both regions. The inner region points are shown as points, the outer region as *. Times for both are shown in minutes.

CHAPTER 4

MODEL SENSITIVITY ANALYSIS

Since this cloud model is a coupled system of non-linear partial differential equations, an effort must be made to determine the sensitivity of the model response to changes in the adjustable parameters used in the representation of various physical processes. The response of the system to variations in the a) subcloud forcing, b) entrainment formulation, and c) microphysics parameterization will be discussed in this section. The July 19 base model run discussed in the preceding chapter is used for these studies. Table 4.1 lists the default values for the variables examined in these sensitivity studies. A full listing may be found in Table 3.2.

4.1. Subcloud forcing

Specification of subcloud forcing is one of the most difficult aspects of cumulus modelling. Usually the cloudbase thermodynamic properties and velocity, in addition to the variability and duration of forcing, are poorly known and must be estimated. All of these may influence cloud lifetime, dynamics and the microphysical development.

table 4.1
Sensitivity run default parameter values

ptime	2700 s
utime	1500 s
w_{cb}	3.0 ms^{-1}
wmult	2.0
avl	800 m
β	0.50
N_{TG}	100 m^{-3}

QUATERNARY LIBRARY

Cloudbase properties were well documented for the clouds discussed in chapter 3, so this examination of the effects on cloud properties due to variation in the subcloud forcing will be limited to the magnitude and duration of the cloudbase forcing.

Simulations were run for a number of cases in order to determine the relationship between these two variables and the modelled cloud properties. Tables 4.2-4.4 contain a summary of those cases considered in this section. Note that the velocity forcing studies are divided into three categories. In the first series of model runs (in table 4.2) a constant velocity of equal magnitude was imposed in both cloud regions for the entire forcing period. In order to test the effects of a spatially non-uniform cloud base forcing the second series (table 4.3) have a cloudbase forcing of variable magnitude in the inner region coupled with a constant forcing in the outer region. The inner region velocity is increased by a constant multiplication factor (here wmult=2X) after 1500 seconds of simulation time. The third series (table 4.4) is an examination of the effect of a variation in wmult. The model runs start with the same initial velocity and vary the multiplication factor for the inner region velocity. The outer region velocity is held at the initial value during the simulation. The time at which the inner region velocity increases is set to 1500 seconds for these simulations. This corresponds to the time for the cloud to reach the maximum liquid water cloud top (about 6 km in the model domain). Variation of this time (stepped from 1000 to 2000 seconds) done in additional model runs changed the overall structure little. The results of these tests are discussed in the following sections.

4.1.1. Sensitivity to variation in forcing velocity magnitude

Table 4.2 shows a direct correlation between cloud properties and the forcing velocity magnitude. A cloudbase velocity (wcb) < 2 ms^{-1} results in very weak cloud formation confined to the first kilometer above cloudbase, but as the velocity is stepped from 2.0 to 3.5 ms^{-1}, maximum cloudtop rises from 4.2 km to 10.6 km, the maximum vertical velocity increases from 4 to 15 ms^{-1}, and the maximum liquid water content increases from 0.2 to > 3.0 g/kg. The outer region properties exhibit similar behavior.

Table 4.3 indicates that doubling the vertical velocity forcing in the inner region after 1500 seconds changes the overall cloud properties little. For initial velocities below 2 ms^{-1} the cloud properties are essentially unchanged. This is somewhat

table 4.2
variation in subcloud forcing; constant cloud base velocity

	w_{cb} ms^{-1}	w_{MAX} ms^{-1}	z_{MAX} m	$q_{c\ MAX}$ g/kg	$q_{s\ MAX}$ g/kg	precip.
inner region	2.0	4	4200	0.2	none	no
	2.5	8	6200	2.8	.01	no
	3.0	12	9800	>3	1.2	yes
	3.5	15	10600	>3	1.4	yes
outer region	2.0	2	3800	.01	none	
	2.5	4	4800	1.2	none	
	3.0	8	9600	2.0	1.2	
	3.5	12	10600	>3	1.2	

surprising, as a constant forcing velocity of 3.5 ms^{-1} (see table 4.2) results in a cloud growing to the tropopause (see figure 3.1), while for an initial velocity of 2.0 ms^{-1}, and a step to 4.0 ms^{-1} at 1500 s, the cloudtop remains at 4200 m. In this case the increase in vertical velocity in the inner region does not yield a larger cloud for two reasons. Examination of the entrainment velocity indicates that the detrainment pattern is similar in both cases ($w_{CB} = 2.0$; wmult $= 1$ & 2). This means that the forcing velocity increase at 1500 seconds is not sufficent to overcome the detrainment field established by the initial velocity. Furthermore, since the velocity increases only in the inner region, no protection exists for the core in the region above cloudbase. This results in rapid drying through entrainment.

In all of these cases, the maximum cloudtop is correlated with the initial vertical velocity. These results suggest that observed cloudtop regimes can be understood as the result of the interaction of the cloud dynamics and microphysics with the environment in response to different cloudbase forcings. From these preliminary tests it appears that, for this sounding, the relationships are as follows:

QUATERNARY LIBRARY

table 4.3
variation in subcloud forcing; wmult=2.0 for inner region

	w_{cb} ms^{-1}	w_{MAX} ms^{-1}	z_{MAX} m	$q_{c\ MAX}$ g/kg	$q_{s\ MAX}$ g/kg	precip.
inner region	2.0	4	4200	0.4	none	no
	2.5	8	7200	>3	.08	no
	3.0	12	10200	>3	>3.0	yes
outer region	2.0	20	3800	.01	none	
	2.5	4	4800	1.2	none	
	3.0	8	10200	2.0	1.8	

a) Fair weather cumulus

The low level clouds (depth $\approx$ 1 km) are driven primarily by the initial forcing. The latent heat release through condensation is not sufficent to overcome the initial dynamically driven detrainment at cloud top.

b) Cumulus congestus

For initial vertical velocities greater than 2 ms^{-1}, the overall cloud structure is driven by latent heat release. The cloud depth then becomes a function of the environmental parameters as well. The cloud initialized at 2.5 ms^{-1} grows to approximately the level of neutral buoyancy for a parcel undergoing moist adiabatic ascent (see figure 3.8 for the virtual temperature sounding). This is reasonable, as little ice (and subsequently no precipitation) is produced in this case, which means that the only reduction in liquid water will be via entrainment and subsequent evaporation. The maximum liquid water entry in tables 4.2 and 4.3 shows that entrainment is important primarily in the outer cloud region for these clouds.

c) Cumulonimbus

Once the initial velocity is around 3 ms^{-1}, sufficent moisture is available at the -10 to -20 °C temperature interval to enable the ice phase to dominate the overall cloud structure. This results in precipitation and an increased cloud depth due to additional latent heat release coupled with updraft unloading.

Photographs (taken from H5) of the general cloud field on July 19 indicate that many of the clouds were near the liquid water cloud top, while a few (of which the study cloud was an example) grew to the cloud top shown for forcing around 3 ms^{-1}. Therefore, the fact that these model simulations indicate that the cloud properties are highly sensitive to threshold velocities, i.e., those corresponding to small 'fair weather' type cumulus, cumulus congestus, and cumulonimbus, may be reflected in the observed cloud field.

4.1.2. Sensitivity to inner region velocity variation

Table 4.4 lists the cloud properties for a variation in the inner region velocity multiplication factor (*wmult* in table 3.2). All of these cases had cloud top above the liquid water cloud top, and precipitated. The maximum snow content was greater than 2 g/kg in all cases. The total precipitation also varies by an order of magnitude, but shows a maximum near wmult $\approx$ 1.5 to 2.0. This shows some correlation with the maximum velocity in the inner region. Although the outer region does not show much variation in the maximum liquid water content, the maximum snow content varies by an order of magnitude, and is correlated with *wmult* .

These processes are interrelated. Figure E.2.1 in Appendix E shows the inner region velocity for wmult = 0.5, 1.5 and 2.5. Although the maximum velocity in each case is 9 ms^{-1}, the duration and vertical extent of the 9 ms^{-1} interval is directly correlated with wmult. Examination of figure E.1.1 (the entrainment velocity for the base run case) indicates that when the velocity is increased to 6.0 ms^{-1} (wmult = 2) the entrainment near cloud base approaches zero. This is a result of the mass flux through cloud base almost exactly offsetting the entrainment required to satisfy mass continuity (see eqn. 2.7) caused by the buoyancy acceleration (which yields a vertical gradient in w). Dropping the inner region velocity (wmult = 0.5) results in more entrainment and a rapid decrease in the vertical velocity. This decrease in vertical velocity results in a

QUATERNARY LIBRARY

table 4.4
variation in subcloud forcing; variable *wmult*

	wmult	w_{MAX} ms^{-1}	z_{MAX} m	$q_{c\ MAX}$ g/kg	$q_{s\ MAX}$ g/kg	precip. rain/graupel mm
inner region	0.5	9	8500	>3	2.0	0.5/0.0
	1.0	12	9800	>3	2.0	0.9/0.0
	1.5	9	10000	>3	>3	5.0/.004
	2.0	12	10200	>3	>3	4.8/.004
	2.5	9	10200	>3	>3	2.3/.001
outer region	0.5	6	8500	2.0	0.2	
	1.0	8	9600	2.0	1.2	
	1.5	6	9800	2.0	1.8	
	2.0	8	10200	2.0	1.8	
	2.5	6	10200	2.4	2.2	

rapid fallout of the small graupel particles, which in turn affects the total precipitation. At wmult = 1.5, the cloud base entrainment is near zero, and the vertical velocity reflects the acceleration of air (due to latent heat release). The increased vertical velocity permits graupel to grow via riming and snow aggregation for a longer period of time. This is shown in the total precipitation. For wmult = 2.5, the cloudbase velocity exceeds the entrainment requirements, and results in additional water being detrained to the outer cloud region (as shown by the higher liquid water in table 4.4) and a much more uniform vertical velocity in the inner region. The total precipitation for this case is less, primarily since fall out of graupel is inhibited in the lower cloud regions. This lowers the graupel riming rate, and allows more snow to form at higher altitude. The total water mass in the graupel category is less then, as the snow to graupel conversion terms, i.e., collection and aggregation, are less efficient at low temperatures.

4.1.3. Subcloud forcing summary

Overall, it appears that the cloud properties (cloudtop height, maximum vertical velocity, precipitation, etc.) are highly sensitive to the subcloud forcing. These tests indicate that for the range of inner region velocities close to the value necessary for reducing the entrainment near cloudbase, the cloud properties are close to that observed for the cumulonimbus on July 19 (see Chapter 3). This range of velocities (wmult $\approx$ 1.5 to 2.0; w $\approx$ 4.5 – 6.0) corresponds to the upper range of vertical velocities observed below cloudbase for that case.

4.2. Entrainment formulation

Table 4.5 lists cloud properties for model runs using different parameterizations of the entrainment with the base run parameters found in table 4.1. The entry labeled *A&K* uses the entrainment assumption of Asai and Kasahara (1967), i.e., that the entrainment necessary to satisfy mass continuity is lateral (see eqn. 2.21). The entry labeled *Cotton* denotes use of Cotton's (1975) flux-weighted entrainment hypothesis based on the values of vertical and entrainment velocity (eqn. 2.24). The remaining entries denote the value of the mixing length (in meters) used in the entrainment formulation developed in chapter 2. This weights the lateral entrainment by the value of the local turbulent kinetic energy (eqn. 2.25).

A comparison of the dynamic, microphysical and thermodynamic properties resulting from these different entrainment formulations will be found in the following sections.

4.2.1. Sensitivity of dynamics and microphysics to entrainment formulations

Table 4.5 shows that the general characteristics of the model simulations are relatively insensitive to the entrainment formulation, the exception being the run incorporating the Asai and Kasahara (1967) form. This entrainment formulation results in a cloud with lower updraft velocity in both the core and outer cloud region, much less snow in both regions, and no precipitation. The Cotton (1975) and turbulent kinetic energy formulations show good agreement in these gross characteristics. The turbulent kinetic energy formulation also shows a strong correlation between the mixing length (avl) and the total precipitation, outer region velocity, liquid and snow mixing ratios.

QUATERNARY LIBRARY

table 4.5
variation in entrainment formulation

	entrainment	w_{MAX} ms^{-1}	z_{MAX} m	$q_{c\ MAX}$ g/kg	$q_{s\ MAX}$ g/kg	precip. rain/graupel mm
inner region	A&K	8	10000	>3	0.6	0.0/0.0
	Cotton	12	10200	>3	>3	6.6/.028
	600 m	12	10200	>3	>3	6.3/.018
	800 m	12	10200	>3	>3	4.8/.004
	1000 m	9	10200	>3	>3	2.4/0.0
outer region	A&K	4	9000	0.6	0.2	
	Cotton	9	10200	>3	3.0	
	600 m	9	10200	>3	2.4	
	800 m	8	10200	2.0	1.8	
	1000 m	6	10000	2.0	1.8	

Maximum cloudtop is nearly independent of these parameters. (see figures E.2.2-E.2.2 in Appendix E)

4.2.2. Sensitivity of thermmodynamic fields to entrainment formulations

Differences in the computed cloud properties do appear when the model output is plotted on a Paluch diagram. Figures 4.1 through 4.5 are plots of the model output from both regions at the H2 penetration level (481 mb; 5200 m in the model domain). The observed data points discussed in section 3.1.4 are also plotted. Note that the limits regarding precipitation and ice in the computation of θ_q imply that the model points are those prior to substantial glaciation (ice less than 0.01 g/kg) in the cloud, and so are representative of the initial cloud growth and liquid cloud dwell phase.

Figure 4.1 is the Paluch diagram in which the model results for the Asai and Kasahara entrainment formulation and the observed points described in Chapter 3 are

compared. This diagram shows an inner model region (at 481 mb) which has undergone more mixing than observed (the cluster of points near $\theta_q \approx 329$ and $q_{tot} \approx 6.2$ g/kg). The outer model region (the points near $\theta_q \approx 326$ and $q_{tot} \approx 2.5$ g/kg) exhibits some mixing from near the penetration level. Figure 4.2 shows the Cotton (1975) formulation. In this case, both the inner and outer regions show little mixing, and most of the computed points lie close to the nearly adiabatic values observed and discussed in Chapter 3. Paluch diagrams at the same level for model simulations using the turbulent kinetic formulation with avl = 600, 800, and 1000 meters are shown in figures 4.3, 4.4 and 4.5. All of these show inner region points which agree reasonbly well with the observations of the nearly adiabatic points (the points near $\theta_q \approx 329$ and $q_{tot} \approx 6.5$ g/kg), and show mixing in the outer model region (the points near $\theta_q \approx 326$ and $q_{tot} \approx 3$ g/kg) from near the penetration level. The turbulent kinetic energy formulation tends to emphasize entrainment (relative to the Cotton formulation) in regions of large gradients of velocity or buoyancy, or in regions of high shear (see equations 2.20, 2.25, and 2.26); e.g., at cloudbase or near cloud top, or at updraft/downdraft interfaces, while the Cotton formulation responds only to the ratio of $\frac{\partial}{\partial z}$w to w. The turbulent kinetic energy formulation includes an additional parameter, i.e., the mixing length *avl*, but these tests indicate that the model results are not extremely sensitive to the choice of mixing length. Both the base run for July 19 and August 6 were run with avl $\approx$ 0.5(brad – arad).

4.3. Sensitivity to microphysical parameterization

The basic microphysical parameterization has been well documented in the literature; e.g., Stephens (1979), Cheng (1981), or Taylor and Baker (1987). In the clouds under consideration precipitation is initiated primarily through the ice phase. Therefore, this series of sensitivity tests is focused on the impact of variation of the important parameters in the ice portion of the microphysical parameterization. In particular, I will examine the effect of a variation of the adjustable parameters controlling the snow and graupel distributions.

QUATERNARY LIBRARY

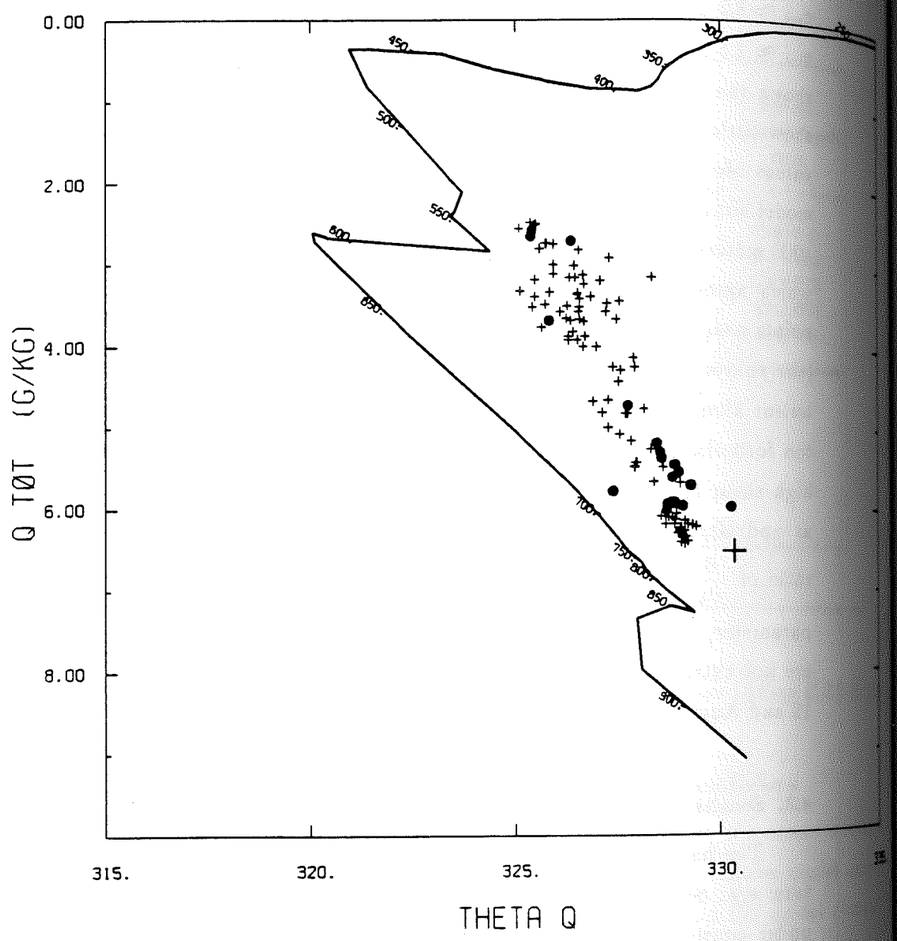


Figure 4.1 Combined Paluch diagram for the inner and outer cloud regions using the Asai and Kasahara entrainment formulation. Cloudbase is denoted by the large +, H2 observations (1 sec. averages - see Chapter 3) are denoted by the small +, and the model computed points (at 60 second intervals) by the filled circles.

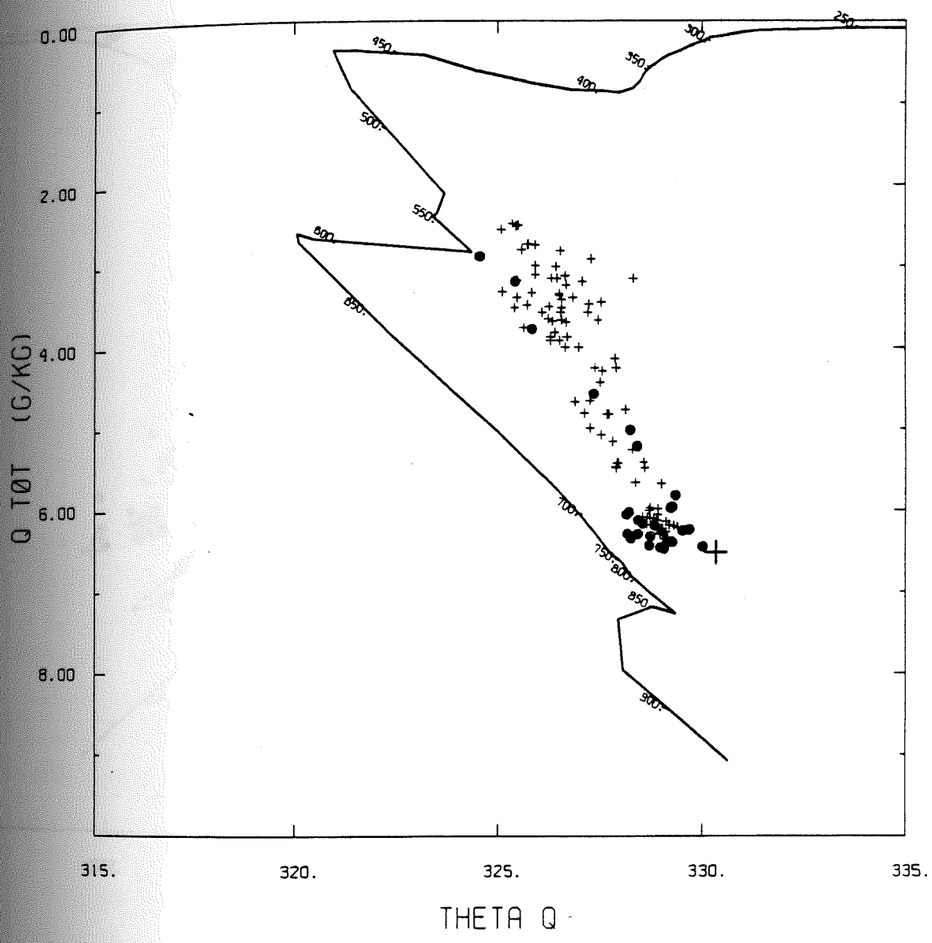


Figure 4.2 Combined Paluch diagram for the inner and outer cloud regions using the Cotton entrainment formulation. Cloudbase is denoted by the large +, H2 observations (1 sec. averages - see Chapter 3) are denoted by the small +, and the model computed points (at 60 second intervals) by the filled circles.

QUATERNARY LIBRARY

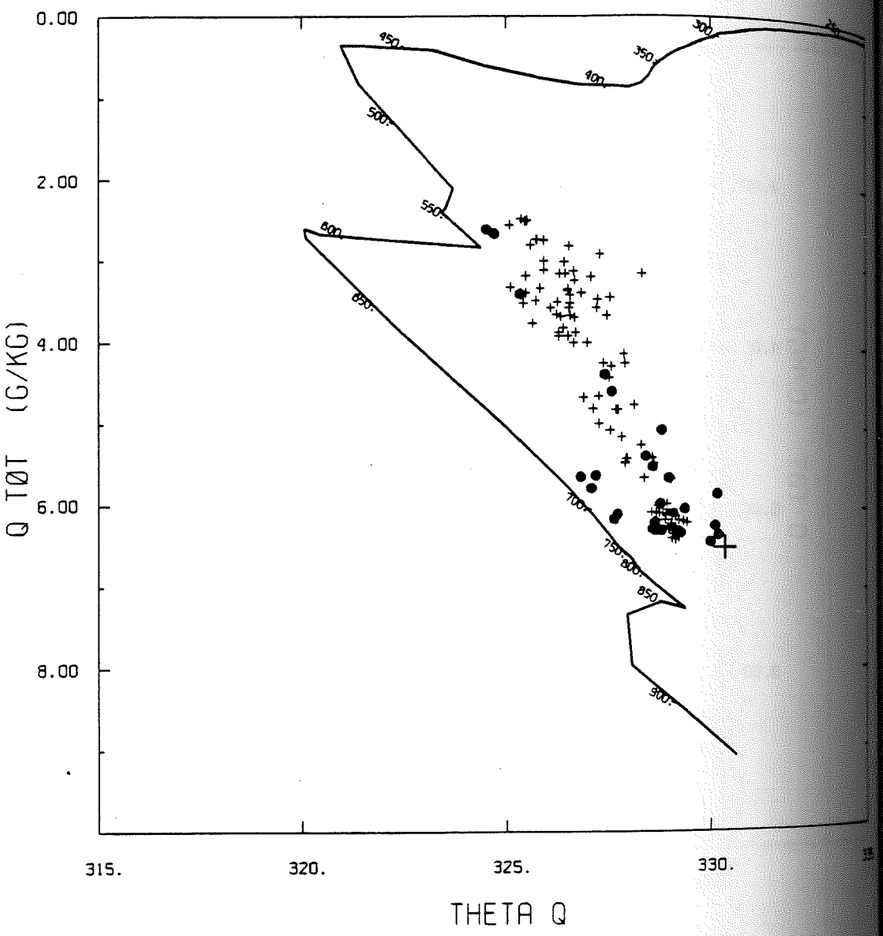


Figure 4.3 Combined Paluch diagram for the inner and outer cloud regions using the TKE entrainment formulation with $avl = 600$ m. Cloudbase is denoted by the large +, H2 observations (1 sec. averages - see Chapter 3) are denoted by the small +, and the model computed points (at 60 second intervals) by the filled circles.

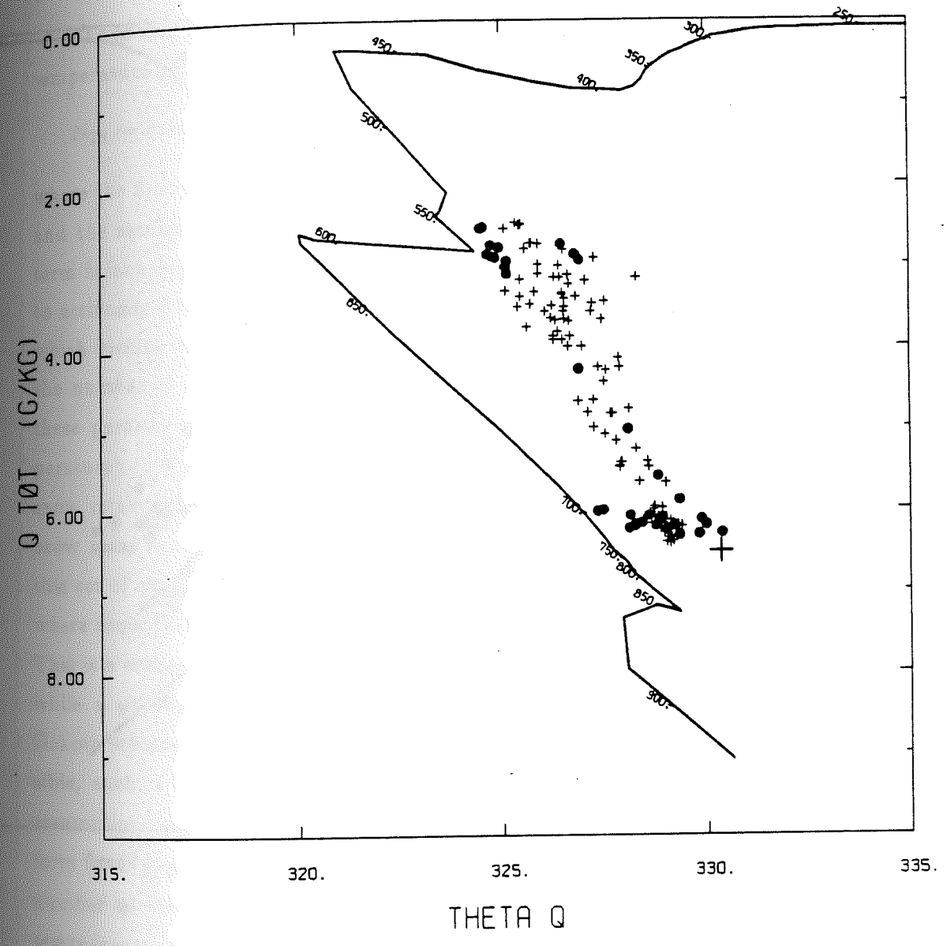


Figure 4.4 Combined Paluch diagram for the inner and outer cloud regions using the TKE entrainment formulation with $avl = 800$ m. Cloudbase is denoted by the large +, H2 observations (1 sec. averages - see Chapter 3) are denoted by the small +, and the model computed points (at 60 second intervals) by the filled circles.

QUATERNARY LIBRARY

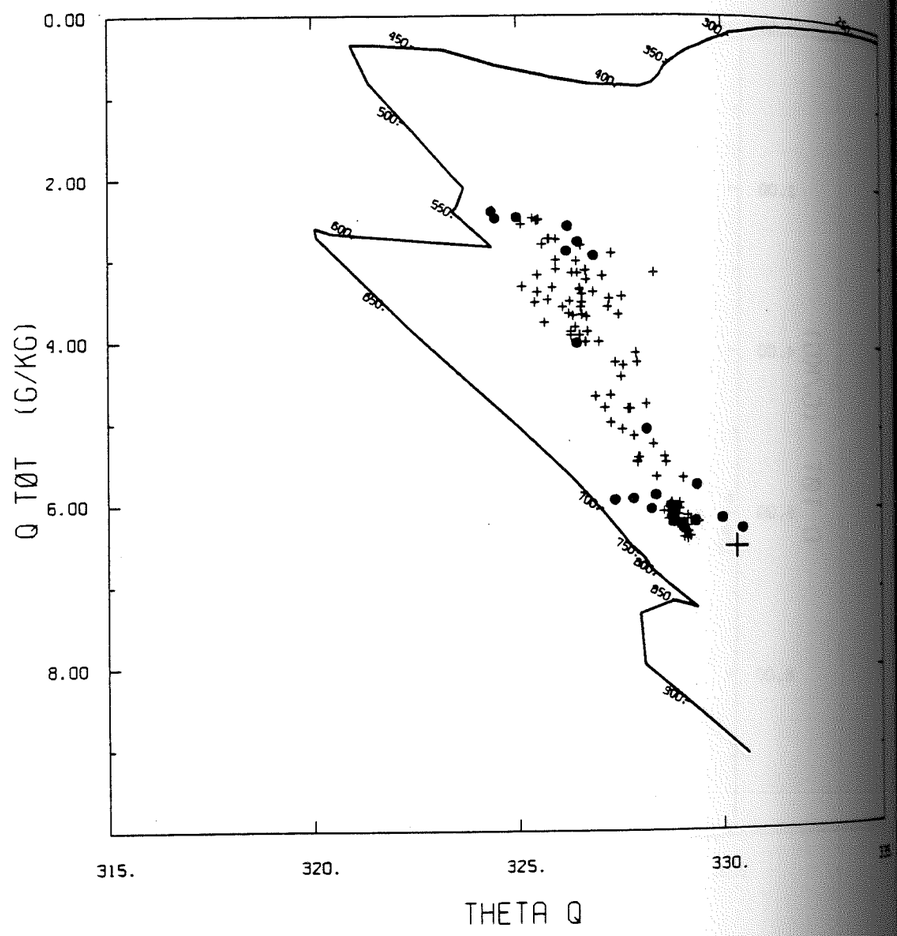


Figure 4.5 Combined Paluch diagram for the inner and outer cloud regions using the TKE entrainment formulation with $avl = 1000$ m. Cloudbase is denoted by the large +, H2 observations (1 sec. averages - see Chapter 3) are denoted by the small +, and the model computed points (at 60 second intervals) by the filled circles.

4.3.1. Sensitivity to snow concentration

4.3.1.1. Snow particle number

The total number of particles in the snow category is defined as a function of temperature and the total snow concentration, i.e.,

$$N_{TS} = \max \left[O_{NS}\, e^{\beta\,(\,273.16\,-\,T\,)}\,,\ \rho_o q_s\,(\,10^6\,) \right] \quad [\,m^{-3}\,], \tag{4.1}$$

where the first term on the right hand side of equation 4.1 is the Fletcher (1969) curve, and the second term limits the maximum snow crystal mass to 0.001 g. This second term is included to alleviate the tendency of the basic Stephens (1979) parameterization to artifically force snow into the graupel category at temperatures above around –12 ° C which results from limiting the maximum snow crystal mass to .001 g, *and* computing the number of snow particles by the Fletcher curve. Simultaneous constraint of both these parameters yields a maximum snow mixing ratio permitted (a function of temperature); in the Stephens parameterization any snow mass in excess of the maximum snow mixing ratio is dumped into the graupel field. Using these criteria, the critical snow mass is reached for a concentration of 0.01 g/m^{-3} at T $\approx$ –12 ° C. The result of this set of conditions is rapid conversion of snow to graupel at the temperature interval where snow first forms, leading to a rapid glaciation of the cloud due to graupel riming. This is a consequence of including ice crystals and snow in the same bulk category, and while a parameterization which distinguishes them (such as a version of the South Dakota scheme; e.g., Lin et al., (1983), Rutledge et al., (1986), or Ferrier (1987)) is possible, such a scheme introduces another set of fields, microphysical transitions and assumptions (such as the snow size distribution *and* the ice particle concentration). The form for N_{TS} given in equation 4.1 (and in equation D.6) is an attempt to decouple the number of snow particles from the exponential character of the Fletcher relation at temperatures above about –12 ° C during the later stages of cloud evolution. This form for N_{TS} is used in all model results presented here. In addition, the snow to graupel transitions have no excess snow mass term as is used in Stephens (1979), or Cheng (1981). Note, however, that the parameterization does include a snow aggregation term and an excess snow riming term in the snow to graupel transitions (see table 2.4 and appendix D).

QUATERNARY LIBRARY

$$v_p = - \frac{130\ \Gamma(4.5)}{6} \left[D_p \right]^{0.5} \quad . \tag{D.5}$$

c) q_s - Snow

Snow crystals are assumed to be formed according to a temperature dependent function based upon Fletcher's (1969) observation that the number of ice crystals increased exponentially with temperature, i.e.,

$$N_{TS} = \max \left[O_{NS}\ e^{\beta\,(\,273.16\,-\,T\,)}\,,\ \rho_o q_s\,(\,10^6\,) \right] \quad [\ m^{-3}\ ] \quad . \tag{D.6}$$

The first term on the right hand side of equation D.6 is the form given by Fletcher (1969) for the ice particle concentration as a function of temperature, while the second term limits the maximum snow crystal mass to 0.001 g. As discussed in section 4.3.1.1, this second term is included in N_{TS} to alleviate the tendency of the basic Stephens (1979) parameterization to artifically force snow into the graupel category at temperatures above around –12 ° C. Such a situation results from limiting the maximum snow crystal mass to .001 g, *and* computing the number of snow particles by the Fletcher relation. The form for N_{TS} given in equation D.6 decouples, albeit in a simple fashion, the number of snow particles from the exponential character of the Fletcher relation at temperatures above about –12 ° C during the later stages of cloud evolution. This form for N_{TS} is used in all model results presented here.

The diameter and fall velocity for snow are computed using the expressions from Stephens (1979). Three mass regimes are considered:

$$D_c = \begin{cases} 16.286\ (\ M_s\ )^{0.5} & M_s < 10^{-10} \\ 6.072\ (\ M_s\ )^{0.5} & 10^{-10} \le M_s \le 10^{-8} \\ 1.585\ (\ M_s\ )^{.417} & M_s > 10^{-8} \end{cases} \quad , \tag{D.7}$$

for a snow particle mass

$$M_s = \frac{q_s\ \rho_o}{N_{TS}} \quad , \tag{D.8}$$

and

$$v_s = - \left[\frac{100000}{p} \right]^{0.5} \begin{cases} 304.\ D_c & M_s < 10^{-10} \\ 1250.\ D_c & 10^{-10} \le M_s \le 10^{-8} \\ 4.838\ D_c^{\,0.25} & M_s > 10^{-8} \end{cases} \quad , \tag{D.9}$$

where the initial term on the right hand side is a correction for the fall velocity at pressure p.

d) q_g - Graupel

Graupel particle diameters are computed using the Stephens (1979) mass categories: light graupel ($M_g < 10^{-6}$), heavy graupel ($1(10^{-6}) \le M_g \le 2.0(10^{-5})$, and hail ($M_g \ge 2(10^{-5})$, for

$$M_g = \frac{q_g\ \rho_o}{N_{TG}} \quad . \tag{D.10}$$

N_{TG} is the total number of graupel particles, and is held constant. The particle diameter is given by;

$$D_g = \left[\frac{q_g\ \rho_o}{\pi\ N_{TG}\ \rho_g} \right]^{0.333} \quad , \tag{D.11}$$

where ρ_g is the graupel particle density;

$$\rho_g = \begin{cases} 100 & M_g < 10^{-6} \\ 600 & 1(10^{-6}) \le M_g \le 2.0(10^{-5}) \\ 900 & M_g \ge 2(10^{-5}) \end{cases} \quad . \tag{D.12}$$

The mass weighted fall velocity is given by

$$v_g = -\Gamma \frac{(4.5)}{6}\ D_g^{\,0.5}\ \kappa_{vg} \quad , \tag{D.13}$$

where

$$\kappa_{vg} = \begin{cases} 59.4 \left[\frac{100000}{p} \right]^{0.5} & \rho_g = 100 \\ \left[\frac{4\ g\ \rho_g}{3\ c_d\ \rho_o} \right]^{0.5} & \rho_g \geq 600 \end{cases} \quad \text{(D.14)}$$

κ_{vg} for $100 < \rho_g < 600$ is the maximum of the two functions calculated in equation D.14.

Source terms

The microphysical parameterization terms discussed in chapter 2 (see figure 2.3, table 2.4 and table 2.5) are presented below. The computer code for these terms is found in subroutines *PARKES* and *PARICE* in Appendix F. Unless noted otherwise, all source terms in this section have units of kg/kg (air) s^{-1}.

D.1 Liquid phase processes

AUTO - autoconversion

This term computes the amount of cloud water, q_c, converted to precipitation water, q_p, via coalescence of cloud droplets. The form used is the Kessler (1969) formula;

$$\text{AUTO} = \begin{cases} \kappa\ (\ q_c - q_{co}\ ) & q_c > q_{co} \\ 0 & q_c < q_{co} \end{cases} , \quad \text{(D.15)}$$

where $\kappa = 10^{-3}\ s^{-1}$ and q_{co} is usually specified in terms of a limiting concentration m^{-3}, i.e., around $\frac{.001}{\rho_o}$kg/kg.

Cotton (1972) computed a time dependent form for the autoconversion consistent with the Kessler (1969) precipitation collection term. Unfortunately, this autoconversion relation was a rather complicated numerical fit to the residual of the stochastic collection equation and the precipitation collection term for two droplet concentrations

(100 cm^{-3} and 300 cm^{-3}). In addition, while the time dependence of autoconversion is well defined for a Lagrangian parcel, a direct extension to an Eulerian grid is not straightforward.

At low droplet concentrations and relatively high liquid water values, the Kessler autoconversion term is a reasonable match for the time average values of the rates computed by Cotton. However, the Kessler form overpredicts for low liquid water concentrations. I have used a modified form of the Kessler form for autoconversion which incorporates, albeit in an intuitive fashion, the main points of the Cotton formulation, i.e., the dependence on droplet concentration, and an allowance for the time dependence of initial droplet coalescence. The Kessler form is used rather than the autoconversion formula derived from the stochastic collection equation by Berry (1968). This is because, as discussed by Cotton (1972), Berry's formulation apparently does not distinguish between autoconversion and accretion and so yields an excessive autoconversion rate. The Orville and Kopp (1977) modified version of the Berry equation also gives autoconversion rates larger than those computed by Cotton (1972) from the stochastic collection equation.

While κ and q_{co} may be readily determined for maritime ($\approx$ 100 cm^{-3}) and moderate continental droplet concentrations ($\approx$ 300 cm^{-3}), the actual values of κ and q_{co} appropriate for use in midlatitude continental clouds are not well established. Aircraft observations of these clouds (e.g., Dye et al. (1986), Jensen (1985), or Dye et al. (1983)) clearly show instances in which the cloud liquid water content is above 4 g/kg (i.e., $\approx$2.5 g m^{-3} at 500 mb), and no precipitation or drizzle size drops are observed. These clouds typically have droplet concentrations around 600 – 1200 cm^{-3}. In such clouds, use of $q_{co} \approx 1$ g/kg is inappropriate and will lead to precipitation formation via coalescence rather than through the ice phase processes.

I have modified the Kessler autoconversion form to include a dependence upon droplet concentration, and to crudely approximate the time dependent behavior observed in the numerical solutions presented by Cotton (1972). The dependence upon droplet concentration is estimated from the observation of no coalescence at high concentrations, and from the time-averaged rates at low-to-moderate droplet concentrations computed from the equations presented in Cotton (1972). (Note that h′ and a′ given in equation 19 of the above paper are inconsistent with the values presented in

LIBRARY

figure 4 of that paper; $h' = e^{8.0}\ m^{-1.75}$ and $a' = 2.847\ (10^4) - 1.024\ (10^4)$ m offer better agreement with the data presented in figure 4.) To reduce the impact of the autoconversion term for clouds with large droplet concentrations, q_{co} is calculated using a simple linear expression;

$$q_{co} = 7.5\ (10^{-6})\ N_{cc} - 2.5\ (10^{-4}) \quad [\ kg/kg\ ] \quad , \tag{D.16}$$

where N_{cc} is the droplet concentration cm^{-3}. Note that in this expression the limit on the autoconversion in determined by the liquid water mixing ratio and not the liquid water concentration (as is usually the case). q_{co} written this way provides a means to distinguish a low liquid water situation ($\rho_o q_l \leq 1.5\ g/m^{-3}$) in a maritime or moderate continental cloud from the case of high droplet concentration, high mixing ratio, as may occur well above cloudbase in a continental cloud. In both instances, the total liquid water concentration (kg/m^{-3}) may be the same due to the air density dependence upon height in the atmosphere. However, while autoconversion in probable for the maritime or moderate continental case, given sufficent time, it is unlikely to happen in the continental case. Even though these cases are degenerate with respect to total liquid water (kg/m^{-3}), a clear distinction exists with respect to liquid water mixing ratio (kg/kg (air)).

κ has not been modified.

COLL - collection

This term represents the collection of cloud droplets by precipitation (accretion). Collection is determined using a continuous collection model and a Marshall-Palmer (1948) distribution for the precipitation. From these one may show;

$$COLL = \frac{130\pi}{4}\,\Gamma(3.5)\left[\frac{6}{\rho_w \pi \Gamma(4)}\right]^{.875} q_c\, N_{or}{}^{.125} E_{pc}\,(\rho_o q_p)^{.875} \quad , \tag{D.17}$$

where N_{or} is the intercept of the precipitation distribution, i.e., $N(D) = N_{or} \exp(-\lambda D)$ and E_{pc} is the collection efficiency modeled on potential flow about the raindrop;

$$E_{pc} = \frac{S_{tk}{}^2}{(S_{tk} + 0.5)^2} \tag{D.18}$$

where (Stephens, 1979),

$$S_{tk} = \frac{1000\ (\pi \rho_w N_{or})^{0.25}\ |\ v_p\ |\ D_c{}^2}{9.0\ \mu\ (\rho_o q_p)^{0.25}} \quad . \tag{D.19}$$

μ is the dynamic viscosity of air (see equation D.47), and v_p is the precipitation fall velocity (equation D.5).

PEVAP - the evaporation of precipitation

Evaporation of precipitation drops is allowed to occur in subsaturated regions using;

$$PEVAP = 2\pi\ N_{or}\left[1 - \frac{q_v}{q_{sp}}\right] G_w(T,p)\left[1 + 0.265\ \frac{\rho_o\ |\ v_p\ |\ D_p}{\mu}\right]\frac{1}{\rho_o} \quad , \tag{D.20}$$

for a saturation vapor mixing ratio q_{sp}. $G_w(T,p)$ is given by;

$$G_w(T,p) = \left[\frac{L_v{}^2}{K_T\ R_v\ T^2} + \frac{R_v\ T}{\psi\ e_w(T)}\right]^{-1} \quad , \tag{D.21}$$

where K_T is the thermal conductivity of air (see equation D.49), R_v is the gas constant for water vapor, and ψ is the molecular diffusivity of air (see equation D.46). PEVAP is derived in a manner analogous to the Stephens (1979) derivation of the ice vapor deposition terms.

D.2 Ice phase processes

Three regimes exist for the ice phase processes; at $T = 273.16$ K, most ice terms become activated, and water and ice co-exist in the cloud until the homogeneous nucleation temperature of liquid water, i.e., $\approx -37\,^\circ$C, is reached. The ice phase terms will be considered according to temperature.

D.2.1 $T < 273.16$ K

Since several terms are set to zero at the homogeneous nucleation temperature, any change resulting when the temperature falls below this threshold will be noted individually.

LIBRARY

SINIT - snow initiation

For a positive supersaturation with respect to ice (Stephens, 1979);

$$\text{SINIT} = \frac{1}{\delta t_m} \min\left(\frac{1}{\rho_o} M_{xtl} N_{TS} , (q_v - q_{si})\right) \quad , \tag{D.22}$$

where $M_{xtl} = 1.0(10^{-12})$ kg is the individual ice crystal mass, and N_{TS} is the total number of snow particles (equation D.6). q_{si} is the saturation vapor mixing ratio with respect to ice. δt_m is the time step for the microphysical processes, and is set to

$$\delta t_m = \max (10 , \delta t) \quad [\text{seconds}] \quad , \tag{D.23}$$

for δt defined in equation 2.28. The second term in equation D.22 insures that the ice crystal production is limited by the available water vapor. In addition, homogeneous nucleation is assumed for $N_{TS} \geq 10^7$. This occurs at about – 37 °C.

If the air is subsaturated with respect to ice, SINIT = 0.0.

SRME - riming of snow

Riming of snow is calculated using an equation similar to the collection term defined in equation D.3. Collection efficencies calculated by Pitter (1977) and Schlamp et al., (1975) indicate that droplets with diameters $< 10\ \mu$ have extremely low probability of being collected. Therefore, if the cloud water diameter, $D_c > 0.00001$, and $q_s > 5(10^{-9})$;

$$\text{SRME} = \min \left[\frac{N_{TS}\, \pi\, D_s^2\, |\, v_s\, |\, E_{sc}\, q_{li}}{4} , \frac{q_{li}}{\delta t_m} \right] \quad , \tag{D.24}$$

where D_s is the snow particle diameter (equation D.7), v_s is the snow fall velocity (equation D.9), and E_{sc} is the collection efficiency calculated identically to E_{pc} in equation D.18, with

$$S_{tk} = \frac{1000\, D_c^2\, |\, v_s\, |}{9\, \mu\, D_s} \quad . \tag{D.25}$$

q_{li} is the amount of water remaining after autoconversion and collection, i.e.,

$$q_{li} = q_c - (\text{AUTO} - \text{COLL})\, \delta t_m \quad . \tag{D.26}$$

If q_c or q_s are outside the bounds noted above, or if $N_{TS} \geq 10^7$, SRME = 0.

CSFZ - freezing of cloud drops to snow

This term accounts for the homogeneous freezing of cloud drops;

$$\text{CSFZ} = \begin{cases} \dfrac{q_c}{\delta t_m} & N_{TS} \geq 10^7 \\ 0 & N_{TS} < 10^7 \end{cases} \quad , \tag{D.27}$$

where N_{TS} is the total number of snow particles calculated from equation D.6.

SMELT - melting of snow

For T < 273.16K no melting of snow is allowed, i.e., SMELT = 0.

PGFZ - freezing of precipitation

Freezing of precipitation via snow crystal contact is calculated using the expression of Stephens (1979);

$$\text{PGFZ} = \pi\, N_{TS}\, D_p^2\, |\, 6.1875\, v_p - v_s\, |\, q_p \quad . \tag{D.28}$$

If $N_{TS} \geq 10^7$, $\text{PGFZ} = \dfrac{q_p}{\delta t_m}$.

GRME - riming of graupel

Riming is calculated whenever $q_g > 5(10^{-9})$ and $D_c > 10^{-5}$. The expression is similar to that used for snow riming;

$$\text{GRME} = \min \left[\frac{3\pi}{7} E_{gc}\, N_{TG}\, |\, v_g\, |\, D_g^2\, q_{li} , \left(\frac{q_{li}}{\delta t_m} - \text{SRME} \right) \right] \quad , \tag{D.29}$$

where the collection efficiency, E_{gc}, is calculated using equation D.18 with

$$S_{tk} = \frac{405\, D_c^2\, |\, v_g\, |}{9\, \mu\, D_g} \quad . \tag{D.30}$$

N_{TG} is the total number of graupel particles, and is fixed in this parameterization, D_g is the graupel particle diameter (equation D.11), v_g is the graupel fall velocity (equation D.13), and q_{li} is defined in equation D.26. The second term in equation D.29 limits

graupel riming to the available cloud water.

GRME = 0 for $N_{TS} \geq 10^7$.

GMELT - melting of graupel

GMELT = 0 for T < 273.16K.

PGCOL - graupel collection of precipitation

The collection term is from Stephens, (1979)

$$PGCOL = \pi \mid v_p - v_g \mid q_p N_{TG} \left[0.5 D_g^2 + 2 D_p D_g + 5 D_p^2 \right] \quad . \tag{D.31}$$

PGCOL = 0 for $N_{TS} \geq 10^7$.

SGCOL - graupel collection of snow

The snow collection term is from Cheng, (1981), and uses a dry collection efficiency of 0.1;

$$SGCOL = \frac{3\pi}{7} 0.10 N_{TG} \mid v_g \mid q_s D_g^2 \quad . \tag{D.32}$$

SGCON - conversion of snow to graupel

$$SGCON = \left[0.01 e^{0.07 (T - 273.16)} \max (0 , q_s - 0.0006) \right] + \left[SRME + 0.001 \frac{q_s}{\delta t_m} \right] Heavi(q_s - 0.0001) Heavi(q_l - 0.001) \quad . \tag{D.33}$$

The first term on the right hand side is the aggregation term of Cheng (1981). The second term limits the snow growth rate via riming in high liquid water situations (Ruttledge and Hobbs, 1983). Heavi is the Heaviside function centered at 0.

GPSHD - precipitation shedding due to wet growth of graupel

Wet growth is calculated using the expression of Cheng, (1981). The rate at which water can freeze on the graupel is;

$$PGWG = \max \left[0 , \frac{-2 \pi N_{TG} F_g D_g \rho_o L_v \psi (q_v - q_{sw}) + K_T (T - 273.16)}{\rho_o L_f + c_w (T - 273.16)} + 10 SGCOL \left\{ 1 - \frac{c_i (T - 273.16)}{L_f + c_w (T - 273.16)} \right\} \right] , \tag{D.34}$$

where F_g is the ventilation factor for graupel, i.e.;

$$F_g = 1 + 0.265 \left[\frac{\rho_o D_g \mid v_g \mid}{\mu} \right]^{0.5} , \tag{D.35}$$

and q_{sw} is the vapor saturation mixing ratio over liquid water. The factor of 10 multiplying SGCOL increases the collection effeciency from 0.1 to 1.0 for wet growth.

If GRME + PGCOL - PGWG > 0, i.e., more water is collected than can freeze, wet growth is assumed and the excess is shed via precipitation.

$$SGCOL = 10 \times SGCOL \quad , \tag{D.36}$$

to change the snow collection efficiency to 1.0. The precipitation water shed is

$$GPSHD = GRME + PGCOL - PGWG + SGCOL \quad . \tag{D.37}$$

VSDEP - vapor deposition on snow

This function is from Stephens (1979). The crystals are assumed hexagonal plates. Once they are large enough, mass riming becomes important and the heat release is included in the vapor deposition rate.

$$VSDEP = \frac{4 D_s SS N_{TS} G_i(T,p) F_s}{\rho_o} - \frac{L_f^2 G_i(T,p) SRME}{R_v K_T T^2} \quad , \tag{D.38}$$

where SS is the vapor supersaturation mixing ratio with respect to ice;

$$SS = \left[\frac{q_v}{q_{si}} - 1 \right] \quad , \tag{D.39}$$

and

$$G_i(T,p) = \left[\frac{L_f^2}{K_T R_v T^2} + \frac{R_v T}{\psi e_i(T)} \right]^{-1} \quad . \tag{D.40}$$

F_s is the ventilation factor for snow;

$$F_s = 1 + 0.229 \left[\frac{\rho_o D_s \mid v_s \mid}{\mu} \right]^{0.5} \quad . \tag{D.41}$$

VGDEP - vapor deposition on graupel

$$VGDEP = 2\pi \, SS \, N_{TG} \, D_g \, G_i(T,p) \, F_g \frac{1}{\rho_o} \quad , \tag{D.42}$$

where the supersaturation SS is defined in equation D.39, and the ventilation factor F_g is given in equation D.35.

D.2.2 $T \geq 273.16$

SINIT, SRME, CSFZ, PGFZ, SGCOL, SGCON, VSDEP and VGDEP are set to 0. The remaining terms are calculated in the following manner;

SMELT - melting of snow

All snow is assumed to melt instantly;

$$SMELT = \frac{q_s}{\delta t_m} \quad . \tag{D.43}$$

GRME - riming of graupel

Riming of graupel is calculated using equations D.29 and D.30.

PGCOL - graupel collection of precipitation

This term is calculated using equation D.20.

GMELT - melting of graupel

Melting of graupel is computed using the expression derived by Stephens (1979);

$$GMELT = \frac{1}{L_f} \left[\left(\frac{K_T \, (T - 273.16)}{\rho_o} + 2\pi \, D_g \, N_{TG} \, F_g \, L_v \, \psi \, (q_v - q_{sw}) \right) \right. \tag{D.44}$$
$$\left. + \, (c_w \, (T - 273.16) \, (GRME + PGCOL)) \right] \quad .$$

The first term on the right hand side is the heat conducted from the surrounding air and from water vapor condensing on the particle with the particle temperature set to 273.16 K. The second term is the heat transfer to the graupel from liquid water accreted onto the particle from riming of cloud droplets and collection of precipitation drops.

GPSHD - precipitation shedding due to wet growth of graupel

All water collected by graupel is assumed to shed, i.e.;

$$GPSHD = GRME + PGCOL \tag{D.45}$$

D.3 Computation of ψ, K_T and μ

The molecular diffusivity, ψ, and the dynamic viscosity, μ, are computed using the expressions from Cheng (1981);

$$\psi = 2.26(10^{-5}) \frac{100000.}{p} \left[\frac{T}{273.16} \right]^{1.81} \quad , \tag{D.46}$$

and

$$\mu = \frac{7.616(10^{-3})}{T + 120} \left[\frac{T}{296.16} \right]^{1.50} \quad . \tag{D.47}$$

The thermal conductivity, K_T is calculated using the expression (Rogers, 1979)

$$K_T = A \, \mu \quad , \tag{D.48}$$

where the constant A has been set here to yield the value of K_T at $T = 273.16$ found in the CRC Handbook, vol. 52 (1972).

$$K_T = 1323.18 \, \mu \quad . \tag{D.49}$$

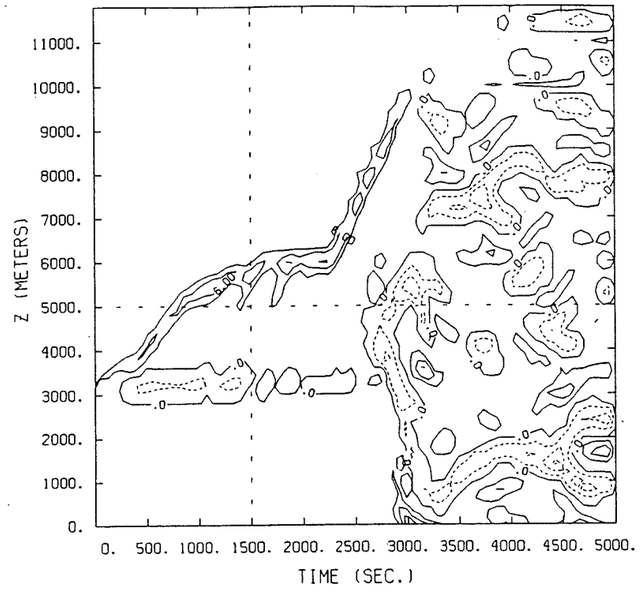
This expression is within 5% of the CRC values for $233 \text{ K} \leq T \leq 293 \text{ K}$.

APPENDIX E

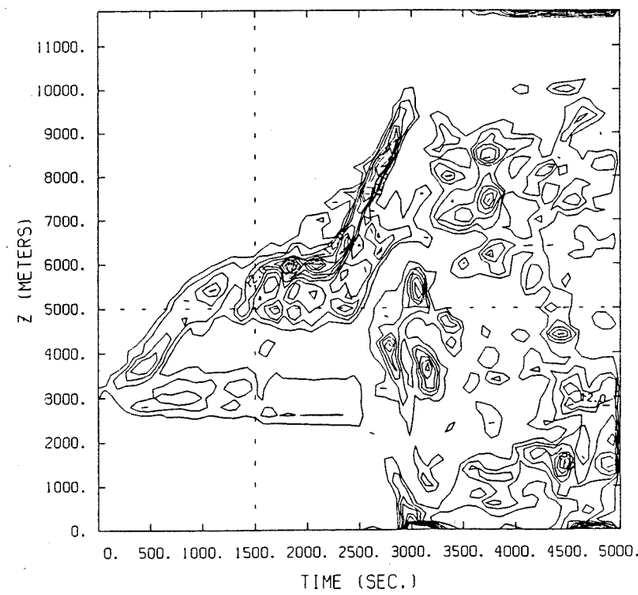
Supplemental Figures

I think it important that the interested reader have the opportunity to examine the model output in some detail. However, including all the potentially relevant figures in the main text of this dissertation would have been too disruptive for most readers, therefore this appendix contains incidental figures upon which much of the discussion in chapters 3 through 6 was based.

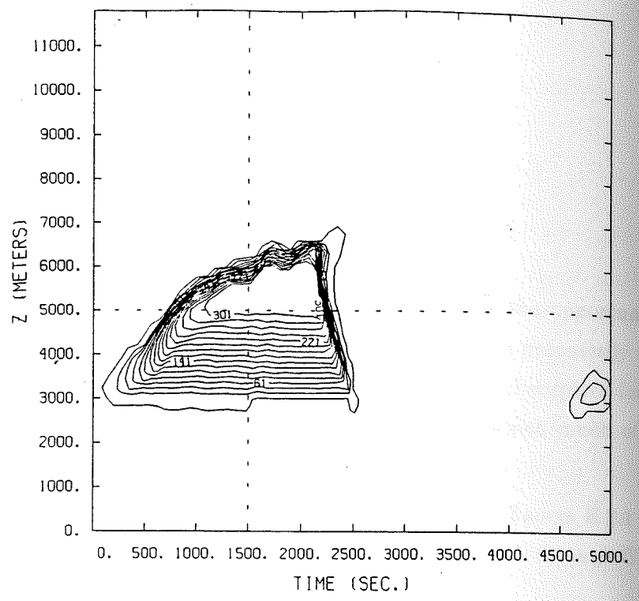
The figures E.1.1 through E.1.23 are from Chapter 3, figures E.2.1 through E.2.27 are relevant to the discussion in Chapter 4, and figures E.3.1 through E.3.5 are used in Chapter 6.



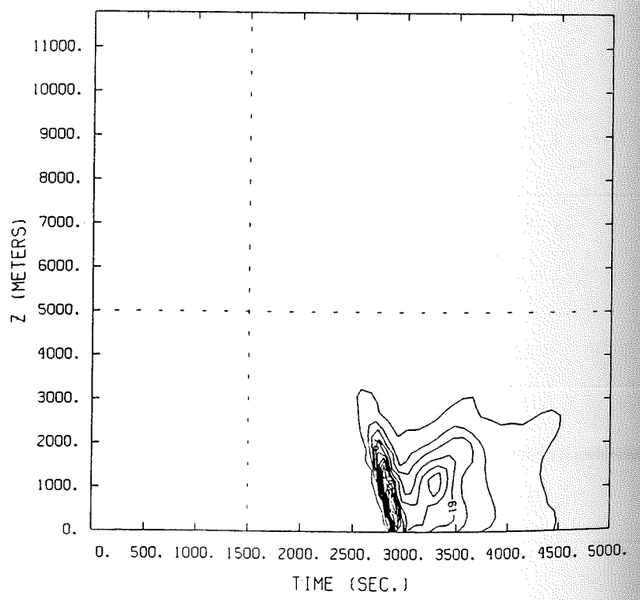
E.1.1 Contour plot of the entrainment velocity in the inner cloud region for the July 19 base run. Contour interval is 3 ms^{-1}; entrainment is shown as dotted contours, detrainment as solid contours.



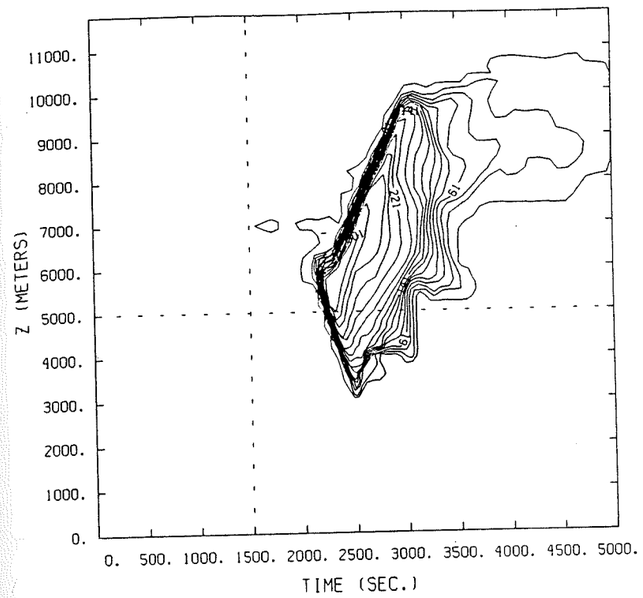
E.1.2 Contour plot of the turbulent kinetic energy in the inner cloud region. Contours are from 0 to 36 J/kg; the contour interval is 3 J/kg.



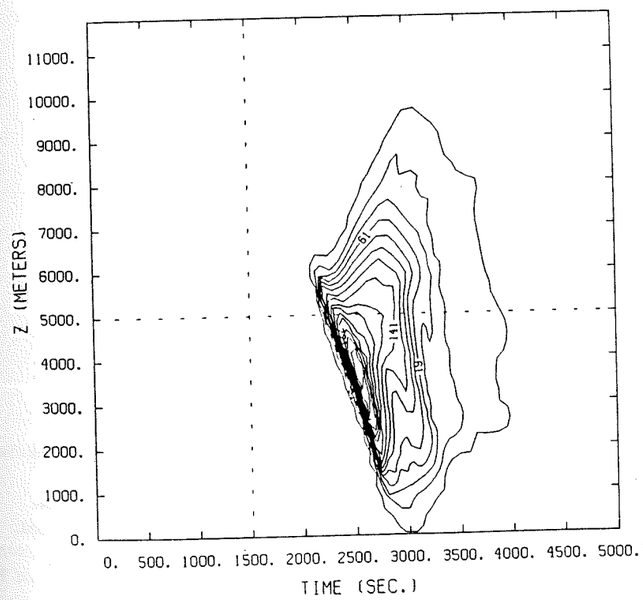
E.1.3 Contour plor of the cloud droplet mixing ratio for the inner cloud region. Contours are from 0.01 to 3.01 g/kg at 0.2 g/kg intervals. The labels are scaled by 100.



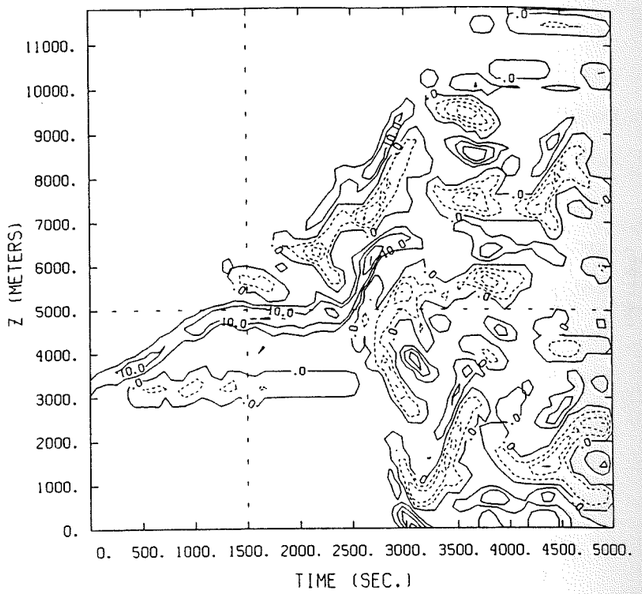
E.1.4 Contour plot of the precipitation mixing ratio for the inner region. Contours are from 0.01 to 3.01 g/kg at 0.2 g/kg intervals. The labels are scaled by 100.



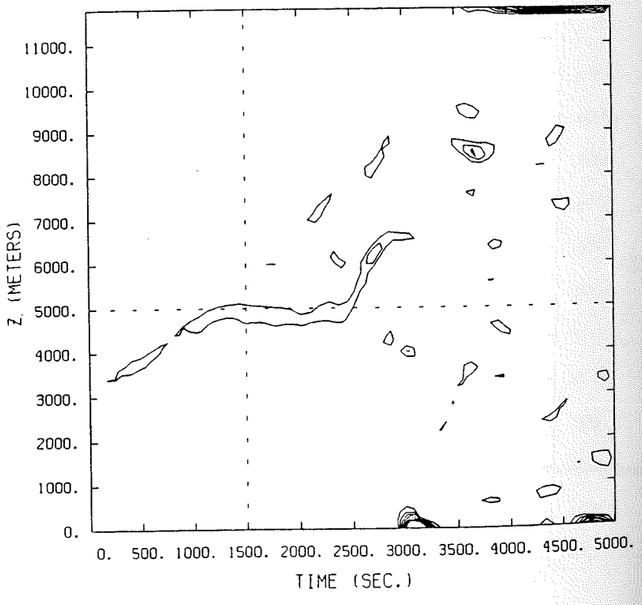
E.1.5 Contour plot of the snow mixing ratio for the inner region. Contours are from 0.01 to 3.01 g/kg at 0.2 g/kg intervals. The labels are scaled by 100.



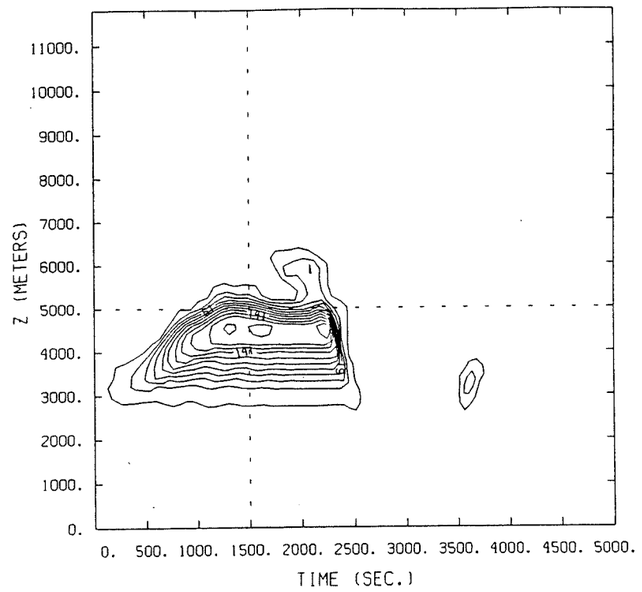
E.1.6 Contour plot of the graupel mixing ratio for the inner region. Contours are from 0.01 to 3.01 g/kg at 0.2 g/kg intervals. The labels are scaled by 100.



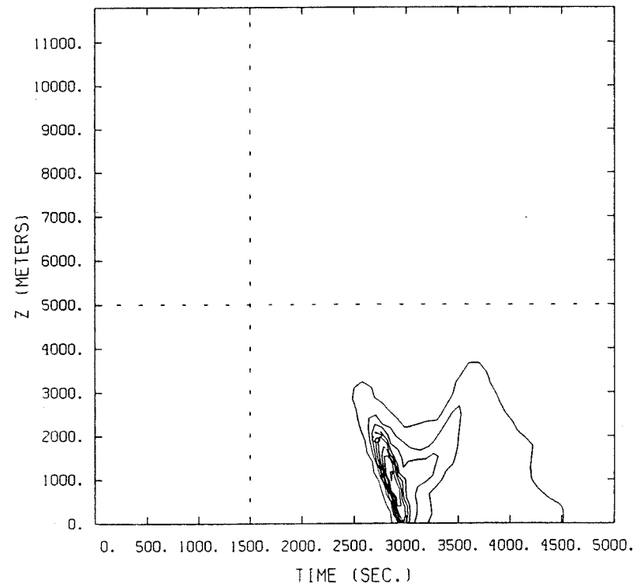
E.1.7 Contour plot of the entrainment velocity in the outer cloud region for the July 19 base run. Contour interval is 5 ms^{-1}; entrainment is shown as dotted contours, detrainment as solid contours.



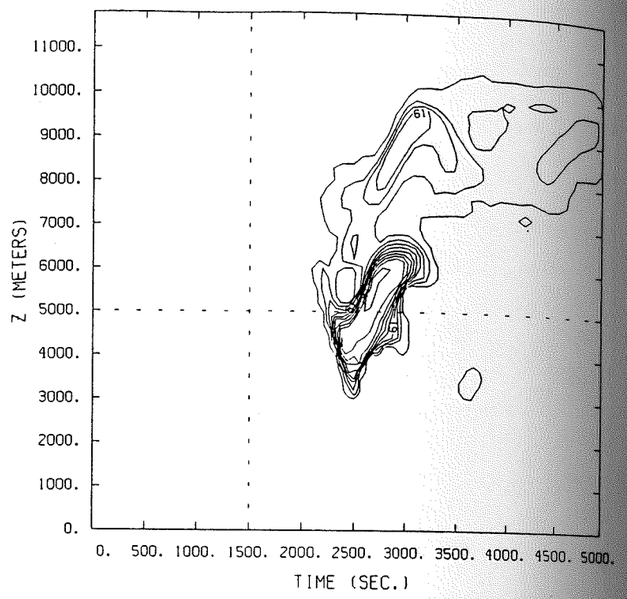
E.1.8 Contour plot of the turbulent kinetic energy in the outer cloud region. Contours are from 0 to 80 J/kg; the contour interval is 8 J/kg.



E.1.9 Contour plot of the cloud droplet mixing ratio for the outer cloud region. Contours are from 0.01 to 3.01 g/kg at 0.2 g/kg intervals. The labels are scaled by 100.

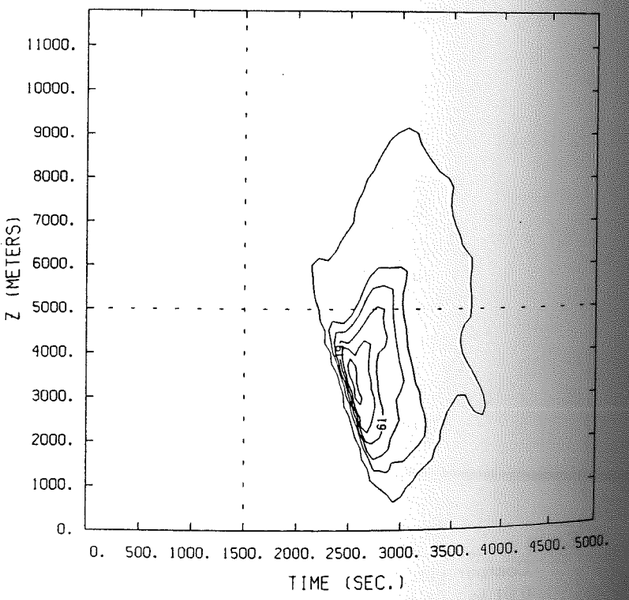


E.1.10 Contour plot of the precipitation mixing ratio for the outer region. Contours are from 0.01 to 3.01 g/kg at 0.2 g/kg intervals. The labels are scaled by 100.



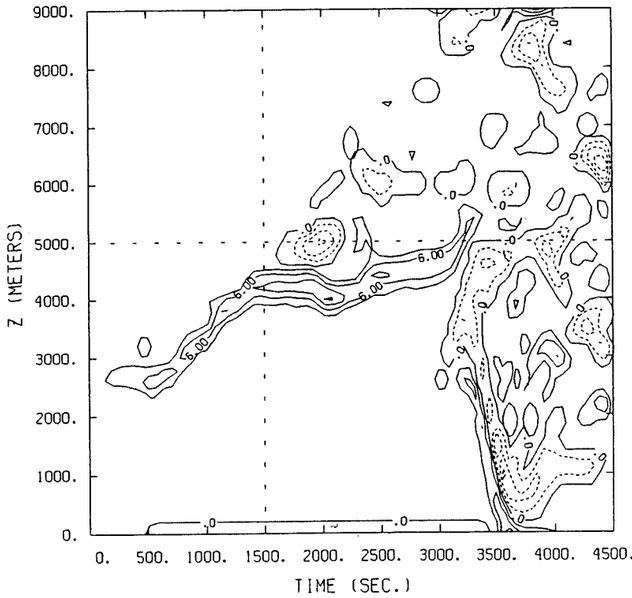
E.1.11

Contour plot of the snow mixing ratio for the outer region. Contours are from 0.01 to 3.01 g/kg at 0.2 g/kg intervals. The labels are scaled by 100.

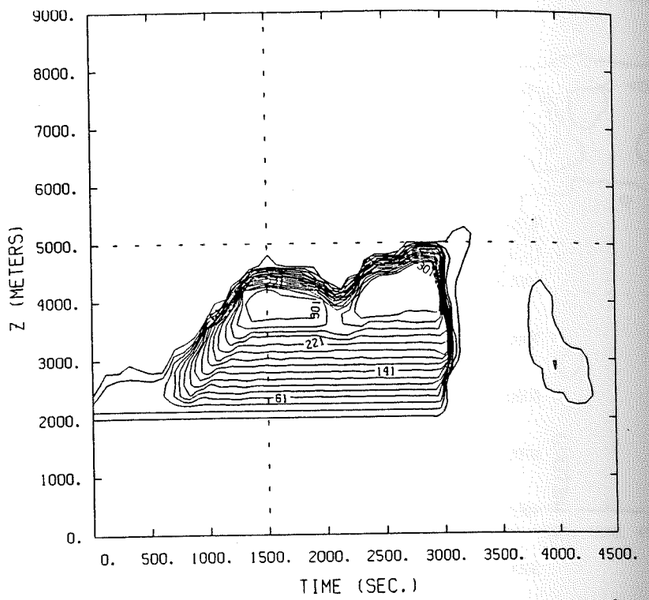


E.1.12

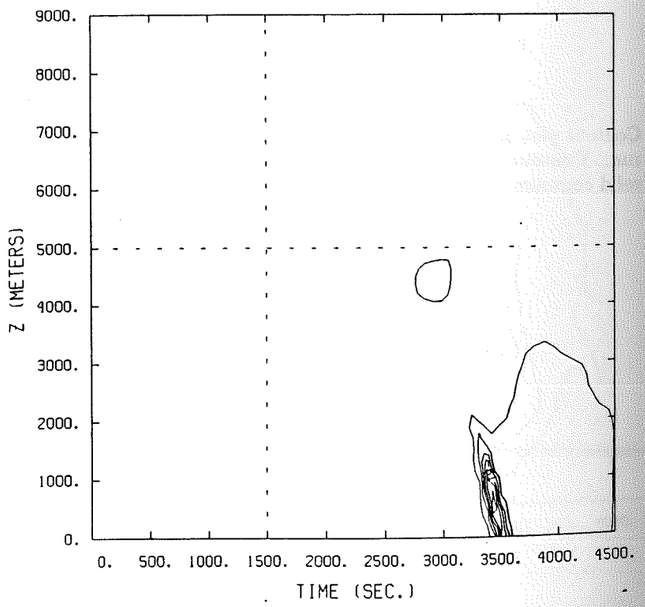
Contour plot of the graupel mixing ratio for the outer region. Contours are from 0.01 to 3.01 g/kg at 0.2 g/kg intervals. The labels are scaled by 100.



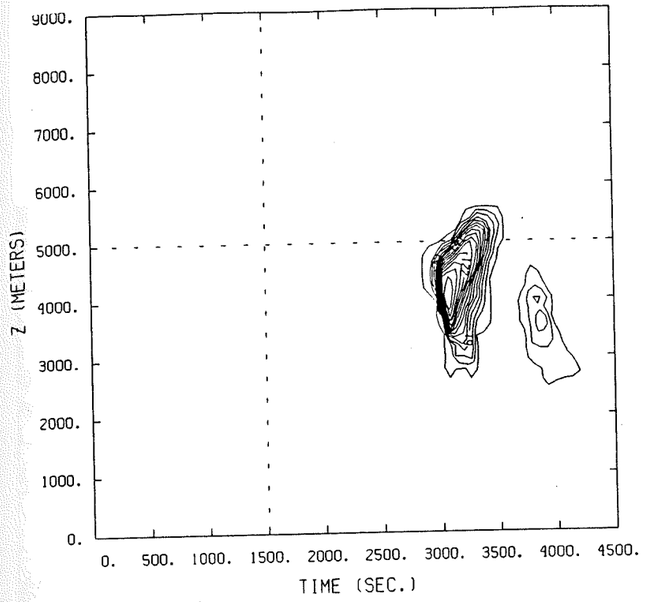
E.1.13 Contour plot of the entrainment velocity in the inner cloud region for the August 6 base run. Contour interval is 3 ms^{-1}; entrainment is for dotted contours, detrainment for solid contours.



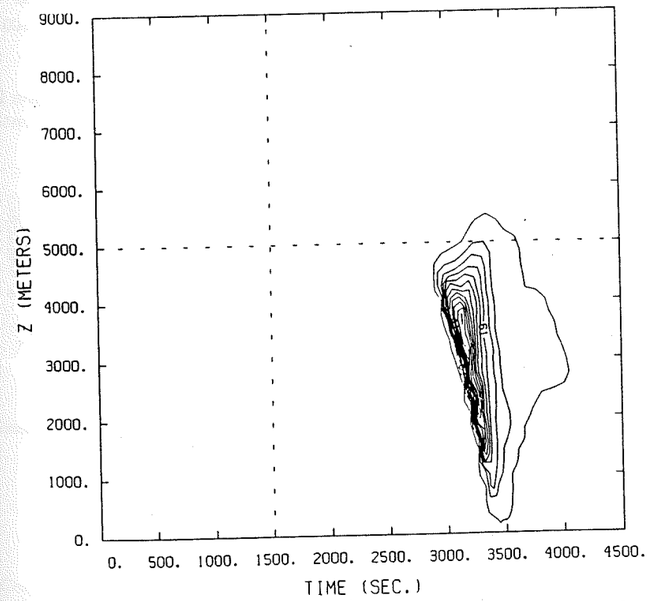
E.1.14 Contour plot of the cloud droplet mixing ratio for the inner cloud region. Contours are from 0.01 to 3.01 g/kg at 0.2 g/kg intervals. The labels are scaled by 100.



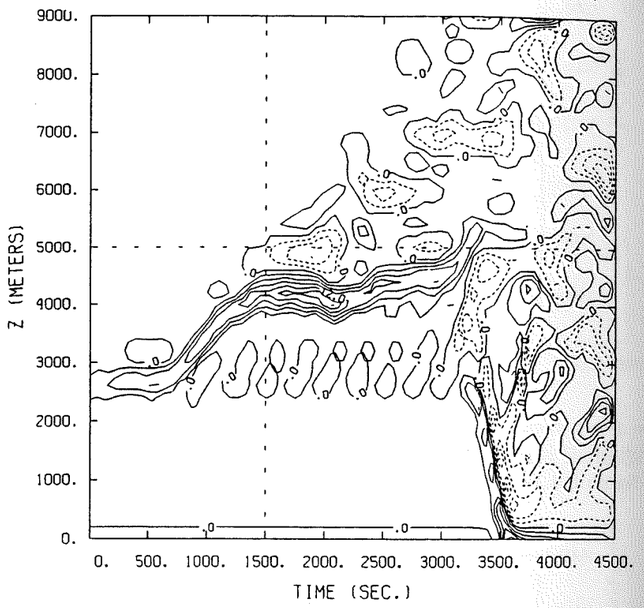
E.1.15 Contour plot of the precipitation mixing ratio for the inner region. Contours and scaling as in figure E.1.14



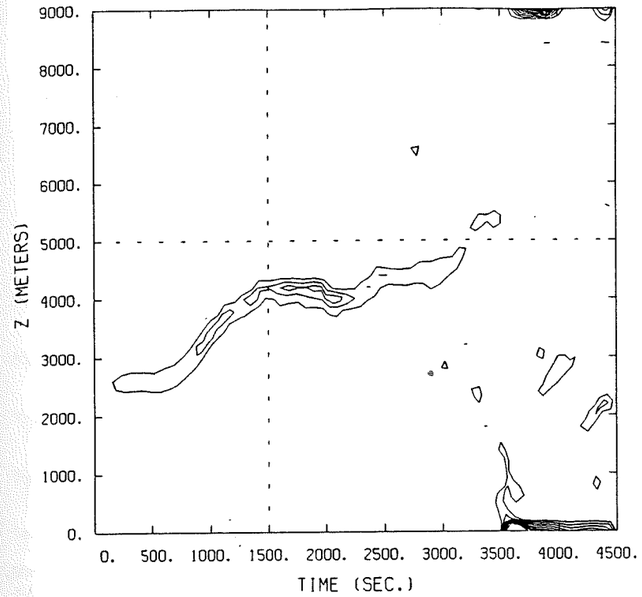
E.1.16 Contour plot of the snow mixing ratio for the inner region. Contours and scaling as in figure E.1.14



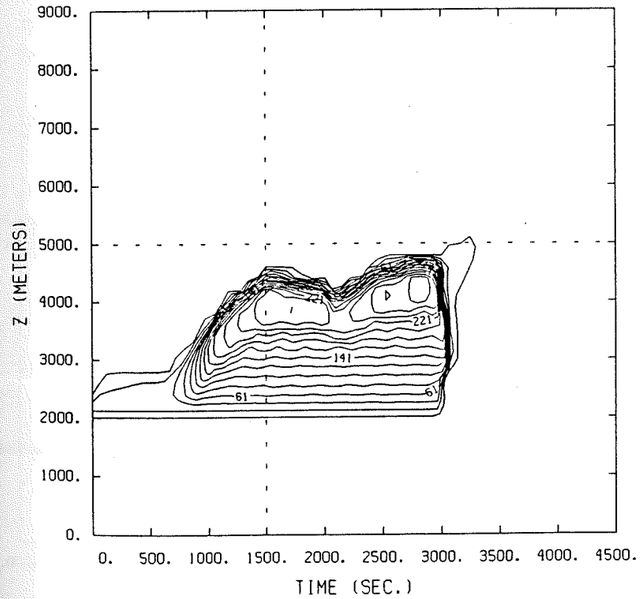
E.1.17 Contour plot of the graupel mixing ratio for the inner region. Contours and scaling as in figure E.1.14



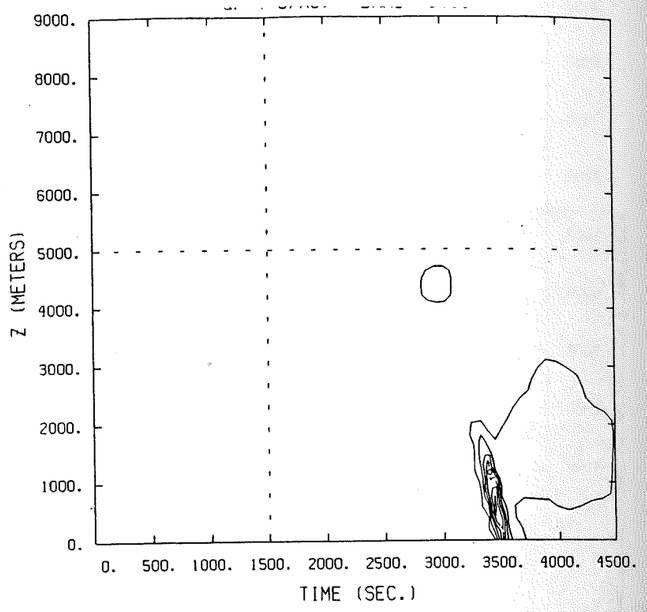
E.1.18 Contour plot of the entrainment velocity in the outer cloud region for the August 6 run. Contour interval is 3 ms^{-1}; entrainment is shown as dotted contours, detrainment as solid contours.



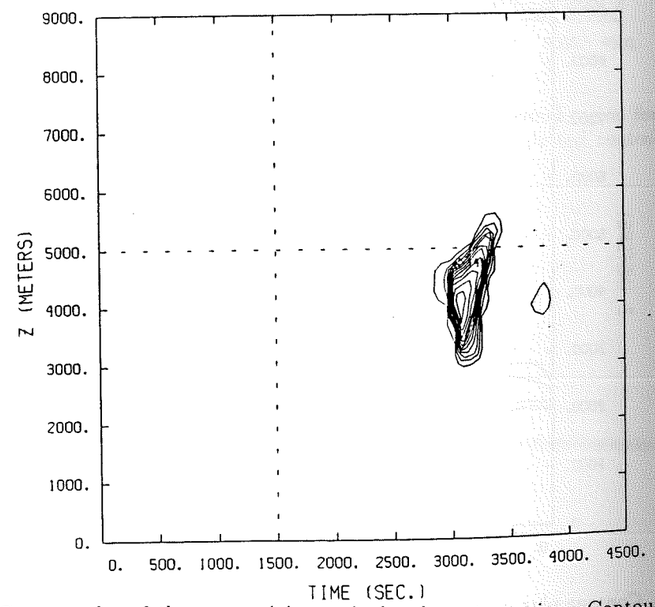
E.1.19 Contour plot of the turbulent kinetic energy in the outer cloud region. Contours are from 0 to 10 J/kg; the contour interval is 1 J/kg.



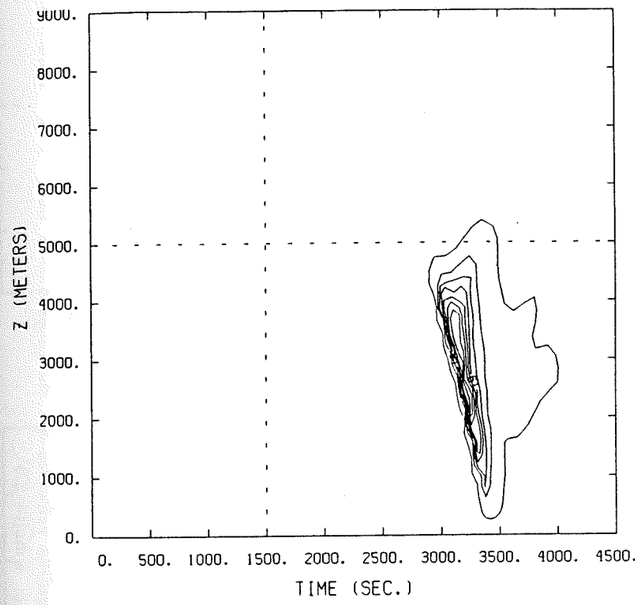
E.1.20 Contour plor of the cloud droplet mixing ratio for the outer cloud region. Contours and scaling as in figure E.1.14



E.1.21 Contour plot of the precipitation mixing ratio for the outer region. Contours and scaling as in figure E.1.14

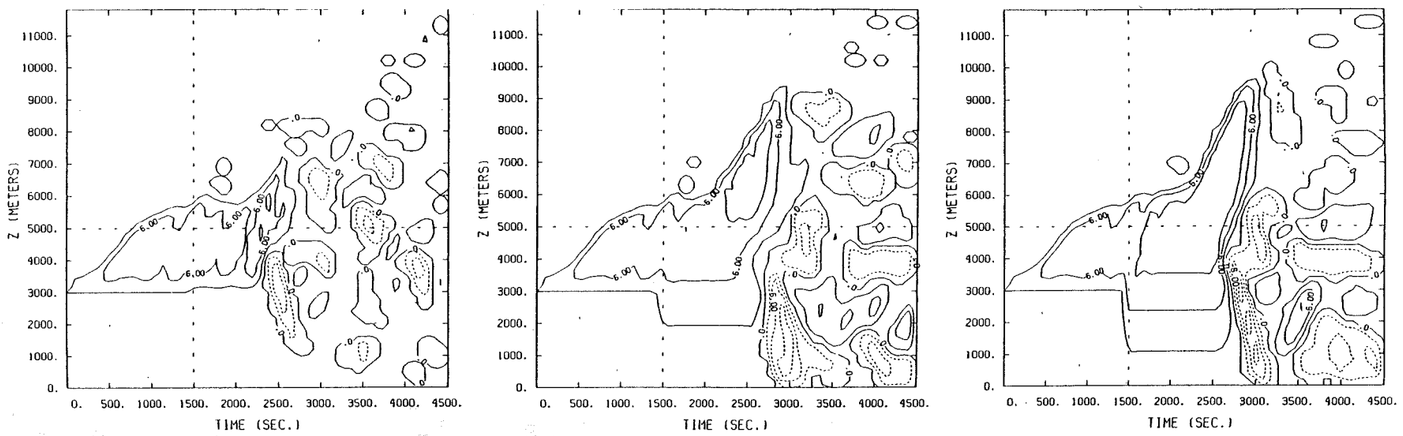


E.1.22 Contour plot of the snow mixing ratio for the outer region. Contours and scaling as in figure E.1.14

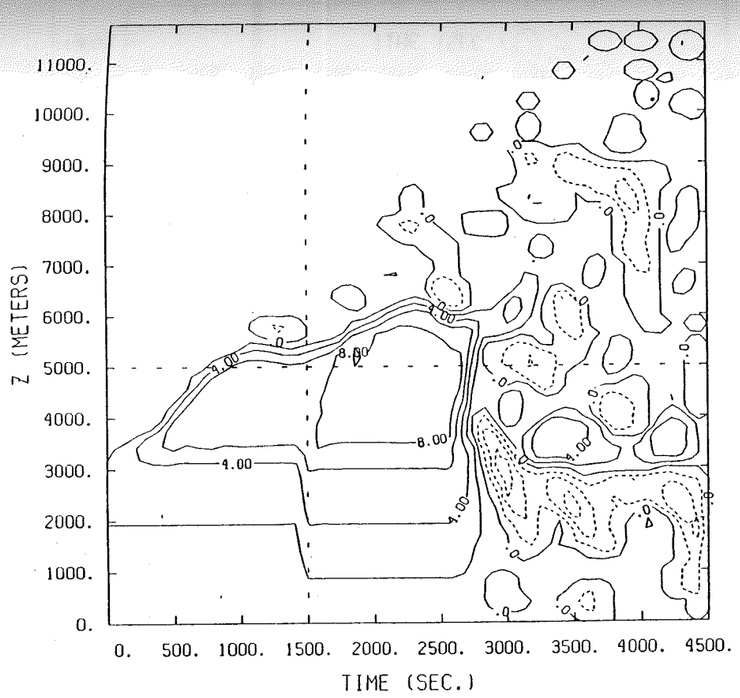


E.1.23 Contour plot of the graupel mixing ratio for the outer region. Contours and scaling as in figure E.1.14

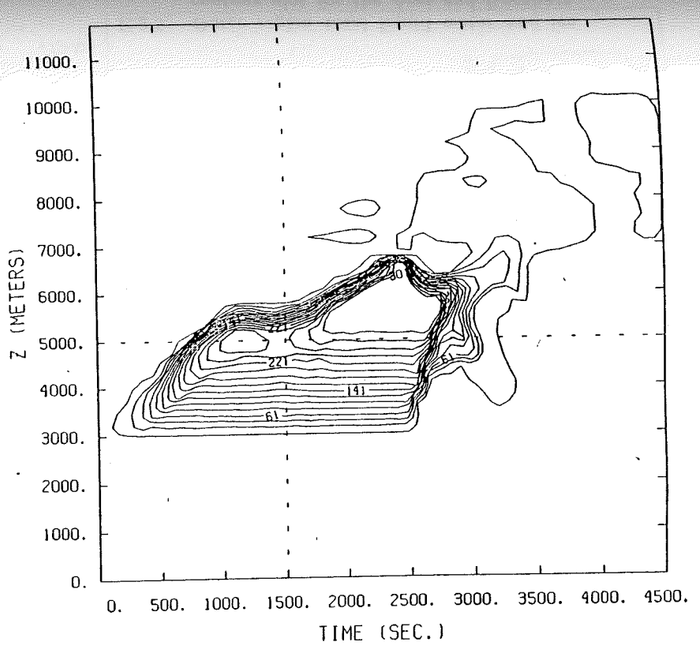
QUATERNARY LIBRARY



E.2.1 Inner region vertical velocity contour plots for a variation in wmult. Contour intervals are 3 ms^{-1}; negative velocities are shown as dashed lines. 4.1a: wmult=0.5, 4.1b: wmult=1.5, 4.1c: wmult=2.5



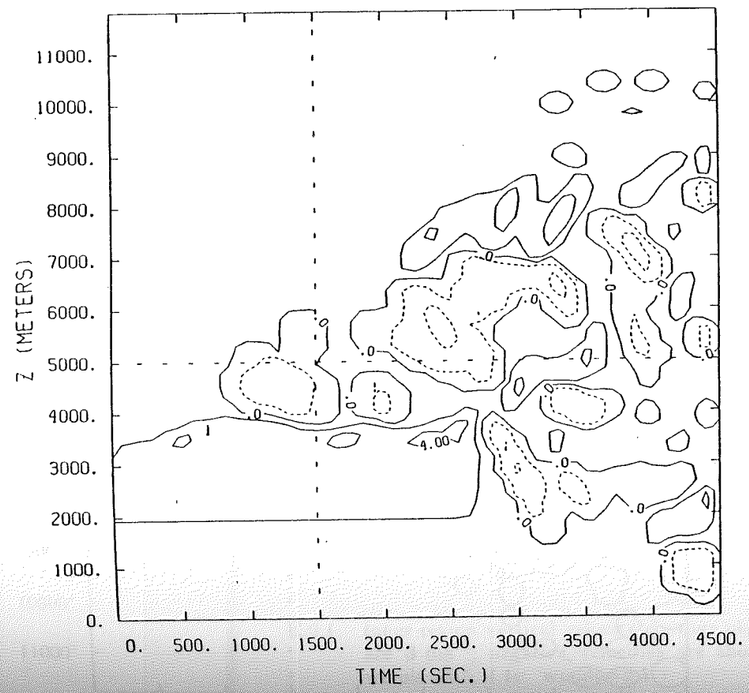
E.2.2 Inner region vertical velocity contour plot for the Asai and Kasahara (1967) entrainment formulation. Contour interval is 2 ms^{-1}.



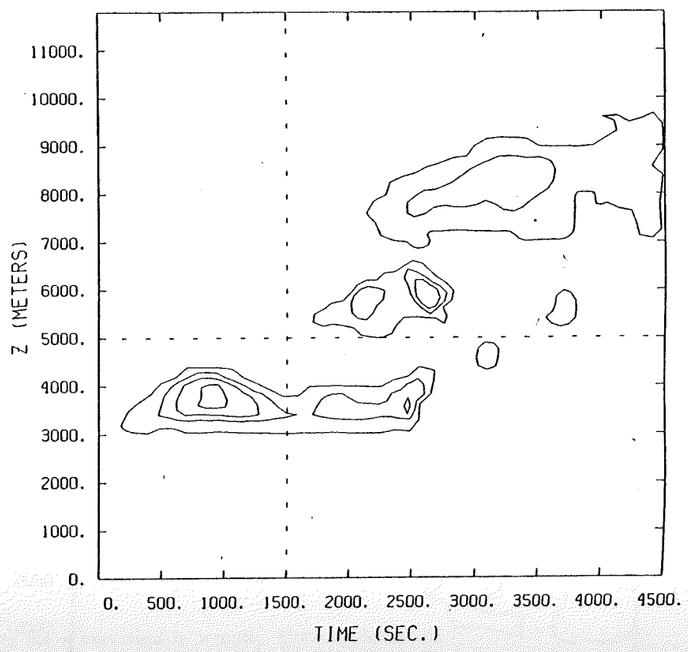
E.2.3 Inner region total liquid water and ice contour plot for the Asai and Kasahara (1967) entrainment formulation. Contours from 0.01 to 3.01 g/kg are plotted; the interval is 0.2 g/kg.

QUATERNARY LIBRARY

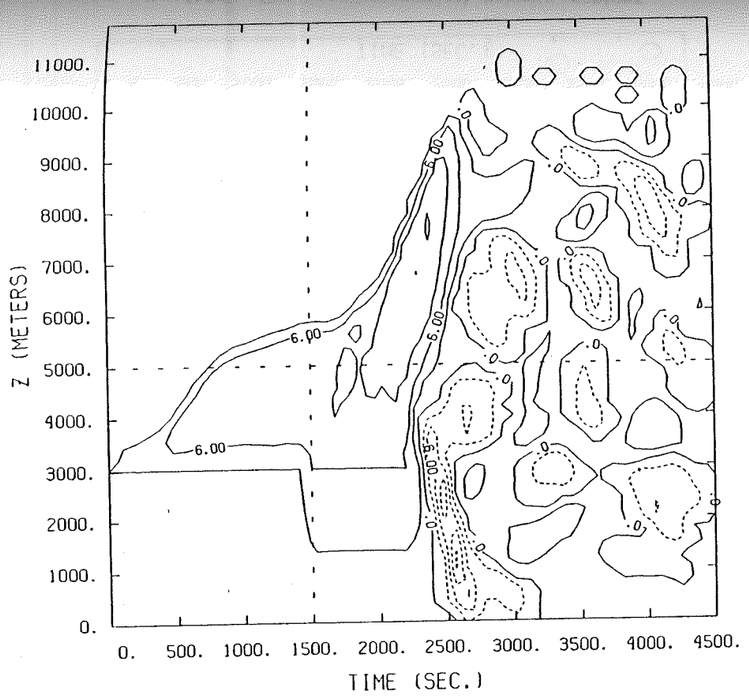
QUATERNARY LIBRARY



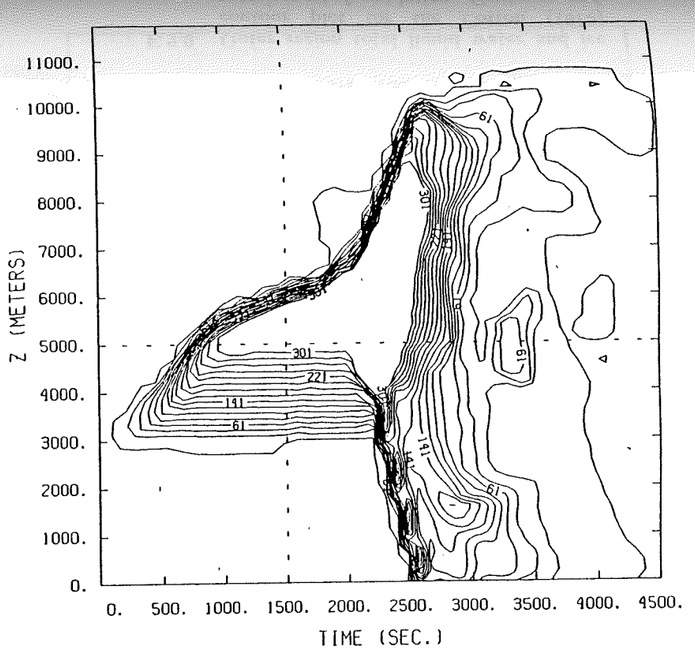
E.2.4 Outer region vertical velocity contour plot for the Asai and Kasahara (1967) entrainment formulation. Contour interval is 2 ms^{-1}.



E.2.5 Outer region total liquid water and ice contour plot for the Asai and Kasahara (1967) entrainment formulation. Contours and interval as in E.2.3.



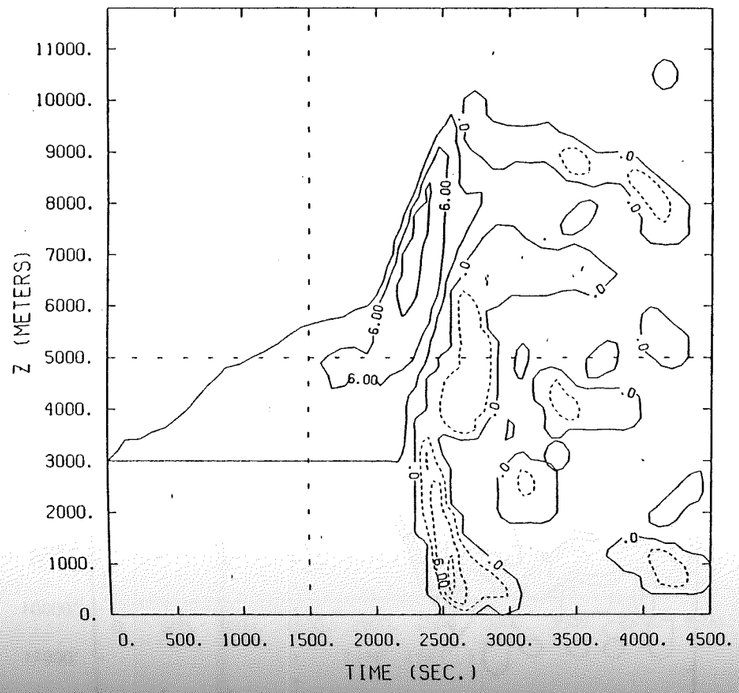
E.2.6 Inner region vertical velocity contour plot for the Cotton (1975) entrainment formulation. Contour interval is 3 ms^{-1}.



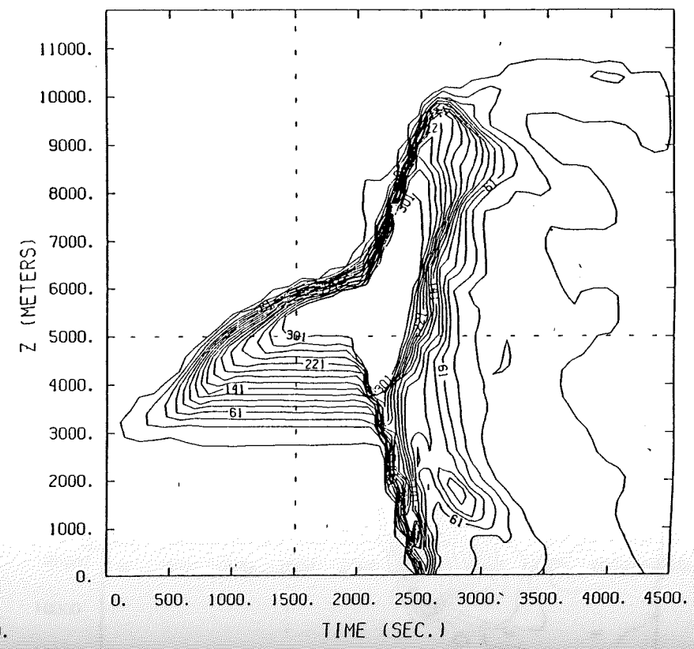
E.2.7 Inner region total liquid water and ice contour plot for the Cotton (1975) entrainment formulation. Contours and interval as in E.2.3.

QUATERNARY LIBRARY

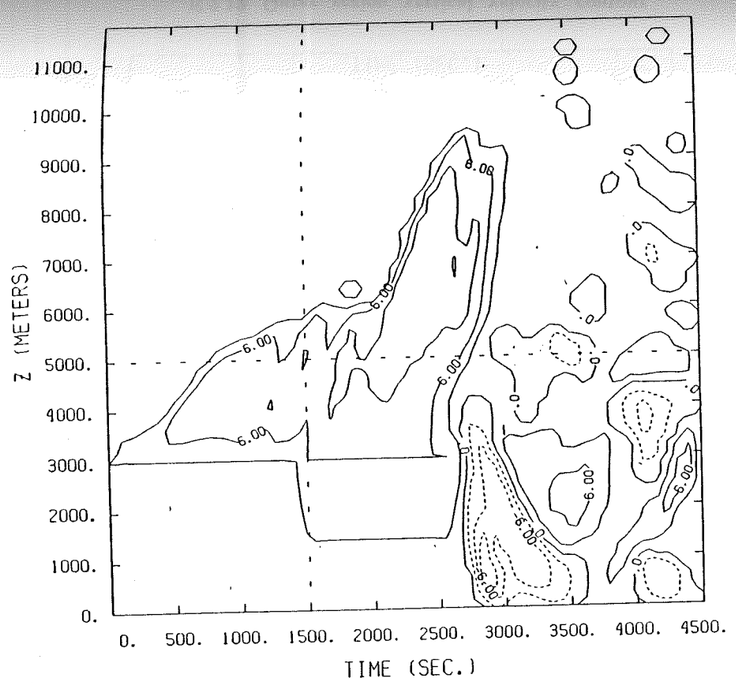
QUATERNARY LIBRARY



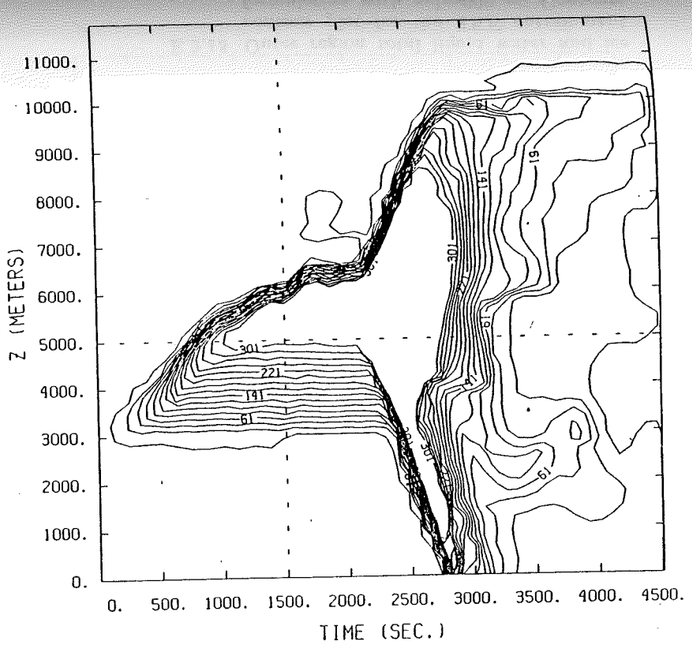
E.2.8 Outer region vertical velocity contour plot for the Cotton (1975) entrainment formulation. Contour interval is 3 ms^{-1}.



E.2.9 Outer region total liquid water and ice contour plot for the Cotton (1975) entrainment formulation. Contours and interval as in E.2.3.



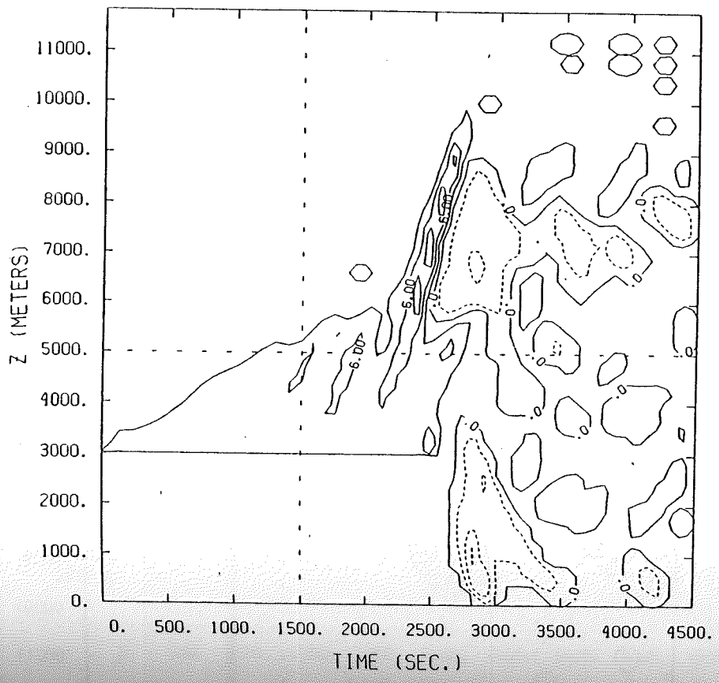
E.2.10 Inner region vertical velocity contour plot for the TKE entrainment formulation with avl=600 m. Contour interval is 3 ms^{-1}.



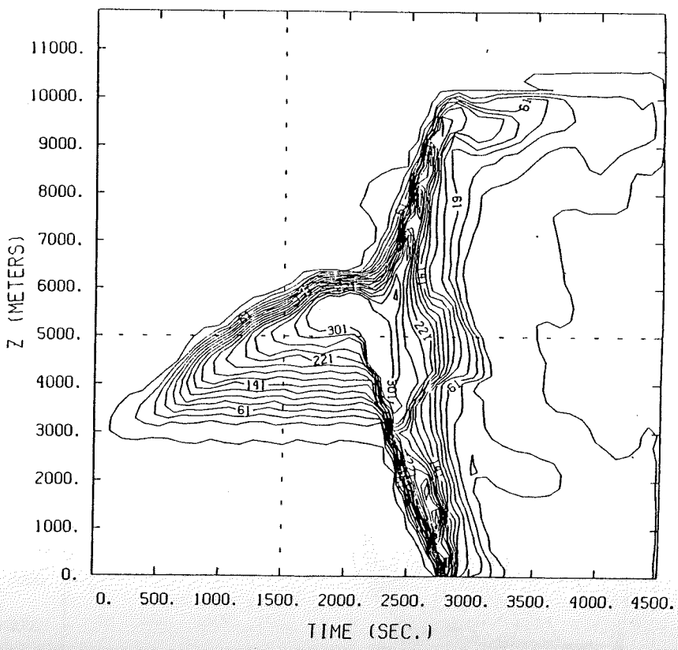
E.2.11 Inner region total liquid water and ice contour plot for the TKE entrainment formulation with avl=600 m. Contours and interval as in E.2.3.

QUATERNARY LIBRARY

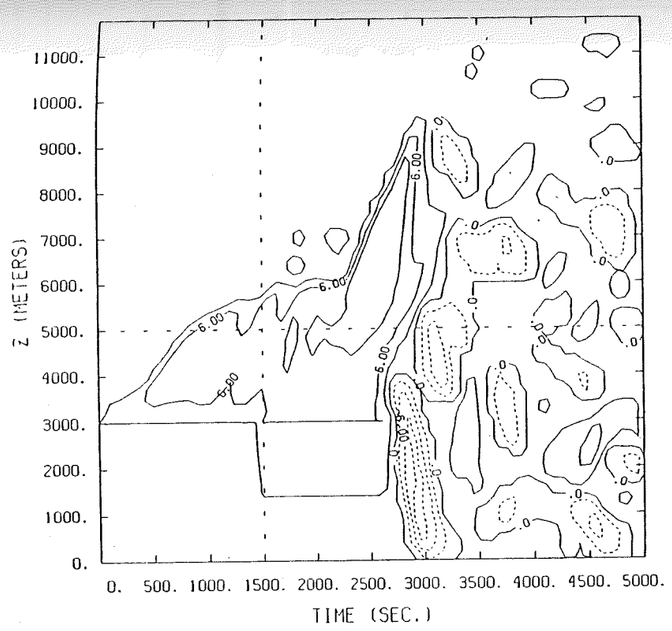
QUATERNARY LIBRARY



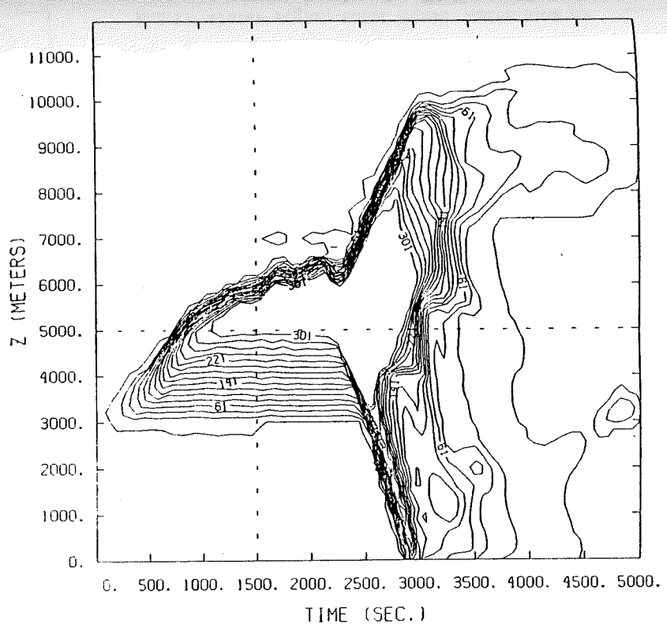
E.2.12 Outer region vertical velocity contour plot for the TKE entrainment formulation with avl=600 m. Contour interval is 3 ms^{-1}.



E.2.13 Outer region total liquid water and ice contour plot for the TKE entrainment formulation with avl=600 m. Contours and interval as in E.2.3.



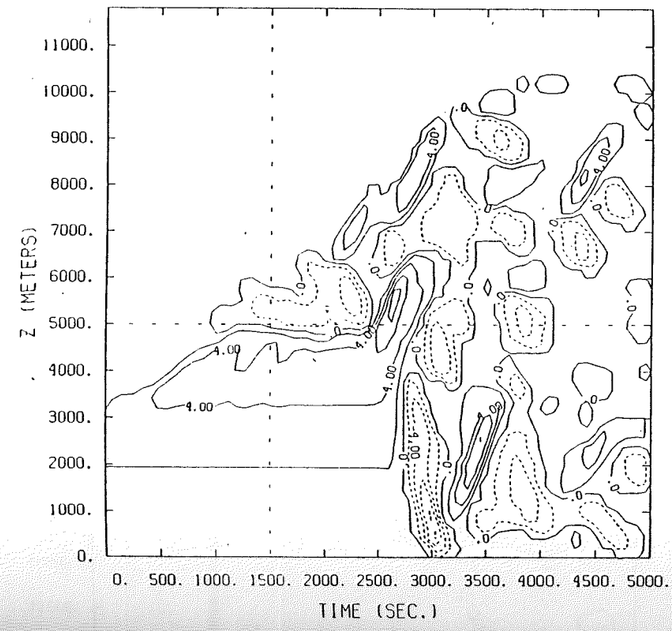
E.2.14 Inner region vertical velocity contour plot for the TKE entrainment formulation with avl=800 m. Contour interval is 3 ms^{-1}.



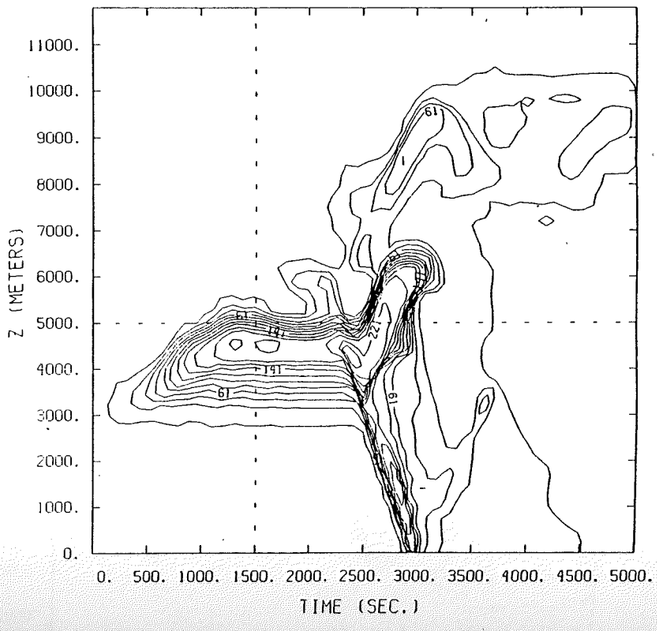
E.2.15 Inner region total liquid water and ice contour plot for the TKE entrainment formulation with avl=800 m. Contours and interval as in E.2.3.

QUATERNARY LIBRARY

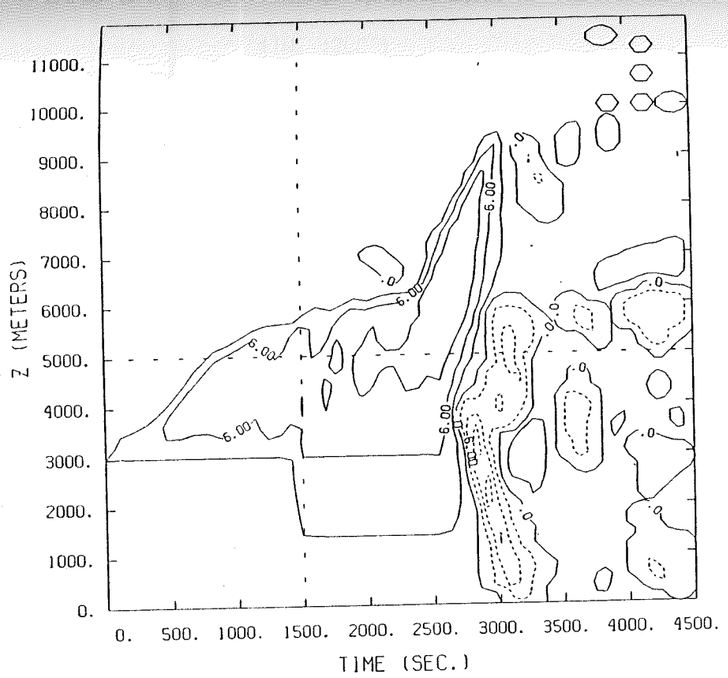
QUATERNARY LIBRARY



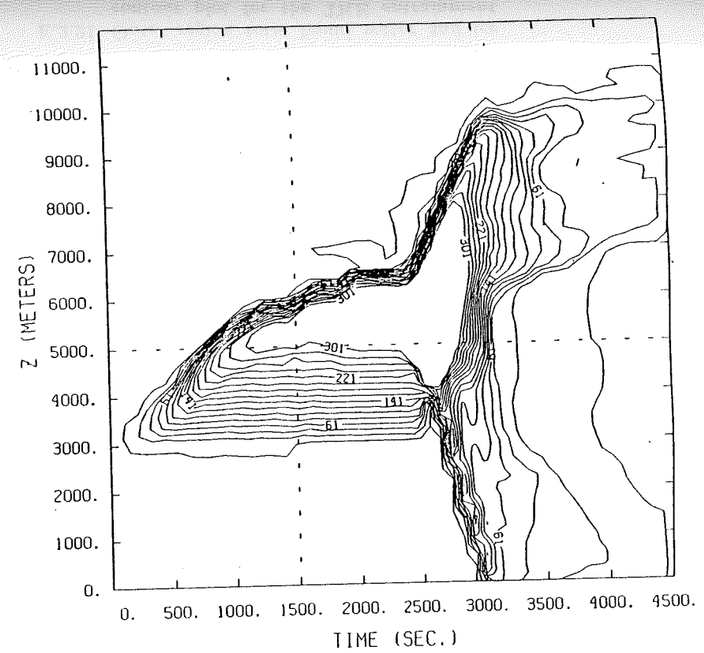
E.2.16 Outer region vertical velocity contour plot for the TKE entrainment formulation with avl=800 m. Contour interval is 3 ms^{-1}.



E.2.17 Outer region total liquid water and ice contour plot for the TKE entrainment formulation with avl=800 m. Contours and interval as in E.2.3.



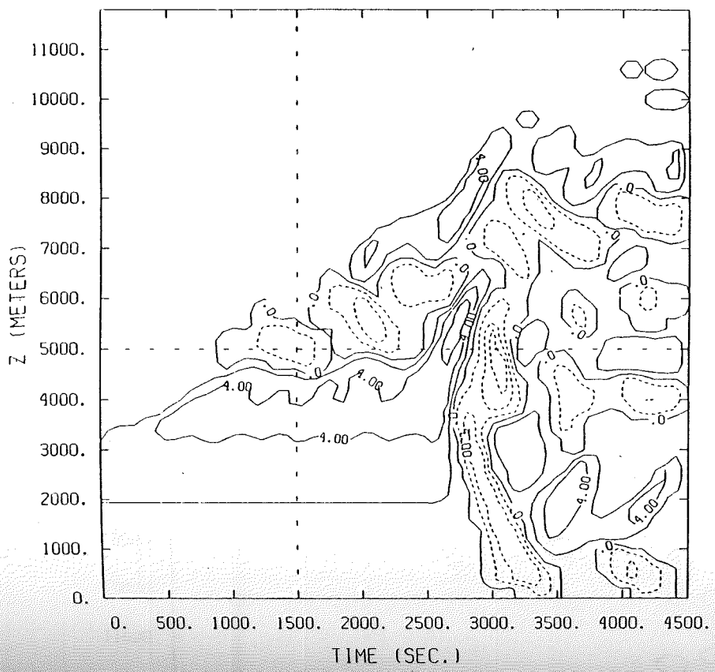
E.2.18 Inner region vertical velocity contour plot for the TKE entrainment formulation with avl=1000 m. Contour interval is 3 ms^{-1}.



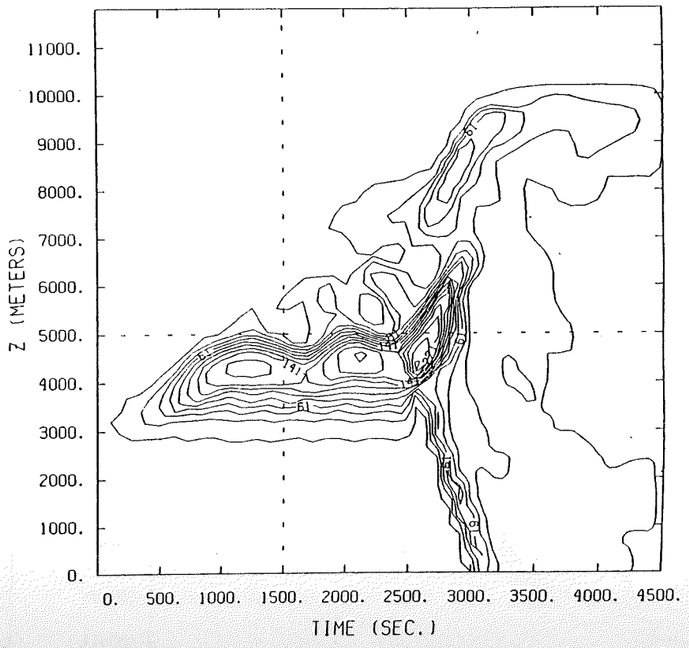
E.2.19 Inner region total liquid water and ice contour plot for the TKE entrainment formulation with avl=1000 m. Contours and interval as in E.2.3.

QUATERNARY LIBRARY

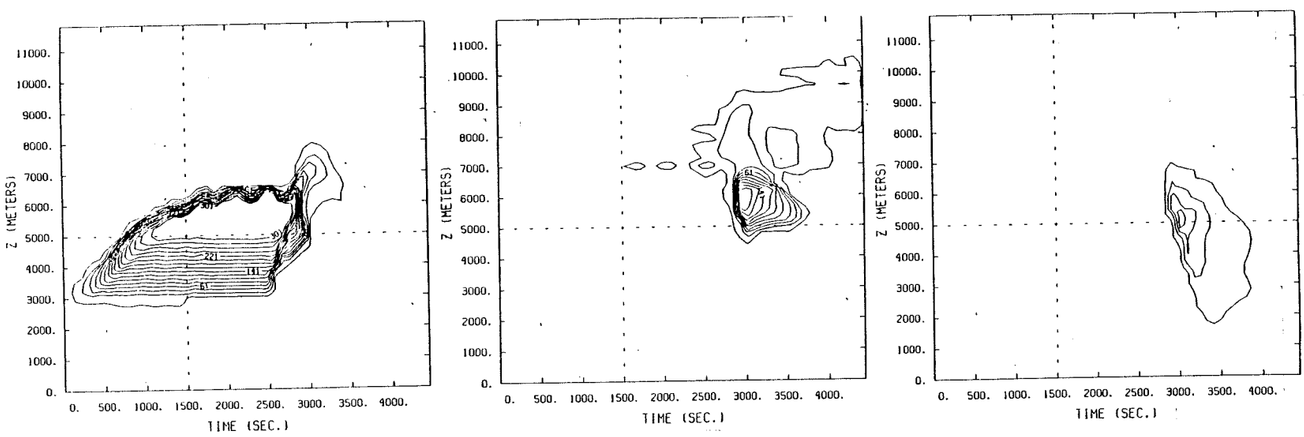
QUATERNARY LIBRARY



E.2.20 Outer region vertical velocity contour plot for the TKE entrainment formulation with avl=1000 m. Contour interval is 3 ms^{-1}.



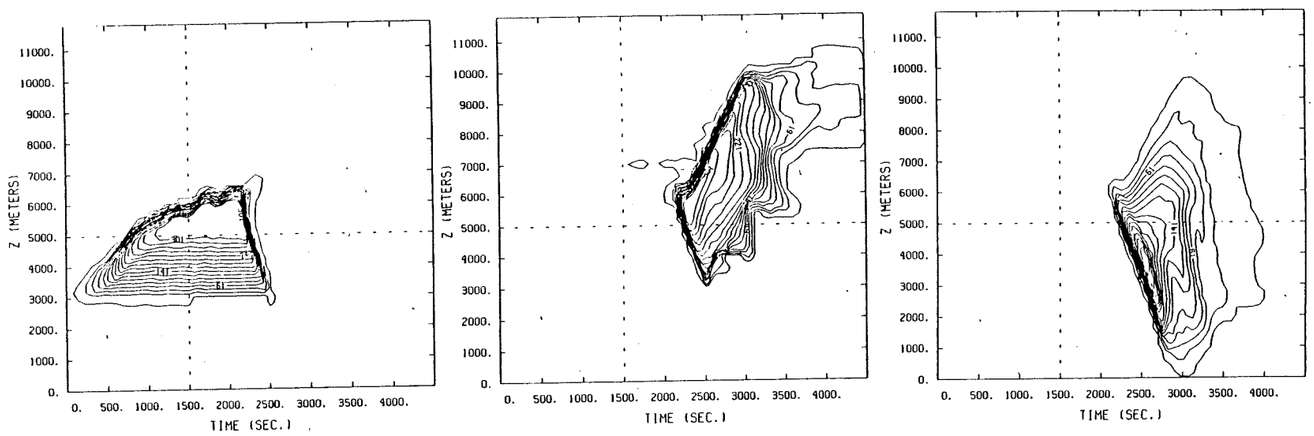
E.2.21 Outer region total liquid water and ice contour plot for the TKE entrainment formulation with avl=1000 m. Contours and interval as in E.2.3.



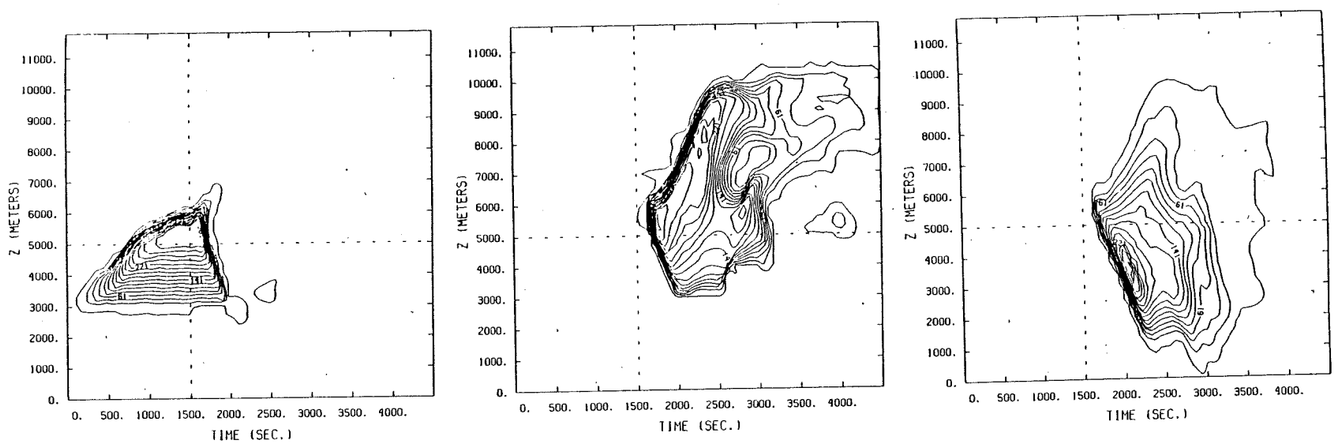
E.2.22 Inner region contour plots for β=0.4. Other parameters are set to the default values listed in table 4.1. (a) the cloud droplet field, (b) the snow field, (c) the graupel field. Contours and interval as in E.2.3.

QUATERNARY LIBRARY

QUATERNARY LIBRARY



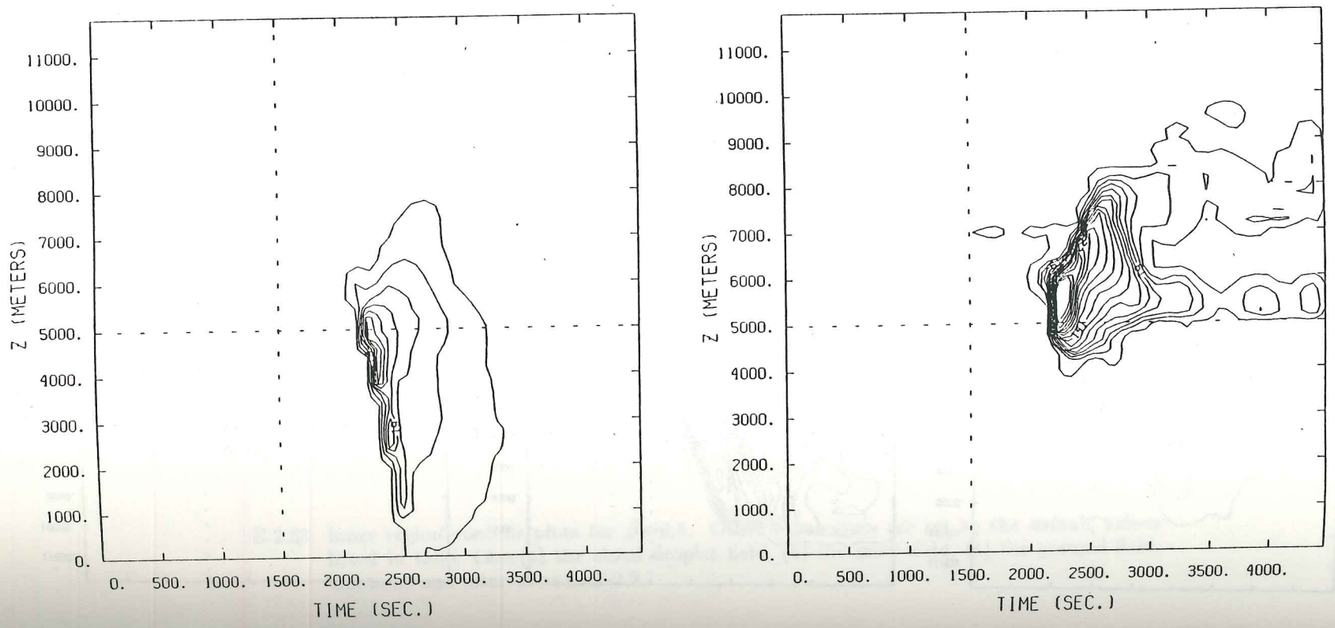
E.2.23 Inner region contour plots for β=0.5. Other parameters are set to the default values listed in table 4.1. (a) the cloud droplet field, (b) the snow field, (c) the graupel field. Contours and interval as in E.2.3.



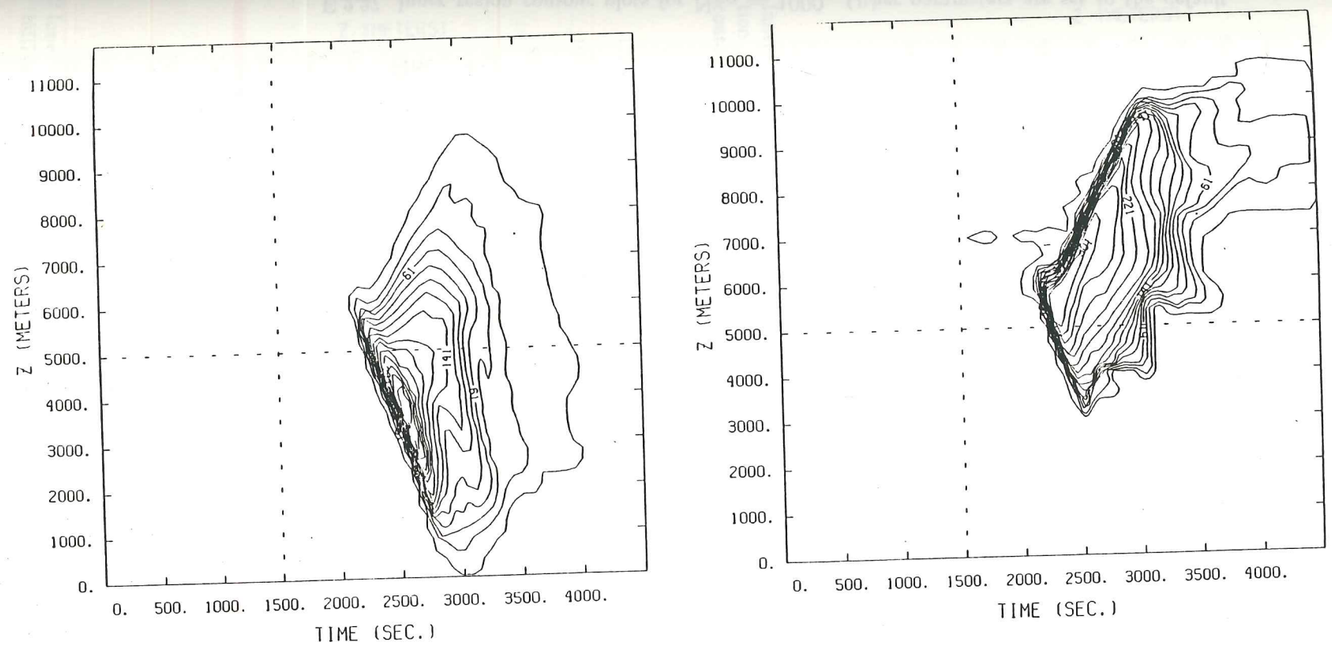
E.2.24 Inner region contour plots for β=0.6. Other parameters are set to the default values listed in table 4.1. (a) the cloud droplet field, (b) the snow field, (c) the graupel field. Contours and interval as in E.2.3.

QUATERNARY LIBRARY

QUATERNARY LIBRARY

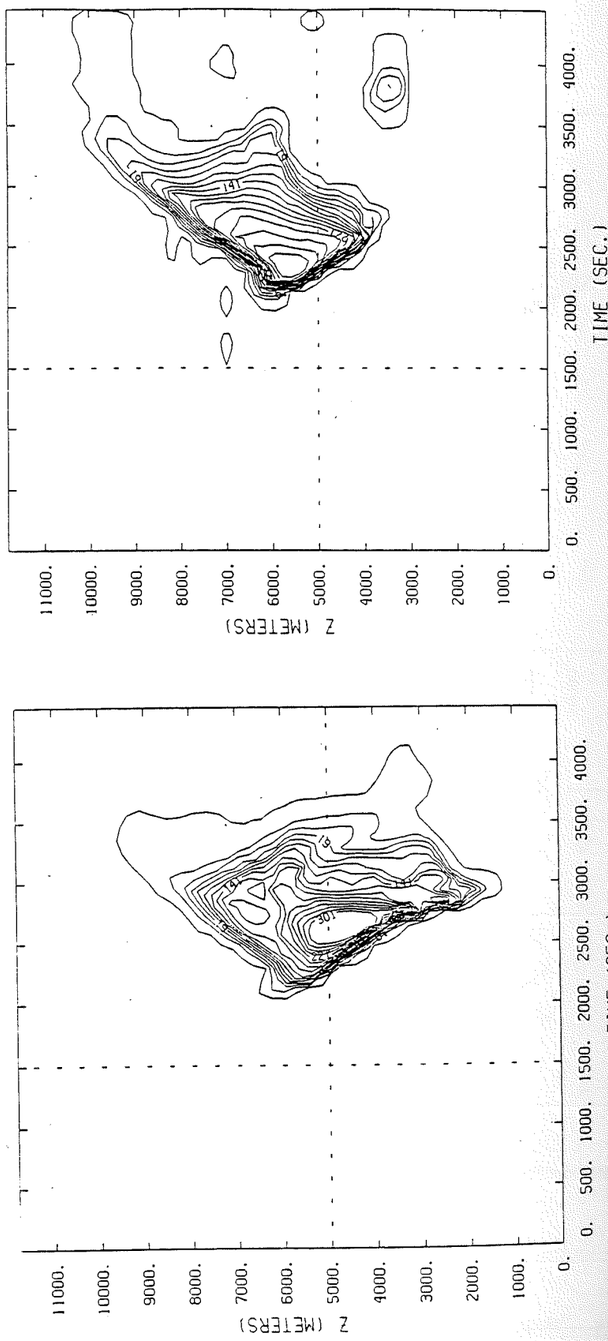


E.2.25 Inner region contour plots for $N_{TG} = 10$. Other parameters are set to the default values listed in table 4.1. (a) graupel field, (b) snow field Contours and interval as in E.2.3.

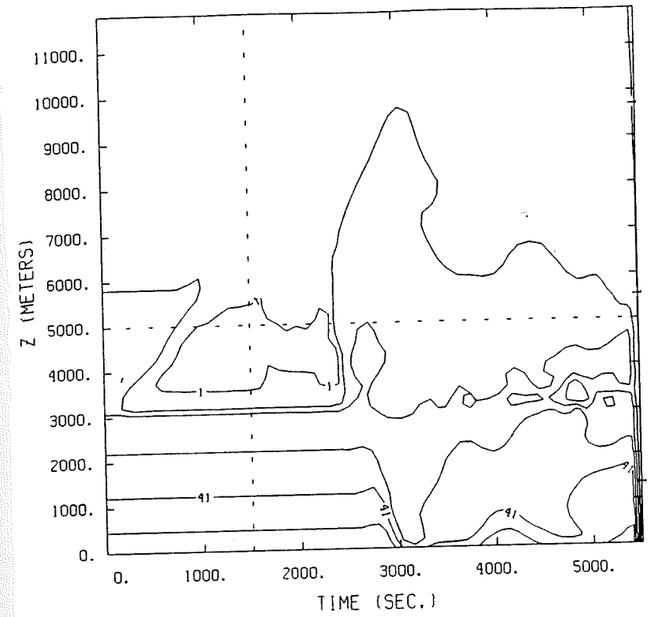


E.2.26 Inner region contour plots for $N_{TG} = 100$. Other parameters are set to the default values listed in table 4.1. (a) graupel field, (b) snow field Contours and interval as in E.2.3.

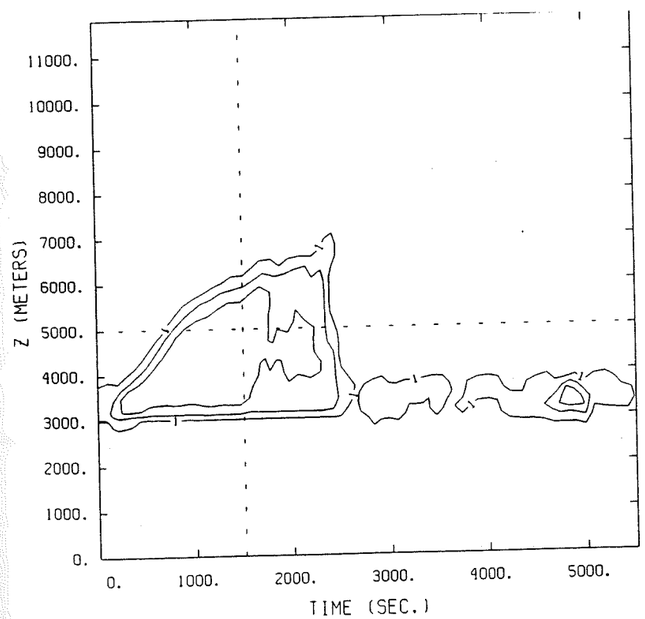
QUATERNARY LIBRARY



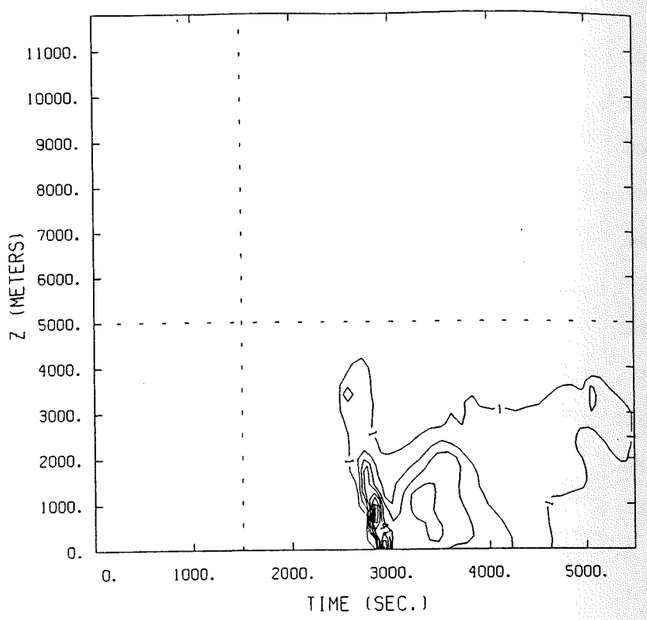
E.2.27 Inner region contour plots for $N_{TG} = 1000$. Other parameters are set to the default values listed in table 4.1. (a) graupel field, (b) snow field Contours and interval as in E.2.3.



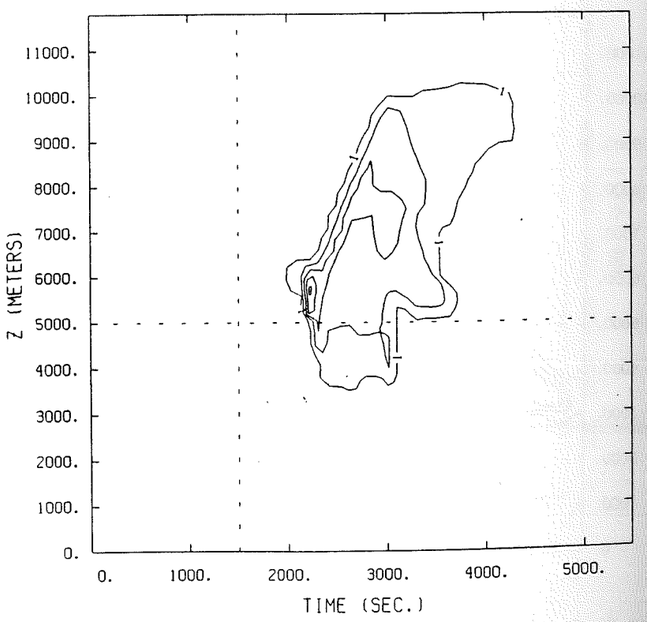
E.3.1 Inner cloud region contour plot of NH_4^+ aerosol for the Continental Background initialization. Contours are from .01 to 1.01 μg/kg; contour interval is 0.1 μg/kg. Note that the contour labels are scaled by 100.



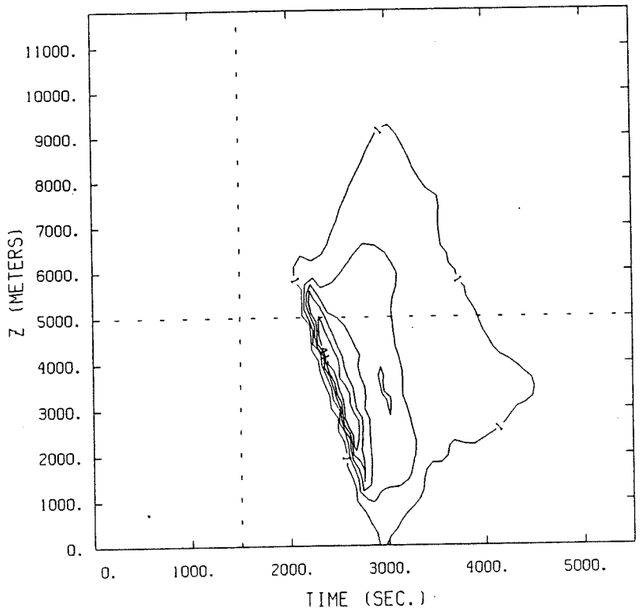
E.3.2 Inner cloud region contour plot of NH_4^+ in cloud droplets for the Continental Background initialization. Contours and scaling as in E.3.1.



E.3.3 Inner cloud region contour plot of NH_4^+ in precipitation for the Continental Background initialization. Contours and scaling as in E.3.1.



E.3.4 Inner cloud region contour plot of NH_4^+ in snow for the Continental Background initialization. Contours and scaling as in E.3.1.



E.3.5 Inner cloud region contour plot of NH_4^+ in graupel for the Continental Background initialization. Contours and scaling as in E.3.1.

APPENDIX F

Model Code

This appendix contains the entire model code (5400 lines) and so is somewhat compressed. The format is CDC CYBER update program library, in that the common blocks are defined using *comdeck statements and then inserted into the relevant subroutines upon compilation (*call statements). The code is documented, and the subroutines discussed in the main text are listed in the table of contents. All subroutines and functions are preceeded by **deck name* lines.

The main program, *Wolken,* sets up the environmental arrays and transfers to subroutine *Clouds,* which computes the cloud fields.

Good luck!


```
*COMDECK GRID
      COMMON/GRID/ TBAR(101),PBAR(101),QBAR(101),WBAR(101),
     1 PHIBAR(101),RHOBAR(101),Z(101),CPT(101),DCPT(101)
*COMDECK CLOUD
      COMMON/CLOUD/TC(101,2),QV(101,2),QC(101,2),QP(101,2),
     1 QS(101,2),QG(101,2),RHOC(101),RHOCI(101),PHIC(101,2),
     2 WC(101,2),UC(101,2),SFLG(101,2),BUOY(101,2)
*COMDECK CHEM
      COMMON/CHEM/ CBAR(101),CC(101,2),CSTOR(101),CENT(101)
*COMDECK DATAIN
      COMMON/DATAIN/ZIN(200),TIN(200),PIN(200),QIN(200),
     1          WIN(200),CIN(200)
*COMDECK VARCON
      COMMON/CONST/ CON1,CON2,CON3,CON4,G,CPV,CW,CPD,EPS,RD,WLV
     1 ,VIL,EWL,GCON,SDAY,PIE,CPO
      COMMON/VAR/ NUM,NLVLS,LDIMS,DZ,TIME,ATIME,PTIME,UTIME,AREAA,
     1 AREAB,ARAD,BRAD,ARI,BRI,ABI,ITNUM,ICOUNT,BIAS,BIX2,BIXI,CBIAS,
     2 BIXC,BIXM,PFALL,SFALL,GFALL,JLEV,JPRT,IFLX,TDUMP,ITD,
     3 WLIM,BLIM,IPPF,CHMFLG,WMULT,LQF,PFT(2),SFT(2),GFT(2),PRATE(2),
     4 SRATE(2),GRATE(2),PHTOT(2),SUFT(2,2),AMFT(2),TKECO,ICHUN,HPFAC
*COMDECK DUMMY
      COMMON/DUMMY/ DU1(101),DU2(101),DU3(101),DU4(101),DU5(101),
     1          DU6(101),DU7(101),DU8(101),DU9(101),DU0(101)
*COMDECK WASSER
      COMMON/WASSER/ AUTO(101,2),COLL(101,2),PEVAP(101,2),
     1          COND(101,2),DC(101,2),DP(101,2),VP(101,2)
*COMDECK WATCON
      COMMON/WATCON/ TNC,ONR,CD,PICON,KAUT,QLLIM,DCCO,DPCO,VPCO
*COMDECK EISCON
      COMMON/EISCON/ TF,ONS,TNG,BETA,CMAS,TGCO
*COMDECK EIS
      COMMON/EIS/ SINIT(101,2),VSDEP(101,2),SRME(101,2),CSFZ(101,2)
     1          ,VGDEP(101,2),GRME(101,2),PGCOL(101,2),PGFZ(101,2)
     2          ,GMELT(101,2),GPSHD(101,2),SMELT(101,2),SGCON(101,2)
     3          ,SGCOL(101,2),DG(101,2),VG(101,2),DS(101,2),VS(101,2)
*COMDECK HBLOK
      COMMON/HBLOK/ HREF,QVS,QWS,QIS,QPS,QGS,TCL,QVC,QWC,QIC
*COMDECK STORS
      COMMON/STORS/ STOR(101,7),TKE(101,2),TEP(101,2),EDCO(4),AVL(2),
     1          RHOO(101,2),D(101,4),AK1
*COMDECK CBLOK
      COMMON/CBLOK/ WA(101),WB(101),EP(101,2),DEL(101,2),
     1          FUP(101,2),FSIDE(101,2),ADCO(3),
     2          UNO(101),ZERO(101),UBCO,HARAD,ARB,COFSI,
     3          DZI,NP,EKA,EKB
*COMDECK ASU
      COMMON/ASU/ PSU(101,2,19)
*COMDECK ANH
      COMMON/ANH/ PNH(101,2,15)
*COMDECK AHP
      COMMON/AHP/ PHP(101,2,13)
*COMDECK SO4
      COMMON/SO4/ QSO2B(101),QSO4B(101),QSO2A(101,2),QSO4A(101,2)
     1          ,QSO4C(101,2),QSO4P(101,2),QSO4S(101,2),QSO4G(101,2)
     2          ,QSB(101)
*COMDECK H4N
      COMMON/H4N/ QNH4B(101),QNH4A(101,2),QNH4C(101,2),QNH4P(101,2)
     1          ,QNH4S(101,2),QNH4G(101,2)
*COMDECK H2O2
      COMMON/H2O2/ QH2O2B(101),QH2O2A(101,2),QH2O2C(101,2),QH2O2P(101,2)
     1          ,QH2O2S(101,2),QH2O2G(101,2)
*COMDECK HP
      COMMON/HP/ HPC(101,2),HPP(101,2)
*COMDECK CHMSTR
      COMMON/CHMSTR/ CHSTR(101,16),CHENT(101,16),CBIASA(101)
*COMDECK WFR
      COMMON/WFR/ WS(101,2),FARR(101),FCON(101),DWPDZ(101,2)
```

```
*DECK WETHUR
      PROGRAM WETHUR(INPUT,OUTPUT,TAPE5,TAPE8,TAPE9,TAPE10,
     1 TAPE11,TAPE12,TAPE13,TAPE14,TAPE15,TAPE16,TAPE17,TAPE18,
     2 TAPE20,TAPE25,TAPE30,TAPE45,TAPE4=INPUT,TAPE6=OUTPUT)
C
C     THIS IS A CYLINDRICAL CLOUD MODEL PROGRAM BASED ON ASAI AND
C     KASAHARA (1967).  TWO CLOUD REGIONS ARE CALCULATED AND
C     INCORPORATE THE STEPHENS (1979) ICE PHASE MICROPHYSICAL
C     PARAMETERIZATION.  ADVECTION IS VIA THE FLUX-CORRECTED
C     ALGORITHM 'ETBFCT' OF BORIS AND BOOK.
C
C
*CALL GRID
*CALL CLOUD
*CALL VARCON
*CALL DATAIN
*CALL DUMMY
*CALL WATCON
*CALL EISCON
*CALL HBLOK
*CALL CHEM
*CALL ASU
*CALL ANH
*CALL AHP
*CALL CHMSTR
C       CONSTANTS ARE IN MKS UNITS
      LOGICAL SFL,WFL
      DATA G/9.807/,CPV/1850./,CPD/1005./,EPS/0.622/,RD/287.05/,
     1 WLV/2.501E+6/,VIL/2.834E+06/,CW/4218./,PIE/3.141592654/,
     2 EWL/0.334E+06/
      DATA LDIMS/101/, CPO/0.33333333/, TF/273.16/, PFALL/0.0/,
     1 SFALL/0.0/, GFALL/0.0/, SDAY/86400./, TIME/0.0/, CMAS/1.0E-12/
      DATA TBAR/101*0.0/, PBAR/101*0.0/, QBAR/101*0.0/, WBAR/101*0.0/,
     1  RHOBAR/101*0.0/, PHIBAR/101*0.0/, Z/101*0.0/, CPT/101*0.0/,
     2  DCPT/101*0.0/, RHOC/101*0.0/, DU1/101*0.0/, DU2/101*0.0/,
     3  DU3/101*0.0/, DU4/101*0.0/, DU5/101*0.0/, DU6/101*0.0/,
     4  DU7/101*0.0/, DU8/101*0.0/, DU9/101*0.0/, DU0/101*0.0/
      DATA TC/202*0.0/, QV/202*0.0/, QC/202*0.0/, QP/202*0.0/,
     1 QS/202*0.0/, QG/202*0.0/, WC/202*0.0/, PHIC/202*0.0/, UC/202*0.0/
      DATA ZIN/200*0.0/, TIN/200*0.0/, PIN/200*0.0/, QIN/200*0.0/,
     1 WIN/200*0.0/, CIN/200*0.0/
      DATA CHSTR/1616*0.0/, CHENT/1616*0.0/, PSU/3838*0.0/,
     1 PNH/3030*0.0/, PHP/2626*0.0/
C     CALCULATE USEFUL CONSTANTS AND INITIALIZE ARRAYS
      CON1=WLV*EPS/CPD
      CON2=RD/CPD
      CON3=WLV/CPD
      CON4=RD/G
      GCON=1./G
      SFL=.FALSE.
      WFL=.FALSE.
C     JLEV IS THE NUMBER OF CYLINDERS TO COMPUTE.
C     JLEV=1 FOR ONLY THE INNER CYLINDER
C        =2 FOR BOTH REGIONS.
      READ(4,*) JLEV
C     BIAS SETS THE INITIAL VALUE IN THE WATER ARRAYS
C     BIXC IS SET TO LIMIT THE ERROR IN T FROM UNCERTAINTIES
C     IN THE WATER VARIABLES TO A VALUE COMPARABLE TO
C     THAT FROM THE ENTHALPY INVERSION (.LT. .00005 K).
C     CBIAS SETS THE CHEMICAL FIELD BIAS
      READ(4,*) BIAS
            CBIAS=.00001*BIAS
            BIX2=2.*BIAS
            BIXX=10.*BIAS
            BIXC=AMAX1(100.*BIAS,1.0E-09)
            BIXL=BIXC*0.5
            BIXM=AMAX1(1000.*BIAS,1.0E-08)
      READ(4,*) ONS,TNG,BETA
      DO 10 I=1,LDIMS
            CBAR(I)=0.0
```



```
            CBIASA(I)=CBIAS
10    CONTINUE
C     READ INITIAL DATA, CONVERT MOISTURE (QIN) TO Q IF NEEDED
      CALL READIN(SFL,WFL)
C     SFL .EQ. TRUE INDICATES THE SOUNDING TO BE USED HAS BEEN
C     READ IN (MFLAG = 0 IN READIN), SO NO INTERPOLATION IS NEEDED.
C     WFL .EQ. TRUE INDICATES THE FINAL INTERPOLATED AND NUMERICALLY
C     DIFFUSED SOUNDING IS WRITTEN TO TAPE 25.  SEE LOOP 310.
      IF(SFL) GO TO 120
C     INTERPOLATE AND SET UP VERTICAL GRID FOR ENVIRONMENT.
C     THE MAXIMUM VERTICAL EXTENT IS 20 KM.
      NLVLS=MIN(LDIMS-2,INT(20000./DZ)+1)
            Z(1)=0.0
            TBAR(1)=TIN(1)
            PBAR(1)=PIN(1)
            WBAR(1)=WIN(1)
            QBAR(1)=QIN(1)
      DO 100 I=1,NUM
            DU1(I)=TIN(I)*(1.0+1.608*QIN(I))/(PIN(I)*(1.+QIN(I)))
100   CONTINUE
      ZIN(1)=0.0
      DO 105 I=2,NUM
            ZIN(I)=CON4*SUMER(PIN,DU1,1,I)
105   CONTINUE
      DO 110 J=2,NLVLS
            Z(J)=Z(J-1)+DZ
            CALL INTEP(Z(J),PBAR(J),ZIN,PIN,NUM,IEP)
            CALL INTEP(PBAR(J),TBAR(J),PIN,TIN,NUM,IET)
            CALL INTEP(PBAR(J),QBAR(J),PIN,QIN,NUM,IEQ)
            CALL INTEP(PBAR(J),WBAR(J),PIN,WIN,NUM,IEW)
            ICON=IEP+IET+IEQ+IEW
            IF(ICON.GE.5) THEN
                  NLVLS=J-1
                  WRITE(6,900) NLVLS
                  GO TO 115
            END IF
110   CONTINUE
115   CONTINUE
C     ADJUST THE ENVIRONMENTAL MOISTURE TO BELOW SATURATION IF
C     NECESSSARY.  QBAR IS SET TO 95% RH (WRT TO LIQUID ABOVE 273.16
C     AND ICE BELOW FREEZING)
      RHC=0.95
      DO 118 I=1,NLVLS
            QBAR(I)=AMAX1(QBAR(I),BIAS)
118   CONTINUE
      DO 119 I=1,NLVLS
            IF(TBAR(I).GT.273.16) THEN
                  QBAR(I)=AMIN1(QBAR(I),0.622*RHC*EW(TBAR(I))
     1                  /(PBAR(I)-RHC*EW(TBAR(I))))
            ELSE
                  QBAR(I)=AMIN1(QBAR(I),0.622*RHC*EI(TBAR(I))
     1                  /(PBAR(I)-RHC*EI(TBAR(I))))
            END IF
119   CONTINUE
C
120   CONTINUE
C
      DO 124 I=1,NLVLS
            DU2(I)=TBAR(I)*(1.0E+05/PBAR(I))**(0.2854)
124   CONTINUE
      DO 125 I=1,NLVLS
            PHIBAR(I)=H(DU2(I),QBAR(I),0.0,0.0)
     1            +0.5*WBAR(I)*WBAR(I)
            RHOBAR(I)=PBAR(I)/(RD*TBAR(I))
            DU0(I)=BIAS
            DU1(I)=0.0
            DU8(I)=TBAR(I)
            DU9(I)=QBAR(I)
125   CONTINUE
C     RELAX TEMPERATURE AND MOISTURE TO REMOVE SMALL DIFFUSIVE
```

```
C     FEATURES
      ZBOTBC=Z(1)-0.5*DZ
      ZTOPBC=Z(NLVLS)+0.5*DZ
      CALL NGRIDE(Z,NLVLS,ZBOTBC,ZTOPBC,1)
      CALL OGRIDE(NLVLS)
      CALL NGRIDE(Z,NLVLS,ZBOTBC,ZTOPBC,1)
      CALL VELOCE(DU1,NLVLS,0.0,0.0,1.0)
      DO 155 J=1,4
       DO 150 I=1,NLVLS
          DU2(I)=RHOBAR(I)*PHIBAR(I)
          DU4(I)=RHOBAR(I)*QBAR(I)
150    CONTINUE
       DO 151 I=1,250
          CALL ETBFCT(DU2,DU2,NLVLS,1.00,1.00)
          CALL ETBFCT(DU4,DU4,NLVLS,1.00,1.00)
151    CONTINUE
       DO 152 I=1,NLVLS
          PHIBAR(I)=DU2(I)/RHOBAR(I)
          QBAR(I)=DU4(I)/RHOBAR(I)
152    CONTINUE
       DO 153 I=1,NLVLS
          TBAR(I)=(PHIBAR(I)-2.501E+06*QBAR(I)-0.5*WBAR(I)*WBAR(I))
     1            /(1005.+4218.*QBAR(I))*(PBAR(I)*1.0E-05)**(0.2854)
153    CONTINUE
       DO 154 I=1,NLVLS
          RHOBAR(I)=PBAR(I)/(RD*TBAR(I))
154    CONTINUE
155    CONTINUE
      WRITE(6,950)
      DO 160 I=1,NLVLS
          WRITE(6,951) LDU8(I),TBAR(I),DU8(I)-TBAR(I),
     1                 DU9(I),QBAR(I),DU9(I)-QBAR(I)
160   CONTINUE
C     CALCULATE CPT (CP*THETA BAR) FOR USE IN THE PRESSURE
C     PERTURBATION TERM, AND WRITE OUT THE ENVIRONMENTAL PARAMETERS
      WRITE(6,1000)
      DO 305 I=1,NLVLS
          THETA=TBAR(I)*(1.0E+05/PBAR(I))**(0.2854)
          CPT(I)=THETA*RHOBAR(I)
          THETV=THETA*(1.0+1.608*QBAR(I))/(1.0+QBAR(I))
          WRITE(6,1010) I,Z(I),PBAR(I),TBAR(I),QBAR(I),WBAR(I)
     1                  ,THETA,THETV,PHIBAR(I),RHOBAR(I),I
305   CONTINUE
      IF(WFL) THEN
        IF(.NOT.SFL) THEN
          DO 310 I=1,NLVLS
             WRITE(25,1012) Z(I),PBAR(I),TBAR(I),QBAR(I),
     1                      WBAR(I)
310        CONTINUE
          WRITE(6,1013)
        ELSE
          WRITE(25,1014) TNG
          DO 311 I=1,NLVLS
             WRITE(25,1014) PBAR(I),RHOBAR(I)
311        CONTINUE
        END IF
          STOP
      ELSE
          DO 312 I=1,NLVLS
             WRITE(25,1012) 0.01*PBAR(I),TBAR(I)-273.16,
     1                      QBAR(I),WBAR(I),0.0
312        CONTINUE
      END IF
       DCPT(1)=(CPT(2)-CPT(1))*0.5/DZ
       DCPT(NLVLS)=(CPT(NLVLS)-CPT(NLVLS-1))*0.5/DZ
      DO 315 I=2,NLVLS-1
          DCPT(I)=(CPT(I+1)-CPT(I-1))*0.5/DZ
315   CONTINUE
C
C     INITILIZE THE CLOUD DOMAIN GRIDS
```



```
      DO 320 J=1,2
      DO 320 I=1,NLVLS
          PHIC(I,J)=PHIBAR(I)
          TC(I,J)=TBAR(I)
          QV(I,J)=QBAR(I)
          WC(I,J)=WBAR(I)
          QC(I,J)=DU0(I)
          QP(I,J)=DU0(I)
          QS(I,J)=DU0(I)
          QG(I,J)=DU0(I)
320   CONTINUE
      CALL CLOUDS
      IF(TIME.GT.0.00) THEN
        TIMEH=TIME/3600.
        WRITE(6,1021) ICOUNT,TIMEH,PFALL,SFALL,GFALL
      END IF
C
900   FORMAT(10X,'INTERPOLATION ENDING AT LEVEL',1X,I3)
950   FORMAT(/,20X,'SOUNDING RELAXATION; TBAR (OLD,NEW,DIFF), QBAR'
     1       ,' (OLD,NEW,DIFF)')
951   FORMAT(10X,I5,2F12.4,F15.6,5X,2F15.6,F15.8)
1000  FORMAT(//,25X,'Z,PRESSURE,TBAR,QBAR,WBAR,THETA,THETAV,PHIBAR',
     1       ',RHOBAR',/)
1010  FORMAT(15X,I5,8F10.2,F10.6,F10.4,2F12.4,F15.2,F10.4,I5)
1012  FORMAT(2F15.4,2F15.6,F12.3)
1013  FORMAT(/,15X,'THIS SOUNDING WRITTEN TO TAPE25',/)
1014  FORMAT(10X,2G15.6)
1021    FORMAT(/,25X,'TOTAL ITERATIONS= ',I5,5X,'TOTAL TIME= ',
     1       F10.2,' HOURS',/,
     2       25X,'TOTAL PRECIPITATION= ',F8.3,' MM RAIN',2X,
     3       F8.3,' MM EQ. SNOW',2X,F8.3,' MM EQ. GRAUPEL',/)
C
      STOP
      END
*DECK READIN
      SUBROUTINE READIN(SFL,WFL)
C     THIS SUBROUTINE READS IN PRES,TEMP,MOISTURE AND VERT. VEL
C     AND CONVERTS MOISTURE TO Q IF NECESSARY.
C     MFLAG IS USED TO DETERMINE THE SOUNDING TYPE AND MOISTURE
C     INPUT IF AN INTERPOLATED SOUNDING IS DESIRED
C        MFLAG      (0) READ IN Z,P,T,Q,W TO BE USED
C                     Z IN METERS
C                     P IN PASCALS
C                     T IN KELVINS
C                     Q IN (KG/KG)
C                     W IN M/S
C        MFLAG     .GE.1 READ IN P,T,Q,W AND INTERPOLATE
C                     P IN MB
C                     T IN C
C                       (1) QIN IS Q (KG/KG)
C                       (2) QIN IS TWB  (C)
C                       (3) QIN IS UW   (%)
C                       (4) QIN IS TD   (C)
C     IWFL SETS VELOCITY INPUT FOR MFLAG .GE.1
C                   IWFL... (1) WIN IS IN M/S
C                           (2) WIN IS IN PA/S
C                           (3) WIN IS IN MB/S
C     RETURNED VARIABLES
C           SFL:    SFL=.TRUE.  MFLAG=0
C                   SFL=.FALSE. MFLAG .GE.1
C           WFL:     WFL=.TRUE.  WSN .EQ. ON
C                   WFL=.FALSE. WSN .EQ. OFF
C                   SEE LOOP 310 IN WETHUR FOR WFL
C
*CALL GRID
*CALL CLOUD
*CALL DATAIN
*CALL VARCON
*CALL WASSER
*CALL WATCON
```

```
*CALL EISCON
      LOGICAL SFL,WFL,CBFL,CHMFLG
      DIMENSION TITLE(80)
      DATA ON/4HON  /,OFF/4HOFF /,BLANK/4H    /,TOP/4HTOP /,
     1    ABAT/4HABAT/,SIDE/4HSIDE/
     2 , TD/4HTDEP/,SS/4HSTDE/,TOPL/4HTOPL/
     3 ,  AUTOK/4HKESS/,AUTOC/4HCOTT/,AUTOB/4HBERR/,AUTOO/4HORVI/
C
      SFL=.FALSE.
      IREAD=5
C
C     JID.GT.1 FOR READING DATA FROM TAPE8.
C
      READ(IREAD,100) MFLAG
      IF(MFLAG.EQ.0) SFL=.TRUE.
      READ(IREAD,100) IWFL
      READ(IREAD,100) JID
      READ(IREAD,110) DZ
      READ(IREAD,110) ATIME
      READ(IREAD,110) PTIME
      READ(IREAD,110) UTIME
      READ(IREAD,105) ITNUM
      READ(IREAD,110) ARAD
      READ(IREAD,110) BRAD
C
      KAUT=1
      READ(IREAD,125) AUTCON
      IF(AUTCON.EQ.AUTOO) KAUT=2
      IF(AUTCON.EQ.AUTOC) KAUT=3
      IF(AUTCON.EQ.AUTOB) KAUT=4
C
      READ(IREAD,120) TNCC
      IF(TNCC.LE.0.0) TNCC=300.0
C       COMPUTE QLLIM FOR AUTOCONVERSION
      QLLIM=AMAX1(0.0, 7.5E-06*TNCC-2.5E-04)
C       CONVERT TNC FROM CM**-3 TO M**-3
      TNC=TNCC*1.0E+06
C       READ PRESSURE PERTURBATION FLAG
      IPPF=1
      READ(IREAD,125) PPF
      IF(PPF.EQ.OFF) IPPF=0
C       READ SOUNDING FLAG
      WFL=.FALSE.
      READ(IREAD,125) WSN
      IF(WSN.EQ.ON) WFL=.TRUE.
C       READ CHEMISTRY FLAG
C       IF ICHUN = 1  USE CONTINENTAL BACKGROUND INITIALIZATION
C                = 2  USE POLLUTED BACKGROUND IN CHEMINT
C                = 3  USE CB WITH (NH4)2(SO4)
C                = 4  USE POL WITH (NH4)2(SO4)
C                = 5  USE CB WITH NO ENTRAINMENT
C                = 6  USE POLL WITH NO ENTRAINMENT
C                = 7  USE CB AND SET H2O2 TO HPFAC*0.5PPBV
C                = 8  USE POL AND SET H2O2 TO HPFAC*0.5PPBV
C                .GT.8  OPEN UNIT (45) AND READ INITIALIZATION
      CHMFLG=.FALSE.
      READ(IREAD,125) CHM
      IF(CHM.EQ.ON) THEN
            CHMFLG=.TRUE.
            READ(IREAD,*) ICHUN
            IF(ICHUN.EQ.7 .OR. ICHUN.EQ.8) THEN
                  READ(IREAD,*) HPFAC
            ELSE
                  HPFAC=1.0
            END IF
      ELSE

            HPFAC=0.0
            ICHUN=99.
      END IF
```



```
      IF(JID.GT.1) IREAD=8
      NUM=0
      WRITE(6,2010) MFLAG,IWFL,JID,ITNUM
      WRITE(6,2000) DZ,ATIME,PTIME
      WRITE(6,2050) ARAD,BRAD
      WRITE(6,2070) AUTCON,KAUT
      WRITE(6,2090) TNC,TNCC
      WRITE(6,2095) PPF,IPPF
      WRITE(6,2096) WSN,WFL
      WRITE(6,3000) CHM,CHMFLG,ICHUN
C
      READ(IREAD,175) (TITLE(I),I=1,80)
      IF(SFL) THEN
            WRITE(6,2097)
            NLVLS=0
            DO 300 I=1,LDIMS
                  READ(IREAD,*,END=305) Z(I),PBAR(I),TBAR(I),
     1                  QBAR(I),WBAR(I)
                  NLVLS=NLVLS+1
300         CONTINUE
305         CONTINUE
            WRITE(6,180) (TITLE(I),I=1,80)
            DO 310 I=1,NLVLS
                  WRITE(6,2098) I,Z(I),PBAR(I),TBAR(I),QBAR(I),
     1                  WBAR(I)
310         CONTINUE
      ELSE
        DO 340 I=1,200
            READ(IREAD,*,END=350) PIN(I),TIN(I),QIN(I),WIN(I),CIN(I)
           NUM=NUM+1
340     CONTINUE
350     CONTINUE
      WRITE(6,180) (TITLE(I),I=1,80)
      WRITE(6,900) NUM
      DO 375 M=1,NUM
            TIN(M)=278.16+TIN(M)
            PIN(M)=100.0*PIN(M)
            WRITE(6,201) PIN(M),TIN(M),QIN(M),WIN(M),CIN(M)
375   CONTINUE
C
      GO TO (600,500,400,575), MFLAG
C       QIN(I)=UW
400   DO 450 J=1,NUM
            EWAT=EW(TIN(J))
            RW=(EPS*EWAT)/(PIN(J)-EWAT)
            QIN(J)=(RW*QIN(J))/(1.+(1.-QIN(J))*RW/EPS)
450   CONTINUE
      GO TO 600
C       QIN(I)=TWB
500   DO 550 J=1,NUM
            PWA=EW(TIN(J))
            QIN(J)=QIN(J)+273.16
            XQ=((EPS*PWA)/(PIN(J)-PWA))-(TIN(J)-QIN(J))
     1        /CON8
            QIN(J)=AMAX1(BIAS,XQ)
550   CONTINUE
      GO TO 600
C       QIN(J)=TD
575   DO 580 J=1,NUM
            PW=EW(QIN(J)+273.16)
            QIN(J)=AMAX1(BIAS,(EPS*PW/(PIN(J)-PW)))
580   CONTINUE
C       QIN=Q
600   CONTINUE
C       CONVERT WIN FROM PA/S OR MB/S TO M/S IF NECESSARY.
C       ENVIRONMENT IS ASSUMED TO BE IN HYDROSTATIC EQUILIBRIUM.
      IF(IWFL.GT.1) THEN
            WINCO=CON4
            IF(IWFL.EQ.3) WINCO=100.*CON4
            DO 625 I=1,NUM
```

```
                WIN(I)=WINCO*TIN(I)*(1.+QIN(I))
     1                *WIN(I)/PIN(I)
625         CONTINUE
      END IF
C
C
C
      END IF
C
      IF(DZ.LT.25.0) THEN
            WRITE(6,1000)
            DZ=100.0
      END IF
      IF(ATIME.LE.0.000) THEN
            WRITE(6,1005)
            ATIME=1800.
      END IF
      IF(ITNUM.LE.0) THEN
            WRITE(6,1010)
            ITNUM=100
      END IF
      IF(ARAD.LE.0.0) THEN
            WRITE(6,1015)
            ARAD=1000.0
      END IF
      IF(BRAD.LE.ARAD) THEN
            WRITE(6,1020)
            BRAD=1.5*ARAD
      END IF
            AREAA=PIE*ARAD*ARAD
            AREAB=PIE*BRAD*BRAD-AREAA
            ARI=1.0/ARAD
            BRI=1.0/BRAD
            ABI=1.0/(BRAD-ARAD)
C
      RETURN
C
C
      ENTRY READCB(CBASE,CBT,CBQ,WCB,RADM,CBFL)
C
C
      READ(4,*) CBASE,CBT,CBQ
      READ(4,*) WCB,WMULT
      READ(4,*) RADM
      READ(4,125) CSN
C
      CBASE=CBASE*100.
      CBT=CBT+273.16
      CBFL=.FALSE.
      IF(CSN.EQ.ON) CBFL=.TRUE.
      WRITE(6,2099) CSN,CBFL
C
      RETURN
C
C
100   FORMAT(15X,I1)
105   FORMAT(15X,I4)
110   FORMAT(15X,F15.5)
120   FORMAT(15X,F15.8)
125   FORMAT(15X,A4)
175   FORMAT(80A1)
180   FORMAT(/,10X,80A1,/)
201   FORMAT(10X,2F15.8,F15.6,F15.5,F15.8)
900   FORMAT(10X,'NUM=',I5)
1000  FORMAT(10X,'DZ NOT INPUT OR LESS THAN 25 M.  DEFAULT'
     1       ,' IS DZ=100.0 M')
1005  FORMAT(10X,'TOTAL INTEGRATION TIME MUST BE GREATER THAN'
     1       ,' 0.00.  DEFAULT IS ATIME=1800.0 S')
1010  FORMAT(10X,'ITERATION LIMIT MUST BE .GE. 1.  DEFAULT IS'
     1       ,' ITNUM=100')
```



```
1015  FORMAT(10X,'INNER REGION RADIUS MUST BE .GT. 0.0.'
     1       ,'  DEFAULT IS ARAD=1000.0 M')
1020  FORMAT(10X,'OUTER REGION RADIUS MUST BE GREATER THAN INNER'
     1       ,' RADIUS.  DEFAULT IS BRAD=1.5*ARAD.')
2010  FORMAT(/,15X,'MFLAG=',I3,5X,'IWFL=',I3,5X,
     1       'JID=',I5,5X,'ITNUM=',I5,/)
2000  FORMAT(15X,'DZ=',F15.5,5X,'ATIME=',F15.5,5X,
     1       'PTIME=',F15.5,/)
2050  FORMAT(15X,'ARAD=',F10.2,' M',5X,'BRAD=',F10.2,' M',/)
2070  FORMAT(15X,'AUTOCONVERSION IS ',A4,';  KAUT IS ',I2,/)
2080  FORMAT(15X,'AUTOCONVERSION THRESHOLD; QLLIM IS',F10.8,
     1       ' KG/KG',/)
2090  FORMAT(15X,'CLOUD DROPLET CONCENTRATION IS',F15.2,' M**-3',
     1       ' OR ',F10.2,' CM**-3',/)
2095  FORMAT(15X,'PRESSURE PERTURBATION IS ',A4, ';IPPF IS',I2,/)
2096  FORMAT(15X,'SOUNDING WRITEOUT IS ',A4, ';WFL IS',L2,/)
2097  FORMAT(15X,'MFLAG=0; NO INTERPOLATION ON INPUT SOUNDING',/)
2098  FORMAT(10X,I5,3F10.3,F12.5,F10.3)
2099  FORMAT(15X,'CLOUD BASE FORCING IS ',A4, ';CBFL IS',L2,/)
3000  FORMAT(15X,'SULFATE CHEMISTRY IS ',A4, ';CHMFLG IS',L2,5X,
     1       'ICHUN=',I5,/)
C
C
      END
*DECK CLOUDS
      SUBROUTINE CLOUDS
C
C     THIS SUBROUTINE COMPUTES THE TIME DEPENDENT
C     CLOUD PROPERTIES.  RHOC IS THE DRY AIR DENSITY
C     AND QV, QL ,QP, QS AND QG ARE MIXING RATIOS
C     RATHER THAN SPECIFIC HUMIDITIES.  THE TOTAL
C     CLOUD DENSITY IS
C           RHOO=RHOC*(1+QV+QL+QP+QS+QG).
C
C     ADVECTION IS VIA THE FLUX CORRECTED TRANSPORT
C     ALGORITHM 'ETBFCT' BY JAY P. BORIS
C     (1976, NRL MEMO REPORT 3237, PP. 79)
C
*CALL GRID
*CALL CLOUD
*CALL VARCON
*CALL DUMMY
*CALL WASSER
*CALL WATCON
*CALL EIS
*CALL EISCON
*CALL HELOK
*CALL STORS
*CALL CBLOK
*CALL CHEM
*CALL WFR
*CALL SO4
*CALL H4N
*CALL HP
C
      COMMON/FORCE/ CBASE,CBT,CBQ,ZCBASE,WCB,RADM
      COMMON/AKO/ AZO,AZM
      DIMENSION TVB(101),TVBAV(101)
      DIMENSION ENTA(101,7)
      DIMENSION BIASA(101),BE(2),BQ(2)
      DIMENSION PFA(101),UW(101),UWA(101),DWW(101),STR(101),ENA(101)
      DIMENSION PHIX(101,2),TX(101,2)
      LOGICAL ESKP,CBFL,CHMFLG
      DATA TPS/1.845270149/, RAD2I/0.707106781/, AK1/0.162/, WTI/10.0/,
     1 CTI/0.0/, DTW/10.0/, ICOUNT/0/, ICFL/0.0/, JT/100/
      DATA PFT/2*0.0/, SFT/2*0.0/, GFT/2*0.0/, SUFT/4*0.0/, AMFT/2*0.0/,
     1 PRATE/2*0.0/, SRATE/2*0.0/, GRATE/2*0.0/, PHTOT/2*0.0/
      DATA UNO/101*1.0/, ZERO/101*0.0/ , WA/101*0.0/, WB/101*0.0/,
     1 TVB/101*0.0/,TVBAV/101*0.0/
      DATA AUTO/202*0.0/, COLL/202*0.0/, PEVAP/202*0.0/, COND/202*0.0/,
```

QUATERNARY LIBRARY

```
     1 SINIT/202*0.0/, VSDEP/202*0.0/, SRME/202*0.0/, CSFZ/202*0.0/,
     2 VGDEP/202*0.0/, GRME/202*0.0/, PGCOL/202*0.0/, PGFZ/202*0.0/,
     3 GMELT/202*0.0/, SMELT/202*0.0/, GPSHD/202*0.0/, SGCOL/202*0.0/,
     4 SGCON/202*0.0/, EP/202*0.0/, DEL/202*0.0/, FUP/202*0.0/,
     5 FSIDE/202*0.0/, TKE/202*0.0/, TEP/202*0.0/, DC/202*0.0/,
     6 DP/202*0.0/, VP/202*0.0/, DS/202*0.0/, VS/202*0.0/, DG/202*0.0/,
     7 VG/202*0.0/, BUOY/202*0.0/, DWPDZ/202*0.0/, WS/202*0.0/,
     8 SFLG/202*0.0/, PHIX/202*0.0/, TX/202*0.0/
      DATA D/404*0.0/, FCON/101*0.0/, FARR/101*0.0/, STOR/707*0.0/,
     1 ENTA/707*0.0/, PPA/101*0.0/, UW/101*0.0/, UWA/101*0.0/,
     2 DWW/101*0.0/, STR/101*0.0/, ENA/101*0.0/
      DATA ONR/8.0E+06/, PICON/8141.592654/
C
      ADCO(1)=2.0/ARAD
      ADCO(2)=2.0*BRAD/(BRAD*BRAD-ARAD*ARAD)
      ADCO(3)=-ARAD*ARAD/(BRAD*BRAD-ARAD*ARAD)
      ARB=ARAD/BRAD
      HARAD=0.5*ARAD
      UBCO=0.5*(BRAD*BRAD-ARAD*ARAD)/BRAD
      COFSI=2.*PIE*DZ
      ZBOTBC=Z(1)-0.5*DZ
      ZTOPBC=Z(NLVLS)+0.5*DZ
      WLIM=.01
      ELIM=.1*WLIM
      NP=NLVLS+1
      DZI=1.0/DZ
C     DCCO,DPCO,VPCO USED IN SUBROUTINE DROPS; PICON AND ONR
C     ARE SET IN A DATA STATEMENT ABOVE.  TNC IS SET IN READIN.
      DCCO=(6./(PICON*TNC))**CPO
      DPCO=1./((PICON*ONR)**0.25)
      VPCO=1.09375*130.*SQRT(PIE)
C
      READ(4,*) JPRT
      READ(4,*) IFLX
      READ(4,*) TDUMP
      READ(4,*) ITD
      READ(4,*) AZO
      READ(4,*) AZM
      READ(4,*) KASA
      READ(4,*) AVL(1)
      AVL(1)=AMAX1(50.,AVL(1))
      AVL(2)=AVL(1)
            WRITE(6,1009) IEDCO,JPRT,IFLX,TDUMP,ITD,AZO,AZM
            WRITE(6,1008) KASA
            WRITE(6,1007) AVL(1),AVL(2)
C     JT AND ICTA ARE USED WHEN ITD=1 TO LIMIT THE PRINTED
C     OUTPUT. SEE AFTER LOOP 500.
      IF(ITD.GT.1) THEN
       ICTA=1+(INT(ATIME/3000.0)+INT(100.0/DZ))*INT(200.0/TDUMP)
      ELSE
       ICTA=1
      END IF
C     READ IN CLOUD BASE PRESSURE, TEMP, MOISTURE, VELOCITY
      CALL READCB(CBASE,CBT,CBQ,WCB,RADM,CBFL)
      CALL INTEP(CBASE,ZCBASE,PBAR,Z,NLVLS,DXZ)
            LEVCB=INT(ZCBASE/DZ+0.5)+1
            WRITE(6,1000) CBASE,CBT,CBQ,WCB,RADM,LEVCB
            CALL TRACINT(LEVCB,0.001)
            IF(CHMFLG) CALL CHEMINT(LEVCB)
C     EDCO IS THE COEFFICIENT ARRAY FOR THE EDDY
C     VISCOSITIES, I.E.; KM(J)=EDCO(J)*D(I,K), WHERE
C           EDCO(J)=(AK1*DELTA)**2./SQRT(2)
C     AND DELTA IS THE APPROPRIATE AVERAGING LENGTH
C     FOR:  J=1 (R=0), J=2 (R=ARAD), J=3 (R=0.5*(ARAD+BRAD)),
C           AND J=4 (R=BRAD).
C     D(I,K) IS THE DEFORMATION TENSOR MAGNITUDE AT J.  KM IS
C     CALCULATED IN FLUXINT AND IS USED FOR THE EDDY FLUX TERMS;
C           WX'=KM*DX/DZ, ETC.
C     THE VALUE OF AK1 IS FROM COTTON (1975).  VALUES IN DATA STMNT;
C      TP3=(2*PIE)**0.3333, RAD2I=1.0/SQRT(2), CPO=1/3, AK1=0.162
```



```
      EDCO(1)=RAD2I*(AK1*(PIE*ARAD*ARAD*DZ)**CPO)**2.
      EDCO(2)=RAD2I*(AK1*TP3*ARAD)**2.
      EDCO(3)=RAD2I*(AK1*(PIE*(BRAD*BRAD-ARAD*ARAD)
     1     *DZ)**CPO)**2.
      EDCO(4)=RAD2I*(AK1*TP3)**2.*(BRAD*BRAD-ARAD*ARAD)
C     AVERAGING LENGTH SHOULD BE NO LARGER THAN SUBCLOUD
C     HEIGHT FOR ISOTROPIC TURBULENCE
      DO 4 I=1,4
            EDCO(I)=AMIN1(EDCO(I),ZCBASE)
4     CONTINUE
C     TKECO USED IN SUBROUTINE FLUXINT.  COMMON/VARCON
      TKECO=0.5*(AZM*AK1*AK1/0.7)**0.666667
C
      DO 7 I=1,NLVLS
            BIASA(I)=BIAS
            RHOO(I,1)=RHOBAR(I)*(1.+QBAR(I))
            RHOO(I,2)=RHOBAR(I)*(1.+QBAR(I))
            RHOC(I)=RHOBAR(I)
            TVB(I)=TBAR(I)*(1.+1.608*QBAR(I))
     1           /(1.+QBAR(I))
7     CONTINUE
      DO 8 I=1,NLVLS
            RHOCI(I)=1.0/RHOBAR(I)
            TVBAV(I)=1.0/TVB(I)
8     CONTINUE
C
      WRITE(11) NLVLS,DZ
      CTI=/TDUMP
      IF(ITD.LE.2) CALL CHART(ITD,0.0)
C     INITIALIZE THE PRESSURE PERTURBATION SUBROUTINES
C     PPCO IS AN ENTRY IN SUBROUTINE PPERT
      CALL PPCO
C     INITIALIZE EULERIAN GRID
      CALL NGRIDE(Z,NLVLS,ZBOTBC,ZTOPBC,1)
      CALL OGRIDE(NLVLS)
      CALL NGRIDE(Z,NLVLS,ZBOTBC,ZTOPBC,1)
C
C     START LOOP FOR TIME DEPENDENT CALCULATIONS
10    CONTINUE
      ICOUNT=ICOUNT+1
C
C     CALCULATE THE SUBCLOUD FORCING.  ITFG IS A FLAG INDICATING
C     THE SUBCLOUD FORCING HAS CHANGED.
      CALL FORCER(BE,BQ,TVB,ITFG,CBFL)
      IF(ITFG.EQ.1) THEN
            DO 14 I=1,NLVLS
                  TVBAV(I)=1.0/TVB(I)
14          CONTINUE
      END IF
C     CALCULATE THE CLOUD TOTAL DENSITY AND BUOYANCY
C     FOR THE NEXT TIME STEP
      DO 15 J=1,JLEV
      DO 15 I=1,NLVLS
            RHOO(I,J)=RHOC(I)*(1.0+QV(I,J)+QC(I,J)
     1           +QP(I,J)+QS(I,J)+QG(I,J)
     2           -4.0*BIAS)
            DU2(I)=(QC(I,J)+QP(I,J)+QS(I,J)+QG(I,J)-4.*BIAS)
     1           /(1.0+QV(I,J))
            DU3(I)=TO(I,J)*(1.0+1.608*QV(I,J))
     1           /(1.0+QV(I,J))
            DU4(I)=TVBAV(I)*(DU3(I)-TVB(I))*(1.0-DU2(I))
     1           -DU2(I)
            BUOY(I,J)=G*RHOC(I)*DU4(I)
15    CONTINUE
C     INITIALIZE THE TURBULENT FLUX ARRAYS AND CALCULATE THE
C     TURBULENT KINETIC ENERGY.  FLUXEM IS AN ENTRY CALL TO
C     SUBROUTINE FLUXINT AND RETURNS THE VERTICAL AND LATERAL
C     EDDY FLUXES FOR MOMENTUM.  FLUXES IS ALSO AN ENTRY CALL
C     TO FLUXINT AND RETURNS THE EDDY FLUXES FOR SCALARS.
      CALL FLUXINT(KASA)
```

```
      IF(JPRT.LT.0 .AND. ICFL.LE.3) THEN
           WRITE(6,1011) TIME,ICOUNT
           DO 82 I=1,NLVLS
                WRITE(6,1012) I,Z(I),(D(I,J),J=1,4)
82         CONTINUE
      END IF
C     CALCULATE THE NEW TIME STEP AND RELEVANT TIMES.
C          WTI     THE TIME MARKER TO CALCULATE MICROPHYSICAL
C                  INTERACTIONS.  THESE ARE UPDATED ABOUT
C                  EVERY 10 SECONDS.
C          CTI     THE TIME MARKER FOR DUMPING THE CLOUD OUTPUT
C                  TO A FILE.  SEE AFTER STATEMENT 600.
      CALL DTSET(DT,DZ,NLVLS,JLEV)
      TIME=TIME+DT
      WTI=WTI+DT
      IF(WTI.GE.DTW) WTI=0.0
      CTI=CTI+DT
      ICFL=ICFL+1
C     FIND VELOCITY AT 0.5*DT FOR ADVECTING SCALAR QUANTITIES
C     AND SOLVING THE CONTINUITY EQUATION.  THE SOURCE TERMS
C     ARE:  DU4 - BUOYANCY,  DU5 - VERTICAL EDDY FLUX,
C          DU8 - LATERAL EDDY FLUX, DU9 - DYNAMIC ENTRAINMENT
      DTH=0.5*DT
C     CALCULATE THE ENTRAINMENT VELOCITIES AND FLUX TERMS
C     FOR THE HALF TIME STEP
      CALL EVELO(WC,10,DWPDZ,DTH)
      DO 40 J=1,JLEV
           CALL FLUXEM(J,DWW,UW)
           IF(IPPF.EQ.1) CALL PPERT(J,PPA,WC,DWW,UW,UWA,ICFL)
           CALL FORSUB(J,WC,WC,1,FARR,FCON)
           CALL ENTRAIN(J,WC,WBAR,DU7,DU0)
           IF(J.EQ.1) THEN
                DO 18 I=1,NLVLS
                     STR(I)=UW(I)
                     ENA(I)=DU0(I)
                     UWA(I)=UW(I)
18              CONTINUE
           END IF
           DO 19 I=1,NLVLS
                DU8(I)=0.5+SIGN(0.5,ABS(BUOY(I,J)-PPA(I))-BLIM)
19         CONTINUE
         DO 20 I=1,NLVLS
           DU1(I)=WC(I,J)
           DU2(I)=WC(I,J)*RHOC(I)
           DU4(I)=DU8(I)*(BUOY(I,J)-PPA(I))
           DU8(I)=ADCO(J)*RHOC(I)*(UW(I)-FLOAT(J-1)*ARB*STR(I))
           DU9(I)=DU7(I)*DWPDZ(I,J)-FLOAT(J-1)*ADCO(3)
     1          *(DU7(I)-ENA(I))*DWPDZ(I,1)
20       CONTINUE
         DO 21 I=1,NLVLS
           DU0(I)=(1.0-FCON(I))*(DU4(I)+DWW(I)+DU8(I)+DU9(I))
     1          +FCON(I)*FARR(I)*DWPDZ(I,J)
21       CONTINUE
           CALL VELOCE(DU1,NLVLS,0.0,0.0,DTH)
           CALL SOURCES(NLVLS,DTH,3,UNO,DU0,0.0,0.0)
           CALL ETBFCT(DU2,DU3,NLVLS,-1.0,1.0)
         DO 22 I=1,NLVLS
           DU5(I)=DU8(I)*RHOC(I)*(1.-FCON(I))+FCON(I)*FARR(I)
22       CONTINUE
         DO 23 I=1,NLVLS
           DU1(I)=0.5+SIGN(0.5,ABS(DU5(I))-WLIM)
23       CONTINUE
         DO 24 I=1,NLVLS
           DU8(I)=DU1(I)*DU5(I)
24       CONTINUE
         IF(J.EQ.1) THEN
           DO 25 I=1,NLVLS
             WA(I)=DU8(I)
             WS(I,1)=DU8(I)
25         CONTINUE
```



```
         ELSE
           DO 26 I=1,NLVLS
             WB(I)=DU3(I)
             WS(I,2)=DU3(I)
26         CONTINUE
         END IF
40     CONTINUE
C
C      CALCULATE THE LATERAL ENTRAINMENT VELOCITY AND FLUX TERMS
       CALL EVELO(WS,ICFL,DWPDZ,DT)
C      UPDATE THE MICROPHYSICAL TRANSITIONS IF NECESSARY
       IF(WTI.EQ.0.0) THEN
           DTWI=1.0/(AMAX1(DT,DTW))
           CALL PARKES(DTWI)
           CALL PARICE(DTWI)
       END IF
       IF(CHMFLG) THEN
           CALL CHEMSET(DTWI,LEVCB)
           CALL SUFLX(DT,LEVCB,WS,VP,VS,VG)
       END IF
           CALL TRACER(DT,1,LEVCB,WS)
C
C      ADVECT MOMENTUM, ENERGY AND WATER.  ALL QUANTITIES ARE
C      REFERENCED TO DRY AIR (RHOC).  CALLS TO SOURCES SET UP
C      THE SOURCE TERM ADDED TO THE TRANSPORT EQUATION.
C
       DO 400 J=1,JLEV
C
C      CALCULATE VELOCITY DEPENDENT ARRAYS.
       IF(J.EQ.1) THEN
           CALL VELOCE(WA,NLVLS,0.0,0.0,DT)
       ELSE
           CALL VELOCE(WB,NLVLS,0.0,0.0,DT)
       END IF
C
C            ADVECT CHEMISTRY VARIABLES
       CALL ADTRAC(J,DT,DWPDZ,WS)
       IF(CHMFLG) CALL ADCHEM(J,DT,VP,VS,VG)
C            MOMENTUM
       CALL FLUXEM(J,DWW,UW)
       IF(IPPF.EQ.1) CALL PPERT(J,PPA,WC,DWW,UW,UWA,ICFL)
       CALL FORSUB(J,WS,WC,1,FARR,FCON)
       CALL ENTRAIN(J,WS,WBAR,DU7,DU0)
C      STORE THE INNER HORIZONTAL EDDY FLUX FOR USE IN THE
C      OUTER REGION CALCULATION.
       IF(J.EQ.1) THEN
           DO 125 I=1,NLVLS
             STOR(I,1)=UW(I)
             ENTA(I,1)=DU0(I)
             UWA(I)=UW(I)
125        CONTINUE
       END IF
       DO 134 I=1,NLVLS
           DU3(I)=0.5+SIGN(0.5,ABS(BUOY(I,J)-PPA(I))-BLIM)
134    CONTINUE
       DO 135 I=1,NLVLS
           DU2(I)=RHOC(I)*WC(I,J)
           DU4(I)=DU8(I)*(BUOY(I,J)-PPA(I))
           DU8(I)=ADCO(J)*RHOC(I)*(UW(I)-FLOAT(J-1)*ARB
     1          *STOR(I,1))
           DU9(I)=DU7(I)*DWPDZ(I,J)-FLOAT(J-1)*ADCO(3)
     1          *(DU7(I)-ENTA(I,1))*DWPDZ(I,1)
135    CONTINUE
       DO 136 I=1,NLVLS
           DU0(I)=(1.0-FCON(I))*(DU4(I)+DWW(I)+DU8(I)+DU9(I))
     1          +FCON(I)*FARR(I)*DWPDZ(I,J)
136    CONTINUE
           CALL SOURCES(NLVLS,DT,3,UNO,DU0,0.0,0.0)
           CALL ETBFCT(DU2,DU1,NLVLS,-1.0,1.0)
       DO 137 I=1,NLVLS
```

QUATERNARY LIBRARY

```
          DU5(I)
137   CONTINUE
      DO 138 I=1,
          DU3(I)
138   CONTINUE
      DO 139 I=1,
          WC(I,J)=DU3(I)*DU5(I)
139   CONTINUE
      IF(JPRT.GE.1 .AND. ICFL.LE.3) THEN
          WRITE(6,1015) TIME,ICOUNT
          DO 140 I=1,NLVLS
              WCZ=DU2(I)*RHOCI(I)
              WRITE(6,1020) I,J,Z(I),WCZ,WC(I,J),
     1                DU4(I),DWW(I),DU8(I),
     2                DU0(I),WS(I,J)
140       CONTINUE
      END IF
C
C             ENERGY
      CALL FORSUB(J,WS,PHIC,2,FARR,FCON)
      CALL FLUXES(PHIC,PHIBAR,J,DU5,DU6)
      CALL ENTRAIN(J,PHIC,PHIBAR,DU8,DU0)
      IF(J.EQ.1) THEN
          DO 155 I=1,NLVLS
              STOR(I,2)=DU6(I)
              ENTA(I,2)=DU0(I)
155       CONTINUE
      END IF
      DO 165 I=1,NLVLS
          DU2(I)=RHOC(I)*PHIC(I,J)
          DU7(I)=-RHOC(I)*(VP(I,J)*QP(I,J)*4218.*TC(I,J)
     1          +2106.*AMIN1(TC(I,J),273.16)
     2          *(VS(I,J)*QS(I,J)+VG(I,J)*QG(I,J)))
          DU8(I)=ADCO(J)*RHOC(I)*(DU6(I)-FLOAT(J-1)*ARB
     1          *STOR(I,2))
          DU9(I)=DU3(I)*DWPDZ(I,J)-FLOAT(J-1)*ADCO(8)
     1          *(DU3(I)-ENTA(I,2))*DWPDZ(I,1)
165   CONTINUE
      DO 166 I=1,NLVLS
          DU0(I)=(DU5(I)+DU8(I)+DU9(I))
166   CONTINUE
          CALL SOURCES(NLVLS,DT,8,UNO,DU0,0.0,0.0)
          CALL SOURCES(NLVLS,DT,1,UNO,DU7,DU7(1),DU7(NLVLS))
            BTE=(PHIBAR(NLVLS))/PHIC(NLVLS,J)
          CALL ETBFCT(DU2,DU1,NLVLS,BE(J),BTE)
      DO 170 I=1,NLVLS
          PHIC(I,J)=DU1(I)*RHOCI(I)*(1.-FCON(I))+FCON(I)*FARR(I)
170   CONTINUE
      IF(JPRT.GE.2 .AND. ICFL.LE.3) THEN
          WRITE(6,1021) TIME,ICOUNT
          DO 171 I=1,NLVLS
              PHZ=DU2(I)*RHOCI(I)
              WRITE(6,1020) I,J,Z(I),
     1              PHZ,PHIC(I,J),DU9(I),DU5(I),DU8(I),
     2              DU7(I),DU0(I)
171       CONTINUE
      END IF
C
C   ADVECT THE WATER VARIABLES
C     VAPOR
      CALL FORSUB(J,WS,QV,8,FARR,FCON)
      CALL FLUXES(QV,QBAR,J,DU5,DU6)
      CALL ENTRAIN(J,QV,QBAR,DU7,DU0)
      IF(J.EQ.1) THEN
          DO 185 I=1,NLVLS
              STOR(I,3)=DU6(I)
              ENTA(I,3)=DU0(I)
185       CONTINUE
      END IF
      DO 195 I=1,NLVLS
```



```
          DU2(I)=RHOC(I)*QV(I,J)
          DU4(I)=RHOC(I)*(PEVAP(I,J)-SINIT(I,J)
     1          -VSDEP(I,J)-VGDEP(I,J))
          DU8(I)=ADCO(J)*RHOC(I)*(DU6(I)-FLOAT(J-1)*ARB
     1          *STOR(I,3))
          DU9(I)=DU7(I)*DWPDZ(I,J)-FLOAT(J-1)*ADCO(3)
     1          *(DU7(I)-ENTA(I,3))*DWPDZ(I,1)
195   CONTINUE
      DO 196 I=1,NLVLS
          DU0(I)=(DU4(I)+DU5(I)+DU8(I)+DU9(I))
196   CONTINUE
          CALL SOURCES(NLVLS,DT,3,UNO,DU0,0.0,0.0)
            BQV=QBAR(NLVLS)/QV(NLVLS,J)
          CALL ETBFCT(DU2,DU1,NLVLS,BQ(J),BQV)
      DO 200 I=1,NLVLS
          QV(I,J)=DU1(I)*RHOCI(I)*(1.-FCON(I))+FCON(I)*FARR(I)
200   CONTINUE
      IF(JPRT.GE.3 .AND. ICFL.LE.3) THEN
          WRITE(6,1030) TIME,ICOUNT
          DO 205 I=1,NLVLS
              QVZ=DU2(I)*RHOCI(I)
              WRITE(6,1020) I,J,Z(I),QVZ,QV(I,J),
     1              DU9(I),DU8(I),DU4(I),
     2              DU5(I),DU0(I)
205       CONTINUE
      END IF
C
C   ADVECT THE LIQUID AND ICE TERMS.  BIASA IS USED FOR THE
C   ENVIRONMENTAL VALUES AND IS SET IN LOOP 7.
C       CLOUD DROPS
C   LQF IS FORCER FLAG IF HAVE NEGATIVELY BUOYANT CLOUDBASE
C   SEE SUBROUTINE FORCER.  LQF IN COMMON/VARCON
      CALL FORSUB(J,WS,QC,LQF,FARR,FCON)
      CALL FLUXES(QC,BIASA,J,DU5,DU6)
      CALL ENTRAIN(J,QC,BIASA,DU7,DU0)
      IF(J.EQ.1) THEN
          DO 215 I=1,NLVLS
              STOR(I,4)=DU6(I)
              ENTA(I,4)=DU0(I)
215       CONTINUE
      END IF
      DO 225 I=1,NLVLS
          DU2(I)=RHOC(I)*QC(I,J)
          DU4(I)=RHOC(I)*(SMELT(I,J)-AUTO(I,J)-COLL(I,J)
     1          -SRME(I,J)-CSFZ(I,J)-GRME(I,J))
          DU8(I)=ADCO(J)*RHOC(I)*(DU6(I)-FLOAT(J-1)*ARB
     1          *STOR(I,4))
          DU9(I)=DU7(I)*DWPDZ(I,J)-FLOAT(J-1)*ADCO(3)
     1          *(DU7(I)-ENTA(I,4))*DWPDZ(I,1)
225   CONTINUE
      DO 226 I=1,NLVLS
          DU0(I)=DU4(I)+DU5(I)+DU8(I)+DU9(I)
226   CONTINUE
          CALL SOURCES(NLVLS,DT,3,UNO,DU0,0.0,0.0)
          CALL ETBFCT(DU2,DU1,NLVLS,1.0,1.0)
      DO 230 I=1,NLVLS
          QC(I,J)=AMAX1(DU1(I)*RHOCI(I),BIASA(I))*(1.-FCON(I))
     1          +FCON(I)*FARR(I)
230   CONTINUE
      IF(JPRT.GE.4 .AND. ICFL.LE.3) THEN
          WRITE(6,1031) TIME,ICOUNT
          DO 235 I=1,NLVLS
              QLZ=DU2(I)*RHOCI(I)
              WRITE(6,1020) I,J,Z(I),QLZ,QC(I,J),
     1              DU9(I),DU8(I),DU4(I)
235       CONTINUE
      END IF
C
C     PRECIPITATION
      CALL FLUXES(QP,BIASA,J,DU5,DU6)
```

QUATERNARY LIBRARY

```
      CALL ENTRAIN(J,QP,BIASA,DU3,DU0)
      IF(J.EQ.1) THEN
            DO 245 I=1,NLVLS
                  STOR(I,5)=DU6(I)
                  ENTA(I,5)=DU0(I)
245         CONTINUE
      END IF
      DO 265 I=1,NLVLS
            DU2(I)=RHOC(I)*QP(I,J)
            DU4(I)=RHOC(I)*(AUTO(I,J)+COLL(I,J)-PEVAP(I,J)
     1            -PGFZ(I,J)-PGCOL(I,J)+GMELT(I,J)+GPSHD(I,J))
            DU7(I)=-RHOC(I)*VP(I,J)*QP(I,J)
            DU8(I)=ADCO(J)*RHOC(I)*(DU6(I)-FLOAT(J-1)*ARB
     1            *STOR(I,5))
            DU9(I)=DU8(I)*DWPDZ(I,J)-FLOAT(J-1)*ADCO(8)
     1            *(DU8(I)-ENTA(I,5))*DWPDZ(I,1)
265   CONTINUE
      DO 266 I=1,NLVLS
            DU0(I)=DU4(I)+DU5(I)+DU8(I)+DU9(I)
266   CONTINUE
            CALL SOURCES(NLVLS,DT,8,UNO,DU0,0.0,0.0)
            CALL SOURCES(NLVLS,DT,1,UNO,DU7,DU7(1),DU7(NLVLS))
            CALL ETBFCT(DU2,DU1,NLVLS,1.0,0.0)
      DO 270 I=1,NLVLS
            QP(I,J)=AMAX1(DU1(I)*RHOCI(I),BIASA(I))
270   CONTINUE
            PRATE(J)=(QP(1,J)-BIAS)*RHOC(1)*ABS(VP(1,J))
            PFT(J)=PRATE(J)*DT+PFT(J)
            PFALL=PFALL+PRATE(J)*DT
            IF(CHMFLG) THEN
              IF(HPP(1,J).LT.0.1 .AND. PRATE(J).GT.BIXC) THEN
                  PHTOT(J)=PRATE(J)*HPP(1,J)*DT+PHTOT(J)
                  WRITE(30,9959) J,TIME,DT,PHTOT(J),HPP(1,J),
     1            PRATE(J),PFT(J),QP(1,J),QSO4P(1,J),QNH4P(1,J),
     2            QHSO8P(1,J)
9959  FORMAT(10X,I5,2F10.3,8G12.4)
              END IF
C               0.333, 0.395 CONVERT QSO4, QHSO8 TO S MIXING RATIO
C               QHSO8 IS A FUNCTION WHICH CALCULATES THE
C               IMPLICIT HSO8 FIELD MIXING RATIO
            SUFT(J,1)=SUFT(J,1)+0.333*QSO4P(1,J)*ABS(VP(1,J))*DT
            SUFT(J,2)=SUFT(J,2)+0.395*QHSO8P(1,J)*ABS(VP(1,J))*DT
            AMFT(J)=AMFT(J)+QNH4P(1,J)*ABS(VP(1,J))*DT
          END IF
      IF(JPRT.GE.5 .AND. ICFL.LE.8) THEN
            WRITE(6,1082) TIME,ICOUNT
            DO 275 I=1,NLVLS
                  QPZ=DU2(I)*RHOCI(I)
                  WRITE(6,1020) I,J,Z(I),QPZ,QP(I,J),
     1                  DU9(I),DU8(I),DU4(I),DU7(I)
275         CONTINUE
      END IF
C
C     SNOW
      CALL FLUXES(QS,BIASA,J,DU5,DU6)
      CALL ENTRAIN(J,QS,BIASA,DU3,DU0)
      IF(J.EQ.1) THEN
            DO 285 I=1,NLVLS
                  STOR(I,6)=DU6(I)
                  ENTA(I,6)=DU0(I)
285         CONTINUE
      END IF
      DO 295 I=1,NLVLS
            DU2(I)=RHOC(I)*QS(I,J)
            DU4(I)=RHOC(I)*(SINIT(I,J)+VSDEP(I,J)+SRME(I,J)
     1            +CSFZ(I,J)-SGCOL(I,J)-SGCON(I,J)-SMELT(I,J))
            DU7(I)=-RHOC(I)*VS(I,J)*QS(I,J)
            DU8(I)=ADCO(J)*RHOC(I)*(DU6(I)-FLOAT(J-1)*ARB
     1            *STOR(I,6))
            DU9(I)=DU3(I)*DWPDZ(I,J)-FLOAT(J-1)*ADCO(8)
```



```
     1            *(DU8(I)-ENTA(I,6))*DWPDZ(I,1)
295   CONTINUE
      DO 296 I=1,NLVLS
            DU0(I)=DU4(I)+DU5(I)+DU8(I)+DU9(I)
296   CONTINUE
            CALL SOURCES(NLVLS,DT,8,UNO,DU0,0.0,0.0)
            CALL SOURCES(NLVLS,DT,1,UNO,DU7,DU7(1),DU7(NLVLS))
            CALL ETBFCT(DU2,DU1,NLVLS,1.0,0.0)
      DO 300 I=1,NLVLS
            QS(I,J)=AMAX1(DU1(I)*RHOCI(I),BIASA(I))
300   CONTINUE
            SRATE(J)=(QS(1,J)-BIAS)*RHOC(1)*ABS(VS(1,J))
            SFT(J)=SFT(J)+SRATE(J)*DT
            SFALL=SFALL+SRATE(J)*DT
      IF(JPRT.GE.6 .AND. ICFL.LE.8) THEN
            WRITE(6,1083) TIME,ICOUNT
            DO 305 I=1,NLVLS
                  QSZ=DU2(I)*RHOCI(I)
                  WRITE(6,1020) I,J,Z(I),QSZ,QS(I,J),
     1                  DU9(I),DU8(I),DU4(I),DU7(I)
305         CONTINUE
      END IF
C
C     GRAUPEL
      CALL FLUXES(QG,BIASA,J,DU5,DU6)
      CALL ENTRAIN(J,QG,BIASA,DU8,DU0)
      IF(J.EQ.1) THEN
            DO 315 I=1,NLVLS
                  STOR(I,7)=DU6(I)
                  ENTA(I,7)=DU0(I)
315         CONTINUE
      END IF
      DO 325 I=1,NLVLS
            DU2(I)=RHOC(I)*QG(I,J)
            DU4(I)=RHOC(I)*(PGFZ(I,J)+VGDEP(I,J)+GRME(I,J)
     1            +PGCOL(I,J)+SGCOL(I,J)+SGCON(I,J)-GMELT(I,J)
     2            -GPSHD(I,J))
            DU7(I)=-RHOC(I)*VG(I,J)*QG(I,J)
            DU8(I)=ADCO(J)*RHOC(I)*(DU6(I)-FLOAT(J-1)*ARB
     1            *STOR(I,7))
            DU9(I)=DU8(I)*DWPDZ(I,J)-FLOAT(J-1)*ADCO(8)
     1            *(DU8(I)-ENTA(I,7))*DWPDZ(I,1)
325   CONTINUE
      DO 326 I=1,NLVLS
            DU0(I)=DU4(I)+DU5(I)+DU8(I)+DU9(I)
326   CONTINUE
            CALL SOURCES(NLVLS,DT,8,UNO,DU0,0.0,0.0)
            CALL SOURCES(NLVLS,DT,1,UNO,DU7,DU7(1),DU7(NLVLS))
            CALL ETBFCT(DU2,DU1,NLVLS,1.0,0.0)
      DO 330 I=1,NLVLS
            QG(I,J)=AMAX1(DU1(I)*RHOCI(I),BIASA(I))
330   CONTINUE
            GRATE(J)=(QG(1,J)-BIAS)*RHOC(1)*ABS(VG(1,J))
            GFT(J)=GFT(J)+GRATE(J)*DT
            GFALL=GFALL+GRATE(J)*DT
      IF(JPRT.GE.7 .AND. ICFL.LE.8) THEN
            WRITE(6,1084) TIME,ICOUNT
            DO 335 I=1,NLVLS
                  QGZ=DU2(I)*RHOCI(I)
                  WRITE(6,1020) I,J,Z(I),QGZ,QG(I,J),
     1                  DU9(I),DU8(I),DU4(I),DU7(I)
335         CONTINUE
      END IF
C
C
400   CONTINUE
C
C
C     CALCULATE THE NEW TEMPERATURE, ADJUST THE WATER VARIABLES AND
C     MIXING RATIOS FOR EQUILIBRIUM
```

QUATERNARY LIBRARY

```
      DTY=1.0/DT
      CALL TSPEC(IERR,PHIC,TC,1,DTY)
      IF(IERR.GT.0) THEN
           CALL CHART(1,DT)
           IF(ITD.GT.2) CALL CHART(ITD,DT)
           STOP
      END IF
C     CHECK TIME
      IF(CTI.GE.TDUMP) THEN
           IF(ITD.EQ.1) THEN
                IF(JT.GE.ICTA) THEN
                     CALL CHART(1,DT)
                     JT=0
                ELSE
                     CALL CHART(2,DT)
                     JT=JT+1
                END IF
           ELSE
                CALL CHART(ITD,DT)
           END IF
           CTI=0.0
           ICFL=0
      END IF
      IF (TIME.LT.ATIME .AND. ICOUNT.LT.ITNUM) GO TO 10
      IF(CTI.NE.0.0) THEN
           CALL CHART(1,DT)
           IF(ITD.GT.2) CALL CHART(ITD,DT)
      END IF
      CALL TRACER(DT,2,LEVCB,WS)
      IF(CHMFLG) CALL SDEP(LEVCB,TPMASS)
C
      RETURN
C
C
 1000 FORMAT(10X,'CLOUDBASE PRESSURE,TEMP,Q',2X,2F15.8,F15.6,/,
     1      10X,'WCB,RADM,LEVCB',2X,2F15.4,I5)
 1009 FORMAT(//,15X,'IEDCO,JPRT,IFLX,TDUMP,ITD,AZO,AZM',3I5,F10.3,
     1      I5,2F10.8)
 1010 FORMAT(25X,'(D(I,J):J=',I2,',4) SET TO 0.00',//)
 1008 FORMAT(15X,'KASA=',I3,' 1 FOR LILLY/SMAGORINSKY,'
     1      ,'2 FOR A&KAS EDDY FLUXES')
 1007 FORMAT(15X,'AVL(1)=',F10.3,10X,'AVL(2)=',F10.3)
 1011 FORMAT(15X,'I,Z,D(I,J), TIME=',F8.3,' IC=',I5)
 1012 FORMAT(10X,I5,F10.3,4F15.5)
 1015 FORMAT(//,15X,'I,J,Z(I),WC;IN,OUT,DU4,DU5,DU8,DU0,WS;  TIME='
     1      ,F10.3,5X,'ICOUNT=',I5)
 1020 FORMAT(5X,2I5,F10.3,7G14.3)
 1021 FORMAT(//,15X,'I,J,Z(I),PHIC;IN,OUT;DU9,DU5,DU8,DU7,DU0;  TIME='
     1      ,F10.3,5X,'ICOUNT=',I5)
 1030 FORMAT(//,15X,'I,J,Z(I),QV;IN,OUT,DU9,DU8,DU4,DU5,DU0;  TIME=',
     1      F10.3,' IC=',I5)
 1031 FORMAT(//,15X,'I,J,Z(I),QL;IN,OUT;DU9,DU8,DU4, TIME=',
     1      F10.3,' IC=',I5)
 1032 FORMAT(//,15X,'I,J,Z(I),QP;IN,OUT;DU9,DU8,DU4,DU7,  TIME=',
     1      F10.3,' IC=',I5)
 1033 FORMAT(//,15X,'I,J,Z(I),QS;IN,OUT;DU9,DU8,DU4,DU7,  TIME=',
     1      F10.3,' IC=',I5)
 1034 FORMAT(//,15X,'I,J,Z(I),QG;IN,OUT;DU9,DU8,DU4,DU7,  TIME=',
     1      F10.3,' IC=',I5)
C
C
      END
*DECK EVELO
      SUBROUTINE EVELO(WX,ICFL,DWPDZ,DT)
C
C     THIS SUBROUTINE CALCULATES THE DYNAMIC ENTRAINMENT VELOCITY
C     UC(I,J) AND THE FLUX CONTRIBUTIONS FROM THE SIDE (FSIDE(I,J))
C     AND UPSTREAM (FUP(I,J)) COMPONENTS.
C
C     ENTRY ENTRAIN RETURNS THE TOTAL FLUX CONTRIBUTION FOR
```



```
C     A VARIABLE 'AX'.
C
*CALL GRID
*CALL CLOUD
*CALL VARCON
*CALL DUMMY
*CALL CBLOK
*CALL STORS
C
      DIMENSION AX(101,2), ABAR(101), ENT(101), ELA(101)
      DIMENSION DWPDZ(101,2),WX(101,2)
C
C     CALCULATE THE LATERAL VELOCITIES FROM THE ANELASTIC CONTINUITY
C     EQUATION USING THE MIDPOINT FLUX DIFFERENCE.  WX IS THE VERTICAL
C     VELOCITY.  UC(I,J) IS THE LATERAL VELOCITY ARRAY.  A POSITIVE
C     VALUE INDICATES DIVERGENCE.
      DO 10 I=2,NLVLS
           DU1(I)=0.50*(RHOC(I)*WX(I,1)+RHOC(I-1)*WX(I-1,1))
           DU2(I)=0.50*(RHOC(I)*WX(I,2)+RHOC(I-1)*WX(I-1,2))
 10   CONTINUE
           DU1(1)=0.0
           DU2(1)=0.0
           DU1(NP)=0.0
           DU2(NP)=0.0
      DO 11 I=1,NLVLS
           DWPDZ(I,1)=DZI*(DU1(I+1)-DU1(I))
           DWPDZ(I,2)=DZI*(DU2(I+1)-DU2(I))
 11   CONTINUE
C     CALCULATE UC AND LIMIT TO .GE. WLIM
      DO 12 I=1,NLVLS
           UC(I,1)=HARAD*DWPDZ(I,1)*RHOCI(I)
           UC(I,2)=UBCO*DWPDZ(I,2)*RHOCI(I)+ARB*UC(I,1)
 12   CONTINUE
      DO 13 I=1,NLVLS
           DU3(I)=0.5+SIGN(0.5,ABS(UC(I,1))-WLIM)
           DU4(I)=0.5+SIGN(0.5,ABS(UC(I,2))-WLIM)
 13   CONTINUE
      DO 14 I=1,NLVLS
           UC(I,1)=DU3(I)*UC(I,1)
           UC(I,2)=DU4(I)*UC(I,2)
 14   CONTINUE
C     EP AND DEL ARE USED TO CALCULATE THE LATERAL FLUX
C     CONTRIUBUTIONS TO THE AVERAGED TRANSPORT EQUATION.
C        EP IS POSITIVE FOR AN UPWARD VELOCITY.
C        DEL IS POSITIVE FOR AN OUTWARD VELOCITY (DIVERGENCE).
      DO 15 J=1,JLEV
      DO 15 I=1,NLVLS
           EP(I,J)=SIGN(1.0,WC(I,J))
           DEL(I,J)=SIGN(1.0,UC(I,J))
 15   CONTINUE
C     IFLX IS THE FLAG FOR THE DYNAMIC ENTRAINMENT CALCULATION AND IS
C     TRANSFERRED IN COMMON/VAR/.
C     IFLX =1  USE LATERAL VALUES, =2 US UPSTREAM VALUES
C          =3  USE A FLUX WEIGHTED COMBINATION OF LATERAL AND
C            UPSTREAM VALUES.
C          =4  USE A TEP MIXING TIME WEIGHTED COMBINATION OF LATERAL
C            AND UPSTREAM VALUES.
C          =5  USE A TKE,KEBAR WEIGHTED COMBINATION OF LATERAL AND
C            UPSTREAM VALUES.
C
      GO TO (40,45,50,60,70), IFLX
C
 40   DO 41 J=1,JLEV
      DO 41 I=1,NLVLS
           FUP(I,J)=ZERO(I)
           FSIDE(I,J)=0.5
 41   CONTINUE
           GO TO 100
 45   DO 46 J=1,JLEV
      DO 46 I=1,NLVLS
```

QUATERNARY LIBRARY

```
      FUP(I,J)=0.5
      FSIDE(I,J)=ZERO(I)
46  CONTINUE
      GO TO 100
50  DO 51 I=2,NLVLS-1
C               LATERAL FLUXES
      DU1(I)=COFSI*ARAD*ABS(UC(I,1))
      DU3(I)=COFSI*BRAD*ABS(UC(I,2))
C               UPWIND FLUXES
      DU2(I)=0.5*AREAA*((1.+EP(I,1))*ABS(WC(I-1,1))
     1       +(1.-EP(I,1))*ABS(WC(I+1,1)))
      DU4(I)=0.5*AREAB*((1.+EP(I,2))*ABS(WC(I-1,2))
     1       +(1.-EP(I,2))*ABS(WC(I+1,2)))
51  CONTINUE
C       BOTTOM LEVEL FLUXES
    DU1(1)=COFSI*ARAD*ABS(UC(1,1))
    DU3(1)=COFSI*BRAD*ABS(UC(1,2))
    DU2(1)=0.5*AREAA*(1.-EP(1,1))*ABS(WC(2,1))
    DU4(1)=0.5*AREAB*(1.-EP(2,1))*ABS(WC(2,2))
C       TOP LEVEL FLUXES
    DU1(NLVLS)=COFSI*ARAD*ABS(UC(NLVLS,1))
    DU3(NLVLS)=COFSI*BRAD*ABS(UC(NLVLS,2))
    DU2(NLVLS)=0.5*AREAA*(1.+EP(NLVLS,1))*ABS(WC(NLVLS-1,1))
    DU4(NLVLS)=0.5*AREAB*(1.+EP(NLVLS,2))*ABS(WC(NLVLS-1,2))
C   CALCULATE THE FLUX RATIOS FOR THE LATERAL FLUX TERMS USING
C   THE FLUX WEIGHTED MEANS OF COTTON (1975, JAS, 32, 548-564).
    DO 52 I=1,NLVLS
      IF((DU1(I)+DU2(I)).EQ.0.0) THEN
        FSIDE(I,1)=0.0
        FUP(I,1)=0.0
      ELSE
        FSIDE(I,1)=0.5*DU1(I)/(DU1(I)+DU2(I))
        FUP(I,1)=0.5*DU2(I)/(DU1(I)+DU2(I))
      END IF
      IF((DU3(I)+DU4(I)).EQ.0.0) THEN
        FSIDE(I,2)=0.0
        FUP(I,2)=0.0
      ELSE
        FSIDE(I,2)=0.5*DU3(I)/(DU3(I)+DU4(I))
        FUP(I,2)=0.5*DU4(I)/(DU3(I)+DU4(I))
      END IF
52  CONTINUE
      GO TO 100
C   WEIGHT THE LATERAL ENTRAINMENT USING THE MIXING TIME CALCULATED
C   FROM THE TURBULENT ENERGY DISSIPATION RATE (TEP) CALCULATED IN
C   SUBROUTINE FLUXINT.  TAU IS THE MIXING TIME FROM DIMENSIONAL
C   ARGUMENTS:  TAU=(L*L/TEP)**0.3333 .  THE LENGTH L USED HERE IS
C   A MIXING LENGTH FOR THE INNER AND OUTER REGIONS BOUNDARIES;
C   AVL(1) AND AVL(2), FROM SUBROUTINE CLOUDS.
C   THE FUNCTIONAL FORM OF FSIDE INCORPORATES THE ASSUMPTION THAT
C   THE BOUNDARY REGION BECOMES ENTIRELY MIXED IN TAU SECONDS, AND
C   MIXES EXPONENTIALLY IN SHORTER TIMES.
60  CONTINUE
    DO 65 J=1,JLEV
     DO 61 I=1,NLVLS
       DU1(I)=AMAX1(TEP(I,J),1.0E-07)
61   CONTINUE
     DO 62 I=1,NLVLS
       DU2(I)=(AVL(J)*AVL(J)/DU1(I))**0.3333
62   CONTINUE
     DO 63 I=1,NLVLS
       FSIDE(I,J)=0.5*(1.0-EXP(-5.*DT/DU2(I)))
63   CONTINUE
     DO 64 I=1,NLVLS
       FUP(I,J)=0.5-FSIDE(I,J)
64   CONTINUE
65  CONTINUE
      GO TO 100
C   WEIGHT THE LATERAL ENTRAINMENT WITH THE TURBULENT KINETIC ENERGY
C   VELOCITY (SQRT(TKE*2)).  TKE IS CALCULATED IN SUBROUTINE FLUXINT
```



```
70  CONTINUE
    DO 75 J=1,JLEV
     DO 71 I=1,NLVLS
       DU1(I)=SQRT(TKE(I,J)*2.)
71   CONTINUE
     DO 72 I=1,NLVLS
       DU2(I)=AMAX1(1.0E-06,ABS(WC(I,J))+DU1(I))
72   CONTINUE
     DO 73 I=1,NLVLS
       FSIDE(I,J)=0.5*DU1(I)/DU2(I)
73   CONTINUE
     DO 74 I=1,NLVLS
       FUP(I,J)=0.5-FSIDE(I,J)
74   CONTINUE
75  CONTINUE
C
100 CONTINUE
C
C   CHECK FOR DETRAINMENT; UC .GT. 0 FORCES FUP=0,FSIDE=0.5
    DO 105 I=1,NLVLS
       FUP(I,1)=0.5*(1.0-DEL(I,1))*FUP(I,1)
       FUP(I,2)=0.5*(1.0-DEL(I,2))*FUP(I,2)
105 CONTINUE
    DO 106 I=1,NLVLS
       FSIDE(I,1)=0.5-FUP(I,1)
       FSIDE(I,2)=0.5-FUP(I,2)
106 CONTINUE
    IF(JPRT.GE.1 .AND. ICFL.LE.3) THEN
      WRITE(6,1000) ICOUNT,TIME
      DO 150 I=1,NLVLS
        WRITE(6,1005) I,Z(I),WC(I,1),WC(I,2),UC(I,1),UC(I,2),
     1                EP(I,1),EP(I,2),DEL(I,1),DEL(I,2),
     2                FSIDE(I,1),FUP(I,1),FSIDE(I,2),FUP(I,2)
150     CONTINUE
    END IF
C
    RETURN
C
C
    ENTRY ENTRAIN(K,AX,ABAR,ENT,ELA)
C
C   ENTRAIN CALCULATES THE DYNAMIC ENTRAINMENT USING THE WEIGHTING
C   FACTORS FSIDE AND FUP CALCULATED IN SUBROUTINE EVELO
C       K     REGION TO BE CALCULATED        I
C       AX    ARRAY OF CLOUD VALUES          I
C       ABAR  ARRAY OF ENVIRONMENTAL VALUES  I
C       ENT   DYNAMIC ENTRAINMENT ARRAY      O
C       ELA   LATERAL ENTRAINMENT ARRAY      O
C   NOTE THAT AX IS A MIXING RATIO.
C       CALCULATE THE LATERAL CONTRIBUTION
    DO 200 I=1,NLVLS
       DU1(I)=(1.+DEL(I,K))*AX(I,K)+(1.-DEL(I,K))
     1        *(FLOAT(2-K)*AX(I,2)+FLOAT(K-1)*ABAR(I))
200 CONTINUE
    IF(K.EQ.1 .AND. JLEV.GT.1)  THEN
       DO 201 I=1,NLVLS
         ENT(I)=0.5*DU1(I)
         ELA(I)=0.5*DU1(I)
201     CONTINUE
       RETURN
    END IF
C   CALCULATE THE UPSTREAM CONTRIBUTION
    DO 205 I=2,NLVLS-1
       DU8(I)=(1.+EP(I,K))*0.5*(1+EP(I-1,K))
       DU9(I)=(1.-EP(I,K))*0.5*(1-EP(I+1,K))
205 CONTINUE
       DU8(1)=(1.+EP(1,K))
       DU9(1)=(1.-EP(1,K))*0.5*(1-EP(2,K))
       DU8(NLVLS)=(1.+EP(NLVLS,K))*0.5*(1+EP(NLVLS-1,K))
       DU9(NLVLS)=(1.-EP(NLVLS,K))
```

QUATERNARY LIBRARY

```
      DO 206 I=1,NLVLS
            DU0(I)=2.0-DU8(I)-DU9(I)
  206 CONTINUE
C       USE LATERAL CONTRIBUTION AT CONVERGENCE ZONE
      DO 207 I=2,NLVLS-1
            DU2(I)=DU8(I)*AX(I-1,K)+DU0(I)*0.5*DU1(I)
     1           +DU9(I)*AX(I+1,K)
  207 CONTINUE
            DU2(1)=DU8(1)*ABAR(1)+DU0(1)*0.5*DU1(1)+DU9(1)*AX(2,K)
            DU2(NLVLS)=DU8(NLVLS)*AX(NLVLS-1,K)
     1           +DU0(NLVLS)*0.5*DU1(NLVLS)
     2           +DU9(NLVLS)*ABAR(NLVLS)
      DO 210 I=1,NLVLS
            ENT(I)=FSIDE(I,K)*DU1(I)+FUP(I,K)*DU2(I)
            ELA(I)=ZERO(I)
  210 CONTINUE
C
      RETURN
C
 1000 FORMAT(//,15X,'I,Z,WC,UC,EP,DEL,((FSIDE,FUP) I=1,2), IC=',I5,
     1 ' TIME=',F8.2/)
 1005 FORMAT(5X,I5,F10.3,4E10.4,3X,4F6.1,3X,2(2F8.4,2X))
C
      END
*DECK FLUXINT
      SUBROUTINE FLUXINT(KASA)
C
C     THIS SUBROUTINE CALCULATES THE EDDY FLUX TERMS USING THE
C     SMAGORINSKY DEFORMATION TENSOR FORM OF COTTON (1975).
C     KASA = 1 FOR SMAGORINSKY, KASA = 2 FOR ASAI AND KASAHARA EDDY TERMS
C
*CALL GRID
*CALL CLOUD
*CALL VARCON
*CALL DUMMY
*CALL STORS
*CALL CBLOK
      COMMON/AKO/ AZO,AZM
      DIMENSION RHOI(101),AX(101,2),ABAR(101)
      DIMENSION DWW(101),UW(101),DWA(101),UA(101)
      COMMON/SL/ SLM(101,2),SVM(101,2)
      EQUIVALENCE (RHOI(1),DU0(1))
C       GAML, GAMI ARE THE PSEUDOADIABATIC GRADIENTS FOR WATER
C       AND ICE / GAMMA (DRY).
      GAML(T,P)=(1.0+5419.34*(EW(T)/(T*P)))/(1.0+8388504.46
     1       *(EW(T)/(T*P*T)))
      GAMI(T,P)=(1.0+6140.91*(EI(T)/(T*P)))/(1.0+1077 1020.40
     1       *(EI(T)/(T*P*T)))
C
      GO TO (50,250), KASA
C       CALCULATE THE DEFORMATION TENSOR MAGNITUDES (D(I,J))
C       FOR:    R=0, ARAD, 0.5*(ARAD+BRAD), BRAD
C               J=1  J=2       J=3         J=4
C       D(I,J) AT R=0.  CALCULATE DW/DZ (DU8) AND DRHO/DZ (DU4)
   50 DO 100 I=1,NLVLS
            RHOI(I)=1.0/RHOO(I,1)
            DU1(I)=WC(I,1)
            DU2(I)=RHOO(I,1)
  100 CONTINUE
      DO 125 I=2,NLVLS-1
            DU8(I)=DIFCAL(DU1,I,DZ,NLVLS,1)
            DU4(I)=DIFCAL(DU2,I,DZ,NLVLS,1)
  125 CONTINUE
            DU8(1)=DIFCAL(DU1,1,DZ,NLVLS,4)
            DU4(1)=DIFCAL(DU2,1,DZ,NLVLS,4)
            DU8(NLVLS)=DIFCAL(DU1,NLVLS,DZ,NLVLS,5)
            DU4(NLVLS)=DIFCAL(DU2,NLVLS,DZ,NLVLS,5)
      DO 130 I=1,NLVLS
            DU5(I)=WC(I,1)*DU4(I)*RHOI(I)+DU8(I)
            STOR(I,1)=DU8(I)
```



```
            D(I,1)=SQRT(DU5(I)*DU5(I)+DU8(I)*DU8(I))
  130 CONTINUE
C       CALCULATE D(I,J) AT R=A (K=1), R=0.5*(A+B) (K=2), R=B(K=3)
      DO 195 K=1,3
            J=MIN(K,2)
           GO TO (140,145,150), K
  140       RAD=ARAD
            DELR=1.0/ARAD
            DO 141 I=1,NLVLS
                  DU4(I)=UC(I,1)
                  DU5(I)=(RHOO(I,2)-RHOO(I,1))*DELR
                  DU6(I)=(WC(I,2)-WC(I,1))*DELR
                  DU7(I)=UC(I,1)*DELR
  141       CONTINUE
            GO TO 160
  145       RAD=0.5*(ARAD+BRAD)
            RADI=1.0/RAD
            DELR=1.0/(BRAD-ARAD)
            DO 146 I=1,NLVLS
                  RHOI(I)=1.0/RHOO(I,2)
                  DU4(I)=0.5*(UC(I,1)+UC(I,2))
                  DU5(I)=(RHOBAR(I)-RHOO(I,1))*DELR
                  DU6(I)=(WBAR(I)-WC(I,1))*DELR
                  DU7(I)=0.5*(UC(I,1)+UC(I,2))*RADI
  146       CONTINUE
            GO TO 160
  150       RAD=BRAD
            RADI=1.0/RAD
            DELR=1.0/(BRAD-ARAD)
            DO 151 I=1,NLVLS
                  DU4(I)=UC(I,2)
                  DU5(I)=(RHOBAR(I)-RHOO(I,2))*DELR
                  DU6(I)=(WBAR(I)-WC(I,2))*DELR
                  DU7(I)=UC(I,2)*RADI
  151       CONTINUE
C       CALCULATE THE VERTICAL DERIVATIVES OVER THE SAME
C       SCALE AS THE HORIZONTAL DERIVATIVES USING A 2 POINT
C       CENTERED DIFFERENCE FOR INTERIOR POINTS AND A FORWARD
C       OR BACKWARD DIFFERENCE AT THE END POINTS.
  160       IS=MAX(INT(0.5*((RAD/DZ)-1.0)),1)
            NPA=NP-IS
            DELA=1.0/(FLOAT(IS)*DZ)
            DELZ=0.5*DELA
            DO 165 I=IS+1,NLVLS-IS
                  DU1(I)=(WC(I+IS,J)-WC(I-IS,J))*DELZ
                  DU2(I)=(RHOO(I+IS,J)-RHOO(I-IS,J))*DELZ
                  DU3(I)=(DU4(I+IS)-DU4(I-IS))*DELZ
  165       CONTINUE
            DO 170 I=1,IS
                  BOI=1.0/(FLOAT(I*I*IS+IS*IS*I)*DZ)
                  SI=FLOAT(I*I)
                  SIS=FLOAT(IS*IS)
                  SNPI=FLOAT((NP-I)*(NP-I))
                  DU1(I)=(SI*WC(I+IS,J)-(SI-SIS)*WC(I,J))*BOI
                  DU2(I)=(SI*RHOO(I+IS,J)-(SI-SIS)*RHOO(I,J)
     1                 -SI*RHOO(1,J))*BOI
                  DU3(I)=(SI*DU4(I+IS)-(SI-SIS)*DU4(I))*BOI
                  DU1(NP-I)=((SI-SIS)*WC(NP-I,J)-SI*WC(NP-I-IS,J))
     1                 *BOI
                  DU2(NP-I)=(SIS*RHOO(NLVLS,J)-(SI-SIS)
     1                 *RHOO(NP-I,J)-SI*RHOO(NP-I-IS,J))*BOI
                  DU3(NP-I)=((SI-SIS)*DU4(NP-I)-SI*DU4(NP-I-IS))
     1                 *BOI
  170       CONTINUE
C       DU8=DU/DR, DU7=U/R, DU1=DW/DZ, DU8=DW/DR, DU3=DU/DZ
            DO 180 I=1,NLVLS
                  DU8(I)=RHOI(I)*DU4(I)*DU5(I)+DU7(I)
     1                 +WC(I,J)*RHOI(I)*DU2(I)+DU1(I)
                  D(ILK+1)=SQRT(DU8(I)*DU8(I)+DU7(I)*DU7(I)
     1                 +DU1(I)*DU1(I)+0.5*(DU6(I)+DU3(I))
```

QUATERNARY LIBRARY

```
     1                *(DU8(I)+DU8(I)))
180        CONTINUE
           GO TO (185,190,185), K
C     STORE DU/DR, DW/DR+DU/DZ, OR DW/DZ FOR THE TKE
C     OR MOMENTUM FLUX CALCULATIONS
185        DO 186 I=1,NLVLS
               STOR(I,K+1)=DU8(I)
               STOR(I,J+4)=DU8(I)+DU3(I)
186        CONTINUE
           GO TO 195
190        DO 191 I=1,NLVLS
               STOR(I,K+1)=DU1(I)
191        CONTINUE
195   CONTINUE
C     CALCULATE THE BUOYANCY CONTRIBUTION TO THE EDDY FLUX
C     CALCULATE TV AND GAMMA REF.  GAMMA(DRY)=G/CPD=9.758E-03
C     SFLG IS A SATURATION FLAG (0=UNSATURATED, 1=SATURATED)
C     SET IN LOOP 900 OF TSPEC.
C     D(I,2) AND D(I,4) ARE STORED IN DU8 AND DU9 FOR USE IN
C     CALCULATING THE TURBULENT KINETIC ENERGY (TKE) AND THE
C     DISSIPATION RATE (TEP) IN LOOP 210.
      DO 197 I=1,NLVLS
           DU1(I)=TC(I,1)*(1.+0.608*QV(I,1)-QC(I,1)-QP(I,1)
     1          -QS(I,1)-QG(I,1)+4*BIAS)
           DU2(I)=TC(I,2)*(1.+0.608*QV(I,2)-QC(I,2)-QP(I,2)
     1          -QS(I,2)-QG(I,2)+4*BIAS)
           DU3(I)=9.758E-03*(1.0-SFLG(I,1)+SFLG(I,1)
     1          *(QC(I,1)*GAML(TC(I,1),PBAR(I))+QS(I,1)
     2          *GAMI(TC(I,1),PBAR(I)))/(QC(I,1)+QS(I,1)))
           DU4(I)=9.758E-03*(1.0-SFLG(I,2)+SFLG(I,2)
     1          *(QC(I,2)*GAML(TC(I,2),PBAR(I))+QS(I,2)
     2          *GAMI(TC(I,2),PBAR(I)))/(QC(I,2)+QS(I,2)))
           DU8(I)=D(I,2)
           DU9(I)=D(I,4)
197   CONTINUE
C     RI=G*(D/DZ(TV)+GAMMA(REF))/(TV*DIJ*DIJ)
C     CORRECTION IS SQRT(1.0-KH/KM*RI)=SQRT(1.-(AZO/AZM)*RI)
      DO 198 I=2,NLVLS-1
           SVM(I,1)=G*(DU3(I)+DIFCAL(DU1,I,DZ,NLVLS,1))/DU1(I)
           SVM(I,2)=G*(DU4(I)+DIFCAL(DU2,I,DZ,NLVLS,1))/DU2(I)
198   CONTINUE
           SVM(1,1)=G*(DU8(1)+DIFCAL(DU1,1,DZ,NLVLS,4))/DU1(1)
           SVM(1,2)=G*(DU4(I)+DIFCAL(DU2,1,DZ,NLVLS,4))/DU2(1)
           SVM(NLVLS,1)=G*(DU3(NLVLS)+DIFCAL(DU1,NLVLS,DZ,NLVLS,5))
     1          /DU1(NLVLS)
           SVM(NLVLS,2)=G*(DU4(NLVLS)+DIFCAL(DU2,NLVLS,DZ,NLVLS,5))
     1          /DU2(NLVLS)
C     CALCULATE KM FROM THE D(I,J) ARRAY.  EDCO(I) IS DEFINED IN
C     SUBROUTINE CLOUDS AND CONTAINS THE APPROPRIATE CONSTANTS AND
C     AVERAGED LENGTH FOR EACH REGION.
C       KM=((K1*DELTA)**2./SQRT(2))*D(I,J)*SQRT(1-(AZO/AZM)*RI)
C     IF (1.-(AZO/AZM)*RI) .LE. 1.0 NO BUOYANCY CORRECTION IS MADE
      DO 205 K=1,4
           LA=(K+1)/2
       DO 200 I=1,NLVLS
           DU5(I)=SVM(LLA)/(AMAX1(EIAS,(D(I,K)*D(I,K))))
200    CONTINUE
       DO 201 I=1,NLVLS
           D(I,K)=EDCO(K)*D(I,K)
     1          *SQRT(AMAX1(1.0,(1.0-(AZO/AZM)*DU5(I))))
201    CONTINUE
205   CONTINUE
C     CALCULATE THE TURBULENT KINETIC ENERGY (TKE) AND THE DISSIPATION
C     RATE (TEP).  DU8 AND DU9 ARE DEFINED IN LOOP 197.
C     THE TKE IS CALCULATED FROM DEARDORFF'S (1973) EQN. TAKEN FROM
C     LIPPS (J.A.S., 34, 1751-1772, 1977, SEE EQNS. 11-14).
C          TKE=0.5*(C*C/CE)**.667*DELTA*DELTA*DIJ*DIJ
C     CE IS SET TO 0.7 BELOW.  TEP IS CALCULATED USING THE FORM FROM
C     TENNEKES AND LUMLEY, EQN. 3.2.5;  TEP=2*KM*SIJ*SIJ, WHERE
C     SIJ=0.5*DIJ.  AVL(1), AVL(2) ARE THE MIXING LENGTHS FOR THE
```



```
C     INNER AND OUTER BOUNDARIES.  TKE AND TEP ARE CALCULATED HERE FOR
C     THE REGION BOUNDARIES.  AVL AND TKECO ARE DEFINED IN CLOUDS.
C     TKECO=0.5*(AZM*AK1*AK1/0.7)**0.66667 FROM COMMON/VARCON.
      DO 210 I=1,NLVLS
           TKE(I,1)=TKECO*AVL(1)*AVL(1)*DU8(I)*DU8(I)
     1          *SQRT(AMAX1(1.0,(1.0-(AZO/AZM)
     2          *SVM(I,2)/(AMAX1(BIAS,(DU8(I)*DU8(I)))))))
           TEP(I,1)=0.5*D(I,1)*DU8(I)*DU8(I)
           TKE(I,2)=TKECO*AVL(2)*AVL(2)*DU9(I)*DU9(I)
     1          *SQRT(AMAX1(1.0,(1.0-(AZO/AZM)
     2          *SVM(I,2)/(AMAX1(BIAS,(DU9(I)*DU9(I)))))))
           TEP(I,2)=0.5*D(I,3)*DU9(I)*DU9(I)
210   CONTINUE
C      STORE DERIVATIVES FOR USE IN ENTRY FLUXEM
      DO 215 I=1,NLVLS
           SVM(I,1)=STOR(I,1)
           SVM(I,2)=STOR(I,3)
           SLM(I,1)=STOR(I,5)
           SLM(I,2)=STOR(I,6)
215   CONTINUE
      RETURN
C
C      ASAI AND KASAHARA EDDY FORMULATION
250   CONTINUE
      DO 255 I=1,NLVLS
           DU1(I)=WC(I,1)
           DU2(I)=WC(I,2)
           DU5(I)=ABS(WC(I,1)-WC(I,2))
           DU6(I)=ABS(WC(I,2)-WBAR(I))
           SLM(I,1)=WC(I,2)-WC(I,1)
           SLM(I,2)=WBAR(I)-WC(I,2)
255   CONTINUE
      DO 260 I=2,NLVLS-1
           DU3(I)=ABS(DIFCAL(DU1,I,DZ,NLVLS,1))
           DU4(I)=ABS(DIFCAL(DU2,I,DZ,NLVLS,1))
260   CONTINUE
           DU3(1)=ABS(DIFCAL(DU1,1,DZ,NLVLS,4))
           DU4(1)=ABS(DIFCAL(DU2,1,DZ,NLVLS,4))
           DU3(NLVLS)=ABS(DIFCAL(DU1,NLVLS,DZ,NLVLS,5))
           DU4(NLVLS)=ABS(DIFCAL(DU2,NLVLS,DZ,NLVLS,5))
      DO 265 I=1,NLVLS
           D(I,1)=DZ*DZ*DU3(I)
           D(I,2)=0.1*ARAD*DU5(I)
           D(I,3)=DZ*DZ*DU4(I)
           D(I,4)=0.1*(ERAD-ARAD)*DU6(I)
265   CONTINUE
      RETURN
C
      ENTRY FLUXEM(JA,DWW,UW)
C
C      FLUXEM CALCULATES THE VERTICAL AND LATERAL MOMENTUM EDDY
C      FLUXES FOR REGION JA.
C           DWW=-D/DZ(RHOC*W'W') = D/DZ (RHOC*KM*(DW/DZ))
C            UW=U'W'
C      DU5=2*(DW/DZ) IN LOOP 300
      DO 300 I=1,NLVLS
           DU5(I)=2.0*D(I,2*JA-1)
           DU6(I)=D(I,2*JA)*SLM(I,JA)
300   CONTINUE
      DO 315 I=1,NLVLS
           DU5(I)=AZM*DU5(I)
           UW(I)=AZM*DU6(I)
315   CONTINUE
C      EVALUATE D/DZ (RHO*W'W') USING 2ND ORDER CENTERED
C      DIFFERENCE FOR SECOND DERIVATIVE.
      DO 320 I=2,NLVLS
           DU1(I)=0.5*(WC(I,JA)+WC(I-1,JA))
           DU2(I)=0.5*(RHOC(I)*DU5(I)+RHOC(I-1)*DU5(I-1))
320   CONTINUE
           DU1(1)=0.0
```

QUATERNARY LIBRARY

```
      DU2(1)=0.0
      DU1(NP)=0.0
      DU2(NP)=0.0
    DO 325 I=1,NLVLS
      DU3(I)=(DU1(I+1)-DU1(I))*(DU2(I+1)-DU2(I))
325  CONTINUE
    DO 330 I=2,NLVLS-1
      DU1(I)=RHOC(I)*DU5(I)*(WC(I+1,JA)-2.*WC(I,JA)
   1        +WC(I-1,JA))
330  CONTINUE
      DU1(1)=RHOC(1)*DU5(1)*(WC(2,JA)-2.*WC(1,JA))
      DU1(NLVLS)=RHOC(NLVLS)*DU5(NLVLS)*(WC(NLVLS-1,JA)
   1          -2.*WC(NLVLS,JA))
    DO 335 I=1,NLVLS
      DWW(I)=DZI*DZI*(0.25*DU3(I)+DU1(I))
335  CONTINUE
    RETURN
C
    ENTRY FLUXES(AX,ABAR,L,DWA,UA)
C
C   FLUXES CALCULATES THE VERTICAL AND LATERAL SCALAR EDDY FLUXES;
C      AX  = CLOUD DOMAIN FIELD                    I
C      ABAR = ENVIRONMENTAL FIELD                  I
C      L   = REGION                                I
C      DWA  = -D/DZ (RHOC*(W'AX'))= D/DZ (RHOC*KA*DAX/DZ)   O
C      UA  =  U'AX' =KA*DAX/DR                     O
C   AZO/AZM IS THE TURBULENT PRANDTL NUMBER.
    DO 410 I=1,NLVLS
      DU5(I)=D(I,2*L-1)
      DU6(I)=D(I,2*L)*(FLOAT(2-L)*(AX(I,2)-AX(I,1))
   1        *ARI+FLOAT(L-1)*(ABAR(I)-AX(I,2))*BRI)
410  CONTINUE
    DO 415 I=1,NLVLS
      DU5(I)=AZO*DU5(I)
      UA(I)=AZO*DU6(I)
415  CONTINUE
C     CALCULATE VERTICAL EDDY FLUX USING 2ND ORDER CENTERED
C     DIFFERENCE FOR SECOND DERIVATIVE.
    DO 420 I=2,NLVLS
      DU1(I)=0.5*(AX(I,L)+AX(I-1,L))
      DU2(I)=0.5*(RHOC(I)*DU5(I)+RHOC(I-1)*DU5(I-1))
420  CONTINUE
      DU1(1)=ABAR(1)
      DU2(1)=0.0
      DU1(NP)=ABAR(NLVLS)
      DU2(NP)=0.0
    DO 425 I=1,NLVLS
      DU3(I)=(DU1(I+1)-DU1(I))*(DU2(I+1)-DU2(I))
425  CONTINUE
    DO 430 I=2,NLVLS-1
      DU1(I)=RHOC(I)*DU5(I)*(AX(I+1,L)-2.*AX(I,L)
   1        +AX(I-1,L))
430  CONTINUE
      DU1(1)=RHOC(1)*DU5(1)*(AX(2,L)-2.*AX(1,L)
   1        +ABAR(1))
      DU1(NLVLS)=RHOC(NLVLS)*DU5(NLVLS)*(AX(NLVLS-1,L)
   1          -2.*AX(NLVLS,L)+ABAR(NLVLS))
    DO 435 I=1,NLVLS
      DWA(I)=DZI*DZI*(0.25*DU3(I)+DU1(I))
435  CONTINUE
    RETURN
    END
*DECK TSPEC
    SUBROUTINE TSPEC(IERR,PHI,TX,ITX,DTI)
C
C   THIS SUBROUTINE SOLVES THE ENTHALPY EQUATION USING A ZERO
C   CROSSING ITERATIVE TECHNIQUE.
C      IERR   ERROR FLAG                           O
C         =0    TEMPERATURE INVERSION  SUCCESSFUL
C         =1    ERROR IN INVERSION
```



```
C      PHI    TOTAL ENERGY ARRAY                   I
C      TX     ITERATED TEMPERATURE ARRAY           O
C      ITX    FLAG FOR TEMPERATURE                 I
C         =0     ONLY CALC TX
C         =1     CALC,TX,WATER VARIABLES
C      DTI    1/DT (DT IS THE CURRENT TIME STEP)   I
C   VARIABLES USED:
C      P      REFERENCE PRESSURE (PA)
C      TG     GUESS TEMPERATURE (K)
C      ISET   GUESS THERMODYNAMICS
C      IFLG   THERMODYNAMICS USED
C   WHERE ISET, IFLG=
C         1, 1   SUBSATURATED, NO WATER OR ICE
C         2, 2   SATURATED, NO ICE
C         3, 3   SATURATED, ICE AND WATER
C         4, 4   SATURATED, NO WATER
C         5, 0   SUBSATURATED, RAIN AND GRAUPEL
C        10-20   ERRORS IN ITERATIVE LOOPS
C   SUBSATURATED, RAIN AND GRAUPEL (ISET=5) YIELDS IFLG=0
C   RATHER THAN IFLG=5 TO KEEP SFLG=0 FOR UNSATURATED AIR
C   SFLG IS USED IN FLUXINT LOOP 197.
C   COND(I,J) IS THE NET CONDENSATION RATE;  QC(NEW)-QC(OLD)*DTI
C
*CALL GRID
*CALL CLOUD
*CALL VARCON
*CALL HELOK
*CALL WASSER
    DIMENSION PHI(101,2),TX(101,2)
    DO 900 J=1,JLEV
    DO 900 I=1,NLVLS
      QPS=QP(I,J)
      QGS=QG(I,J)
      QVS=QV(I,J)
      QWS=QC(I,J)
      QIS=QS(I,J)
      QTOT=QVS+QWS+QIS+QPS+QGS
      THG=(PHI(I,J)-2.501E+06*QVS-0.5*WC(I,J)*WC(I,J)
   1      +0.344E+06*(QGS+QIS))
   2      /(1005.+4218.*QTOT)
      TKG=THG*(PBAR(I)*1.0E-05)**(0.2854)
      HREF=H(TKG,QVS,QWS+QPS,QIS+QGS)
      TG=TX(I,J)-1.5
        ISET=3
        IF((QGS+QIS).LE.BIXL) ISET=2
        IF((QPS+QWS).LE.BIXL) ISET=4
        IF(TG.GE.273.16) ISET=2
        IF((QWS+QIS).LE.BIXL) ISET=5
        IF((QWS+QPS+QIS+QGS).LE.BIXL) ISET=1
      DELTAT=0.5
      TCL=TG-DELTAT
      HOLD=(1005.+4218.*(QVS+QWS+QPS))*TCL
   1      +4218.*(QIS+QGS)*AMIN1(TCL,273.16)
   2      +2.501E+06*QVS-0.334E+06*(QIS+QGS)
C
    GO TO (100,200,300,400,500) ISET
C
C     SUBSATURATED
100  IF((QPS+QGS).GT.BIXL) GO TO 500
    QWC=BIAS
    QIC=BIAS
    QVC=AMAX1(QTOT-QWC-QIC-QPS-QGS,BIAS)
    TCL=(HREF-2.501E+06*QVC)/(1005.+4218.*QVC)
C     CHECK FOR SATURATION
    IF(QVC.LE.(0.622*EW(TCL)/(PBAR(I)-EW(TCL)))) THEN
        IFLG=1
        GO TO 850
    END IF
    TCL=TCL-10.*DELTAT
    HOLD=(1005.+4218.*(QVS+QWS+QPS))*TCL+2.501E+06*QVS
```

QUATERNARY LIBRARY

```
C     SATURATED, NO ICE
200   CONTINUE
      QIC=BIAS
      DO 250 K=1,100
           TCL=TCL+DELTAT
           QVC=AMIN1((QTOT-QPS-QGS-QIC-BIAS),
     1       (0.622*EW(TCL)/(PBAR(I)-EW(TCL))))
           QWC=QTOT-QPS-QGS-QIC-QVC
           HFUNC=(1005.+4218.*QTOT)*TCL+2.501E+06*QVC
           IF(ABS(HFUNC-HREF).LE.0.05) THEN
                IFLG=2
                GO TO 850
           END IF
           IF(((HFUNC-HREF)/(HOLD-HREF)).LE.0.00)
     1          DELTAT=0.5*DELTAT
           HOLD=HFUNC
250   CONTINUE
      IFLG=10
      WRITE(6,1000) I,J,ISET,IFLG,TCL,HOLD,HREF,DELTAT
           GO TO 850
C     SATURATED, WATER AND ICE
300   WA=AMAX1(0.0,(QWS+QPS-BIX2)/(QWS+QIS+QPS+QGS-4.*BIAS))
      EA=AMAX1(0.0,(QIS+QGS-BIX2)/(QWS+QIS+QPS+QGS-4.*BIAS))
      DO 350 K=1,100
           TCL=TCL+DELTAT
           QVC=AMIN1((QTOT-QPS-QGS-BIX2)
     1       ,(0.622*(EA*EI(TCL)+WA*EW(TCL))
     2       /(PBAR(I)-EA*EI(TCL)-WA*EW(TCL))))
           QWC=(QTOT-QPS-QGS-QVC)*WA+BIAS
           QIC=(QTOT-QPS-QGS-QVC)*EA+BIAS
           HFUNC=(1005.+4218.*(QVC+QWC+QPS))*TCL
     1       +4218.*(QIC+QGS)*AMIN1(TCL,273.16)
     2       +2.501E+06*QVC-0.334E+06
     3       *(QIC+QGS)
           IF(ABS(HFUNC-HREF).LE.0.05) THEN
                IF(TCL.LT.238.0) THEN
                     QTOT=QVS+QWS+QIS+QPS+QGS
                     DELTAT=0.5
                     TCL=TG-DELTAT
                     HOLD=H(TCL,QVS,QWS+QPS,QIS+QGS)
                     GO TO 400
                END IF
                IF(TCL.GE.273.16) THEN
                     QTOT=QVS+QWS+QIS+QPS+QGS
                     DELTAT=0.5
                     TCL=TG-DELTAT
                     HOLD=H(TCL,QVS,QWS+QPS,QIS+QGS)
                     GO TO 200
                END IF
                IFLG=8
                GO TO 850
           END IF
           IF(((HFUNC-HREF)/(HOLD-HREF)).LT.0.00)
     1          DELTAT=0.5*DELTAT
           HOLD=HFUNC
350   CONTINUE
      IFLG=15
      WRITE(6,1000) I,J,ISET,IFLG,TCL,HOLD,HREF,DELTAT
           GO TO 850
C     SATURATED ICE, NO WATER
400   CONTINUE
      QWC=BIAS
      DO 450 K=1,100
           TCL=TCL+DELTAT
           QVC=AMIN1((QTOT-QGS-QPS-QWC-BIAS),
     1       (0.622*EI(TCL)/(PBAR(I)-EI(TCL))))
           QIC=QTOT-QGS-QVC-QPS-QWC
           HFUNC=(1005.+4218.*QVC)*TCL
     1       +4218.*(QIC+QGS)*AMIN1(TCL,273.16)
     2       +2.501E+06*QVC-0.334E+06
```



```
     3       *(QIC+QGS)
C
           IF(ABS(HFUNC-HREF).LE.0.05) THEN
                IFLG=4
                GO TO 850
           END IF
           IF(((HFUNC-HREF)/(HOLD-HREF)).LT.0.00)
     1          DELTAT=0.5*DELTAT
           HOLD=HFUNC
450   CONTINUE
      IFLG=20
      WRITE(6,1000) I,J,ISET,IFLG,TCL,HOLD,HREF,DELTAT
           GO TO 850
C     SUBSATURATED, RAIN AND GRAUPEL
500   IF((QPS+QGS).LE.BIXL) GO TO 100
      QWC=BIAS
      QIC=BIAS
      QVC=AMAX1(QTOT-QPS-QGS-QWC-QIC,BIAS)
      TCL=(HREF-2.501E+06*QVC+0.334E+06*QGS)
     1   /(1005.+4218.*QTOT)
C     CHECK FOR TCL ABOVE FREEZING
C     -4218.*273.16+0.334E+06=-818188.88
      IF(TCL.GT.273.16)
     1       TCL=(HREF-2.501E+06*QVC-818188.88*QGS)
     2         /(1005.+4218.*(QVC+QPS))
C     CHECK FOR SATURATION
      EWX=((QGS/(QPS+QGS))*EI(TCL)+(QPS/(QPS+QGS))
     1   *EW(TCL))
      IF(QVC.GT.(0.622*EWX/(PBAR(I)-EWX))) THEN
           QTOT=QVS+QWS+QIS+QPS+QGS
           DELTAT=0.5
           TCL=TG-DELTAT
           HOLD=H(TCL,QVS,QWS+QPS,QIS+QGS)
           GO TO 300
      END IF
      IFLG=0
C
850   CONTINUE
C
           IF(IFLG.GE.10) THEN
                WRITE(6,1005) ICOUNT,TIME
                WRITE(6,1006) QVS,QWS,QPS,QIS,QGS,TG
                WRITE(6,1007) QVC,QWC,QIC
                IERR=1
           END IF
           IF(ITX.EQ.1) THEN
                QV(I,J)=QVC
                QC(I,J)=QWC
                QS(I,J)=QIC
                SFLG(I,J)=AMAX1(0.0,SIGN(1.0,(FLOAT(IFLG)-1.5)))
                COND(I,J)=(QWC-QWS)*DTI
           END IF
           TX(I,J)=TCL
C
900   CONTINUE
      RETURN
C
C
1000  FORMAT(/,10X,'ERROR IN TSPEC, I,J,ISET,IFLG,TCL,HOLD,HREF,DELTAT'
     1    ,/,25X,4I5,F12.4,2F15.8,F15.6)
1005  FORMAT(25X,'ICOUNT,TIME',I5,F15.5)
1006  FORMAT(25X,'QVS,QWS,QPS,QIS,QGS,TG',5E10.4,F10.3)
1007  FORMAT(25X,'QVC,QWC,QIC',3E10.4,/)
      END
*DECK BESSN
      FUNCTION BESSN(X,N)
C
C     THIS FUNCTION CALCULATES THE BESSEL FUNCTION OF INTEGER
C     ORDER N AT X USING THE IMSL LIBRARY FUNCTION MMBSJN.
C          X      ARGUMENT OF BESSEL FUNCTION    I
```

QUATERNARY LIBRARY

```
C           N      ORDER OF BESSEL FUNCTION       I
C    NOTE THAT N MUST BE .LE. 25 UNLESS THE DIMENSION OF BJ IS
C    INCREASED.
C
      DIMENSION BJ(26)
C
      IF(N.GT.25) THEN
            WRITE(6,1000) N
            STOP
      END IF
      CALL MMBSJN(X,N+1,BJ,IRI)
      IF(IRI.EQ.129 .OR. IRI.EQ.130) THEN
            WRITE(6,1005) IRI,X,N
            STOP
      END IF
C
      BESSN=BJ(N+1)
      RETURN
C
1000  FORMAT(10X,'***** ERROR IN BESSN *****  N=',I5)
1005  FORMAT(10X,'***** ERROR IN BESSN *****  IRI,X,N=',I5,G10.4,I5)
C
      END
*DECK BMODE
      SUBROUTINE BMODE(ARAD,BRAD,XNJ,XNI)
C
C    BMODE COMPUTES THE ORDER AND ZERO OF THE BESSEL MODE FOR
C    TO OUTER REGION PRESSURE PERTURBATION USING THE IMSL
C    FUNCTIONS MMBSJR (REAL ORDER), AND MMBSJN (INTEGER ORDER)
C           XNJ     ZERO OF BESSEL FCN RETURNED
C           XNI     ORDER OF BESSEL FCN RETURNED
C    THE RADIAL DEPENDENCE IS ALSO CALCULATED.
C
      DIMENSION BJ(100),WK(200)
      DATA CP/0.8333333333/, CM/-0.3333333333/
C
      DO 55 I=1,100
            BJ(I)=0.0
55    CONTINUE
      DO 56 I=1,200
            WK(I)=0.0
56    CONTINUE
      ALPHA=(ARAD+BRAD)/(2.*BRAD)
      DELTAX=1.0
      XN=2.0-DELTAX
      YN=XN**(-.333333333)
      YN2=YN*YN
      BETA=XN+1.8557571*XN**(0.333333)+YN*(1.033150-YN2*(.00892
     1   +YN2*(.0908-YN2*.043)))
      FRAC=XN-FLOAT(INT(XN))
      NR=INT(XN)+1
      ARG=ALPHA*BETA
         CALL MMBSJR(ARG,FRAC,NR,BJ,WK,IRR)
      FOLD=XN*BJ(NR)-ARG*BJ(NR-1)
      IF(ABS(FOLD).LE.1.0E-05) GO TO 200
      DO 100 I=1,100
            XN=XN+DELTAX
            YN=XN**(-.333333333)
            YN2=YN*YN
            BETA=XN+1.8557571*XN**(0.333333)+YN*(1.033150-YN2
     1         *(.00892+YN2*(.0908-YN2*.043)))
            FRAC=XN-FLOAT(INT(XN))
            NR=INT(XN)+1
            ARG=ALPHA*BETA
               CALL MMBSJR(ARG,FRAC,NR,BJ,WK,IRR)
            FNEW=XN*BJ(NR)-ARG*BJ(NR-1)
            IF(ABS(FNEW).LE.1.0E-05) GO TO 200
            IF(FNEW/FOLD .LT. 0.00) DELTAX=-0.5*DELTAX
            FOLD=FNEW
100   CONTINUE
```



```
         WRITE(6,1040)
            STOP
200   CONTINUE
      BJF=BJ(NR)
      XNI=FLOAT(INT(XN+0.5))
      XNJ=XNI+1.8557771*(XNI)**CP+1.033150*(XNI)**CM-.00897/XNI
      NRN=INT(XNI)+1
         CALL MMBSJN(ALPHA*XNJ,NRN,BJ,IEX)
      WRITE(6,1050)
      WRITE(6,1051) I,XN,BETA,BJF,XNI,XNJ,BJ(NRN)
      BJMI=1./BJF
      BJNI=1.0/BJ(NRN)
      FRAC=XN-FLOAT(INT(XN))
      NR=INT(XN)+1
      WRITE(6,1100) XNI,XNJ
      WRITE(6,1101)
      DO 250 I=1,25
            ARGR=FLOAT(I)*0.05
            ARG=ARGR*BETA
            ARGN=ARGR*XNJ
               CALL MMBSJR(ARG,FRAC,NR,BJ,WK,IRR)
            BJG=BJ(NR)
               CALL MMBSJN(ARGN,NRN,BJ,IRR)
            WRITE(6,1105) I,ARGR,ARG,BJG,BJG*BJMI,ARGN,BJ(NRN),
     1                  BJ(NRN)*BJNI
250   CONTINUE
      RETURN
C
1040  FORMAT(15X,'*** BMODE ERROR IN LOOP 100 ***')
1050  FORMAT(20X,'BMODE ITERATED SOLUTION; XN,KN,BJ(KN*(A+B)/2B,XN)',
     1       '  XNI,XNJ,BJ(XNJ*(A+B)/2B,XNI),')
1051  FORMAT(10X,I5,2(3F15.5,5X))
1100  FORMAT(30X,'RADIAL DEPENDENCE OF OUTER REGION',
     1     ' BESSEL MODE; INTEGER MODE RETURNED, XNI,XNJ=',2F12.5)
1101  FORMAT(10X,'I,R/B, (REAL,INT):  KN*R/B,BJ(KN*R/B,XN),BJ(KN*',
     1     'R/B,XN)/BJ(KN*(A+B)/2B,XN)')
1105  FORMAT(10X,I5,F10.4,2(3F13.6,5X))
      END
*DECK EPDS
      SUBROUTINE EPDS(X,N,ALPH,BETH,GAMM,DELT,DZ,BBC,UBC)
C
C    THIS SUBROUTINE SOLVES THE EQUATION
C     ALPH*(D/DZ(D/DZ (X))) + BETH*D/DZ (X) - GAMM*X = DELT
C    FOR ALPH(Z), BETH(Z), GAMM(Z) AND F(Z) ON A VERTICAL
C    GRID WITH THE UPPER AND LOWER BOUNDARY CONDITIONS
C    SPECIFIED BY EITHER X OR DX/DZ AT THE TOP AND BOTTOM.
C    THE EQUATION IS SOLVED USING A MARCHING TECHNIQUE.
C    SEE RICHTMYER AND MORTON (1967), PP.198-201, OR
C    CHENG(1981), PP. 206-207.
C    ARGUMENTS:
C           X      SOLUTION ARRAY                O
C           N      NUMBER OF GRID POINTS           I
C           ALPH   REAL ARRAY                    I
C           BETH   REAL ARRAY                    I
C           GAMM   REAL ARRAY                    I
C           DELT   FORCING FUNCTION ARRAY          I
C           DZ     VERTICAL GRID SIZE            I
C           BBC    LOWER BOUNDARY COND.            I
C           UBC    UPPER BOUNDARY COND.            I
C    THE BOUNDARY CONDITIONS ARE ESTABLISHED BY A CALL TO
C    ENTRY PDBCO(KNB,KNU)
C
      DIMENSION X(N),ALPH(N),BETH(N),GAMM(N),DELT(N)
      DIMENSION A(101),B(101),C(101),D(101),E(101),F(101)
      DIMENSION ZA(101),ZB(101)
      LOGICAL LBC,TBC
      COMMON/EPDST/ LBC,TBC,EL,FL
C
C     CALCULATE THE COEFFICIENT ARRAYS FOR THE FINITE DIFFERENCE
C     SOLUTION OF -A(I)*X(I+1)+B(I)*X(I)-C(I)*X(I-1)=D(I)
```

QUATERNARY LIBRARY

```
      DO 100 I=1,N
            ZA(I)=1.0/ALPH(I)
  100 CONTINUE
      DO 101 I=1,N
            ZB(I)=DZ*ZA(I)
  101 CONTINUE
C     NOTE:  ZB(I)*BETH(I)=BETH(I)*DZ/ALPH(I)
      DO 102 I=1,N
            A(I)=1.0+(0.5*ZB(I)*BETH(I))
            B(I)=2.+GAMM(I)*DZ*DZ*ZA(I)
            C(I)=1.0-(0.5*ZB(I)*BETH(I))
            D(I)=-DZ*ZB(I)*DELT(I)
  102 CONTINUE
C     CALCULATE THE LOWER BOUNDARY E AND F.
      IF(LBC) THEN
            EL=0.0
            FL=BBC
      ELSE
            EL=1.0
            FL=-BBC*DZ
      END IF
      E(1)=A(1)/(B(1)-C(1)*EL)
      DO 105 I=2,N
            E(I)=A(I)/(B(I)-C(I)*E(I-1))
  105 CONTINUE
C     NOTE:  E(I)/A(I)=1.0/(B(I)-C(I)*E(I-1))
      F(1)=(D(1)+C(1)*FL)*E(1)/A(1)
      DO 110 I=2,N
            F(I)=(D(I)+C(I)*F(I-1))*E(I)/A(I)
  110 CONTINUE
C     FROM UPPER BC CALCULATE X(N) AND MARCH BACKWARDS
C     WITH THE SOLUTION
      IF(TBC) THEN
            X(N)=UBC*E(N)+F(N)
      ELSE
            X(N)=(E(N)*UBC*DZ+F(N))/(1.-E(N))
      END IF
      DO 120 I=N-1,1,-1
            X(I)=X(I+1)*E(I)+F(I)
  120 CONTINUE
      RETURN
C
      ENTRY PDBCO(KNB,KNU)
C
C     ENTRY PDBCO SETS THE BOUNDARY CONDITION FLAGS FOR EPDS.
C           KNB    FLAG FOR LOWER B.C.            I
C                  =1 FOR X
C                  =2 FOR DX/DZ
C           KNU    FLAG FOR UPPER B.C.            I
C                  =1 FOR X
C                  =2 FOR DX/DZ
C
      IF(KNB.EQ.1) THEN
            LBC=.TRUE.
      ELSE
            LBC=.FALSE.
      END IF
      IF(KNU.EQ.1) THEN
            TBC=.TRUE.
      ELSE
            TBC=.FALSE.
      END IF
        RETURN
C
      END
*DECK PPERT
      SUBROUTINE PPERT(J,PPA,W,DWW,UW,UWA,ICFX)
C
C     THIS SUBROUTINE CALCULATES THE PRESSURE PERTURBATION
C     PI FROM THE DIAGNOSTIC EQUATION
```



```
C     DEL*DEL (PI) = D/DZ B - F(Z)
C     FOR BUOYANCY B AND A FORCING FUNCTION F(Z).
C           J      REGION                  I
C           PPA    CP*THETA*RHO*D/DZ PI    O
C           W      VELOCITY ARRAY          I
C           DWW    -D/DZ(RHO*W'W') ARRAY   I
C           UW     U'W' ARRAY              I
C           UWA    INNER REGION U'W' ARRAY I
C     ENTRY PPCO CALCULATES USEFUL CONSTANTS AND INITIALIZES
C     THE BESSEL MODE ARRAYS.  PPCO IS CALLED FROM SUBROUTINE
C     CLOUDS.
C
*CALL GRID
*CALL CLOUD
*CALL VARCON
*CALL STORS
*CALL CBLOK
*CALL DUMMY
      DIMENSION PPA(101),W(101,2),DWW(101),UW(101),UWA(101)
      DIMENSION F(101)
      COMMON /AKX/ AKAA(101),AKBB(101),ACA,ACB,BCA,BCB,BCC,BCD,RI,
     1  UUA(101),DUA(101),DUUA(101),AJFNC(2),AJFNI(2),ACX(2),
     2  BAI,BA,BBA,BAB,BBB
      COMMON/EPDST/ LBC,TBC,EL,FL
C
      GO TO (100,200), J
C
C     INNER REGION
  100 DO 105 I=1,NLVLS
            DU1(I)=BUOY(I,1)+DWW(I)
            DU2(I)=RHOC(I)*WC(I,1)*WC(I,1)
            DU3(I)=RHOC(I)*(3.*UC(I,1)*UC(I,1)-2.*D(I,2)*UC(I,1)
     1           *ARI)
            DU4(I)=RHOC(I)*(UC(I,1)*WC(I,1)+UW(I))
  105 CONTINUE
      DO 106 I=2,NLVLS-1
            DU5(I)=DU1(I)-DU1(I-1)
            DU6(I)=(DU2(I+1)-2.*DU2(I)+DU2(I-1))*DZI*DZI
            DU7(I)=DU4(I)-DU4(I-1)
  106 CONTINUE
            DU5(1)=DU1(1)
            DU6(1)=(DU2(2)-2.*DU2(1))*DZI*DZI
            DU7(1)=DU4(1)
            DU5(NLVLS)=DU1(NLVLS)-DU1(NLVLS-1)
            DU6(NLVLS)=(-2.*DU2(NLVLS)+DU2(NLVLS-1))*DZI*DZI
            DU7(NLVLS)=DU4(NLVLS)-DU4(NLVLS-1)
      DO 107 I=1,NLVLS-1
            DU8(I)=(DU5(I+1)+DU5(I))*0.5*DZI
            DU9(I)=(DU7(I+1)+DU7(I))*0.5*DZI
  107 CONTINUE
            DU8(NLVLS)=DU5(NLVLS)*0.5*DZI
            DU9(NLVLS)=DU7(NLVLS)*0.5*DZI
      DO 108 I=1,NLVLS
            F(I)=0.5*ARAD*ARAD*(DU8(I)-DU6(I))-DU3(I)-2.*ARAD*DU9(I)
  108 CONTINUE
      DO 109 I=1,NLVLS
            F(I)=9.9502E-04*AJFNI(J)*F(I)
            DU0(I)=AKAA(I)
  109 CONTINUE
       EBC=ARAD*ARAD*0.5*BUOY(1,1)*AJFNI(J)*9.9502E-04/CPT(1)
       UBC=0.0
      GO TO 300
C
C     OUTER REGION
  200 DO 205 I=1,NLVLS
            DU1(I)=BBB*UC(I,2)-BAB*UC(I,1)
            DU2(I)=ARAD*BRAD*(BBB*UC(I,1)-BAB*UC(I,2))
  205 CONTINUE
      DO 206 I=1,NLVLS
            DU3(I)=-2.*D(I,4)*(DU1(I)-BRI*BRI*DU2(I))
```

```
      DO 100 I=1,N
            ZA(I)=1.0/ALPH(I)
  100 CONTINUE
      DO 101 I=1,N
            ZB(I)=DZ*ZA(I)
  101 CONTINUE
C       NOTE:  ZB(I)*BETH(I)=BETH(I)*DZ/ALPH(I)
      DO 102 I=1,N
            A(I)=1.0+(0.5*ZB(I)*BETH(I))
            B(I)=-2.+GAMM(I)*DZ*DZ*ZA(I)
            C(I)=1.0-(0.5*ZB(I)*BETH(I))
            D(I)=-DZ*ZB(I)*DELT(I)
  102 CONTINUE
C       CALCULATE THE LOWER BOUNDARY E AND F.
      IF(LBC) THEN
            EL=0.0
            FL=BBC
      ELSE
            EL=1.0
            FL=-BBC*DZ
      END IF
      E(1)=A(1)/(B(1)-C(1)*EL)
      DO 105 I=2,N
            E(I)=A(I)/(B(I)-C(I)*E(I-1))
  105 CONTINUE
C       NOTE:  E(I)/A(I)=1.0/(B(I)-C(I)*E(I-1))
      F(1)=(D(1)+C(1)*FL)*E(1)/A(1)
      DO 110 I=2,N
            F(I)=(D(I)+C(I)*F(I-1))*E(I)/A(I)
  110 CONTINUE
C       FROM UPPER BC CALCULATE X(N) AND MARCH BACKWARDS
C       WITH THE SOLUTION
      IF(TBC) THEN
            X(N)=-UBC*E(N)+F(N)
      ELSE
            X(N)=(E(N)*UBC*DZ+F(N))/(1.-E(N))
      END IF
      DO 120 I=N-1,1,-1
            X(I)=X(I+1)*E(I)+F(I)
  120 CONTINUE
      RETURN
C
      ENTRY PDBCO(KNB,KNU)
C
C       ENTRY PDBCO SETS THE BOUNDARY CONDITION FLAGS FOR EPDS.
C             KNB    FLAG FOR LOWER B.C.          I
C                    =1 FOR X
C                    =2 FOR DX/DZ
C             KNU    FLAG FOR UPPER B.C.          I
C                    =1 FOR X
C                    =2 FOR DX/DZ
C
      IF(KNB.EQ.1) THEN
            LBC=.TRUE.
      ELSE
            LBC=.FALSE.
      END IF
      IF(KNU.EQ.1) THEN
            TBC=.TRUE.
      ELSE
            TBC=.FALSE.
      END IF
        RETURN
C
      END
*DECK PPERT
      SUBROUTINE PPERT(J,PPA,W,DWW,UW,UWA,ICFX)
C
C     THIS SUBROUTINE CALCULATES THE PRESSURE PERTURBATION
C     PI FROM THE DIAGNOSTIC EQUATION
```



```
C       DEL*DEL (PI) = D/DZ B - F(Z)
C     FOR BUOYANCY B AND A FORCING FUNCTION F(Z).
C             J       REGION                  I
C             PPA     CP*THETA*RHO*D/DZ PI    O
C             W       VELOCITY ARRAY          I
C             DWW     -D/DZ(RHO*W'W') ARRAY   I
C             UW      U'W' ARRAY              I
C             UWA     INNER REGION U'W' ARRAY I
C     ENTRY PPCO CALCULATES USEFUL CONSTANTS AND INITIALIZES
C     THE BESSEL MODE ARRAYS.  PPCO IS CALLED FROM SUBROUTINE
C     CLOUDS.
C
*CALL GRID
*CALL CLOUD
*CALL VARCON
*CALL STORS
*CALL CBLOK
*CALL DUMMY
      DIMENSION PPA(101),W(101,2),DWW(101),UW(101),UWA(101)
      DIMENSION F(101)
      COMMON /AKX/ AKAA(101),AKBB(101),ACA,ACB,BCA,BCB,BCC,BCD,RI,
     1   UUA(101),DUA(101),DUUA(101),AJFNC(2),AJFNI(2),ACX(2),
     2   BAI,BA,BBA,BAB,BBB
      COMMON/EPDST/ LBC,TBC,EL,FL
C
      GO TO (100,200), J
C
C       INNER REGION
  100 DO 105 I=1,NLVLS
            DU1(I)=BUOY(I,1)+DWW(I)
            DU2(I)=RHOC(I)*WC(I,1)*WC(I,1)
            DU3(I)=RHOC(I)*(8.*UC(I,1)*UC(I,1)-2.*D(I,2)*UC(I,1)
     1           *ARI)
            DU4(I)=RHOC(I)*(UC(I,1)*WC(I,1)+UW(I))
  105 CONTINUE
      DO 106 I=2,NLVLS-1
            DU5(I)=DU1(I)-DU1(I-1)
            DU6(I)=(DU2(I+1)-2.*DU2(I)+DU2(I-1))*DZI*DZI
            DU7(I)=DU4(I)-DU4(I-1)
  106 CONTINUE
            DU5(1)=DU1(1)
            DU6(1)=(DU2(2)-2.*DU2(1))*DZI*DZI
            DU7(1)=DU4(1)
            DU5(NLVLS)=DU1(NLVLS)-DU1(NLVLS-1)
            DU6(NLVLS)=(-2.*DU2(NLVLS)+DU2(NLVLS-1))*DZI*DZI
            DU7(NLVLS)=DU4(NLVLS)-DU4(NLVLS-1)
      DO 107 I=1,NLVLS-1
            DU8(I)=(DU5(I+1)+DU5(I))*0.5*DZI
            DU9(I)=(DU7(I+1)+DU7(I))*0.5*DZI
  107 CONTINUE
            DU8(NLVLS)=DU5(NLVLS)*0.5*DZI
            DU9(NLVLS)=DU7(NLVLS)*0.5*DZI
      DO 108 I=1,NLVLS
            F(I)=0.5*ARAD*ARAD*(DU8(I)-DU6(I))-DU3(I)-2.*ARAD*DU9(I)
  108 CONTINUE
      DO 109 I=1,NLVLS
            F(I)=9.9502E-04*AJFNI(J)*F(I)
            DU0(I)=AKAA(I)
  109 CONTINUE
      BBC=ARAD*ARAD*0.5*BUOY(1,1)*AJFNI(J)*9.9502E-04/CPT(1)
      UBC=0.0
      GO TO 300
C
C       OUTER REGION
  200 DO 205 I=1,NLVLS
            DU1(I)=BBB*UC(I,2)-BAB*UC(I,1)
            DU2(I)=ARAD*ERAD*(BBB*UC(I,1)-BAB*UC(I,2))
  205 CONTINUE
      DO 206 I=1,NLVLS
            DU3(I)=-2.*D(I,4)*(DU1(I)-ERI*ERI*DU2(I))
```

QUATERNARY LIBRARY

```
          DU4(I)=-2.*D(I,2)*(DU1(I)-ARI*ARI*DU2(I))
206   CONTINUE
      DO 207 I=1,NLVLS
          DU5(I)=BUOY(I,2)+DWW(I)
          DU6(I)=RHOC(I)*WC(I,2)*WC(I,2)
          DU7(I)=RHOC(I)*(UC(I,2)*UC(I,2)+2.*(DU1(I)*(BRAD*UC(I,2)
     1          -ARAD*UC(I,1))-DU2(I)*(BRI*UC(I,2)-ARI*UC(I,1))
     2          +DU3(I)-DU4(I)))
          DU8(I)=RHOC(I)*(BRAD*(UC(I,2)*WC(I,2)+UW(I))-ARAD
     1          *(UC(I,1)*WC(I,2)+UWA(I)))
207   CONTINUE
      DO 208 I=2,NLVLS-1
          DU1(I)=DU5(I)-DU5(I-1)
          DU2(I)=(DU6(I+1)-2.*DU6(I)+DU6(I-1))*DZI*DZI
          DU3(I)=DU8(I)-DU8(I-1)
208   CONTINUE
          DU1(1)=DU5(1)
          DU2(1)=(DU6(2)-2.*DU6(1))*DZI*DZI
          DU3(1)=DU8(1)
          DU1(NLVLS)=DU5(NLVLS)-DU5(NLVLS-1)
          DU2(NLVLS)=(-2.*DU6(NLVLS)+DU6(NLVLS-1))*DZI*DZI
          DU3(NLVLS)=DU8(NLVLS)-DU8(NLVLS-1)
      DO 209 I=1,NLVLS-1
          DU4(I)=(DU1(I+1)+DU1(I))*0.5*DZI
          DU5(I)=(DU3(I+1)+DU3(I))*0.5*DZI
209   CONTINUE
          DU4(NLVLS)=DU1(NLVLS)*0.5*DZI
          DU5(NLVLS)=DU3(NLVLS)*0.5*DZI
      DO 210 I=1,NLVLS
          F(I)=0.5*BA*(DU4(I)-DU2(I))-DU7(I)-2.*DU5(I)
210   CONTINUE
      DO 211 I=1,NLVLS
          F(I)=9.9502E-04*AJFNI(J)*F(I)
          DU0(I)=AKBB(I)
211   CONTINUE
       BBC=0.5*BA*BUOY(1,2)*AJFNI(J)*9.9502E-04/CPT(1)
       UBC=0.0
C
300   CALL EPDS(DU1,NLVLS,CPT,DCPT,DU0,F,DZ,BBC,UBC)
C
C       P1 IS RETURNED IN DU1, CALCULATE CPT*D/DZ (P1)
      DUL=EL*DU1(1)+FL
      DUT=0.0
       DU3(1)=DU1(1)-DUL
       DU3(NLVLS+1)=DUT-DU1(NLVLS)
      DO 305 I=2,NLVLS
          DU3(I)=DU1(I)-DU1(I-1)
305   CONTINUE
      DO 306 I=1,NLVLS
          DU2(I)=(DU3(I+1)+DU3(I))*0.5*DZI
306   CONTINUE
      DO 307 I=1,NLVLS
          DU2(I)=DU2(I)*(0.5+SIGN(0.5,ABS(DU2(I))-1.0E-10))
307   CONTINUE
      DO 310 I=1,NLVLS
          PPA(I)=1005.*CPT(I)*2.*ACX(J)*AJFNC(J)*DU2(I)
310   CONTINUE
      IF(ICFXLE.8 .AND. JPRT.GE.1) THEN
        DO 890 I=2,NLVLS-1
          DU3(I)=(DU1(I+1)-2.*DU1(I)+DU1(I-1))*DZI*DZI
890     CONTINUE
          DU3(1)=(DU1(2)-2.*DU1(1)+DUL)*DZI*DZI
          DU3(NLVLS)=(DUT-2.*DU1(NLVLS)+DU1(NLVLS-1))*DZI*DZI
        WRITE(6,1100) BBC,UBC
        DO  900 I=1,NLVLS
          SIG=CPT(I)*DU3(I)+DCPT(I)*DU2(I)-DU0(I)*DU1(I)
          WRITE(6,1105) I,CPT(I),DU3(I),DCPT(I),DU2(I),DU0(I)
     1                 ,DU1(I),SIG,F(I)
900     CONTINUE
      END IF
```



```
      RETURN
C
      ENTRY PPCO
C
      ACA=-6./(ARAD*ARAD)
      ACB=4./ARAD
      BCA=1./(BRAD-ARAD)
      BCB=ARAD*BCA
      BCC=BRAD*BCA
      BCD=4./(ARAD+BRAD)
      RI=2./(ARAD+BRAD)
      BA=BRAD*BRAD-ARAD*ARAD
      BAI=1.0/BA
      BAB=ARAD*BAI
      BBB=BRAD*BAI
      BBA=BRAD*BRAD+ARAD*ARAD
C     CALCULATE THE OUTER REGION BESSEL MODE.
      CALL BMODE(ARAD,BRAD,BK,XK)
      NK=INT(XK)
C     2.405 IS K(0,1) FOR J0
      BKA=2.405/ARAD
      BKB=BK/BRAD
      DO 400 I=1,NLVLS
          AKAA(I)=BKA*BKA*CPT(I)
          AKBB(I)=BKB*BKB*CPT(I)
400   CONTINUE
C     SET BOUNDARY CONDITION FLAGS FOR PRESSURE PERTURBATION
C     EQUATION SOLUTION
      CALL PDBCO(2,1)
C     CALCULATE THE INTEGRATED BESSEL FUNCTION FROM R=0 TO R=ARAD
C     AND R=ARAD TO R=BRAD
      DO 410 I=1,21
          DU1(I)=ARAD*FLOAT(I-1)*0.05
          DU2(I)=ARAD+(BRAD-ARAD)*FLOAT(I-1)*0.05
410   CONTINUE
      DO 415 I=1,21
          DU3(I)=DU1(I)*BESSN(BKA*DU1(I),0)
          DU4(I)=DU2(I)*BESSN(BKB*DU2(I),NK)
415   CONTINUE
      AJFNC(1)=SUMER(DU1,DU3,1,21)
      AJFNC(2)=SUMER(DU2,DU4,1,21)
      AJFNI(1)=1.0/AJFNC(1)
      AJFNI(2)=1.0/AJFNC(2)
      ACX(1)=1.0/(ARAD*ARAD)
      ACX(2)=BAI
       WRITE(6,1110) AJFNC(1),AJFNC(2),ACX(1),ACX(2)
C
1100  FORMAT(/,20X,'IN PPERT;  CPT,D2/DZ2(P), DCPT,DP/DZ, DU0,F, SIG,F',
     1  5X,'(BBC,UBC)=',2G15.6)
1105  FORMAT(10X,I5,4(2G12.4,3X))
1110  FORMAT(10X,'AJFNC,ACX',2G15.5,5X,2G15.5)
      END
*DECK EW
      FUNCTION EW(TZ)
C
C     THIS FUNCTION CALCULATES THE PARTIAL PRESSURE OF VAPOR OVER
C     WATER AND ICE.  THE INPUT TEMPERATURES (TZ) IS IN KELVIN.
C     EW AND EI ARE IN PASCALS.
C
      REAL TEI(9),PEI(9)
      DATA TEI/253.16,248.16,243.16,238.16,233.16,228.16,223.16,218.16,213.16,
     1        208.16,203.16,198.16,
     1        193.16,173.16/
      DATA PEI/103.173671,37.964242,12.829779,3.963072,1.080,0.2614,
     1        0.05468,9.665E-03,1.402E-03/
C
C       THIS FORMULA IS FROM BOLTON (1980) AND IS ACCURATE
C       TO 0.3% FROM 238.16 TO 308.16.  THE FIT BEYOND THIS
C       RANGE IS LESS ACCURATE (0.32% AT 223.16) WHEN COMPARED
C       TO THE FORMULA OF LOWE AND FICK (1974).
      EW=611.2*EXP((17.67*TZ-4826.7373)/(TZ-29.66))
```

QUATERNARY LIBRARY

```
      RETURN
C
      ENTRY EI(TZ)
C
C     FOR TZ BETWEEN 273.16 AND 223.16 EI IS CALCULATED FROM
C     THE FORMULA OF LOWE AND FICK (1974) FROM PRUPPACHER AND
C     KLETT, P.625.  THE FACTOR OF 100. CONVERTS FROM MB TO PA.
C     FOR TZ BELOW 223.16 EI IS DETERMINED VIA INTERPOLATION OF
C     THE DATA FROM IRIBARNE AND GODSON (1981,P.246).  EI IN ARRAY
C     PEI FOR T=253.16,248.16,233.16 AND 223.16 IS CALCULATED USING
C     THE LOWE AND FICK FORMULA.
      IF(TZ.LT.223.16) THEN
            CALL INTEP(TZ,EI,TEI,PEI,9,IZ)
            RETURN
      END IF
      T=AMIN1((TZ-273.16) ,0.0)
        EI=100.*(6.109177956
     1    +T*(5.034698970E-01
     2    +T*(1.886013408E-02
     3    +T*(4.176223716E-04
     4    +T*(5.824720280E-06
     5    +T*(4.838803174E-08
     6    +T*(1.838826904E-10)))))))
      RETURN
      END
*DECK H
      FUNCTION H(T,QV,QW,QI)
C
C     THIS FUNCTION CALCULATES THE ENTHALPY OF A SYSTEM OF DRY
C     AIR, WATER VAPOR (QV), LIQUID (QW), AND ICE (QI) AT T (K).
C
        H=(1005.+4218.*(QV+QW))*T+4218.*QI*AMIN1(T,273.16)
     1   +2.501E+06*QV-0.334E+06*QI
C
      RETURN
      END
*DECK FORCER
      SUBROUTINE FORCER(BE,BQ,TV,IT,CBFL)
C
C     THIS SUBROUTINE CALCULATES THE SUBCLOUD FORCING.
C     PTIME .LT. 0.00 SETS A VELOCITY SUBCLOUD FORCING
C     FOR TIME=ABS(PTIME). A VALUE OF PTIME
C     .GT. 0.0 YIELDS A MOISTURE AND ENERGY PERTURBATION
C     FOR PTIME SECONDS.   PTIME .EQ. 0.00 USES THE INPUT
C     WBAR FOR THE CLOUD INITIALIZATION.
C           BE    THE ENERGY LOWER BC                O
C           BQ    THE WATER VAPOR LOWER BC           O
C           TV    THE ENVIRONMENTAL VIRTUAL TEMP.    O
C           IT    FLAG FOR NEW TV                    O
C                 =1 FOR NEW TV
C                 =0 FOR NO CHANGE
C           CBFL  FLAG FOR SUBCLOUD VELOCITY FORCING  I
C                 =T FORCING ON
C                 =F GAUSSIAN PERTURBATION
C     ENTRY FORSUB CALCULATES THE SUBCLOUD FORCING FOR
C     CBFL .EQ. TRUE
C
C
*CALL GRID
*CALL CLOUD
*CALL VARCON
*CALL DUMMY
*CALL DATAIN
*CALL CBLOK
C
      COMMON/FORCS/ JT,IPSET,JTA,JTB,RTIME,RTIMI,CBL,JTX,WCBX(2)
      COMMON/FORCE/ CBASE,CBT,CBQ,ZCBASE,WCB,RADM
C
      LOGICAL IPSET,CBL,CBFL
      DIMENSION FARR(101), W(101,2), X(101,2),FCON(101)
```



```
      DIMENSION DP(100),DW(100),DQ(100)
      DIMENSION BE(2),BQ(2),TV(101),JTFG(10)
C
      EQUIVALENCE (TIN(1),DP(1)),(ZIN(1),DW(1)),(CIN(1),DQ(1))
C
C
      IT=0
      BE(1)=1.0
      BE(2)=1.0
      BQ(1)=1.0
      BQ(2)=1.0
C
      IF(PTIME.EQ.0.00) RETURN
C
      IF(TIME.EQ.0.00) THEN
            RTIME=50.0
            RTIMI=1.0/RTIME
            IPSET=.TRUE.
            CBL=CBFL
            DO 50 I=1,100
                  WIN(I)=0.0
                  PIN(I)=0.0
                  QIN(I)=0.0
                  DP(I)=0.0
                  DW(I)=0.0
                  DQ(I)=0.0
 50         CONTINUE
C
      THCB=CBT*(1.0E+05/CBASE)**(0.2854)
      THCBX=THCB
      PCB=H(THCB,CBQ,0.,0.)+0.5*WCB*WCB
C     PCB=H(CBT,CBQ,0.,0.)+0.5*WCB*WCB+G*ZCBASE*(1./(1+CBQ))
      JTX=INT(ZCBASE/DZ+0.5)+1
      JT=JTX/2
      JTY=JTX
      TXL=CBT
      QVXL=CBQ
      QLXL=0.0
      TQVB=THETAQ(PBAR(JTX),CBT,CBQ,0.0)
      DO 60 I=1,10
        JTFG(I)=0
        THCBV=THCBX*(1.0+1.608*QVXL)*(1.-QLXL)/(1.+QVXL)
        THBV=TBAR(JTY)*((1.0E+05/PBAR(JTY))**(0.2854))*(1.0
     1     +1.608*QBAR(JTY))/(1.+QBAR(JTY))
          WRITE(6,9000) I,THCBV,THBV
9000        FORMAT(10X,'IN FORCER; THCBV,THBV',I5,2F15.4)
        IF(THCBV.GT.THBV) THEN
            JTFG(I)=1
            IF(JTFG(I-1).EQ.1 .AND. I.GE.2) THEN
              JTY=JTY-1
              GO TO 65
            END IF
          END IF
            JTY=JTY+1
            CALL QTHET(TXL,QVXL,QLXL,TQVB,CBQ,PBAR(JTY))
            THCBX=TXL*((1.0E+05/PBAR(JTY))**(0.2854))
 60   CONTINUE
C
 65   CONTINUE
      JFG=0
C     LQF IS THE FLAG TO SET FORSUB FOR QL IN CLOUDS; IN COMMON/VARCON
      LQF=5
      IF(JTY.NE.JTX) THEN
            JFG=1
            LQF=4
            WRITE(6,1003) JTX,JTY
1003          FORMAT(10X,'CLOUDBASE IS NEGATIVELY BUOYANT; SUBCLOUD'
     1             ,' FORCING MOVED FROM LEV',I4,' TO LEV',I4)
            JTX=JTY
      END IF
```

QUATERNARY LIBRARY

```
C
      IF(JLEV.EQ.1) THEN
            RADX=ARAD
      ELSE
            RADX=BRAD
      END IF
      JTA=INT((ZCBASE+RADM*RADX)/DZ)+1
      JTB=INT((ZCBASE+(RADM+0.5)*RADX)/DZ)+1
      JTB=MIN(JTB,NLVLS-8)
      JTA=MIN(JTA,JTB-8)
      TEBASE=THETAE(CBASE,CBT,CBQ)
      WRITE(6,1000) CBASE,CBT,CBQ,WCB,ZCBASE,JT,TEBASE
      WRITE(6,1001) CBFL,CEL
C
      IF(CEL) THEN
            DO 75 J=1,JLEV
            DO 75 I=1,NLVLS
                  WC(I,J)=WBAR(I)
                  PHIC(I,J)=PHIBAR(I)
                  QV(I,J)=QBAR(I)
                  TC(I,J)=TBAR(I)
75          CONTINUE
            WCEX(1)=WCB
            WCEX(2)=WCB
            DO 80 J=1,JLEV
            DO 80 I=1,JTX
                  WC(I,J)=WCEX(J)*FLOAT(I)/(FLOAT(JTX))
                  PHIC(I,J)=PCB
                  IF(Z(I).GT.ZCBASE .AND. JFG.EQ.1) THEN
                    CALL QTHET(TXL,QVXL,QLXL,TQVB,CBQ,PBAR(I))
                    QV(I,J)=QVXL
                    QC(I,J)=QLXL
                  ELSE
                    QV(I,J)=CBQ
                    QC(I,J)=BIAS
                  END IF
80          CONTINUE
            SIGDZ=-DZ*DZ/(RADM*RADM*RADX*RADX)
            ELIMA=ALOG(BIAS)
            DO 85 J=1,JLEV
C           DO 85 I=JTX,NLVLS
            DO 85 I=JTX+1,NLVLS
                  ESIG=EXP(AMAX1(FLOAT(I-JTX)*FLOAT(I-JTX)
     1                        *SIGDZ,ELIMA))
                  QV(I,J)=AMAX1(QBAR(I),ESIG
     1                  *0.622*EW(TC(I,J))/(PBAR(I)-EW(TC(I,J))))
                  WC(I,J)=AMAX1(WBAR(I),WCEX(J)*ESIG)
                  THG=TC(I,J)*(1.0E+05/PBAR(I))**(0.2854)
                  PHIC(I,J)=H(THG,QV(I,J),0.0,0.0)+0.5*WC(I,J)
     1                  *WC(I,J)
C                 PHIC(I,J)=H(TC(I,J),QV(I,J),0.,0.)+G*Z(I)
C     1                  *(1./(1.+QV(I,J)))
C     2                  +0.5*WC(I,J)*WC(I,J)
85          CONTINUE
C
            WRITE(6,1002) JTX,Z(JTX),WCB,PCB,CBQ,CBT
            GO TO 250
C
      END IF
C
C
      DO 200 J=1,JLEV
C
      SIGDZ=-DZ*DZ/(RADM*RADM*RADX*RADX)
      DO 105 I=1,JTX
            WC(I,J)=WCB*EXP(FLOAT(I-JTX)*FLOAT(I-JTX)*SIGDZ)
            QV(I,J)=CBQ
            PHIC(I,J)=PCB
105   CONTINUE
      DO 125 I=JTX+1,4*JT
```



```
            WC(I,J)=WCB*EXP(FLOAT(I-JTX)*FLOAT(I-JTX)*SIGDZ)
            THG=(PHIC(I,J)-0.5*WC(I,J)*WC(I,J)-2.501E+06*QV(I,J))
     1        /(1005.+4218.*QV(I,J))
            QV(I,J)=(0.622*EW(TC(I,J))/(PBAR(I)-EW(TC(I,J))))
     1            *EXP(FLOAT(I-JTX)*FLOAT(I-JTX)*SIGDZ)
            PHIC(I,J)=H(THG,QV(I,J),0.0,0.0)+0.5*WC(I,J)*WC(I,J)
C           PHIC(I,J)=H(TC(I,J),QV(I,J),0.0,0.0)+0.5*WC(I,J)*WC(I,J)
C     1            +G*Z(I)
125   CONTINUE
C
C
200   CONTINUE
C
C
250   CONTINUE
      WRITE(6,1005)
      DO 256 J=1,JLEV
      DO 256 I=1,NLVLS
            THG=(PHIC(I,J)-0.5*WC(I,J)*WC(I,J)-2.501E+06*QV(I,J))
     1        /(1005.+4218.*(QV(I,J)+QC(I,J)))
            TC(I,J)=THG*(PBAR(I)*1.0E-05)**(0.2854)
C           THG=TC(I,J)*(1.0E+05/PBAR(I))**(0.2854)
            THGV=THG*(1.+1.608*QV(I,J)-QC(I,J))/(1.+QV(I,J))
            WRITE(6,1010) I,J,Z(I),WC(I,J),QV(I,J),QC(I,J),TC(I,J),
     1                  PHIC(I,J),THG,THGV
256   CONTINUE
C
C
C
C     STORE THE INITIAL ENVIRONMENTAL  AND SUBCLOUD PERTUBRATION
C     FOR ABS(PTIME) .GT. 0.00
            DO 300 I=1,JTX
                  DP(I)=PHIC(I,1)-PHIBAR(I)
                  DW(I)=WC(I,1)-WBAR(I)
                  DQ(I)=QV(I,1)-QBAR(I)
300         CONTINUE
C
      IF(CEL) RETURN
            BE(1)=PHIBAR(1)/PHIC(1,1)
            BE(2)=PHIBAR(1)/PHIC(1,2)
            BQ(1)=QBAR(1)/QV(1,1)
            BQ(2)=QBAR(1)/QV(1,2)
            RETURN
C
C
      ENTRY FORSUB(K,W,X,MO,FARR,FCON)
C
C     FORSUB CALCULATES THE VELOCITY FORCING
C           K       REGION                  I
C           W       VELOCITY ARRAY          I
C           X       ADVECTED ARRAY          I
C           MO      ADVECTED ARRAY DESIG.   I
C                 =1 DW
C                 =2 DP
C                 =3 DQ
C                 =4 SET SUBCLOUD VALUE AT X
C                 =5 SET FARR,FCON = 0
C           FARR    FORCING ARRAY           O
C           FCON    FORCING CONTRIBUTION    O
C
      DO 500 I=1,NLVLS
            FCON(I)=ZERO(I)
            FARR(I)=ZERO(I)
500   CONTINUE
      IF(MO.GE.5) RETURN
      IF(.NOT.CEL) RETURN
      IF(.NOT.IPSET) RETURN
C
      IF(TIME.GT.PTIME) THEN
            IPSET=.FALSE.
```

QUATERNARY LIBRARY

```
      RETURN
    END IF
    KPFLG=0
    DO 510 I=1,JTX
      FCON(I)=UNO(I)
      FARR(I)=X(I,K)
510  CONTINUE
    IF(MO.GE.4) RETURN
    IF(MO.EQ.1 .AND. TIME.GT.UTIME) THEN
      IF(K.EQ.1) THEN
        DO 515 I=1,JTX
          FARR(I)=WMULT*WCEX(1)*FLOAT(I)/(FLOAT(JTX))
515      CONTINUE
      END IF
    END IF
    IF(MO.EQ.2) THEN
      DO 520 I=1,JTX
        FARR(I)=AMIN1(FARR(I),PCB)
520    CONTINUE
      IF(TIME.GT.0.95*PTIME .OR. KPFLG.EQ.1) THEN
        DO 521 I=1,JTX
          FARR(I)=PHIBAR(I)+4218.*QP(I,K)*TC(I,K)
     1          +2106.*AMIN1(TC(I,K),273.16)*QG(I,K)
521    CONTINUE
      END IF
    END IF
    IF(MO.EQ.8) THEN
      IF(TIME.GT.0.95*PTIME .OR. KPFLG.EQ.1) THEN
        DO 525 I=1,JTX
          FARR(I)=QBAR(I)
525    CONTINUE
      END IF
      IF(KPFLG.EQ.1) IPSET=.FALSE.
    END IF
C
      RETURN
C
1000  FORMAT(10X,'FORCING PARAMETERS, CBASE,CBT,CBQ,WCB,ZCB,JT,TEB'
     1    ,/,10X,2F12.3,F12.5,2F12.3,I5,F12.5)
1001  FORMAT(10X,'IN FORCER, CBFL IS',L2,' CEL IS',L2,/)
1002  FORMAT(10X,'CLOUDBASE VELOCITY FORCING: LEV,Z(LEV),PHIC,WC,QV,TC'
     1    ,/,10X,I5,F12.3,F20.2,F12.3,F12.5,F10.3,/)
1005  FORMAT(10X,'SUBCLOUD PERTURBATION; I,J,Z,WC,QV,QC,TC,PHIC,THETA,'
     1    ,'THETAV')
1010  FORMAT(5X,2I5,2F10.3,2F15.5,F12.3,F15.3,2F12.4)
2000  FORMAT(10X,'DECAY IN FORSUB, TIME,BFAC',2F15.6,/,
     1    30X, 'I,K,FCON,FARR,50*BFAC*DU2, F-50*B*D')
2001  FORMAT(10X,2I5,4G15.4)
2002  FORMAT(10X,'DECAY MODE, I,K,FARR',2I5,F15.4)
2010  FORMAT(10X,'GROWTH IN FORSUB, TIME,BFAC',2F15.6,/,
     1    30X, 'I,K,FCON,FARR,50*BFAC*DU2, F+50*B*D')
C
    END
*DECK QTHET
    SUBROUTINE QTHET(TC,QVC,QLC,TQBSE,QBSE,PB)
C
C   THIS SUBROUTINE ITERATES A KNOWN THETAQ AND TOTAL WATER
C   AT PRESSURE PB TO FIND THE TEMPERATURE AND LIQUID WATER.
C         TC    ITERATED TEMPERATURE  (K)      O
C         QVC   ITERATED VAPOR AT PB  (KG/KG)  O
C         QLC   ITERATED LIQUID AT PB (KG/KG)  O
C         TQBSE THETA Q               (K)      I
C         QBSE  TOTAL WATER           (KG/KG)  I
C         PB    PRESSURE              (PA)     I
C
C
    T=170.0
    DELTAT=5.0
C
    QSA=AMIN1(QBSE,0.622*EW(T)/(PB-EW(T)))
```



```
    QLS=QBSE-QSA
    TFUNC=THETAQ(PB,T,QSA,QLS)
    IF(ABS(TFUNC-TQBSE) .LE. 0.0001) GO TO 300
    TQOLD=TFUNC
    DO 200 J=1,100
      T=T+DELTAT
      QSA=AMIN1(QBSE,0.622*EW(T)/(PB-EW(T)))
      QLS=QBSE-QSA
      TFUNC=THETAQ(PB,T,QSA,QLS)
      IF(ABS(TFUNC-TQBSE) .LE. 0.0001) GO TO 300
      R=(TFUNC-TQBSE)/(TQOLD-TQBSE)
      IF(R.LT.0.00) DELTAT=-0.5*DELTAT
      TQOLD=TFUNC
200    CONTINUE
C
300  TC=T
    QVC=QSA
    QLC=QLS
C
    RETURN
    END
*DECK SUFLX
    SUBROUTINE SUFLX(DT,LEVCB,WS,VP,VS,VG)
C
C   THIS SUBROUTINE CALCULATES THE SULFUR DEPOSITION AND CLOUDBASE
C   SULFUR FLUX INTO THE CLOUD DOMAIN
C
*CALL GRID
*CALL CLOUD
*CALL VARCON
*CALL CBLOK
*CALL CHEM
*CALL DUMMY
*CALL SO4
*CALL H4N
C
    COMMON/SRST/ SIDES(101),UPSS(101),SCBF(2),SCBTA,SCBTB
   1      ,SUTOT,SCBA,SCBB,STOTL(101),STOT(101,2),STVF(2)
   2      ,STOTD(101),USTOT(101),SCBVF(2),SCBVT,VSTOT(2)
   3      ,AMTOT(101,2),AMVF(2),SO2VT,SO4AT,SO2LT
    DIMENSION WS(101,2),VP(101,2),VS(101,2),VG(101,2)
    DIMENSION SO2VF(2),SO4AF(2),SO4LF(2)
C
    DATA SIDES/101*0.0/, UPSS/101*0.0/, SUTOT/0.0/, SCBF/2*0.0/
    DATA SCBA/0.0/, SCBB/0.0/, STOTL/101*0.0/, STOT/202*0.0/
    DATA SCBTA/0.0/, SCBTB/0.0/, STOTD/101*0.0/, USTOT/101*0.0/
    DATA SCBVT/0.0/, SCBVF/2*0.0/, VSTOT/2*0.0/, STVF/2*0.0/
    DATA STVT/0.0/, AMVF/2*0.0/, AMTOT/202*0.0/, AMCBT/0.0/
    DATA SO2VT/0.0/, SO4AT/0.0/, SO4LT/0.0/
C
C
C   COMPUTE THE SULFUR MIXING RATIO FROM SO2 AND SO4.
C   QS=0.5*QSO2 AND 0.888*QSO4 BASED ON MOLECULAR WEIGHT.
C   QHSO8 IS AN IMPLICIT FIELD
    DO 105 J=1,JLEV
      SCBF(J)=(0.5*QSO2A(LEVCB,J)+0.333*(QSO4A(LEVCB,J)
   1        +QSO4C(LEVCB,J)))*WS(LEVCB,J)+0.888
   2        *(QSO4P(LEVCB,J)*(WS(LEVCB,J)+VP(LEVCB,J))
   3        +QSO4S(LEVCB,J)*(WS(LEVCB,J)+VS(LEVCB,J))
   4        +QSO4G(LEVCB,J)*(WS(LEVCB,J)+VG(LEVCB,J)))
   5        +0.895*(QHSO8P(LEVCB,J)*(WS(LEVCB,J)+VP(LEVCB,J))
   6        +QHSO8C(LEVCB,J)*WS(LEVCB,J))
      SCBVF(J)=(0.5*QSO2A(LEVCB,J)
   1        +0.333*(QSO4A(LEVCB,J)+QSO4C(LEVCB,J))
   2        +0.895*QHSO8C(LEVCB,J))*WS(LEVCB,J)
      SO2VF(J)=0.5*QSO2A(LEVCB,J)*WS(LEVCB,J)
      SO4AF(J)=0.333*QSO4A(LEVCB,J)*WS(LEVCB,J)
      SO4LF(J)=(0.333*QSO4C(LEVCB,J)
   2        +0.895*QHSO8C(LEVCB,J))*WS(LEVCB,J)
      STVF(J)=(0.5*QSO2A(NLVLS,J)+0.333*(QSO4A(NLVLS,J)
```

QUATERNARY LIBRARY

```
     1              +QSO4C(NLVLS,J)))*WS(NLVLS,J)+0.333
     2              *(QSO4P(NLVLS,J)*(WS(NLVLS,J)+VP(NLVLS,J))
     3              +QSO4S(NLVLS,J)*(WS(NLVLS,J)+VS(NLVLS,J))
     4              +QSO4G(NLVLS,J)*(WS(NLVLS,J)+VG(NLVLS,J)))
     5              +0.895*(QHSO8P(NLVLS,J)*(WS(NLVLS,J)+VP(NLVLS,J))
     6              +QHSO8C(NLVLS,J)*WS(NLVLS,J))
           AMVF(J)=(QNH4A(LEVCB,J)+QNH4C(LEVCB,J))*WS(LEVCB,J)
      DO 105 I=1,NLVLS
           STOT(I,J)=0.5*QSO2A(I,J)+0.333*(QSO4A(I,J)+QSO4C(I,J)
     1              +QSO4P(I,J)+QSO4S(I,J)+QSO4G(I,J))+0.895
     2              *(QHSO8C(I,J)+QHSO8P(I,J))
           AMTOT(I,J)=QNH4A(I,J)+QNH4C(I,J)+QNH4P(I,J)+QNH4S(I,J)
     1              +QNH4G(I,J)
  105 CONTINUE
C
      IF(TIME.LE.PTIME) THEN
        SO2VT=SO2VT+DT*(ARAD*ARAD*SO2VF(1)+(BRAD*BRAD-ARAD*ARAD)
     1      *SO2VF(2))
        SO4AT=SO4AT+DT*(ARAD*ARAD*SO4AF(1)+(BRAD*BRAD-ARAD*ARAD)
     1      *SO4AF(2))
        SO4LT=SO4LT+DT*(ARAD*ARAD*SO4LF(1)+(BRAD*BRAD-ARAD*ARAD)
     1      *SO4LF(2))
      END IF
C
C
C     COMPUTE THE POSITIVE VERTICAL SULFUR FLUX THROUGH CLOUDBASE
      SCBA=SCBA+0.5*(1.0+EP(LEVCB,1))*SCBF(1)*DT
      SCBB=SCBB+0.5*(1.0+EP(LEVCB,2))*SCBF(2)*DT
C
C     COMPUTE THE TOTAL VERTICAL SULFUR FLUX THROUGH CLOUDBASE
      SCBTA=SCBTA+SCBF(1)*DT
      SCBTB=SCBTB+SCBF(2)*DT
      SCBVT=SCBVT+DT*(ARAD*ARAD*SCBVF(1)+(BRAD*BRAD-ARAD*ARAD)
     1    *SCBVF(2))
      AMCBT=AMCBT+DT*(ARAD*ARAD*AMVF(1)+(BRAD*BRAD-ARAD*ARAD)
     1    *AMVF(2))
C     COMPUTE THE TOTAL VERTICAL SULFUR FLUX THROUGH DOMAIN TOP
      STVT=STVT+DT*(ARAD*ARAD*STVF(1)+(BRAD*BRAD-ARAD*ARAD)
     1    *STVF(2))
C     COMPUTE THE ENTRAINED SULFUR MASS ABOVE CLOUD BASE
C     THE MINUS SIGN CORRECTS FOR UC .LT. 0 FOR ENTRAINMENT
C     DU7 FROM ENTRAIN IS THE ENTRAINED TOTAL MIXING RATIO
      CALL ENTRAIN(JLEV,STOT,QSB,DU7,DU0)
      DO 115 I=LEVCB,NLVLS
           SIDES(I)=SIDES(I)-(1.-DEL(I,2))*FSIDE(I,2)*UC(I,2)
     1              *QSB(I)*DT
           UPSS(I)=UPSS(I)-(1.-DEL(I,2))*(DU7(I)-FSIDE(I,2)
     1            *QSB(I))*UC(I,2)*DT
  115 CONTINUE
      IF(TIME.LT.PTIME) THEN
           LBOT=LEVCB
      ELSE
           LBOT=1
      END IF
C     COMPUTE THE TOTAL SULFUR DIVERGENCE ABOVE CLOUD BASE
      DO 120 I=LBOT,NLVLS
           STOTL(I)=STOTL(I)+UC(I,2)*DU7(I)*RHOC(I)*DT
           STOTD(I)=STOTD(I)+UC(I,2)*(DU7(I)-QSB(I))*RHOC(I)*DT
  120 CONTINUE
C
C     COMPUTE THE LATERAL EDDY FLUX
      CALL FLUXES(STOT,QSB,JLEV,DU0,DU6)
      DO 125 I=LBOT,NLVLS
           USTOT(I)=USTOT(I)+DU6(I)*DT
           VSTOT(2)=VSTOT(2)+DU0(I)*DT
  125 CONTINUE
      CALL FLUXES(STOT,QSB,1,DU0,DU6)
      DO 130 I=LBOT,NLVLS
           VSTOT(1)=VSTOT(1)+DU0(I)*DT
  130 CONTINUE
```



```
C
      RETURN
C
C
      ENTRY SDEP(LEVCB,TPMASS)
C
C
C     COMPUTE THE UPWARD SULFUR MASS THROUGH CLOUDBASE
      SCBA=PIE*ARAD*ARAD*RHOC(LEVCB)*SCBA
      SCBB=PIE*(BRAD*BRAD-ARAD*ARAD)*RHOC(LEVCB)*SCBB
C     COMPUTE THE TOTAL SULFUR MASS THROUGH CLOUDBASE
      SCBVT=SCBVT*PIE*RHOC(LEVCB)
      SCBTA=PIE*ARAD*ARAD*RHOC(LEVCB)*SCBTA
      SCBTB=PIE*(BRAD*BRAD-ARAD*ARAD)*RHOC(LEVCB)*SCBTB
      WRITE(6,1100)
      WRITE(6,1105)  SCBA,SCBB,SCBA+SCBB
      WRITE(6,1110)  SCBTA,SCBTB,SCBTA+SCBTB,SCBVT,STVT
C     COMPUTE THE INTEGRATED ENTRAINED SULFUR MASS THROUGH THE
C     OUTER CLOUD BOUNDARY
      TSIDE=0.0
      TSTOT=0.0
      TUPSS=0.0
      SRTCA=0.0
      SRTCB=0.0
      SBASC=0.0
      AMRCA=0.0
      AMRCB=0.0
      DO 525 I=LEVCB,NLVLS
           TSIDE=TSIDE+SIDES(I)*RHOC(I)
           TSTOT=TSTOT+STOTL(I)*RHOC(I)
           TUPSS=TUPSS+UPSS(I)*RHOC(I)
           SRTCA=SRTCA+(STOT(I,1)-QSB(I))*RHOC(I)
           SRTCB=SRTCB+(STOT(I,2)-QSB(I))*RHOC(I)
           SBASC=SBASC+QSB(I)*RHOC(I)
           AMRCA=AMRCA+(AMTOT(I,1)-QNH4B(I))*RHOC(I)
           AMRCB=AMRCB+(AMTOT(I,2)-QNH4B(I))*RHOC(I)
  525 CONTINUE
           TSIDE=TSIDE*2.*PIE*BRAD*DZ
           TSTOT=TSTOT*2.*PIE*BRAD*DZ
           TUPSS=TUPSS*2.*PIE*BRAD*DZ
           SRTC=DZ*PIE*(ARAD*ARAD*SRTCA+(BRAD*BRAD-ARAD*ARAD
     1          *SRTCB)
           SBASC=SBASC*2.*PIE*BRAD*DZ
C     COMPUTE THE INTEGRATED SULFUR DIVERGENCE IN THE MODEL DOMAIN
      SDTOT=0.0
      SRTOTA=0.0
      SRTOTB=0.0
      SBASE=0.0
      SLTOT=0.0
      USTOTL=0.0
      AMRESA=0.0
      AMRESB=0.0
      DO 530 I=1,NLVLS
           SLTOT=SLTOT+STOTL(I)*RHOC(I)
           SDTOT=SDTOT+STOTD(I)*RHOC(I)
           SRTOTA=SRTOTA+(STOT(I,1)-QSB(I))*RHOC(I)
           SRTOTB=SRTOTB+(STOT(I,2)-QSB(I))*RHOC(I)
           SBASE=SBASE+QSB(I)*RHOC(I)
           USTOTL=USTOTL+USTOT(I)*RHOC(I)
           AMRESA=AMRESA+(AMTOT(I,1)-QNH4B(I))*RHOC(I)
           AMRESB=AMRESB+(AMTOT(I,2)-QNH4B(I))*RHOC(I)
  530 CONTINUE
           SLTOT=SLTOT*2.*PIE*BRAD*DZ
           SDTOT=SDTOT*2.*PIE*BRAD*DZ
           SRTOT=DZ*PIE*(ARAD*ARAD*SRTOTA+(BRAD*BRAD-ARAD*ARAD)
     1          *SRTOTB)
           SBASE=SBASE*2.*PIE*BRAD*DZ
           USTOTL=USTOTL*2.*PIE*BRAD*DZ
C     TOTAL SULFUR MASS INTO CLOUD
      TSMASS=SCBA+SCBB+TSIDE+TUPSS
```

QUATERNARY LIBRARY

```
C     NET SULFUR MASS INTO CLOUD
      VSTT=PIE*DZ*(ARAD*ARAD*VSTOT(1)+(BRAD*BRAD-ARAD*ARAD)*VSTOT(2))
      TSNET=SCBTA+SCBTB-TSTOT-USTOTL-VSTT
C     TOTAL SULFUR IN PRECIPITATION AT LOWEST LEVEL IN DOMAIN
C     SUFT IS INTEGRATED QSO4P AT LEVEL 1
      TSPMAS=(SUFT(1,1)*ARAD*ARAD+SUFT(2,1)*(BRAD*BRAD-ARAD*ARAD))
     1     *PIE*RHOC(1)
      TSOMAS=(SUFT(1,2)*ARAD*ARAD+SUFT(2,2)*(BRAD*BRAD-ARAD*ARAD))
     1     *PIE*RHOC(1)
      TSRES=TSPMAS-SDTOT-SRTOT+USTOTL+VSTT
      AMCBT=AMCBT*PIE*RHOC(LEVCB)
      AMPT=(AMFT(1)*ARAD*ARAD+AMFT(2)*(BRAD*BRAD-ARAD*ARAD))
     1     *PIE*RHOC(1)
      AMRESD=PIE*DZ*(AMRESA*ARAD*ARAD+(BRAD*BRAD-ARAD*ARAD)*AMRESB)
      AMRESC=PIE*DZ*(AMRCA*ARAD*ARAD+(BRAD*BRAD-ARAD*ARAD)*AMRCB)
      SO2VT=PIE*RHOC(LEVCB)*SO2VT
      SO4AT=PIE*RHOC(LEVCB)*SO4AT
      SO4LT=PIE*RHOC(LEVCB)*SO4LT
C
      WRITE(6,1200) TSIDE,TUPSS,TSTOT
      WRITE(6,1206) TSMASS,TSNET,TSPMAS,TSOMAS
      WRITE(6,1215) SRTC,SRTOT,SBASC,SBASE,SLTOT
      WRITE(6,1220) SDTOT,TSRES,USTOTL,VSTT
      WRITE(6,1225) SCBTA+SCBTB,SLTOT,SRTC
      WRITE(6,1230) AMCBT,AMPT,AMRESD,AMRESC
      WRITE(6,1235) SO2VT,SO4AT,SO4LT
C
C
      RETURN
C
C
C
 1100 FORMAT(/,25X,'SULFUR MASSES IN KG')
 1105 FORMAT(10X,'TOTAL UPWARD SULFUR MASS THROUGH CLOUDBASE; INNER',
     1     G15.4,3X,'OUTER',G15.4,3X,'TOTAL',G15.4)
 1110 FORMAT(10X,'NET SULFUR MASS THROUGH CLOUDBASE; INNER',
     1 G15.5,3X,'OUTER',G15.5,/,10X,'TOTAL',G15.5,5X,'SO2,SO4A',
     2 G15.5,5X,'DOMAIN TOP',G15.5)
 1200 FORMAT(/,10X,'INTEGRATED TSIDE,TSTOT,TUPSS',3G15.4)
 1205 FORMAT(10X,'SULFUR MASS INTO CLOUD; TOTAL, NET',2G15.4,
     1     /,10X,'TOTAL PRECIP AT LEV==1; SO4, HSO3',2G15.4)
 1215 FORMAT(10X,'RESIDUAL SULFUR; ABOVE CLOUD BASE',G15.5,' DOMAIN'
     1     ,G15.5,/,10X,'BASE SULFUR; ABOVE CLOUD BASE',G15.5
     2     ,' DOMAIN',G15.5,5X,'DOMAIN ENTRAINED SU',G15.5)
 1220 FORMAT(/,10X,'ENTRAINED DOMAIN RESIDUAL',G15.5,5X,'TOTAL RESIDUAL'
     1     ,G15.5,/,10X,'EDDY FLUX TOTAL; SIDE',G15.5,/,10X
     2     ,'VERTICAL',G15.5)
 1225 FORMAT(10X,'NET; THROUGH CLOUDBASE',G15.5,5X,'LATERAL DIVERGENCE'
     1     ,G15.5,/,10X,'RESIDUAL IN CLOUD',G15.5)
 1230 FORMAT(10X,'AMMONIUM; CLOUD BASE NET',G15.5,5X,'PRECIPITATION',
     1     G15.5,/,10X,'DOMAIN RESIDUAL',G15.5,5X,'CLOUD RESIDUAL',
     2     G15.5)
 1235 FORMAT(10X,'SO2VT,SO4AT,SO4LT; 0 TO PTIME',3G15.5)
C
      END
*DECK QHSO8
      FUNCTION QHSO8C(I,J)
C
C     CALCULATE THE HSO8 MIXING RATIO IN CLOUD WATER OR PRECIPITATION
C     THIS IS AN IMPLICIT FIELD
C
*CALL GRID
*CALL CLOUD
*CALL SO4
*CALL HP
C
      IF(HPC(I,J).LT.0.10 .AND. HPC(I,J).GT.0.0) THEN
           QHSO8C=QC(I,J)*1.984E-16*EXP(5133./TC(I,J))*PBAR(I)
     1          *QSO2A(I,J)/HPC(I,J)
      ELSE
```



```
           QHSO8C=0.0
      END IF
C
      RETURN
C
      ENTRY QHSO8P(K,L)
      IF(HPP(K,L).LT.0.10 .AND. HPP(K,L).GT.0.0) THEN
           QHSO8P=QP(I,J)*1.984E-16*EXP(5133./TC(I,J))*PBAR(I)
     1          *QSO2A(I,J)/HPP(K,L)
      ELSE
           QHSO8C=0.0
      END IF
C
      RETURN
C
      END
*DECK TRACER
      SUBROUTINE TRACER(DT,ITX,LEVCB,WS)
C
C     COMPUTE THE FLUXES FOR A PASSIVE TRACER
*CALL GRID
*CALL CLOUD
*CALL VARCON
*CALL CBLOK
*CALL CHEM
*CALL DUMMY
C
      COMMON/TRST/ SIDE(101),UPS(101),CBA,CBB,ETOT,SIDEC(101),UPSC(101)
     1          ,CTOT,CCBA,CCBB,CTOTL(101),CBT,ETOTC,CTT,CCTT,FTOT
     2          ,TOTLC,TOTEC,TOTVC(2),CRESA,CRESB,CRESCA,CRESCB
     3          ,CCBT
      DIMENSION WS(101,2)
C
      DATA SIDE/101*0.0/, UPS/101*0.0/, CBA/0.0/, CBB/0.0/
      DATA ETOT/0.0/, SIDEC/101*0.0/, UPSC/101*0.0/, CTOT/0.0/
      DATA CCBA/0.0/, CCBB/0.0/, CTOTL/101*0.0/, FTOT/0.0/
      DATA CBT/0.0/,ETOTC/0.0/,CTT/0.0/,CCTT/0.0/,CCBT/0.0/
      DATA TOTLC/0.0/, TOTEC/0.0/, TOTVC/2*0.0/
      DATA CRESA/0.0/, CRESB/0.0/, CRESCA/0.0/, CRESCB/0.0/
C
      GO TO (10,500), ITX
C
   10 CONTINUE
      CBA=CBA+0.5*(1.0+EP(LEVCB,1))*WS(LEVCB,1)*DT
      CBB=CBB+0.5*(1.0+EP(LEVCB,2))*WS(LEVCB,2)*DT
      CBT=CBT+(ARAD*ARAD*WS(LEVCB,1)+(BRAD*BRAD-ARAD*ARAD)*WS(LEVCB,2))
     1  *DT
      CTT=CTT+(ARAD*ARAD*WS(NLVLS,1)+(BRAD*BRAD-ARAD*ARAD)*WS(NLVLS,2))
     1  *DT
C
C     THE MINUS SIGN CORRECTS FOR UC .LT. 0 FOR ENTRAINMENT
      DO 100 I=LEVCB,NLVLS
           SIDE(I)=SIDE(I)-(1.-DEL(I,2))*FSIDE(I,2)*UC(I,2)*DT
           UPS(I)=UPS(I)-(1.-DEL(I,2))*FUP(I,2)*UC(I,2)*DT
           ETOTC=ETOTC+UC(I,2)*DT*RHOC(I)
  100 CONTINUE
      DO 110 I=1,NLVLS
           ETOT=ETOT+UC(I,2)*DT*RHOC(I)
  110 CONTINUE
      IF(TIME.LT.PTIME) THEN
           LBOT=LEVCB
      ELSE
           LBOT=1
      END IF
      DO 111 I=LBOT,NLVLS
           FTOT=FTOT+UC(I,2)*DT*RHOC(I)
  111 CONTINUE
C     TRACER FLUXES
      CCBA=CCBA+0.5*(1.0+EP(LEVCB,1))*WS(LEVCB,1)*DT*CC(LEVCB,1)
      CCBB=CCBB+0.5*(1.0+EP(LEVCB,2))*WS(LEVCB,2)*DT*CC(LEVCB,2)
```

QUATERNARY LIBRARY

```
      CCBT=CCBT+(ARAD*ARAD*WS(LEVCB,1)*CC(LEVCB,1)+(BRAD*BRAD-ARAD*ARAD)
     1   *WS(LEVCB,2)*CC(LEVCB,2))*DT
      CCTT=CCTT+(ARAD*ARAD*WS(NLVLS,1)*CC(NLVLS,1)+(BRAD*BRAD-ARAD*ARAD)
     1   *WS(NLVLS,2)*CC(NLVLS,2))*DT
C
      CALL ENTRAIN(JLEV,CC,CBAR,DU7,DU0)
      DO 115 I=LEVCB,NLVLS
            SIDEC(I)=SIDEC(I)-(1.-DEL(I,2))*FSIDE(I,2)*UC(I,2)
     1           *CBAR(I)*DT
            UPSC(I)=UPSC(I)-(1.-DEL(I,2))*(DU7(I)-FSIDE(I,2)
     1           *CBAR(I))*UC(I,2)*DT
  115 CONTINUE
      DO 120 I=1,NLVLS
            CTOTL(I)=CTOTL(I)+UC(I,2)*DU7(I)*RHOC(I)*DT
  120 CONTINUE
      CALL FLUXES(CC,CBAR,JLEV,DU0,DU6)
      DO 121 I=LBOT,NLVLS
            TOTVC(2)=TOTVC(2)+DU0(I)*DT
            TOTEC=TOTEC+DU6(I)*RHOC(I)*DT
            TOTLC=TOTLC+UC(I,2)*DU7(I)*RHOC(I)*DT
  121 CONTINUE
      CALL FLUXES(CC,CBAR,1,DU0,DU6)
      DO 122 I=LBOT,NLVLS
            TOTVC(1)=TOTVC(1)+DU0(I)*DT
  122 CONTINUE
C
      RETURN
C
  500 CONTINUE
C
      STOT=0.0
      UTOT=0.0
      DO 510 I=1,NLVLS
            STOT=STOT+SIDE(I)*RHOC(I)
            UTOT=UTOT+UPS(I)*RHOC(I)
  510 CONTINUE
            STOT=AMAX1(2.*PIE*BRAD*DZ*STOT,BIAS)
            UTOT=AMAX1(2.*PIE*BRAD*DZ*UTOT,BIAS)
            TOTL=STOT+UTOT
C
      CBA=PIE*ARAD*ARAD*RHOC(LEVCB)*CBA
      CBB=PIE*(BRAD*BRAD-ARAD*ARAD)*RHOC(LEVCB)*CBB
      CBTOT=AMAX1(CBA+CBB,BIAS)
      ETOT=2.*PIE*BRAD*DZ*ETOT
      FTOT=2.*PIE*BRAD*DZ*FTOT
      ETOTC=2.*PIE*BRAD*DZ*ETOTC
      CBT=PIE*CBT*RHOC(LEVCB)
      CTT=PIE*CTT*RHOC(LEVCB)
C
      WRITE(6,1000)
      WRITE(6,1005) ETOT,FTOT
      WRITE(6,1010) STOT,UTOT,STOT/UTOT,STOT/TOTL,UTOT/TOTL
      WRITE(6,1015) CBA,CBB,CBTOT
      WRITE(6,1020) STOT/CBTOT,UTOT/CBTOT,TOTL/CBTOT
      WRITE(6,1025) CBT,ETOTC,CTT
C
      CCBA=PIE*ARAD*ARAD*RHOC(LEVCB)*CCBA
      CCBB=PIE*(BRAD*BRAD-ARAD*ARAD)*RHOC(LEVCB)*CCBB
      TOTLC=TOTLC*2.*PIE*BRAD*DZ
      TOTEC=TOTEC*2.*PIE*BRAD*DZ
      TTVC=DZ*PIE*(ARAD*ARAD*TOTVC(1)+(BRAD*BRAD-ARAD*ARAD)*TOTVC(2))
      SIDEX=0.0
      UPSCX=0.0
      CTOTX=0.0
      CRESA=0.0
      CRESB=0.0
      CRESCA=0.0
      CRESCB=0.0
      DO 520 I=1,NLVLS
            SIDEX=SIDEX+SIDEC(I)*RHOC(I)
```



```
            CTOTX=CTOTX+CTOTL(I)*RHOC(I)
            UPSCX=UPSCX+UPSC(I)*RHOC(I)
            CRESA=CRESA+(CC(I,1)-CBAR(I))*RHOC(I)
            CRESB=CRESB+(CC(I,2)-CBAR(I))*RHOC(I)
  520 CONTINUE
      DO 521 I=LEVCB,NLVLS
            CRESCA=CRESCA+(CC(I,1)-CBAR(I))*RHOC(I)
            CRESCB=CRESCB+(CC(I,2)-CBAR(I))*RHOC(I)
  521 CONTINUE
C
C
      SIDEX=SIDEX*2.*PIE*BRAD*DZ
      UPSCX=UPSCX*2.*PIE*BRAD*DZ
      CTOTX=CTOTX*2.*PIE*BRAD*DZ
      CRESD=PIE*DZ*(ARAD*ARAD*CRESA+(BRAD*BRAD-ARAD*ARAD)*CRESB)
      CRESC=PIE*DZ*(ARAD*ARAD*CRESCA+(BRAD*BRAD-ARAD*ARAD)*CRESCB)
      CCBT=PIE*RHOC(LEVCB)*CCBT
      CCTT=PIE*RHOC(NLVLS)*CCTT
C
      WRITE(6,1100) CCBA,CCBB,CCBA+CCBB,CCBT,CCTT
      WRITE(6,1101) SIDEX,UPSCX,CTOTX
      WRITE(6,1102) TOTLC,TOTEC,TTVC
      WRITE(6,1103) CRESD,CRESC
  525 CONTINUE
C
      RETURN
C
      ENTRY TRACINIT(LEVCB,TCONC)
C
C     PASSIVE TRACER INITIALIZATION
      DO 600 I=1,NLVLS
            CBAR(I)=CBIAS
            CC(I,1)=CBIAS
            CC(I,2)=CBIAS
  600 CONTINUE
      DO 610 I=1,LEVCB
            CBAR(I)=TCONC
            CC(I,1)=TCONC
            CC(I,2)=TCONC
  610 CONTINUE
      WRITE(6,1200)
      DO 615 I=1,NLVLS
            WRITE(6,1205) I,Z(I),CBAR(I),CC(I,1),CC(I,2)
  615 CONTINUE
C
      RETURN
C
 1000 FORMAT(/,30X,'ENTRAINMENT TOTALS IN KG')
 1005 FORMAT(10X,'TOTAL LATERAL ENTRAINED MASS',G15.5,5X,
     1      'DOMAIN ENTRAINED MASS',G15.5)
 1010 FORMAT(10X,'STOT,UTOT,S/U,S/TOT/U/TOT',2E12.4,3F12.4)
 1015 FORMAT(10X,'CBA,CBB,CBTOT',3E15.4)
 1020 FORMAT(10X,'ST/CBT,UT/CBT,TOT/CBT',3F15.5,/)
 1025 FORMAT(10X,'INTEGRATED NET MASS; CLOUDBASE',G15.5,5X,'LATERAL',
     1      G15.5,5X,'DOMAIN TOP',G15.5)
 1100 FORMAT(10X,'TRACER THROUGH CLOUDBASE: CCBA,CCBB,CCTOT',3G15.5,
     1      /,10X,'NET; CLOUDBASE',G15.5,5X,'DOMAIN TOP',G15.5)
 1101 FORMAT(10X,'ENTRAINMENT; SIDE',G15.5,5X,'UPSTREAM',G15.5,
     1      'TOTAL SIDE',G15.5)
 1102 FORMAT(10X,'DOMAIN ENTRAINMENT',G15.5,5X,'EDDY FLUX TOTAL',G15.5,
     1      'VERTICAL FLUX TOTAL',G15.5)
 1103 FORMAT(10X,'RESIDUAL CONCENTRATION; DOMAIN',G15.5,'  CLOUD',
     1      G15.5)
 1105 FORMAT(20X,'Z,CTOTL,SIDEC,UPSC,S+U,CBAR,CC(1,2)')
 1110 FORMAT(10X,I5,F10.2,4E13.4,5X,3E13.4,I5)
 1200 FORMAT(/,10X,'INITIAL TRACER CONCENTRATION; CBAR, CC(I,1), ',
     1      'CC(I,2)')
 1205 FORMAT(10X,I5,F10.2,3G15.6)
C
      END
```

QUATERNARY LIBRARY

```
*DECK CHEMSET
      SUBROUTINE CHEMSET(DTI,LEVCB)
C
C     THIS SUBROUTINE CALCULATES THE SOURCE TERMS FOR THE SULFATE,
C     AMMONIA AND HYDROGEN PEROXIDE FIELDS.  THE HYDROGEN ION
C     CONCENTRATION IN THE LIQUID PHASE IS ALSO CALCULATED.
C           DTI    1/CURRENT TIME STEP            I
C           LEVCB  CLOUD BASE LEVEL               I
C
*CALL GRID
*CALL CLOUD
*CALL VARCON
*CALL DUMMY
*CALL WASSER
*CALL EIS
*CALL WATCON
*CALL EISCON
*CALL ASU
*CALL ANH
*CALL AHP
*CALL SO4
*CALL H4N
*CALL H2O2
*CALL HP
*CALL WFR
*CALL CBLOK
C
C     CALCULATE H+ IN LIQUID PHASE FIELDS
C           DU8=4.*KSO2*KHSO2*PSO2, RD/RSO2=287.05/129.88
      DO 120 J=1,JLEV
      DO 100 I=1,NLVLS
            DU1(I)=(2.*10.4106*(QSO4C(I,J)-CBIAS)
     1            -55.4364*(QNH4C(I,J)-CBIAS))/QC(I,J)
            DU2(I)=(2.*10.4106*(QSO4P(I,J)-CBIAS)
     1            -55.4364*(QNH4P(I,J)-CBIAS))/QP(I,J)
            DU3(I)=2.453E-15*EXP(5133./TC(I,J))
     1            *PBAR(I)*(QSO2A(I,J)-CBIAS)
            DU4(I)=SIGN(0.5,EIXL-QC(I,J))+0.5
            DU5(I)=SIGN(0.5,EIXL-QP(I,J))+0.5
100   CONTINUE
C
C           DU4 IS A FLAG FOR SUFFICENT LIQUID WATER TO CALCULATE PH
      DO 120 I=1,NLVLS
            HPC(I,J)=1000.*DU4(I)+(1.-DU4(I))
     1            *0.5*(DU1(I)+SQRT(DU1(I)*DU1(I)+4.0E-14
     2            +DU3(I)))
            HPP(I,J)=1000.*DU5(I)+(1.-DU5(I))
     1            *0.5*(DU2(I)+SQRT(DU2(I)*DU2(I)+4.0E-14
     2            +DU3(I)))
120   CONTINUE
C     DETERMINE IF ACTIVATION IS OCCURRING AT CLOUDBASE
C     AND COMPUTE NUCLEATION SCAVENGING FOR SO4C AND NH4C
      DO 155 J=1,JLEV
            DO 150 I=1,NLVLS
                  PSU(I,J,8)=ZERO(I)
                  PNH(I,J,1)=ZERO(I)
150         CONTINUE
            DO 155 I=LEVCB+1,LEVCB+8
                  PSU(I,J,8)=0.55*(QSO4A(I,J)-CBIAS)*DTI
     1                  *AMAX1(0.0,SIGN(1.0,COND(I,J)))
                  PNH(I,J,1)=0.55*(QNH4A(I,J)-CBIAS)*DTI
     1                  *AMAX1(0.0,SIGN(1.0,COND(I,J)))
155         CONTINUE
C
C     CALCULATE THE SULFATE SOURCE TERMS
      DO 210 J=1,JLEV
      DO 200 I=1,NLVLS
            DU1(I)=(QSO4C(I,J)-CBIAS)/QC(I,J)
            DU2(I)=(QSO4P(I,J)-CBIAS)/QP(I,J)
            DU3(I)=(QSO4S(I,J)-CBIAS)/QS(I,J)
```



```
            DU4(I)=(QSO4G(I,J)-CBIAS)/QG(I,J)
            DU5(I)=1.0/TC(I,J)
200   CONTINUE
      DO 210 I=1,NLVLS
C                 COND(I,J) IS THE CONDENSATION RATE COMPUTED
C                 IN SUBROUTINE TSPEC
            PSU(I,J,1)=AMIN1((1.567E-16/HPC(I,J))*EXP(5133.*DU5(I))
     1            *PBAR(I)*(QSO2A(I,J)-CBIAS)*COND(I,J)
     2            ,(QSO2A(I,J)-CBIAS)*DTI)
C           H2O2 OXIDATION FROM MARTIN (1984)
            PSU(I,J,2)=(QSO2A(I,J)-CBIAS)*QC(I,J)*((1.033E-28
     1            /HPC(I,J))*EXP(7693.*DU5(I))*PBAR(I)*PBAR(I)
     2            *(1.+2.39E-05/HPC(I,J)))
     3            + PBAR(I)*(QSO2A(I,J)-CBIAS)*(QH2O2C(I,J)
     4            -CBIAS)*2.79E-05*EXP(3168.6*DU5(I))
     5            /(0.1+HPC(I,J))
            PSU(I,J,4)=2.40E-10*QC(I,J)*RHOC(I)*(QSO4A(I,J)-CBIAS)
     1            /(AMAX1(1.0E-12,DC(I,J)*DC(I,J)))
            PSU(I,J,5)=16.2874*(QSO4A(I,J)-CBIAS)*ONR*DP(I,J)**3.5
            PSU(I,J,6)=0.0063*DS(I,J)*DS(I,J)*(QSO4A(I,J)-CBIAS)
     1            *TNSCAL(TC(I,J),QS(I,J)*RHOC(I))*ABS(VS(I,J))
            PSU(I,J,7)=DU1(I)*(AUTO(I,J)+COLL(I,J))
            PSU(I,J,8)=DU1(I)*(SRME(I,J)+CSFZ(I,J))
            PSU(I,J,9)=(QSO2A(I,J)-CBIAS)*QP(I,J)*((1.033E-28
     1            /HPP(I,J))*EXP(7693.*DU5(I))*PBAR(I)*PBAR(I)
     2            *(1.+2.39E-05/HPP(I,J)))
     3            + PBAR(I)*(QSO2A(I,J)-CBIAS)*(QH2O2P(I,J)
     4            -CBIAS)*2.79E-05*EXP(3168.6*DU5(I))
     5            /(0.1+HPP(I,J))
            PSU(I,J,10)=DU3(I)*SMELT(I,J)
            PSU(I,J,11)=(1.567E-16/HPP(I,J))*EXP(5133.*DU5(I))
     1            *(QSO2A(I,J)-CBIAS)*PEVAP(I,J)
            PSU(I,J,12)=DU1(I)*GRME(I,J)
            PSU(I,J,18)=-.064627*TNG*ABS(VG(I,J))*DG(I,J)
     1            *DG(I,J)*(QSO4A(I,J)-CBIAS)
            PSU(I,J,14)=DU3(I)*(SGCOL(I,J)+SGCON(I,J))
            PSU(I,J,15)=DU4(I)*(GPSHD(I,J)+GMELT(I,J))
            PSU(I,J,16)=DU2(I)*(PGCOL(I,J)+PGFZ(I,J))
            PSU(I,J,17)=DU2(I)*PEVAP(I,J)
            PSU(I,J,18)=QSO4C(I,J)*DTI*(0.5
     1            -SIGN(0.5,DC(I,J)-1.E-06))
            PSU(I,J,19)=QSO4S(I,J)*DTI*(0.5
     1            -SIGN(0.5,DS(I,J)-1.E-06))
210   CONTINUE
      DO 230 J=1,JLEV
      DO 220 I=1,NLVLS
            DU1(I)=(QSO2A(I,J)-CBIAS)*DTI
            DU2(I)=(QSO4A(I,J)-CBIAS)*DTI
            DU3(I)=(QSO4C(I,J)-CBIAS)*DTI
            DU4(I)=(QSO4S(I,J)-CBIAS)*DTI
220   CONTINUE
      DO 230 I=1,NLVLS
C           SO2 SINKS
            PSU(I,J,2)=AMIN1(PSU(I,J,2),DU1(I)-PSU(I,J,1))
            PSU(I,J,9)=AMIN1(PSU(I,J,9),DU1(I)-PSU(I,J,1)
     1            -PSU(I,J,2))
            PSU(I,J,11)=AMIN1(PSU(I,J,11),DU1(I)-PSU(I,J,1)
     1            -PSU(I,J,2)-PSU(I,J,9))
C           SO4C SINKS
            PSU(I,J,3)=AMIN1(PSU(I,J,3),DU2(I))
            PSU(I,J,4)=AMIN1(PSU(I,J,4),DU2(I)-PSU(I,J,3))
            PSU(I,J,5)=AMIN1(PSU(I,J,5),DU2(I)-PSU(I,J,3)
     1            -PSU(I,J,4))
            PSU(I,J,6)=AMIN1(PSU(I,J,6),DU2(I)-PSU(I,J,3)
     1            -PSU(I,J,4)-PSU(I,J,5))
            PSU(I,J,13)=AMIN1(PSU(I,J,13),DU2(I)-PSU(I,J,3)
     1            -PSU(I,J,4)-PSU(I,J,5)-PSU(I,J,6))
C           SO4L SINKS
            PSU(I,J,7)=AMIN1(PSU(I,J,7),DU3(I))
            PSU(I,J,8)=AMIN1(PSU(I,J,8),DU3(I)-PSU(I,J,7))
```

QUATERNARY LIBRARY

```
            PSU(I,J,12)=AMIN1(PSU(I,J,12),DU3(I)-PSU(I,J,7)
     1                -PSU(I,J,8))
            PSU(I,J,18)=AMIN1(PSU(I,J,18),DU3(I)-PSU(I,J,7)
     1                -PSU(I,J,8)-PSU(I,J,12))
C             SO4P SINKS
            PSU(I,J,16)=AMIN1(PSU(I,J,16),(QSO4P(I,J)-CBIAS)*DTI)
            PSU(I,J,17)=AMIN1(PSU(I,J,17),(QSO4P(I,J)-CBIAS)*DTI
     1                -PSU(I,J,16))
C             SO4S SINKS
            PSU(I,J,10)=AMIN1(PSU(I,J,10),DU4(I))
            PSU(I,J,14)=AMIN1(PSU(I,J,14),DU4(I)-PSU(I,J,10))
            PSU(I,J,19)=AMIN1(PSU(I,J,19),DU4(I)-PSU(I,J,10)
     1                -PSU(I,J,14))
C             SO4G SINKS
            PSU(I,J,15)=AMIN1(PSU(I,J,15),(QSO4G(I,J)-CBIAS)*DTI)
230   CONTINUE
C
C     AMMONIA SOURCE TERMS
      DO 310 J=1,JLEV
      DO 300 I=1,NLVLS
            DU1(I)=(QNH4C(I,J)-CBIAS)/QC(I,J)
            DU2(I)=(QNH4P(I,J)-CBIAS)/QP(I,J)
            DU3(I)=(QNH4S(I,J)-CBIAS)/QS(I,J)
            DU4(I)=(QNH4G(I,J)-CBIAS)/QG(I,J)
300   CONTINUE
      DO 310 I=1,NLVLS
            PNH(I,J,2)=2.40E-10*QC(I,J)*RHOC(I)*(QNH4A(I,J)-CBIAS)
     1           /(AMAX1(1.0E-12,DC(I,J)*DC(I,J)))
            PNH(I,J,3)=16.2874*(QNH4A(I,J)-CBIAS)*ONR*DP(I,J)**3.5
            PNH(I,J,4)=0.0068*DS(I,J)*DS(I,J)*(QNH4A(I,J)-CBIAS)
     1           *TNSCAL(TC(I,J),QS(I,J)*RHOC(I)*ABS(VS(I,J))
            PNH(I,J,5)=DU1(I)*(AUTO(I,J)+COLL(I,J))
            PNH(I,J,6)=DU1(I)*(SRME(I,J)+CSFZ(I,J))
            PNH(I,J,7)=DU3(I)*SMELT(I,J)
            PNH(I,J,8)=DU1(I)*GRME(I,J)
            PNH(I,J,9)=.064627*TNG*ABS(VG(I,J))*DG(I,J)
     1           *DG(I,J)*(QNH4A(I,J)-CBIAS)
            PNH(I,J,10)=DU3(I)*(SGCOL(I,J)+SGCON(I,J))
            PNH(I,J,11)=DU4(I)*(GPSHD(I,J)+GMELT(I,J))
            PNH(I,J,12)=DU2(I)*(PGCOL(I,J)+PGFZ(I,J))
            PNH(I,J,18)=DU2(I)*PEVAP(I,J)
            PNH(I,J,14)=(QNH4C(I,J)-CBIAS)*DTI*(0.5
     1           -SIGN(0.5,DC(I,J)-1.E-06))
            PNH(I,J,15)=(QNH4S(I,J)-CBIAS)*DTI*(0.5
     1           -SIGN(0.5,DS(I,J)-1.E-06))
310   CONTINUE
      DO 330 J=1,JLEV
      DO 320 I=1,NLVLS
            DU1(I)=(QNH4A(I,J)-CBIAS)*DTI
            DU2(I)=(QNH4C(I,J)-CBIAS)*DTI
            DU3(I)=(QNH4S(I,J)-CBIAS)*DTI
320   CONTINUE
      DO 330 I=1,NLVLS
C             NH4C SINK TERMS
            PNH(I,J,2)=AMIN1(PNH(I,J,2),DU1(I)-PNH(I,J,1))
            PNH(I,J,3)=AMIN1(PNH(I,J,3),DU1(I)-PNH(I,J,1)
     1           -PNH(I,J,2))
            PNH(I,J,4)=AMIN1(PNH(I,J,4),DU1(I)-PNH(I,J,1)
     1           -PNH(I,J,2)-PNH(I,J,3))
            PNH(I,J,9)=AMIN1(PNH(I,J,9),DU1(I)-PNH(I,J,1)
     1           -PNH(I,J,2)-PNH(I,J,3)-PNH(I,J,4))
C             NH4L SINK TERMS
            PNH(I,J,5)=AMIN1(PNH(I,J,5),DU2(I))
            PNH(I,J,6)=AMIN1(PNH(I,J,6),DU2(I)-PNH(I,J,5))
            PNH(I,J,8)=AMIN1(PNH(I,J,8),DU2(I)-PNH(I,J,5)
     1           -PNH(I,J,6))
            PNH(I,J,14)=AMIN1(PNH(I,J,14),DU2(I)-PNH(I,J,5)
     1           -PNH(I,J,6)-PNH(I,J,8))
C             NH4P SINK TERMS
            PNH(I,J,12)=AMIN1(PNH(I,J,12),(QNH4P(I,J)-CBIAS)*DTI)
```



```
            PNH(I,J,18)=AMIN1(PNH(I,J,18),(QNH4P(I,J)-CBIAS)*DTI
     1           -PNH(I,J,12))
C             NH4S SINK TERMS
            PNH(I,J,7)=AMIN1(PNH(I,J,7),DU3(I))
            PNH(I,J,10)=AMIN1(PNH(I,J,10),DU3(I)-PNH(I,J,7))
            PNH(I,J,15)=AMIN1(PNH(I,J,15),DU3(I)-PNH(I,J,7)
     1           -PNH(I,J,10))
C             NH4G SINK TERMS
            PNH(I,J,11)=AMIN1(PNH(I,J,11),(QNH4G(I,J)-CBIAS)*DTI)
330   CONTINUE
C
C     HYDROGEN PEROXIDE SOURCE TERMS
      DO 410 J=1,JLEV
      DO 400 I=1,NLVLS
            DU1(I)=(QH2O2C(I,J)-CBIAS)/QC(I,J)
            DU2(I)=(QH2O2P(I,J)-CBIAS)/QP(I,J)
            DU3(I)=(QH2O2S(I,J)-CBIAS)/QS(I,J)
            DU4(I)=(QH2O2G(I,J)-CBIAS)/QG(I,J)
            DU5(I)=1.0/TC(I,J)
400   CONTINUE
      DO 410 I=1,NLVLS
            PHP(I,J,1)=DU1(I)*(AUTO(I,J)+COLL(I,J))
            PHP(I,J,2)=DU1(I)*(SRME(I,J)+CSFZ(I,J))
            PHP(I,J,8)=DU3(I)*SMELT(I,J)
            PHP(I,J,4)=DU1(I)*GRME(I,J)
            PHP(I,J,5)=DU3(I)*(SGCOL(I,J)+SGCON(I,J))
            PHP(I,J,6)=DU4(I)*(GPSHD(I,J)+GMELT(I,J))
            PHP(I,J,7)=DU2(I)*(PGCOL(I,J)+PGFZ(I,J))
            PHP(I,J,8)=(QH2O2C(I,J)-CBIAS)*DTI*(0.5
     1           -SIGN(0.5,DC(I,J)-1.E-06))
            PHP(I,J,9)=(QH2O2S(I,J)-CBIAS)*DTI*(0.5
     1           -SIGN(0.5,DS(I,J)-1.E-06))
            PHP(I,J,10)=AMAX1(1.15590E-07*EXP(7.03E+03*DU5(I))
     1           *(QH2O2A(I,J)-CBIAS)*COND(I,J)
     2           ,-(QH2O2C(I,J)-CBIAS)*DTI)
     3           *(0.5-SIGN(0.5,COND(I,J)))
     4           +(0.5+SIGN(0.5,COND(I,J)))*1.15590E-07
     5           *EXP(7.03E+03*DU5(I))*(QH2O2A(I,J)-CBIAS)
     6           *COND(I,J)
            PHP(I,J,11)=AMIN1(1.15590E-07*EXP(7.08E+03*DU5(I))
     1           *(QH2O2A(I,J)-CBIAS)*PEVAP(I,J)
     2           ,(QH2O2P(I,J)-CBIAS)*DTI)
C       MARTIN OXIDATION
C           CHANGE 0.53=34/64 TO 0.35=34/96 FOR S(VI) SOURCE TERM
            PHP(I,J,12)=0.53*PBAR(I)*(QSO2A(I,J)-CBIAS)*(QH2O2C(I,J)
     1           -CBIAS)*2.79E-05*EXP(3168.6*DU5(I))
     2           /(0.1+HPC(I,J))
            PHP(I,J,13)=0.53*PBAR(I)*(QSO2A(I,J)-CBIAS)*(QH2O2P(I,J
     1           -CBIAS)*2.79E-05*EXP(3168.6*DU5(I))
     2           /(0.1+HPP(I,J))
410   CONTINUE
      DO 430 J=1,JLEV
      DO 420 I=1,NLVLS
            DU1(I)=(QH2O2C(I,J)-CBIAS)*DTI
            DU2(I)=(QH2O2P(I,J)-CBIAS)*DTI
            DU3(I)=(QH2O2S(I,J)-CBIAS)*DTI
420   CONTINUE
      DO 430 I=1,NLVLS
C             H2O2C SINK TERMS
            PHP(I,J,11)=AMIN1(PHP(I,J,10),(QH2O2A(I,J)-CBIAS)*DTI)
C             H2O2L SINK TERMS
            PHP(I,J,1)=AMIN1(PHP(I,J,1),DU1(I))
            PHP(I,J,2)=AMIN1(PHP(I,J,2),DU1(I)-PHP(I,J,1))
            PHP(I,J,4)=AMIN1(PHP(I,J,4),DU1(I)-PHP(I,J,1)
     1           -PHP(I,J,2))
            PHP(I,J,8)=AMIN1(PHP(I,J,8),DU1(I)-PHP(I,J,1)
     1           -PHP(I,J,2)-PHP(I,J,4))
            PHP(I,J,12)=AMIN1(PHP(I,J,12),DU1(I)-PHP(I,J,1)
     1           -PHP(I,J,2)-PHP(I,J,4)-PHP(I,J,8))
C             H2O2P SINK TERMS
```

QUATERNARY LIBRARY

```
          PHP(I,J,7)=AMIN1(PHP(I,J,7),DU2(I))
          PHP(I,J,11)=AMIN1(PHP(I,J,11),DU2(I)-PHP(I,J,7))
          PHP(I,J,13)=AMIN1(PHP(I,J,13),DU2(I)-PHP(I,J,7)
     1              -PHP(I,J,11))
C           H2O2S SINK TERMS
          PHP(I,J,3)=AMIN1(PHP(I,J,3),DU3(I))
          PHP(I,J,5)=AMIN1(PHP(I,J,5),DU3(I)-PHP(I,J,3))
          PHP(I,J,9)=AMIN1(PHP(I,J,9),DU3(I)-PHP(I,J,3)
     1              -PHP(I,J,5))
C           H2O2G SINK TERMS
          PHP(I,J,6)=AMIN1(PHP(I,J,6),(QH2O2G(I,J)-CBIAS)*DTI)
430   CONTINUE
C
      RETURN
C
      ENTRY CHEMINT(LEVCB)
C
C     INITIALIZE THE CHEMISTRY FIELDS
      DO 501 I=1,NLVLS
          QSO2B(I)=CBIAS
          QSO4B(I)=CBIAS
          QNH4B(I)=CBIAS
          QH2O2B(I)=CBIAS
501   CONTINUE
      DO 505 J=1,JLEV
      DO 505 I=1,NLVLS
          QSO2A(I,J)=CBIAS
          QSO4A(I,J)=CBIAS
          QSO4C(I,J)=CBIAS
          QSO4P(I,J)=CBIAS
          QSO4S(I,J)=CBIAS
          QSO4G(I,J)=CBIAS
          QNH4A(I,J)=CBIAS
          QNH4C(I,J)=CBIAS
          QNH4P(I,J)=CBIAS
          QNH4S(I,J)=CBIAS
          QNH4G(I,J)=CBIAS
          QH2O2A(I,J)=CBIAS
          QH2O2C(I,J)=CBIAS
          QH2O2P(I,J)=CBIAS
          QH2O2S(I,J)=CBIAS
          QH2O2G(I,J)=CBIAS
505   CONTINUE
C
C     ICHUN=1  INITIALIZATION FOR CONTINENTAL BACKGROUND;
C       QSO2 AT 2 PPBV, QH2O2 AT 0.5 PPBV, QNH4 AT 0.9 PPBV
C     ICHUN=2  INITIALIZATION FOR POLLUTED ATMOSPHERE;
C       QSO2 AT 10 PPBV, QH2O2 AT 0.5 PPBV, QNH4 AT 4.5 PPBV
C       (FROM HEGG, ET AL., 1986B, JGR,91,14403-416)
C     QSO2, QSO4 AND QNH4 DECAY EXPONENTIALLY; QH2O2 IS CONSTANT.
C     THE SCALE HEIGHT IS 2KM FOR SO2, 3.5KM FOR SO4 AND NH4.
C     ICHUN=3,4 CB OR POLLUTED ATMOSPHERE WITH NH4)2 SO4
C     ICHUN-5,6 CB OR POLLUTED ATMOSPHERE WITH NO ENTRAINMENT
C     ICHUN=7  SAME AS ICHUN=1 WITH H2O2=HPFAC*0.5 PPBV
C     ICHUN=8  SAME AS ICHUN=2 WITH H2O2=HPFAC*0.5 PPBV
C     ICHUN .GT. 8  READ INPUT FIELDS FROM UNIT 45
      IF(ICHUN.LE.8) THEN
        LTP=NLVLS
        IF(ICHUN.EQ.1 .OR. ICHUN.EQ.5 .OR. ICHUN.EQ.7) THEN
            SO2I=4.419E-09
            SO4I=3.10E-09
            H4NI=5.8E-10
            H2O2I=5.869E-10
        END IF
        IF(ICHUN.EQ.2 .OR. ICHUN.EQ.6 .OR. ICHUN.EQ.8) THEN
            SO2I=2.1E-08
            SO4I=1.60E-08
            H4NI=3.0E-09
            H2O2I=5.869E-10
        END IF
```



```
        IF(ICHUN.EQ.3) THEN
            SO2I=4.419E-09
            SO4I=3.10E-09
            H4NI=1.16E-09
            H2O2I=5.869E-10
        END IF
        IF(ICHUN.EQ.4) THEN
            SO2I=2.1E-08
            SO4I=1.60E-08
            H4NI=6.0E-09
            H2O2I=5.869E-10
        END IF
        IF(ICHUN.EQ.5 .OR. ICHUN.EQ.6) LTP=LEVCB
        IF(ICHUN.GE.7) H2O2I=H2O2I*HPFAC
        DO 510 I=1,LTP
            QSO2B(I)=AMAX1(SO2I*EXP(-0.001*Z(I)*.5),CBIAS)
            QSO2A(I,1)=AMAX1(SO2I*EXP(-0.001*Z(I)*.5),CBIAS)
            QSO2A(I,2)=AMAX1(SO2I*EXP(-0.001*Z(I)*.5),CBIAS)
C
            QSO4B(I)=AMAX1(SO4I*EXP(-0.001*Z(I)*.286),CBIAS)
            QSO4A(I,1)=AMAX1(SO4I*EXP(-0.001*Z(I)*.286),CBIAS)
            QSO4A(I,2)=AMAX1(SO4I*EXP(-0.001*Z(I)*.286),CBIAS)
C
            QNH4B(I)=AMAX1(H4NI*EXP(-0.001*Z(I)*.286),CBIAS)
            QNH4A(I,1)=AMAX1(H4NI*EXP(-0.001*Z(I)*.286),CBIAS)
            QNH4A(I,2)=AMAX1(H4NI*EXP(-0.001*Z(I)*.286),CBIAS)
C
            QH2O2B(I)=AMAX1(H2O2I,CBIAS)
            QH2O2A(I,1)=AMAX1(H2O2I,CBIAS)
            QH2O2A(I,2)=AMAX1(H2O2I,CBIAS)
510     CONTINUE
      ELSE
        DO 515 I=1,NLVLS
            READ(45,END=516) QSO2B(I),QSO4B(I),QNH4B(I),
     1          QH2O2B(I),QSO2A(I,1),QSO4A(I,1),QNH4A(I,1),
     2          QH2O2A(I,1)
515       CONTINUE
516   CONTINUE
        DO 517 I=1,NLVLS
            QSO2A(I,2)=QSO2C(I,1)
            QSO4A(I,2)=QSO4C(I,1)
            QNH4A(I,2)=QNH4C(I,1)
            QH2O2A(I,2)=QH2O2C(I,1)
517     CONTINUE
      END IF
      WRITE(6,1000) ICHUN
      DO 518 I=1,NLVLS
          WRITE(6,1005) I,QSO2B(I),QSO2A(I,1),QSO4B(I),QSO4A(I,1),
     1              QNH4B(I),QNH4A(I,1),QH2O2B(I),QH2O2A(I,1)
518   CONTINUE
C
C     QSB IS THE BACKGROUND SULFUR MIXING RATIO
      DO 520 I=1,NLVLS
          QSB(I)=0.5*QSO2B(I)+0.333*QSO4B(I)
520   CONTINUE
C     CALCULATE H+ IN LIQUID PHASE FIELDS
C           DU3=4.*KSO2*KHSO2*PSO2, RD/RSO2=287.05/129.83
      DO 530 J=1,JLEV
      DO 525 I=1,NLVLS
          DU1(I)=(2.*10.4106*(QSO4C(I,J)-CBIAS)
     1          -55.4364*(QNH4C(I,J)-CBIAS))/QC(I,J)
          DU2(I)=(2.*10.4106*(QSO4P(I,J)-CBIAS)
     1          -55.4364*(QNH4P(I,J)-CBIAS))/QP(I,J)
          DU3(I)=2.453E-15*EXP(5133/TC(I,J))
     1          *PBAR(I)*(QSO2A(I,J)-CBIAS)
          DU4(I)=SIGN(0.5,BIXL-QC(I,J))+0.5
          DU5(I)=SIGN(0.5,BIXL-QP(I,J))+0.5
525   CONTINUE
      DO 530 I=1,NLVLS
          HPC(I,J)=1000.*DU4(I)+(1.-DU4(I))
```

QUATERNARY LIBRARY

```
     1              *0.5*(DU1(I)+SQRT(DU1(I)*DU1(I)+4.0E-14
     2              +DU8(I)))
            HPP(I,J)=1000.*DU5(I)+(1.-DU5(I))
     1              *0.5*(DU2(I)+SQRT(DU2(I)*DU2(I)+4.0E-14
     2              +DU8(I)))
  580  CONTINUE
C
 1000  FORMAT(//,20X,'SO2,SO4,NH4,H2O2;  B,C',10X,'ICHUN=',I5)
 1005  FORMAT(5X,I5,4(1P,2E12.3,3X))
C
      RETURN
C
      END
*DECK ADCHEM
      SUBROUTINE ADCHEM(J,DT,VP,VS,VG)
C
C     THIS IS THE ADVECTION SUBROUTINE CALLED FROM SUBROUTINE CLOUDS
C     AND USED TO ADVECT THE CHEMISTRY FIELDS
C          J      REGION (1=INNER, 2=OUTER)      I
C          DT     CURRENT TIME STEP              I
C          VP     PRECIP. VELOCITY ARRAY         I
C          VS     SNOW VELOCITY ARRAY            I
C          VG     GRAUPEL VELOCITY ARRAY         I
C
*CALL GRID
*CALL CLOUD
*CALL VARCON
*CALL DUMMY
*CALL CBLOK
*CALL ASU
*CALL ANH
*CALL AHP
*CALL SO4
*CALL H4N
*CALL H2O2
*CALL CHMSTR
C
      DIMENSION VP(101,2),VS(101,2),VG(101,2)
      DIMENSION SRCE(101),VSRCE(101)
C
C     SULFATE
C       SO2
C     THE FACTOR 0.67 ON SOURCE TERMS PSU(I,J,2), PSU(I,J,9) COVERT
C     SO4 (MW 96) TO SO2 (MW 64) IN THE LIQUID PHASE.  SEE HEGG, ET AL.,
C     1984; JGR, VOL. 89, 7133-7147.  NOTE THEY HAVE SINK TERMS FOR SO2
C     HERE ARE WRITTEN FOR SO4
      DO 100 I=1,NLVLS
         SRCE(I)=RHOC(I)*(PSU(I,J,11)-0.667*(PSU(I,J,9)
     1           +PSU(I,J,2))-PSU(I,J,1))
  100 CONTINUE
      CALL ADVECT(J,QSO2A,QSO2B,1,SRCE,DT)
C       SO4C
      DO 105 I=1,NLVLS
         SRCE(I)=RHOC(I)*(PSU(I,J,19)+PSU(I,J,18)+PSU(I,J,17)
     1           -PSU(I,J,13)-PSU(I,J,6)-PSU(I,J,5)-PSU(I,J,4)
     2           -PSU(I,J,3))
  105 CONTINUE
      CALL ADVECT(J,QSO4A,QSO4B,2,SRCE,DT)
C       SO4L
      DO 110 I=1,NLVLS
         SRCE(I)=RHOC(I)*(PSU(I,J,2)+PSU(I,J,3)
     1           +PSU(I,J,4)-PSU(I,J,7)
     2           -PSU(I,J,8)+PSU(I,J,10)-PSU(I,J,12)-PSU(I,J,18))
  110 CONTINUE
      CALL ADVECT(J,QSO4C,CBIASA,3,SRCE,DT)
C       SO4P
      DO 115 I=1,NLVLS
         SRCE(I)=RHOC(I)*(PSU(I,J,5)+PSU(I,J,7)+PSU(I,J,9)
     1           +PSU(I,J,15)-PSU(I,J,16)-PSU(I,J,17))
         VSRCE(I)=-RHOC(I)*VP(I,J)*QSO4P(I,J)
```



```
  115 CONTINUE
      CALL VADVCT(J,QSO4P,CBIASA,4,SRCE,VSRCE,DT)
C       SO4S
      DO 120 I=1,NLVLS
         SRCE(I)=RHOC(I)*(PSU(I,J,6)+PSU(I,J,8)-PSU(I,J,10)
     1           -PSU(I,J,14)-PSU(I,J,19))
         VSRCE(I)=-RHOC(I)*VS(I,J)*QSO4S(I,J)
  120 CONTINUE
      CALL VADVCT(J,QSO4S,CBIASA,5,SRCE,VSRCE,DT)
C       SO4G
      DO 125 I=1,NLVLS
         SRCE(I)=RHOC(I)*(PSU(I,J,12)+PSU(I,J,13)+PSU(I,J,14)
     1           -PSU(I,J,15)+PSU(I,J,16))
         VSRCE(I)=-RHOC(I)*VG(I,J)*QSO4G(I,J)
  125 CONTINUE
      CALL VADVCT(J,QSO4G,CBIASA,6,SRCE,VSRCE,DT)
C
C     AMMONIA
C       NH4C
      DO 200 I=1,NLVLS
         SRCE(I)=RHOC(I)*(PNH(I,J,15)+PNH(I,J,14)+PNH(I,J,13)
     1           -PNH(I,J,9)-PNH(I,J,4)-PNH(I,J,8)-PNH(I,J,2)
     2           -PNH(I,J,1))
  200 CONTINUE
      CALL ADVECT(J,QNH4A,QNH4B,7,SRCE,DT)
C       NH4L
      DO 205 I=1,NLVLS
         SRCE(I)=RHOC(I)*(PNH(I,J,1)+PNH(I,J,2)-PNH(I,J,5)
     1           -PNH(I,J,6)+PNH(I,J,7)-PNH(I,J,8)-PNH(I,J,14))
  205 CONTINUE
      CALL ADVECT(J,QNH4C,CBIASA,8,SRCE,DT)
C       NH4P
      DO 210 I=1,NLVLS
         SRCE(I)=RHOC(I)*(PNH(I,J,3)+PNH(I,J,5)+PNH(I,J,11)
     1           -PNH(I,J,12)-PNH(I,J,13))
         VSRCE(I)=-RHOC(I)*VP(I,J)*QNH4P(I,J)
  210 CONTINUE
      CALL VADVCT(J,QNH4P,CBIASA,9,SRCE,VSRCE,DT)
C       NH4S
      DO 215 I=1,NLVLS
         SRCE(I)=RHOC(I)*(PNH(I,J,4)+PNH(I,J,6)-PNH(I,J,7)
     1           -PNH(I,J,10)-PNH(I,J,15))
         VSRCE(I)=-RHOC(I)*VS(I,J)*QNH4S(I,J)
  215 CONTINUE
      CALL VADVCT(J,QNH4S,CBIASA,10,SRCE,VSRCE,DT)
C       NH4G
      DO 220 I=1,NLVLS
         SRCE(I)=RHOC(I)*(PNH(I,J,8)+PNH(I,J,9)+PNH(I,J,10)
     1           -PNH(I,J,11)+PNH(I,J,12))
         VSRCE(I)=-RHOC(I)*VG(I,J)*QNH4G(I,J)
  220 CONTINUE
      CALL VADVCT(J,QNH4G,CBIASA,11,SRCE,VSRCE,DT)
C
C     HYDROGEN PEROXIDE
C       H2O2C
      DO 800 I=1,NLVLS
         SRCE(I)=RHOC(I)*(PHP(I,J,11)-PHP(I,J,10)+PHP(I,J,9))
  800 CONTINUE
      CALL ADVECT(J,QH2O2A,QH2O2B,12,SRCE,DT)
C       H2O2L
      DO 805 I=1,NLVLS
         SRCE(I)=RHOC(I)*(-PHP(I,J,1)-PHP(I,J,2)
     1           +PHP(I,J,3)-PHP(I,J,4)-PHP(I,J,8)
     2           +PHP(I,J,10)-PHP(I,J,12))
  805 CONTINUE
      CALL ADVECT(J,QH2O2C,CBIASA,13,SRCE,DT)
C       H2O2P
      DO 810 I=1,NLVLS
         SRCE(I)=RHOC(I)*(PHP(I,J,1)+PHP(I,J,6)
     1           -PHP(I,J,7)-PHP(I,J,11)-PHP(I,J,13))
```

QUATERNARY LIBRARY

```
            VSRCE(I)=-RHOC(I)*VP(I,J)*QH2O2P(I,J)
310   CONTINUE
      CALL VADVCT(J,QH2O2P,CBIASA,14,SRCE,VSRCE,DT)
C         H2O2S
      DO 315 I=1,NLVLS
            SRCE(I)=RHOC(I)*(PHP(I,J,2)-PHP(I,J,3)
     1            -PHP(I,J,5)-PHP(I,J,9))
            VSRCE(I)=-RHOC(I)*VS(I,J)*QH2O2S(I,J)
315   CONTINUE
      CALL VADVCT(J,QH2O2S,CBIASA,15,SRCE,VSRCE,DT)
C         H2O2G
      DO 320 I=1,NLVLS
            SRCE(I)=RHOC(I)*(PHP(I,J,4)+PHP(I,J,5)
     1            -PHP(I,J,6)+PHP(I,J,7))
            VSRCE(I)=-RHOC(I)*VG(I,J)*QH2O2G(I,J)
320   CONTINUE
      CALL VADVCT(J,QH2O2G,CBIASA,16,SRCE,VSRCE,DT)
C
      RETURN
      END
*DECK ADVECT
      SUBROUTINE ADVECT(J,XC,XBAR,LV,SRCE,DT)
C
C     THIS SUBROUTINE SETS UP THE NECESSARY ARRAYS AND
C     MAKES THE CALLS TO ETBFCT FOR ADVECTION
C           J      REGION                    I
C           XC     ARRAY TO BE ADVECTED      I
C           XBAR   ENVIRONMENT ARRAY         I
C           LV     FLAG FOR SUBCLOUD FORCING I
C           SRCE   SOURCE ARRAY              I
C           DT     CURRENT TIME STEP         I
C     ENTRY VADVCT INCORPORATES AN ADDITIONAL SOURCE TERM
C     FOR FALLING HYDROMETEORS
*CALL GRID
*CALL CLOUD
*CALL VARCON
*CALL CBLOK
*CALL DUMMY
*CALL CHMSTR
*CALL WFR
C
      DIMENSION SRCE(101),XC(101,2),XBAR(101)
      DIMENSION SRC(101),XCA(101,2),XBARA(101),VSRC(101)
C
C       SET SUBCLOUD FORCING FOR SO2C,SO4C,NH4C,H2O2C
C       SEE SUBROUTINE ADCHEM FOR LV VALUES
      IF(LV.LE.2 .OR. LV.EQ.7 .OR. LV.EQ.12) THEN
            LX=4
      ELSE
            LX=5
      END IF
      CALL FORSUB(J,WS,XC,LX,FARR,FCON)
      CALL FLUXES(XC,XBAR,J,DU5,DU8)
      CALL ENTRAIN(J,XC,XBAR,DU3,DU0)
      IF(J.EQ.1) THEN
            DO 100 I=1,NLVLS
                  CHSTR(I,LV)=DU8(I)
                  CHENT(I,LV)=DU0(I)
100         CONTINUE
      END IF
      DO 105 I=1,NLVLS
            DU2(I)=RHOC(I)*XC(I,J)
            DU8(I)=ADCO(J)*RHOC(I)*(DU8(I)-FLOAT(J-1)*ARB
     1            *CHSTR(I,LV))
            DU9(I)=DU8(I)*DWPDZ(I,J)-FLOAT(J-1)*ADCO(3)
     1            *(DU8(I)-CHENT(I,LV))*DWPDZ(I,1)
105   CONTINUE
      DO 110 I=1,NLVLS
            DU0(I)=SRCE(I)+DU5(I)+DU8(I)+DU9(I)
110   CONTINUE
```



```
C
      CALL SOURCES(NLVLS,DT,3,UNO,DU0,0.0,0.0)
      CALL ETBFCT(DU2,DU1,NLVLS,1.0,1.0)
      DO 115 I=1,NLVLS
            XC(I,J)=(1.-FCON(I))*AMAX1(DU1(I)*RHOCI(I),CBIAS)
     1            +FCON(I)*FARR(I)
115   CONTINUE
      RETURN
C
      ENTRY VADVCT(K,XCA,XBARA,LVA,SRC,VSRC,DTX)
C
      CALL FLUXES(XCA,XBARA,K,DU5,DU8)
      CALL ENTRAIN(K,XCA,XBARA,DU3,DU0)
      IF(K.EQ.1) THEN
            DO 200 I=1,NLVLS
                  CHSTR(I,LVA)=DU8(I)
                  CHENT(I,LVA)=DU0(I)
200         CONTINUE
      END IF
      DO 205 I=1,NLVLS
            DU2(I)=RHOC(I)*XCA(I,K)
            DU8(I)=ADCO(J)*RHOC(I)*(DU8(I)-FLOAT(J-1)*ARB
     1            *CHSTR(I,LVA))
            DU9(I)=DU8(I)*DWPDZ(I,K)-FLOAT(K-1)*ADCO(3)
     1            *(DU8(I)-CHENT(I,LVA))*DWPDZ(I,1)
205   CONTINUE
      DO 210 I=1,NLVLS
            DU0(I)=SRC(I)+DU5(I)+DU8(I)+DU9(I)
210   CONTINUE
C
      CALL SOURCES(NLVLS,DTX,3,UNO,DU0,0.0,0.0)
      CALL SOURCES(NLVLS,DTX,1,UNO,VSRC,VSRC(1),VSRC(NLVLS))
      CALL ETBFCT(DU2,DU1,NLVLS,1.0,1.0)
      DO 215 I=1,NLVLS
            XCA(I,J)=AMAX1(DU1(I)*RHOCI(I),CBIAS)
215   CONTINUE
      RETURN
C
      END
*DECK DTSET
      SUBROUTINE DTSET(DT,DZ,NLVLS,JLEV)
C
C     THIS SUBROUTINE CALCULATES THE MAXIMUM TIME INCREMENT
C     ACCORDING TO THE FORMULA
C           DT=ALPHA*ABS(DZ/VMAX)
C     WHERE ALPHA MUST BE .LE. 0.5 FOR POSITIVITY. HERE ALPHA
C     IS SET TO 0.25.
C
*CALL GRID
*CALL CLOUD
*CALL WASSER
*CALL EIS
      VMAX=10.0
      DO 150 J=1,JLEV
      DO 125 I=1,NLVLS
            VMAX=AMAX1(ABS(VP(I,J)),ABS(VG(I,J)),
     1            ABS(VS(I,J)),ABS(WC(I,J)),VMAX)
125   CONTINUE
150   CONTINUE
        DT=0.25*(DZ/VMAX)
      RETURN
      END
*DECK DIFCAL
      FUNCTION DIFCAL(A,I,H,NUM,IK)
C
C     THIS FUNCTION CALCULATES THE FIRST DERIVATIVE OF A
C     TABULATED FUNCTION.  THE DERIVATIVE FUNCTIONS FOR
C     IK=2-8 ARE TAKEN FROM THE BOEING MATH LIBRARY, VOL. 5A.
C     SEE THE INTRODUCTION TO THE CHAPTER ON DIFFERENTIATION.
C     THE REMAINDER HAVE BEEN DERIVED FROM TAYLOR SERIES
```

QUATERNARY LIBRARY

```
C   EXPANSIONS OF F(X+H).
C         A     FUNCTION ARRAY
C         I     POINT AT WHICH DERIVATIVE IS EVALUATED
C         H     STEP SIZE BETWEEN POINTS
C         NUM   ARRAY SIZE
C         IK    FIRST DERIVATIVE CALCULATED BY
C               =1 CENTERED SECOND ORDER DERIVATIVE
C                 USING 2 POINTS
C               =2 CENTERED FOURTH ORDER DERIVATIVE
C                 USING 4 POINTS
C               =3 CENTERED SIXTH ORDER DERIVATIVE
C                 USING 6 POINTS
C               =4 FOR FORWARD DIFFERENCE USING
C                 3 POINTS (SECOND ORDER)
C               =5 FOR BACKWARD DIFFERENCE USING
C                 3 POINTS (SECOND ORDER)
C               =6 FOR 4 POINT FORWARD DIFFERENCE
C                 SECOND ORDER DERIVATIVE
C               =7 FOR 4 POINT BACKWARD DIFFERENCE
C                 SECOND ORDER DERIVATIVE
      DIMENSION A(NUM)
C
      GO TO (10,20,30,40,50,60,70), IK
10      DIFCAL=(A(I+1)-A(I-1))/(2.*H)
        RETURN
C
20      DIFCAL=(A(I-2)-8.*A(I-1)+8.*A(I+1)-A(I+2))/(12.*H)
        RETURN
C
30      DIFCAL=(-A(I-3)+9.*A(I-2)-45.*A(I-1)+45.*A(I+1)
     1          -9.*A(I+2)+A(I+3))/(60.*H)
        RETURN
C
40      DIFCAL=(-3.*A(I)+4.*A(I+1)-A(I+2))/(2.*H)
        RETURN
C
50      DIFCAL=(A(I-2)-4.*A(I-1)+3.*A(I))/(2.*H)
        RETURN
C
60      DIFCAL=(-A(I+2)+6.*A(I+1)-3.*A(I)-2.*A(I-1))/(6.*H)
        RETURN
C
70      DIFCAL=(2.*A(I+1)+3.*A(I)-6.*A(I-1)+A(I-2))/(6.*H)
        RETURN
C
      END
*DECK CHART
      SUBROUTINE CHART(KA,DT)
C
C     THIS SUBROUTINE WRITES CLOUD PARAMETERS
C     KA  =1   WRITE SELECTED VARIABLES TO TAPE11,TAPE12
C           AND TO OUTPUT (TAPE6).
C         =2   WRITE SELECTED VARIABLES TO TAPE11,TAPE12.
C         =3   DUMP EVERYTHING TO TAPES 15-18.
C
*CALL GRID
*CALL CLOUD
*CALL VARCON
*CALL WASSER
*CALL EIS
*CALL CELOK
*CALL STORS
*CALL CHEM
*CALL SO4
*CALL HNO3
*CALL H2O2
*CALL HP
*CALL ASU
C
      LOGICAL CHMFLG
```



```
C
      GO TO (100,200,800), KA
C
100   CONTINUE
      IF(PFT(1).GT.BIXX .AND. PHTOT(1).GT.0.0) THEN
          PHTA=ALOG10(PHTOT(1)/PFT(1))
      ELSE
          PHTA=10.
      END IF
      IF(PFT(2).GT.BIXX .AND. PHTOT(2).GT.0.0) THEN
          PHTB=ALOG10(PHTOT(2)/PFT(2))
      ELSE
          PHTB=10.
      END IF
      WRITE(30,9800) TIME,PFT(1),PHTOT(1),PHTA,PFT(2),PHTOT(2),PHTB
9800  FORMAT(/,5X,'TIME,(PFT,PHTOT,PHTA);1,2',F10.3,2(3G12.4,5X))
      WRITE(11) TIME,DT,ICOUNT,PFT(1),PFT(2),PRATE(1),PRATE(2),GFT(1),
     1     GFT(2),GRATE(1),GRATE(2),PHTA,PHTB
      DO 125 J=1,JLEV
          WRITE(6,1000)
          DO 125 I=1,NLVLS
              WRITE(6,1005) I,J,Z(I),TC(I,J),PHIC(I,J),
     1             WC(I,J),UC(I,J),TKE(I,J),TEP(I,J),
     2             FUP(I,J),FSIDE(I,J)
              WRITE(6,1010) QV(I,J),QC(I,J),
     1             QP(I,J),QS(I,J),QG(I,J),
     2             CC(I,J),DT,TIME,ICOUNT
              WRITE(12) TIME,Z(I),TC(I,J),PHIC(I,J),
     1             WC(I,J),UC(I,J),TKE(I,J),
     2             QV(I,J),QC(I,J),QP(I,J),QS(I,J),
     3             QG(I,J)
              WRITE(16) TIME,Z(I),WC(I,J),QC(I,J),CC(I,J)
125   CONTINUE
      IF(CHMFLG) THEN
        DO 130 J=1,JLEV
          WRITE(6,1100)
          DO 130 I=1,NLVLS
              WRITE(6,1105) I,J,Z(I),QSO2A(I,J),QSO4A(I,J)
     1             ,QSO4C(I,J),QSO4P(I,J),QSO4S(I,J)
     2             ,QSO4G(I,J)
              WRITE(6,1110) QNH4A(I,J),QNH4C(I,J),QNH4P(I,J)
     1             ,QNH4S(I,J),QNH4G(I,J),-ALOG10(HPC(I,J))
              WRITE(6,1110) QH2O2A(I,J),QH2O2C(I,J),QH2O2P(I,J)
     1             ,QH2O2S(I,J),QH2O2G(I,J)
     2             ,-ALOG10(HPP(I,J))
              WRITE(17) TIME,Z(I),QSO2A(I,J),QSO4A(I,J)
     1             ,QSO4C(I,J),QSO4P(I,J),QSO4S(I,J)
     2             ,QSO4G(I,J),-ALOG10(HPC(I,J))
     3             ,-ALOG10(HPP(I,J))
     4             ,QNH4A(I,J),QNH4C(I,J)
     5             ,QNH4P(I,J),QNH4S(I,J),QNH4G(I,J)          ))
     6             ,QH2O2A(I,J),QH2O2C(I,J),QH2O2P(I,J)
     7             ,QH2O2S(I,J),QH2O2G(I,J)
130     CONTINUE
      END IF
C
      RETURN
C
200   CONTINUE
      IF(PFT(1).GT.BIXX .AND. PHTOT(1).NE.0.0) THEN
          PHTA=ALOG10(PHTOT(1)/PFT(1))
      ELSE
          PHTA=10.
      END IF
      IF(PFT(2).GT.BIXX .AND. PHTOT(2).NE.0.0) THEN
          PHTB=ALOG10(PHTOT(2)/PFT(2))
      ELSE
          PHTB=10.
      END IF
      WRITE(30,9800) TIME,PFT(1),PHTOT(1),PHTA,PFT(2),PHTOT(2),PHTB
```

QUATERNARY LIBRARY

```
      WRITE(11) TIME,DT,ICOUNT,PFT(1),PFT(2),PRATE(1),PRATE(2),GFT(1),
     1     GFT(2),GRATE(1),GRATE(2),PHTA,PHTB
      DO 225 J=1,JLEV
         DO 225 I=1,NLVLS
            WRITE(12) TIME,Z(I),TC(I,J),PHIC(I,J),
     1              WC(I,J),UC(I,J),TKE(I,J),
     2              QV(I,J),QC(I,J),QP(I,J),QS(I,J),
     3              QG(I,J),CC(I,J)
  225 CONTINUE
      IF(CHMFLG) THEN
         DO 230 J=1,JLEV
         DO 230 I=1,NLVLS
            WRITE(17) TIME,Z(I),QSO2A(I,J),QSO4A(I,J)
     1              ,QSO4C(I,J),QSO4P(I,J),QSO4S(I,J)
     2              ,QSO4G(I,J),-ALOG10(HPC(I,J))
     3              ,-ALOG10(HPP(I,J))
     4              ,QNH4A(I,J),QNH4C(I,J)
     5              ,QNH4P(I,J),QNH4S(I,J),QNH4G(I,J)
     6              ,QH2O2A(I,J),QH2O2C(I,J),QH2O2P(I,J)
     7              ,QH2O2S(I,J),QH2O2G(I,J)
            WRITE(18) TIME,I,J,Z(I),PSU(I,J,1),PSU(I,J,2)
     1              ,PSU(I,J,8),PSU(I,J,4),PSU(I,J,9)
     2              ,PSU(I,J,11),PSU(I,J,18)
  230    CONTINUE
      END IF
C
      RETURN
C
  300    ZT=TIME/60.
         WRITE(15,2000) ZT,PFALL,SFALL,GFALL
         DO 250 K=1,NLVLS
            PR=PBAR(K)*0.01
            ZL=Z(K)*0.001
            WRITE(16) PR,ZL,TC(K,1),TC(K,2),QV(K,1),
     1              QV(K,2),QC(K,1),QC(K,2),QP(K,1),
     2              QP(K,2),QS(K,1),QS(K,2),QG(K,1),
     3              QG(K,2),PHIC(K,1),PHIC(K,2),
     4              WC(K,1),WC(K,2),UC(K,1),UC(K,2)
            WRITE(17) PR,ZL,AUTO(K,1),AUTO(K,2),
     1              COLL(K,1),COLL(K,2),PEVAP(K,1),
     2              PEVAP(K,2)
            WRITE(18) PR,ZL,SINIT(K,1),SINIT(K,2),
     1              VSDEP(K,1),VSDEP(K,2),SRME(K,1),
     2              SRME(K,2),CSFZ(K,1),CSFZ(K,2),
     3              VGDEP(K,1),VGDEP(K,2),GRME(K,1),
     4              GRME(K,2),PGCOL(K,1),PGCOL(K,2),
     5              PGFZ(K,1),PGFZ(K,2),GMELT(K,1),
     6              GMELT(K,2),SMELT(K,1),SMELT(K,2),
     7              SGCON(K,1),SGCON(K,2)
  250    CONTINUE
      END IF
      RETURN
C
 1000 FORMAT(///,25X,'I,J,Z,TC,PHIC,WC,UC,TKE,TEP,FUP,FSIDE'
     1     ,/,30X,'QV,QC,QP,QS,QG,CC,DT,TIME,ICOUNT',//)
 1005 FORMAT(5X,2I5,2F10.8,F15.8,4F12.6,2F7.4)
 1010 FORMAT(15X,F12.6,5E12.6,2F10.3,I5/)
 1100 FORMAT(//,25X,'CHEMISTRY FIELDS: SO2C,SO4C,L,P,S,G; NH4C,L,P,S,G;'
     1     ,', HPC; H2O2C,L,P,S,G; HPF')
 1105 FORMAT(10X,2I5,F10.8,6G15.5)
 1110 FORMAT(30X,5G15.5,5X,F10.4)
 2000 FORMAT(2X,'TIME',F8.2,' M',2X,'PFALL,SFALL,GFALL',3F10.8,/)
C
      END
*DECK WIL
      FUNCTION WIL(TZ)
C
C     THIS FUNCTION CALCULATES THE LATENT HEAT OF MELTING
C     USING A FUNCTION FROM PRUPPACHER AND KLETT, P.89.
C     THE FACTOR OF 4.1868E+03 IS TO CONVERT FROM CAL/G TO J/KG
```



```
C
      T=TZ-273.16
      WIL=(79.7+0.485*T-(2.5E-3)*T**2.)*4.1868E+03
      RETURN
C
C     THE FORMULA FOR THE LATENT HEAT OF VAPORIZATION
C     IS FROM BOLTON (1980)
C
      ENTRY WVL(TZ)
C
      T=TZ-273.16
      WVL=(2.501-0.00237*T)*1.0E+06
      RETURN
      END
*DECK PARICE
      SUBROUTINE PARICE(DTI)
C
C     THIS SUBROUTINE CALCULATES THE ICE TERMS FOR AN
C     ICE PARAMETERIZATION FROM STEVENS (CSU ATMOSPHERIC
C     SCIENCE DEPT. PAPER NO. 319, 1979) WITH SOME
C     MODIFICATIONS FROM CHENG (1981, U.W. THESIS).  WET GROWTH
C     IS INCLUDED FOR GRAUPEL.
C
*CALL GRID
*CALL CLOUD
*CALL VARCON
*CALL WASSER
*CALL EIS
*CALL EISCON
C
C       GRMCON=3*PIE/7.
      DATA GRMCON/1.346396852/, TPI/6.283185308/
C
      CALL HAIL
      CALL SNOW
C
      DTX=1.0/DTI
C
      DO 500 J=1,JLEV
      DO 400 I=1,NLVLS
C       CALCULATE PSI,K,MU AND THE VENTILATION FACTORS
C       FOR SNOW AND GRAUPEL.
      CALL PSIKMU(PBAR(I),TC(I,J),PSI,TK,DMU)
      FRES=1.+0.229*SQRT(RHOC(I)*DS(I,J)*ABS(VS(I,J))/DMU)
      FREG=1.+0.265*SQRT(RHOC(I)*DG(I,J)*ABS(VG(I,J))/DMU)
      IF(TC(I,J).GT.TF) THEN
         SINIT(I,J)=0.0
         SRMB(I,J)=0.0
         CSFZ(I,J)=0.0
         SMELT(I,J)=AMAX1(0.0,(QS(I,J)-BIAS)*DTI)
         PGFZ(I,J)=0.0
         IF(QG(I,J)*RHOC(I) .LE. AMAX1(BDX,1.0E-09*TNG)
     1        .OR.
     2      QC(I,J).LT.0.00001) THEN
            GRME(I,J)=0.0
         ELSE
            QLI=AMAX1(0.0,QC(I,J)-DTX*(AUTO(I,J)-COLL(I,J)
     1           -BIAS))
            STK=405.0*DC(I,J)*DC(I,J)*ABS(VG(I,J))
     1           /(9.*DMU*DG(I,J))
            EGR=STK*STK/((STK+0.5)*(STK+0.5))
            GRME(I,J)=AMIN1(GRMCON*EGR*TNG
     1              *ABS(VG(I,J))*DG(I,J)*DG(I,J)*QLI
     2              ,QLI*DTI)
         END IF
         IF(QP(I,J).LT.BDX .OR. QG(I,J).LT.BDX) THEN
            PGCOL(I,J)=0.0
         ELSE
            PGCOL(I,J)=PIE*ABS(VP(I,J)-VG(I,J))*QP(I,J)
     1           *TNG*(0.5*DG(I,J)*DG(I,J)+2.
```

QUATERNARY LIBRARY

```
     2                    *DP(I,J)*DG(I,J)+5.*DP(I,J)*DP(I,J))
          END IF
          IF(QG(I,J).LT.BIXL) THEN
               GMELT(I,J)=AMAX1(0.0,(QG(I,J)-BIAS)*DTI)
          ELSE
               QSW=EPS*EW(TF)/(PBAR(I)-EW(TF))
               DTF=TC(I,J)-TF
C              1/.334E+06=2.994E-06
               GMELT(I,J)=(TK*DTF*RHOCI(I)+2.501E+06
     1                    *PSI*(QV(I,J)-QSW))*TPI
     2                    *DG(I,J)*TNG*FREG*2.994E-06
     3                    +(CW*DTF*(GRME(I,J)
     4                    +PGCOL(I,J)))*2.994E-06
               GMELT(I,J)=AMAX1(0.0,AMIN1(GMELT(I,J)
     1                    ,(QG(I,J)-BIAS)*DTI))
          END IF
          SGCOL(I,J)=0.0
          SGCON(I,J)=0.0
          GPSHD(I,J)=GRME(I,J)+PGCOL(I,J)
          VSDEP(I,J)=0.0
          VGDEP(I,J)=0.0
          GO TO 400
     END IF
C
C    TC(I,J) .LE. 273.16
C
     TNS=TNSCAL(TC(I,J),QS(I,J)*RHOC(I))
     QSI=EPS*EI(TC(I,J))/(PBAR(I)-EI(TC(I,J)))
     SS=(QV(I,J)/QSI)-1.0
     IF(SS.LT.0.00) THEN
          SINTT(I,J)=0.0
          SRME(I,J)=0.0
          CSFZ(I,J)=0.0
          SMELT(I,J)=0.0
          GO TO 100
     END IF
C
     SINTT(I,J)=DTI*AMIN1(CMAS*TNS*RHOCI(I),AMAX1(0.0,(QV(I,J)
     1         -BIAS-QSI)))
C
     IF(TNS.GT.1.0E+07) THEN
          SRME(I,J)=0.0
          CSFZ(I,J)=AMAX1(0.0,(QC(I,J)-BIAS)*DTI)
          SMELT(I,J)=0.0
          PGFZ(I,J)=AMAX1(0.0,(QP(I,J)-BIAS)*DTI)
          GRME(I,J)=0.0
          GMELT(I,J)=0.0
          PGCOL(I,J)=0.0
          GO TO 150
     END IF
C
     QLI=AMAX1(0.0,QC(I,J)-DTX*(AUTO(I,J)-COLL(I,J)-BIAS))
C
C       SNOW TRANSITIONS
C
C      LIMIT SNOW RIMING TO CONCENTRATIONS .GT. 0.01/LITER
C      OR FOR CLOUD DROPS .GT. 10 MICRONS DIAMETER
     IF(QS(I,J)*RHOC(I) .LE. AMAX1(1.5E-09,1.5E-10*TNS)   .OR.
     1  DC(I,J).LT.0.00001) THEN
          EFR=0.0
          SRME(I,J)=0.0
     ELSE
          STK=1000.*DC(I,J)*DC(I,J)*ABS(VS(I,J))
     1       /(9.*DMU*DS(I,J))
          EFR=STK*STK/((STK+0.5)*(STK+0.5))
          SRME(I,J)=AMIN1((TNS*PIE*DS(I,J)*DS(I,J)*ABS(VS(I,J))
     1              *EFR*QLI*0.25),QLI*DTI)
     END IF
C
     CSFZ(I,J)=0.0
```



```
     SMELT(I,J)=0.0
C
C       GRAUPEL TRANSITIONS
C
100  CONTINUE
     IF(QP(I,J).LT.BIXL) THEN
          PGFZ(I,J)=0.0
     ELSE
          PGFZ(I,J)=AMIN1(PIE*TNS*DP(I,J)*DP(I,J)
     1              *ABS(6.1875*VP(I,J)-5.*VS(I,J))
     2              *QP(I,J),(QP(I,J)-BIAS)*DTI)
     END IF
C
     IF(QG(I,J)*RHOC(I) .LE. AMAX1(BIXX,1.0E-09*TNG)  .OR.
     1  DC(I,J).LT.0.00001) THEN
          GRME(I,J)=0.0
     ELSE
          STK=405.0*DC(I,J)*DC(I,J)*ABS(VG(I,J))
     1       /(9.*DMU*DG(I,J))
          EGR=STK*STK/((STK+0.5)*(STK+0.5))
          GRME(I,J)=AMIN1(GRMCON*EGR*TNG
     1              *ABS(VG(I,J))*DG(I,J)
     2              *DG(I,J)*QLI,AMAX1(0.,(QLI*DTI-SRME(I,J))))
     END IF
C
     GMELT(I,J)=0.0
C
C    GRAUPEL COLLECTION TERMS.  THE COLLECTION EFFICIENCY FOR
C    SNOW COLLECTION IS ASSUMED 0.1 FOR DRY GROWTH.
     IF(QP(I,J).LT.BIXL .OR. QG(I,J).LT.BIXL) THEN
          PGCOL(I,J)=0.0
     ELSE
          PGCOL(I,J)=PIE*ABS(VP(I,J)-VG(I,J))*QP(I,J)*TNG
     1              *(0.5*DG(I,J)*DG(I,J)+2.*DP(I,J)*DG(I,J)
     2              +5.*DP(I,J)*DP(I,J))
     END IF
C
150  CONTINUE
     IF(QS(I,J).LT.BIXL .OR. QG(I,J).LT.BIXL) THEN
          SGCOL(I,J)=0.0
     ELSE
          SGCOL(I,J)=GRMCON*0.1*TNG*ABS(VG(I,J))
     1              *(QS(I,J)-BIAS)*DG(I,J)*DG(I,J)
          SGCOL(I,J)=AMIN1(SGCOL(I,J),(QS(I,J)-BIAS)*DTI)
     END IF
C
C    SNOW TO GRAUPEL CONVERSION. THE FIRST TERM IS AGGREGATION, THE
C    SECOND IS DUE TO EXCESS SNOW RIMING, AND THE LAST LIMITS THE
C    SNOW MIXING RATIO, I.E., THE MAXIMUM SNOW CRYSTAL MASS.  SEE
C    CHENG, P.27 OR STEPHENS, P.52.   THE AGGREGATION TERM IS FROM
C    LIN, ET AL., J.CLIM.AP.MET., VOL. 22, 1065-1092, EQ. 38
C
     SGCON(I,J)=0.01*EXP(0.07*(TC(I,J)-TF))*AMAX1(0.0,QS(I,J)-0.0006)
     1         +(SRME(I,J)+0.001*(QS(I,J)-BIAS)*DTI)
     2         *HEAVI(QS(I,J)-0.0001)*HEAVI(QC(I,J)-.0010)
C
     SGCON(I,J)=AMIN1(SGCON(I,J), (QS(I,J)-BIAS)*DTI-SGCOL(I,J))
C
C    WET GROWTH OF GRAUPEL. THE FACTOR OF 10*SGCOL CHANGES THE
C    COLLECTION EFFICIENCY TO 1.0 FOR WET GROWTH.
     IF(GRME(I,J)+PGCOL(I,J).LE.0.0 .OR. TNS.GT.1.0E+07) THEN
          GPSHD(I,J)=0.0
     ELSE
          DFT=TC(I,J)-TF
          QSW=EPS*EW(TC(I,J))/(PBAR(I)-EW(TC(I,J)))
          PGWG=AMAX1(0.0,(-TPI*TNG*FREG*DG(I,J)*(RHOC(I)
     1         *2.501E+06*PSI*(QV(I,J)-QSW)+TK*DFT))
     2         /(RHOC(I)*0.334E+06+4218.*DFT)
     3         +10.0*SGCOL(I,J)*(1.0-(2106.*DFT)/(0.334E+06
     4         +4218.*DFT)))
```

QUATERNARY LIBRARY

```
            GPSHD(I,J)=AMAX1(0.0,GRME(I,J)+PGCOL(I,J)-PGWG)
            IF(GPSHD(I,J).GT.0.0) THEN
                SGCOL(I,J)=AMIN1(10.*SGCOL(I,J)
     1                  ,(QS(I,J)-BIAS)*DTI-SGCON(I,J))
                GPSHD(I,J)=GPSHD(I,J)+SGCOL(I,J)
            END IF
        END IF
C
C     VAPOR DEPOSITION OR SUBLIMATION
C
        GTP=1./((2.8345E+06*WIL(TC(I,J))/(TK*461.51*TC(I,J)
     1   *TC(I,J)))+(461.51*TC(I,J)/(PSI*EI(TC(I,J)))))
C
C             SNOW
        IF(QS(I,J).LT.BIXX) THEN
            VSDEP(I,J)=0.0
        ELSE
            VSDA=4.*DS(I,J)*SS*TNS*GTP*FRES*RHOCI(I)
            VSDEP(I,J)=VSDA-2.8345E+06*WIL(TC(I,J))*GTP
     1          *SRME(I,J)/(461.51*TK*TC(I,J)*TC(I,J))
        END IF
C
C             GRAUPEL
        IF(QG(I,J).LT.BIXX .OR. GRME(I,J).GT.0.0) THEN
            VGDEP(I,J)=0.0
        ELSE
            VGDEP(I,J)=TPI*SS*TNG*DG(I,J)*GTP*FREG*RHOCI(I)
        END IF
C
400   CONTINUE
500   CONTINUE
C
      RETURN
      END
*DECK PARKES
      SUBROUTINE PARKES(DTI)
C
C     THIS SUBROUTINE CALCULATES THE LIQUID WATER TERMS FOR
C     A KESSLER BULK WATER PARAMETERIZATION.
C     AUTOCONVERSION IS CALCULATED USING THE KESSLER FORM
C     USING A CUTOFF, QLLIM, CALCULATED FROM THE DROPLET
C     CONCENTRATION.  SEE TAYLOR, PHD DISSERTATION, U.W.,
C     (1987), APPENDIX D, EQN D.16.
C     COLLECTION IS MODELED ASSUMING CONTINUOUS COLLECTION
C     OF A MARSHALL-PALMER PRECIPITATION DISTRIBUTION.
C     THE EQUATION IS FROM COTTON (1972, MWR, 100, 757-763), AND
C     ASSUMES THE VELOCITY EQUATION USED IN DROPS.
C          DTI = 1/(CURRENT TIME STEP)       I
C
*CALL GRID
*CALL CLOUD
*CALL VARCON
*CALL WASSER
*CALL WATCON
C
      DATA STCON/44240.28808/, COLCO/2.155209816/, PECO/5.026548246E+07/
C
      CALL DROPS
C
C     AUTOCONVERSION
      DO 100 J=1,JLEV
      DO 100 I=1,NLVLS
          AUTO(I,J)=0.001*AMAX1(0.0,(QC(I,J)-QLLIM))
100   CONTINUE
C
C     COLLECTION AND EVAPORATION
C     COLCO = (130*PI*GAMMA(3.5)/4)*ONR**.125
C          *(6/(RHO(WATER)*PI*GAMMA(4))**.875
C     PEVAP IS EVAPORATION OR CONDENSATION OF VAPOR ONTO
C     PRECIPITATION DROPS.
```



```
C     STCON=1000.*(PICON*ONR)**0.25/9.0
C     COLCO=.29553118*ONR**0.125
C     PECO=2*PI*ONR
C   TOTAL PRECIPITATION DROPS=ONR*DP
      DO 200 J=1,JLEV
      DO 200 I=1,NLVLS
          IF(QP(I,J).LT.BIXL) THEN
              COLL(I,J)=0.0
              PEVAP(I,J)=0.0
          ELSE
              CALL PSIKMU(PBAR(I),TC(I,J),PSI,TK,DMU)
              STK=STCON*ABS(VP(I,J))*DC(I,J)*DC(I,J)
     1          /(DMU*(RHOC(I)*QP(I,J))**.25)
              ERC=STK*STK/((STK+0.5)*(STK+0.5))
              GWTP=1.0/((WVL(TC(I,J))*WVL(TC(I,J))
     1          /(TK*461.51*TC(I,J)*TC(I,J))
     2          +461.51*TC(I,J)*TC(I,J)/(PSI*EW(TC(I,J)))))
              QSP=EPS*EW(TC(I,J))/(PBAR(I)-EW(TC(I,J)))
C
              COLL(I,J)=AMIN1(COLCO*ERC*QC(I,J)
     1              *(RHOC(I)*QP(I,J))**.875
     2              ,AMAX1(0.0,((QC(I,J)-BIAS)*DTI
     3              -AUTO(I,J))))
              PEVAP(I,J)=-AMIN1((PECO*DP(I,J)*DP(I,J)
     1              *(1.0-QV(I,J)/QSP)*GWTP
     2              *(1.0+0.265*SQRT(RHOC(I)*ABS(VP(I,J)))
     3              *DP(I,J)/DMU))*RHOCI(I))
     4              ,AMAX1(0.0,(QP(I,J)-BIAS)*DTI))
          END IF
200   CONTINUE
      RETURN
      END
*DECK DROPS
      SUBROUTINE DROPS
C
C     THIS SUBROUTINE CALCULATES CLOUD AND PRECIPITATION
C     DROP DIAMETERS AND FALL VELOCITIES.
C
C
*CALL GRID
*CALL CLOUD
*CALL VARCON
*CALL WASSER
*CALL WATCON
C
C     DCCO,DPCO AND VPCO ARE SET IN CLOUDS.
C          DCCO=(6./(PI*RHOWATER*TNC))**CPO
C          DPCO=1./((PI*RHOWATER*ONR)**0.25)
C          VPCO=1.09375*130.*SQRT(PI)
C
      DO 500 J=1,JLEV
      DO 400 I=1,NLVLS
          IF(QC(I,J).LE.BIXL) THEN
              DC(I,J)=0.0
              GO TO 300
          END IF
          DC(I,J)=DCCO*(QC(I,J)*RHOC(I))**CPO
C
C     THE EQUATIONS FOR THE CLOUD DROPLET DIAMETER, PRECIPITATION
C     DIAMETER AND VELOCITY ARE FROM CHENG, PP.15-16, 202.  THE
C     CLOUD DROPLET DIAMETER IS CALCULATED ASSUMING THE NUMBER
C     OF CLOUD DROPLETS REMAINS CONSTANT.  THE PRECIPITATION
C     DIAMETER AND VELOCITY ARE CALCULATED FROM A MARSHALL-
C     PALMER DISTRIBUTION (NP=ONR*EXP(-LAM*D)) ASSUMING THAT
C     ONR IS CONSTANT.
C
300           IF(QP(I,J).LE.BIXL) THEN
                  DP(I,J)=0.0
                  VP(I,J)=0.0
                  GO TO 400
```

QUATERNARY LIBRARY

```
          END IF
          DP(I,J)=DPCO*(QP(I,J)*RHOC(I))**0.25
C
C    THE PRECIPITATION VELOCITY IS ASSUMED TO BE THE MASS
C    WEIGHTED VELOCITY, I.E., VP=(GAMMA(4.5)/6)*VP(DP)
C    WHERE DP IS THE MEAN DIAMETER FROM THE MARSHALL-
C    PALMER DISTRIBUTION.  VP(DP) IS THE EMPIRICAL
C    FORMULA OF SPILHAUS (1948), J. MET., VOL. 5, 108-110
C    (FROM LOPEZ (1973), JAS, VOL. 30, 1354-1358, EQ. 17).
C    THIS IS USED AS IT APPEARS TO GIVE MORE REASONABLE (V.LE.
C    16 M/S)  VELOCITIES THAN A STOKES FLOW USING CD.
C           GAMMA(4.5)/6=1.09375*SQRT(PIE)
C
          VP(I,J)=VPCO*SQRT(DP(I,J))
C
400   CONTINUE
500   CONTINUE
C
      RETURN
      END
*DECK HAIL
      SUBROUTINE HAIL
C
C    THIS SUBROUTINE CALCULATES A GRAUPEL MASS (GMAS),
C    DIAMETER (DG), AND FALL VELOCITY (VG) USING THE
C    FORMULAS FROM STEPHENS, PP.37-39.  THE EQUATIONS HAVE
C    BEEN MODIFIED TO INCLUDE LINEAR TRANSITIONS BETWEEN THE
C    THREE MASS CATEGORIES.
*CALL GRID
*CALL CLOUD
*CALL VARCON
*CALL EIS
*CALL EISCON
C
C       VKCO=SQRT(4.*100.*G/(3.*CD))
C     DATA CD/0.45/,VKCO/58.91/
C
      DO 500 J=1,JLEV
      DO 400 I=1,NLVLS
          IF(QG(I,J).LE.BIXX) THEN
              GMAS=0.0
              VG(I,J)=0.0
              DG(I,J)=0.0
                GO TO 400
          END IF
C
C    CALCULATE THE TOTAL NUMBER OF GRAUPEL, MASS AND DIAMETER
C    ASSUMING THE SLOPE INTECEPT OF THE DISTRIBUTION IS CONSTANT.
C
      CD=0.45
      GRHO=100.
      GMAS=QG(I,J)*RHOC(I)/(TNG)
      IF(GMAS.GE.1.0E-06) THEN
          GRHO=600.
          IF(GMAS.GE.2.0E-05) THEN
              GRHO=900.
              CD=0.6
          END IF
      END IF
      DG(I,J)=(QG(I,J)*RHOC(I)/(PIE*GRHO*TNG))**0.3333333
C
C    CONVERT THE DIAMETER TO CGS FOR THE VELOCITY
C    EQUATION.  VG=.01*VK*SQRT(100.*DG)
C
      IF(GRHO.EQ.100.) THEN
          VK=59.40*SQRT(100000./PBAR(I))
      ELSE
C             VK=VKCO*SQRT(GRHO/RHOC(I))
          VK=SQRT(4.*GRHO*G*RHOC(I)/(3.*CD))
      END IF
```



```
C
C       1.939=GAMMA(4.5)/6
      VG(I,J)=1.939*VK*SQRT(DG(I,J))
C
400   CONTINUE
500   CONTINUE
      RETURN
      END
*DECK SNOW
      SUBROUTINE SNOW
C
C    THIS SUBROUTINE CALCULATES THE DIAMETER AND FALL
C    VELOCITY FOR SNOW PARTICLES USING THE EQUATIONS
C    FROM STEVENS, PP.24-25. SMAS IS CONVERTED TO GRAMS
C    FOR USE IN THE EQUATIONS AND THE RESULTANT
C    VELOCITY AND DIAMETER ARE CONVERTED TO MKS FROM
C    CGS.
C
*CALL GRID
*CALL CLOUD
*CALL VARCON
*CALL EIS
*CALL EISCON
C
      DO 500 J=1,JLEV
      DO 400 I=1,NLVLS
          IF(QS(I,J).LE.BIX2) THEN
              DS(I,J)=0.0
              VS(I,J)=0.0
                GO TO 400
          END IF
C     SNOW MASS IN CGS
      SM=1000.0*QS(I,J)*RHOC(I)/TNSCAL(TC(I,J),QS(I,J)*RHOC(I))
      IF(SM.LT.(1.0E-07)) THEN
          DCM=51.5*SQRT(SM)
          VCM=304.0*DCM
      ELSE
          IF(SM.GT.(1.0E-05)) THEN
              DCM=8.89*SM**0.417
              VCM=153.0*DCM**0.25
          ELSE
              DCM=19.2*SQRT(SM)
              VCM=1250.0*DCM
          END IF
      END IF
C
C    CONVERT DIAMETER AND VELOCITY TO MKS.  THE VELOCITY ALSO
C    HAS A PRESSURE CORRECTION.
C
      DS(I,J)=DCM*0.01
      VS(I,J)=VCM*0.01*SQRT(1.0E+05/PBAR(I))
C
400   CONTINUE
500   CONTINUE
      RETURN
C
      END
*DECK TNSCAL
      FUNCTION TNSCAL(T,QS)
C
C    THIS FUNCTION CALCULATES THE TOTAL NUMBER OF
C    SNOW PARTICLES PER M*M*M USING THE FLETCHER
C    FORMULA. THE SECOND TERM LIMITS THE MAXIMUM
C    SNOW MASS TO 1.0E-06 KG.  QS=KG/M**3
C
*CALL VARCON
*CALL EISCON
C
C     TNSCAL=ONS*EXP((TF-T)*BETA)
      TNSCAL=AMAX1(ONS*EXP((TF-T)*BETA), QS*1.0E+06)
```

QUATERNARY LIBRARY

```
      RETURN
      END
*DECK HEAVI
      FUNCTION HEAVI(X)
C
C     THIS IS THE HEAVISIDE FUNCTION CENTERED AT 0.
C
      HEAVI=0.5+SIGN(0.5,X)
C
      RETURN
      END
*DECK PSIKMU
      SUBROUTINE PSIKMU(P,T,PSI,TK,DMU)
C
C     THIS SUBROUTINE CALCULATES THE MOLECULAR DIFFUSIVITY
C     (PSI), THE THERMAL CONDUCTIVITY (TK), AND THE DYNAMIC
C     VISCOSITY (DMU) FOR A GIVEN TEMPERATURE AND PRESSURE.
C     THE FORMULAS FOR MU AND PSI ARE FROM CHENG, P.31.  THE
C     THERMAL CONDUCTIVITY IS CALCULATED USING KA=A*DMU (ROGERS,P.74)
C     FOR 'A' SET TO YIELD THE CRC KA AT 273.16.  THE ERRORS (COMPARED
C     TO THE CRC, VOL. 52, VALUES) ARE LESS
C     THAN 5% OVER THE RANGE -40 TO 20 C.
C
C     DMU=7.616E-03/(T+120)*(T/296.16)**1.5
C     PSI=2.26E-05*(100000./P)*(T/278.16)**1.81
C
      DMU=(1.494E-06/(T+120.))*T*SQRT(T)
      TK=1828.18*DMU
      PSI=(8.79E-05/P)*T**1.81
C
      RETURN
      END
*DECK RKFOUR
      SUBROUTINE RKFOUR(J,TO,H,XO,X,FCN)
C
C     FOURTH ORDER RUNGE-KUTTA SCHEME FROM APPLIED NUMERICAL
C     ANALYSIS BY C.F. GERALD, P.288
C
      XK1=H*FCN(XO,TO,J)
      XK2=H*FCN(XO+XK1/2.,TO+H/2.,J)
      XK3=H*FCN(XO+XK2/2.,TO+H/2.,J)
      XK4=H*FCN(XO+XK3,TO+H,J)
      X=XO+(XK1+2.*XK2+2.*XK3+XK4)/6.
      RETURN
      END
*DECK INTEP
      SUBROUTINE INTEP(XP,P,X,F,N,IER)
C
C     FROM GRAHAM HILL, PUBLICATIONS OF THE
C     DOMINION ASTROPHYSICAL OBSERVATORY, VOL. XVI,
C     NO. 6
C
C     THIS SUBROUTINE INTERPOLATES A FUNCTION P FOR A
C     GIVEN ARGUMENT VALUE XP USING A TABLE OF N VALUES
C     (X,F).  INTEP USES A SPLINE INTERPOLATION SCHEME
C     BASED ON HERMITE POLYNOMNIALS.
C
C     FOR RANDOM VALUES OF XP
C     CALL INTEP(XP,P,X,F,N,IER) OR AFTER FIRST CALL TO INTEP WITH
C     MONOTONICALLY INCREASING OR DECREASING VALUES OF XP CONSISTENT WITH
C     THE X VECTOR, CALL EINTEP(XP,P,X,F,N,IER).
C
C          XP          -THE CHOSEN ARGUMENT VALUE
C          P           -THE RESULTANT INTERPOLATED VALUE
C          X           -THE VECTOR OF INDEPENDENT VALUES
C          F           -THE VECTOR OF FUNCTIONS OR DEPENDENT VALUES
C          N           -THE NUMBER OF POINTS IN THE (X,F) VECTORS
C          IER         -THE RESULTANT ERROR PARAMETER
C
C     REMARKS
```



```
C          IF XP IS BEYOND EITHER EXTREME IN THE VECTOR X, THE VALUE OF
C          F AT THE EXTREME IS ADOPTED AND IER IS SET TO 2.
C
      REAL LP1,LP2,L1,L2
      DIMENSION F(N),X(N)
5     IER=1
      IO=1
      IUP=0
      IF(X(2).LT.X(1)) IUP=1
      N1=N-1
C
      IF((XP.GE.X(N).AND.IUP.EQ.0).OR.(XP.LE.X(N).AND.IUP.EQ.1))
     1 THEN
            P=F(N)
            IER=2
            RETURN
      ELSE
      IF((XP.LE.X(1)).AND.IUP.EQ.0).OR.(XP.GE.X(1).AND.IUP.EQ.1))
     1 THEN
            P=F(1)
            IER=2
            RETURN
      END IF
      END IF
C
      ENTRY EINTEP(XP,P,X,F,N,IER)
C
      DO 1 I=IO,N
            IF(XP.LT.X(I).AND.IUP.EQ.0) GO TO 2
            IF(XP.GT.X(I).AND.IUP.EQ.1) GO TO 2
1     CONTINUE
      GO TO 5
2     I=I-1
      IF(LEQ.(IO-1)) GO TO 4
      IO=I+1
      LP1=1./(X(I)-X(I+1))
      LP2=1./(X(I+1)-X(I))
      IF(I.EQ.1) FP1=(F(2)-F(1))/(X(2)-X(1))
      IF(I.EQ.1) GO TO 3
      FP1=(F(I+1)-F(I-1))/(X(I+1)-X(I-1))
3     IF(I.GE.N1) FP2=(F(N)-F(N-1))/(X(N)-X(N-1))
      IF(I.GE.N1) GO TO 4
      FP2=(F(I+2)-F(I))/(X(I+2)-X(I))
4     XPI1=XP-X(I+1)
      XPI=XP-X(I)
      L1=XPI1*LP1
      L2=XPI*LP2
      P=F(I)*(1.-2.*LP1*XPI)*L1**2.+F(I+1)*(1.-2.*LP2*XPI1)
     1      *L2**2.+FP2*XPI1*L2**2.+FP1*XPI*L1**2.
      RETURN
      END
*DECK SUMER
      FUNCTION SUMER(X,S,M,N)
C
C     RUDOLF LOESER,8 SEP 64, FROM CHRIS SNEDON'S PROGRAM MOOG
C     THIS FUNCTION IS CALLED HELENA IN MOOG.
C     SUMER IS MODIFIED TO START AT AN ARBITRARY POINT.
C     AS(I) = INTEGRAL FROM X(I-1) TO X(I) OF S(X), FOR I FROM M TO N.
C     UPON RETURN, THE VALUE OF THE FUNCTION WILL BE THE SUM OF ALL
C     AS(I)'S.
C
C
*CALL DUMMY
C
      DIMENSION X(N),S(N)
      DIMENSION AS(101)
      EQUIVALENCE (AS,DUO)
      EQUIVALENCE (ISS,SS),(ISP,SP),(ISM,SM)
C
C     SET VALUE OF FIRST COMPONENT INTEGRAL.
```

QUATERNARY LIBRARY

```
      AS(1)=0.
C     COMPUTE SECOND COMPONENT INTEGRAL,USING SPECIALLY FITTED PARABOLIC
C     SEGMENT
C
      TL=X(M+1)-X(M)
      TH=X(M+2)-X(M)
      AS(2)=TL*TL*((4.*S(M+1)+2.*S(M))/TL-(S(M+2)-S(M))/TH)/6.
C     INITIALIZE SUMER
      ANS=AS(2)
C     LOOP OVER REMAINING INTEGRALS
      DO 110 I=M+2,N
C     IF THIS IS THE LAST INTEGRAL,USE LOWER PARABOLIC SEGMENT.
            K=I-M+1
            IF(LEQ.N) GO TO 106
C     DECIDE WHETHER TO USE THE UPPER OR LOWER PARABOLIC SEGMENT FOR
C     THE CURRENT INTEGRAL
            SP=(S(I+1)-S(I))/(X(I+1)-X(I))
            SS=(S(I)-S(I-1))/(X(I)-X(I-1))
            SM=(S(I-1)-S(I-2))/(X(I-1)-X(I-2))
            IF(ISS-ISM) 101,103,103
101   IF(ISS-ISP) 105,102,102
102   IF(SS) 106,105,105
103   IF(ISS-ISP) 104,104,106
104   IF(SS) 105,106,106
C     SET INDEX TO THE APPROPRIATE PARABOLA
105   J=I
      GO TO 107
106   J=I-1
107   TL=X(J)-X(J-1)
      TH=X(J+1)-X(J-1)
      TU=X(J+1)-X(J)
      FL=S(J)-S(J-1)
C     SELECT FORMULA
      IF(J.NE.I) GO TO 109
      FX=S(I+1)-S(I-1)
      AS(K)=TL*(S(I-1)+(8.*FL*TH*TH-2.*FL*TH*TL-FX*TL*TL)/(6.*TH*TU))
      GO TO 110
C     COMPUTE WITH LOWER PARABOLA
109   FX=S(I)-S(I-1)
      AS(K)=TU*(S(I-1)+(FL*TU*TU+2.*FX*TL*TU+3.*FX*TL*TL)/(6.*TH*TL))
C     UPDATE SUMER
110   ANS=ANS+AS(K)
      SUMER=ANS
      RETURN
      END
*DECK THETAS
      FUNCTION THETAE(P,T,R)
C
C     THESE FUNCTIONS CALCULATE THETA E AND THETA Q FOR SATURATED
C     AND UNSATURATED PARCELS.  THE FORMULAS FOR THETA AND THETAE
C     ARE FROM BOLTON (1980).
C     ENTRY THETAQ CALCULATES THETA Q.
C     FORMULAS FOR THESE ARE FROM PALUCH (1979), JAS
C     VOL.36, PP. 2467-2478.
C
*CALL VARCON
C
      TPO(X)=0.2854*(1.-0.28*X)
C
      RT=EPS*EW(T)/(P-EW(T))
      TL=T
      IF(R.LT.RT) TL=TSAT(P,T,R)
      THETAE=T*((1.0E+05/P)**TPO(R))*
     1    EXP((3.376/TL-.00254)*R*(1+.81*R)*1000.)
      RETURN
C
      ENTRY THETAQ(P,T,RV,RL)
C
      RT=EPS*EW(T)/(P-EW(T))
      IF(RV.LT.RT) THEN
```



```
            TL=TSAT(P,T,RV)
            PFAC=((T/TL)**(1./TPO(RV)))*(1.0E+05/P)
          FACTOR=(PFAC**(-1.12*RV))*EXP(RV*WLV/
     1          (TL*(CPD+CW*RV)))
            EP=(P*RV)/(0.622+RV)
            PO=0.286*(1.0-0.26*RV)
      ELSE
          FACTOR=EXP(RV*WLV/(T*(CPD+(CW*(RV+RL)))))
            EP=EW(T)
            PO=RD/(CPD+CW*(RV+RL))
      END IF
C
      THETAQ=FACTOR*T*(1.0E+05/(P-EP))**PO
      RETURN
      END
*DECK TSAT
      FUNCTION TSAT(P,T,R)
C
C     THIS FUNCTION CALCULATES THE TEMPERATURE AT WHICH A
C     PARCEL BECOMES SATURATED USING THE FORMULA FROM BOLTON
C     (1980),MWR, VOL.108,NO.7, 1046-1053.  AS THE FORMULA
C     USES E IN 'MB' A CONVERSION FACTOR OF 0.01 IS NEEDED.
C
      E=0.01*(R*P/(0.622+R))
      TSAT=(2840./(3.5*ALOG(T)-ALOG(E)-4.805))+55.
      RETURN
      END
*DECK CP
      FUNCTION CP(R)
C
C     THIS FUNCTION CALCULATES THE SPECIFIC HEAT OF MOIST AIR
C     FOR A MIXING RATIO  'R'.
C       CP=(CPD+CPV*R)/(1.+R)
      CP=(1005.+1850.*R)/(1.+R)
      RETURN
C
C     THE FORMULA FOR CW(T) IS FROM PRUPPACHER AND KLETT,
C     P.89.  THE FACTOR OF 4.1868E+03 CONVERTS FROM
C     CAL/G TO J/KG
C
      ENTRY CPW(TZ)
C
      T=TZ-273.16
      IF(T.LE.0.00) THEN
            CPW=(1.0074+(8.29E-05)*T*T)*4.1868E+03
      ELSE
            CPW=(.9979+3.1E-06*(T-35.)*(T-35.)+3.8E-09*
     1          (T-35.)*(T-35.)*(T-35.)*(T-35.))*4.1868E+03
      END IF
      RETURN
C
C     THE FORMULA FOR CI(T) IS FROM PRUPPACHER AND KLETT
C     P.89.
C
      ENTRY CPI(TZ)
C
      T=TZ-273.16
      IF(T.GT.0.00) T=0.00
      CPI=(0.508+0.00175*T)*4.1868E+03
      RETURN
C
      END
*DECK FCT
      SUBROUTINE ETEFCT(RHOO,RHON,N,LBC,RBC)
C
C
C     ***************************************************************************
C
C     ETEFCT(RHOO,RHON,N,LBC,RBC)
C     ORIGINATOR-J.P. BORIS        OCT.1975
```

QUATERNARY LIBRARY

```
C
C   THIS VERSION HAS BEEN MODIFIED TO COMPILE ON THE CYBER 180-855
C   AND USES THE CYBER SIGN AND ABS INTRINSIC FUNCTIONS IN LOOP 21
C   RATHER THAN MASK1,-2 AND -3 AS IN THE ORIGINAL CODE.
C   GRT (AUGUST 28, 1985)
C
C   LOOP OPTIMIZATION FOR CYBER STARTED OCTOBER 1, 1985. (GRT)
C
C
C   DESCRIPTION:  THIS ROUTINE SOLVES GENERALIZED CONTINUITY EQUATIONS
C   OF THE FORM   DRHO/DT=-DIV(RHO*V)+SOURCES   IN EITHER
C   CARTESIAN, CYLINDRICAL OR SPHERICAL GEOMETRY.  THE FINITE-DIFFERENCE
C   GRID CAN BE EULERIAN, SLIDING REZONE, OR LAGRANGIAN AND CAN BE
C   ARBITRARILY SPACED.  THE ALGORITHM USED IS A LOW-PHASE-ERROR FCT
C   ALGORITHM, VECTORIZED AND OPTIMIZED FOR SPEED.  A COMPLETE
C   DESCRIPTION APPEARS IN NRL MEMO REPORT #3237, "FLUX-CORRECTED
C   TRANSPORT MODULES FOR SOLVING GENERALIZED CONTINUITY EQUATIONS."
C
C   ARGUMENTS:  IN THIS ROUTINE THE RIGHT BOUNDARY AT RADR IS HALF A
C   CELL BEYOND THE LAST GRID POINT N AT RADN(N), AND THE LEFT BOUNDARY
C   AT RADL IS HALF A CELL BEFORE THE FIRST GRID POINT AT RADN(1).
C   RHOO     REAL ARRAY(N)   GRID POINT DENSITIES AT START OF STEP   I
C   RHON     REAL ARRAY(N)   GRID POINT DENSITIES AT END OF STEP     O
C   N       INTEGER          NUMBER OF INTERIOR GRID POINTS          I
C   LBC      REAL            LEFT BOUNDARY CONDITION FACTOR          I
C   RBC      REAL            RIGHT BOUNDARY CONDITION FACTOR         I
C
C   LANGUAGE AND LIMITATIONS:  THE SUBROUTINE ETBFCT IS A MULTIPLE-
C   ENTRY FORTRAN ROUTINE IS SINGLE PRECISION (60 BITS CYBER).  THE
C   PARAMETER STATEMENT IS USED TO SET SYMBOLICALLY THE INTERNAL ARRAY
C   DIMENSIONS.  UNDERFLOWS ARE POSSIBLE WHEN THE FUNCTION BEING
C   TRANSPORTED HAS MANY ZEROES.  THE CALCULATIONS GENERALLY MISCONSERVE
C   BY ONE OR TWO BITS PER CYCLE.  THE RELATIVE PHASE AND AMPLITUDE
C   ERRORS (FOR SMOOTH FUNCTIONS) ARE TYPICALLY SEVERAL PERCENT FOR
C   CHARACTERISTIC LENGTHS OF 1-2 CELLS (WAVELENGTHS OF ORDER 10 CELLS).
C
C   ENTRY POINTS:  OGRIDE, NGRIDE, VELOCE, SOURCE, CONSRE.  SEE # 11
C   OF THE DETAILED DOCUMENTION (OR THE LISTING BELOW) FOR THE
C   EXPLANATION AND USE OF THE ARGUMENTS TO THESE OTHER ENTRIES.
C
C   NO AUXILIARY OR LIBRARY ROUTINES ARE CALLED BY ETBFCT.
C
C   MODIFICATIONS:
C         GEOMETRY ADDITIONS IN NGRIDE TO INCLUDE THE
C         CYLINDRICAL GEOMETRY FOR THE CLOUD MODEL.
C              ALPHA= 4       INNER CYLINDER  (ARAD)
C                     5       OUTER REGION    (BRAD)
C         THE AREAS AND VOLUMES ARE COMPUTED WITH THE
C         CLOUD AREAS IN COMMON /VARCON/, I.E., AREAA
C         AND AREAB, CALCULATED IN SUBROUTINE READIN.
C              (NOVEMBER 14, 1985...GRT)
C         TERP(N) AND TERM(2) ARE SET TO 1.0E+75
C         TO FORCE THE ANTIDIFFUSION TO REMOVE MOST
C         OF THE DIFFUSION AT THE BOUNDARIES.
C              (GRT 12-85)
C
C   ********************************************************************
C
*CALL VARCON
C
      REAL    RHOO(N), RHON(N)
      REAL    U(N), RADN(N), C(N), D(N), RHO(N)
      INTEGER  ALPHA
C
      PARAMETER (NPT=202)
      LOGICAL  LSOURC
      REAL    RBC, LBC
      REAL    SOURCE(NPT), SCRH(NPT), RHOT(NPT), DIFF(NPT)
      REAL    ADUGTH(NPT), FLXH(NPT), NULH(NPT), MULH(NPT)
      REAL    LNRHOT(NPT), FSGN(NPT), FABS(NPT), EPSH(NPT)
```



```
      REAL    LORHOT(NPT), TERP(NPT), TERM(NPT), ADUDTH(NPT)
      REAL    LO(NPT), LN(NPT), LH(NPT), RLO(NPT), RLN(NPT)
      REAL    RNH(NPT), ROH(NPT), RLH(NPT), AH(NPT)
C
      COMMON/ADVECT/ADUGTH,NULH,MULH,ADUDTH,LO,LN,LH,
     1          RLO,RLN,RNH,ROH,RLH,AH
C
      EQUIVALENCE (EPSH(1), SCRH(1))
      EQUIVALENCE (FLXH(1), SCRH(1)), (FSGN(1), RHOT(1))
      EQUIVALENCE (FABS(1), SCRH(1)), (TERP(1), SOURCE(1))
      EQUIVALENCE (TERM(1), SOURCE(1))
C
C
C     MASK1,2,3 ARE HEXADECIMAL FOR 32 BIT MACHINE
C     DATA MASK1,MASK2,MASK3 /Z"80000000",0.998,Z"7FFFFFFF" /
      DATA PI/3.141592654/,FTP1/4.188790207/
      DATA ROH/NPT*1.0/,SOURCE/NPT*0.0/, LSOURC/.FALSE./
C
C CALCULATE THE DIFFUSIVE AND CONVECTIVE FLUXES
      NP=N+1
      DO 11 I=2,N
      FLXH(I)=0.5*ADUDTH(I)*(RHOO(I)+RHOO(I-1))
   11 DIFF(I)=NULH(I)*(RHOO(I)-RHOO(I-1))
      RHOL=RHOO(1)*LBC
      RHOR=RHOO(N)*RBC
      DIFF(1)=NULH(1)*(RHOO(1)-RHOL)
      DIFF(NP)=NULH(NP)*(RHOR-RHOO(N))
      FLXH(1)=0.5*ADUDTH(1)*(RHOO(1)+RHOL)
      FLXH(NP)=0.5*ADUDTH(NP)*(RHOR+RHOO(N))
C
C CALCULATE LAMBDAO*RHOT, THE TRANSPORTED MASS ELEMENTS.
      DO 12 I=1,N
   12 LORHOT(I)=LO(I)*RHOO(I)-FLXH(I+1)+FLXH(I)
C
C ADD IN THE SOURCE TERMS AS APPROPRIATE.
      IF(.NOT. LSOURC) GO TO 14
      DO 13 I=1,N
   13 LORHOT(I)=LORHOT(I)+SOURCE(I)
C
C CALCULATE THE PHOENICAL ANTIDIFFUSIVE FLUXES HERE.
   14 RHOT(1)=LORHOT(1)*RLO(1)
      DO 17 I=2,N
      RHOT(I)=LORHOT(I)*RLO(I)
   17 FLXH(I)=MULH(I)*(RHOT(I)-RHOT(I-1))
      FLXH(1)=MULH(1)*(RHOT(1)-LBC*RHOT(1))
      FLXH(NP)=MULH(NP)*(RBC*RHOT(N)-RHOT(N))
C
C DIFFUSE THE SOLUTION RHOT USING OLD FLUXES.
      LNRHOT(1)=LORHOT(1)+DIFF(2)-DIFF(1)
      RHOT(1)=LNRHOT(1)*RLN(1)
      DO 20 I=2,N
      LNRHOT(I)=LORHOT(I)+DIFF(I+1)-DIFF(I)
C
C CALCULATE THE TRANSPORTED/DIFFUSED DENSITY AND GRID DIFFERENCES.
      RHOT(I)=LNRHOT(I)*RLN(I)
   20 DIFF(I)=RHOT(I)-RHOT(I-1)
      DIFF(1)=RHOT(1)-LBC*RHOT(1)
      DIFF(NP)=RBC*RHOT(N)-RHOT(N)
C
C CALCULATE THE SIGN AND MAGNITUDE OF THE ANTIDIFFUSIVE FLUX.
      DO 21 I=1,NP
      FABS(I)=ABS(FLXH(I))
   21 FSGN(I)=SIGN(1.0,DIFF(I))
C
C CALCULATE THE FLUX-LIMITING CHANGES ON THE RIGHT AND THE LEFT
      DO 23 I=1,N+1
   23 TERP(I)=FSGN(I)*LN(I)*DIFF(I+1)
      TERP(N)=1.0E+75
      TERP(NP)=1.0E+75
      DO 24 I=1,NP
```

QUATERNARY LIBRARY

```
 24   FABS(I)=AMIN1(TERP(I),FABS(I))
      DO 25 I=3,NP
 25   TERM(I)=FSGN(I)*LN(I-1)*DIFF(I-1)
      TERM(1)=1.0E+75
      TERM(2)=1.0E+75
C
C CORRECT THE FLUXES COMPLETELY NOW.
      DO 35 I=1,NP
 35   DIFF(I)=AMIN1(FABS(I),TERM(I))
      DO 36 I=1,NP
 36   FLXH(I)=AMAX1(0.0, DIFF(I))
      DO 37 I=1,NP
 37   FLXH(I)=FSGN(I)*FLXH(I)
C
C CALCULATE THE NEW FLUX-CORRECTED DENSITIES.
      DO 41 I=1,N
      SOURCE(I)=0.0
 41   RHON(I)=RLN(I)*(LNRHOT(I)-FLXH(I+1)+FLXH(I))
      LSOURC=.FALSE.
      RETURN
C
C     ------------------------------------------------------
C
      ENTRY VELOCE(U, N, UL, UR, DT)
C
C     **********************************************************************
C
C     VELOCE (U,N,UL,UR,DT)
C     DESCRIPTION:  THIS ENTRY CALCULATES ALL VELOCITY-DEPENDANT COEFFS.
C
C     ARGUMENTS:
C     U       REAL ARRAY(N)   FLOW VELOCITY AT THE GRID POINT        I
C     N       INTEGER         NUMBER OF INTERIOR GRID POINTS         I
C     UL      REAL            VELOCITY OF FLOW AT LEFT BOUNDARY      I
C     UR      REAL            VELOCITY OF FLOW AT RIGHT BOUNDARY     I
C     DT      REAL            STEPSIZE FOR THE TIME INTEGRATION      I
C
C     **********************************************************************
C
C
C CALCULATE THE INTERFACE AREA X VELOCITY DIFFERENTIAL X DT.
      NP=N+1
      DTH=0.5*DT
      DO 101 I=2,N
 101  ADUDTH(I)=AH(I)*DTH*(U(I)+U(I-1))-ADUGTH(I)
      ADUDTH(1)=AH(1)*DT*UL-ADUGTH(1)
      ADUDTH(NP)=AH(NP)*DT*UR-ADUGTH(NP)
C
C CALCULATE THE HALF-CELL EPSILON (V*DT/DX)
      DO 104 I=1,NP
      EPSH(I)=ADUDTH(I)*RLH(I)
C
C NEXT CALCULATE THE DIFFUSION AND ANTIDIFFUSION COEFFICIENTS.
C VARIATION WITH EPSILON MEANS FOURTH-ORDER ACCURATE PHASES.
      NULH(I)=0.1666666667+0.3333333333*EPSH(I)*EPSH(I)
      MULH(I)=0.25-0.5*NULH(I)
      NULH(I)=LH(I)*NULH(I)
 104  MULH(I)=LH(I)*MULH(I)
      RETURN
C
C
C     ------------------------------------------------------
C
      ENTRY NGRIDE(RADN, N, RADL, RADR, ALPHA)
C
C     **********************************************************************
C
C     NGRIDE(RAQDN, N, RADL, RADR, ALPHA)
C     DESCRIPTION:  THIS ENTRY SETS NEW GEOMETRY VARIABLES AND COEFFS.
C
```



```
C     ARGUMENTS:
C     RADN    REAL ARRAY(N)   NEW GRID POINT POSITIONS                I
C     N       INTEGER         NUMBER OF INTERIOR GRID POINTS         I
C     RADL    REAL            POSITION OF THE LEFT BOUNDARY          I
C     RADR    REAL            POSITION OF THE RIGHT BOUNDARY         I
C     ALPHA   INTEGER           =1  FOR CARTESIAN GEOMETRY           I
C                               =2  FOR CYLINDRICAL GEOMETRY         I
C                               =3  FOR SPHERICAL GEOMETRY           I
C                               =4  FOR INNER CLOUD REGION           I
C                               =5  FOR OUTER CLOUD REGION           I
C     **********************************************************************
C
C CALCULATE THE NEW HALF-CELL POSITIONS AND GRID CHANGES.
      NP=N+1
      DO 202 I=2,N
 202  RNH(I)=0.5*(RADN(I)+RADN(I-1))
      RNH(1)=RADL
      RNH(NP)=RADR
C
C CALCULATE THE THREE COORDINATE SYSTEMS
      GO TO (203, 206, 209, 220, 230), ALPHA
C
C CARTESIAN COORDINATES.
 203  DO 204 I=1,NP
 204  AH(I)=1.0
      DO 205 I=1,N
 205  LN(I)=RNH(I+1)-RNH(I)
      GO TO 250
C
C CYLINDRICAL COORDINATES.
 206  DO 207 I=1,NP
      DIFF(I)=RNH(I)*RNH(I)
 207  AH(I)=PI*(ROH(I)+RNH(I))
      DO 208 I=1,N
 208  LN(I)=PI*(DIFF(I+1)-DIFF(I))
      GO TO 250
C
C SPHERICAL COORDINATES
 209  DO 210 I=1,NP
      DIFF(I)=RNH(I)*RNH(I)*RNH(I)
 210  SCRH(I)=(ROH(I)+RNH(I))*ROH(I)
      DO 211 I=1,NP
 211  AH(I)=FTPI*(SCRH(I)+RNH(I)*RNH(I))
      DO 212 I=1,N
 212  LN(I)=FTPI*(DIFF(I+1)-DIFF(I))
      GO TO 250
C
C CLOUD GEOMETRY IS RIGHT CYLINDRICAL WITH THE MAIN
C AXIS PARALLEL TO THE FLUID FLOW.
C    INNER CYLINDER OF RADIUS ARAD
 220  AH(NP)=AREAA
      DO 225 I=1,N
           AH(I)=AREAA
 225       LN(I)=AREAA*(RNH(I+1)-RNH(I))
      GO TO 250
C
C    OUTER CYLINDER OF RADIUS BRAD
 230  AH(NP)=AREAB
      DO 235 I=1,N
           AH(I)=AREAB
 235       LN(I)=AREAB*(RNH(I+1)-RNH(I))
C
C NOW THE GEOMETRIC VARIABLES WHICH ARE SYSTEM INDEPENDANT.
 250  DO 255 I=2,N
 255  LH(I)=0.5*(LN(I)+LN(I-1))
      LH(1)=LN(1)
      LH(NP)=LN(N)
      DO 260 I=1,N
 260  RLN(I)=1.0/LN(I)
      DO 265 I=1,NP
```

QUATERNARY LIBRARY

```
285  ADUGTH(I)=AH(I)*(RNH(I)-ROH(I))
     DO 270 I=2,N
270  RLH(I)=0.5*(RLN(I)+RLN(I-1))
     RLH(1)=RLN(1)
     RLH(NP)=RLN(N)
     RETURN
C
C
C    ------------------------------------------------------
C
     ENTRY SOURCES(N, DT, MODES, C, D, DL, DR)
C
C    **********************************************************************
C
C    SOURCES(N, DT, MODES, C, D, DL, DR)
C    DESCRIPTION:  THIS ENTRY ACCUMULATES DIFFERENT SOURCE TERMS.
C
C    ARGUMENTS:
C    N       INTEGER        NUMBER OF INTERIOR GRID POINTS          I
C    DT      REAL           STEPSIZE FOR THE TIME INTEGRATION       I
C    MODES   INTEGER        =1  COMPUTES +DIV(D)                    I
C                           =2  COMPUTES +C*GRAD(D)                 I
C                           =3  ADDS +D TO THE SOURCES              I
C    C       REAL ARRAY(N)  ARRAY OF SOURCE VARIABLES AT GRID PTS   I
C    D       REAL ARRAY(N)  ARRAY OF SOURCE VARIABLES AT GRID PTS   I
C    DL      REAL           LEFT BOUNDARY VALUE OF D                I
C    DR      REAL           RIGHT BOUNDARY VALUE OF D               I
C
C    **********************************************************************
C
     NP=N+1
     DTH=0.5*DT
     DTQ=0.25*DT
     GO TO (310, 320, 330), MODES
C
C +DIV(D) IS COMPUTED CONSERVATIVELY AND ADDED TO THE SOURCES.
310  DO 311 I=2,N
311  SCRH(I)=DTH*AH(I)*(D(I)+D(I-1))
     SCRH(1)=DT*AH(1)*DL
     SCRH(NP)=DT*AH(NP)*DR
     DO 312 I=N,1,-1
312  SOURCE(I)=SOURCE(I)+SCRH(I+1)-SCRH(I)
     LSOURC=.TRUE.
     RETURN
C
C +C*GRAD(D) IS COMPUTED EFFICIENTLY AND ADDED TO THE SOURCES.
320  DO 321 I=2,N
321  SCRH(I)=DTQ*(D(I)+D(I-1))
     SCRH(1)=DTH*DL
     SCRH(NP)=DTH*DR
     DO 323 I=N,1,-1
     DIFF(I)=SCRH(I+1)-SCRH(I)
323  SOURCE(I)=SOURCE(I)+C(I)*(AH(I+1)+AH(I))*DIFF(I)
     LSOURC=.TRUE.
     RETURN
C
C +D IS ADDED TO THE SOURCES IN AN EXPLICIT FORMULATION.
330  DO 331 I=1,N
331  SOURCE(I)=SOURCE(I)+DT*LO(I)*D(I)
     LSOURC=.TRUE.
     RETURN
C
C    ------------------------------------------------------
C
     ENTRY OGRIDE(N)
C
C    **********************************************************************
C
C    OGRIDE(N)
C    DESCRIPTION:  THIS ENTRY COPIES OLD GRID AND GEOMETRY VARIABLES
```



```
C
C    ARGUMENTS:
C    N       INTEGER        NUMBER OF INTERIOR GRID POINTS          I
C
C    **********************************************************************
C
C COPY THE PREVIOUSLY NEW GRID VALUES TO BE USED AS THE OLD GRID
     NP=N+1
     DO 401 I=1,N
     LO(I)=LN(I)
401  RLO(I)=RLN(I)
     DO 402 I=1,NP
402  ROH(I)=RNH(I)
     RETURN
C
C
C    ------------------------------------------------------
C
     ENTRY CONSRE(RHO, N, CSUM)
C
C    **********************************************************************
C
C    CONSRE(RHO, N, CSUM)
C    DESCRIPTION:  THIS ENTRY COMPUTES THE OSTENSIBLY CONSERVED SUM.
C
C    ARGUMENTS:
C    RHO     REAL ARRAY(N)  GRID POINT VALUES FOR CONSERVATION SUM  I
C    N       INTEGER        NUMBER OF INTERIOR GRID POINTS          I
C    CSUM    REAL           VALUE OF THE CONSERVATION SUM OF RHO    O
C
C    **********************************************************************
C
C COMPUTE THE OSTENSIBLY CONSERVED TOTAL MASS (BEWARE YOUR B.C.)
     CSUM=0.0
     DO 501 I=1,N
501  CSUM=CSUM+LN(I)*RHO(I)
     RETURN
C
C
     END
*DECK ADTRAC
     SUBROUTINE ADTRAC(J,DT,DWPDZ,WS)
C
C    THIS IS AN ADVECTION ROUTINE FOR A CONSERVED TRACER
C
*CALL GRID
*CALL CLOUD
*CALL CHEM
*CALL VARCON
*CALL DUMMY
*CALL CBLOK
C
     DIMENSION DWPDZ(101,2),WS(101,2),FARR(101),FCON(101)
C            TRACER
     CALL FORSUB(J,WS,CC,4,FARR,FCON)
     CALL FLUXES(CC,CBAR,J,DU5,DU8)
     CALL ENTRAIN(J,CC,CBAR,DU3,DU0)
     IF(J.EQ.1) THEN
          DO 100 I=1,NLVLS
               CSTOR(I)=DU8(I)
               CENT(I)=DU0(I)
100       CONTINUE
     END IF
C
     DO 105 I=1,NLVLS
          DU2(I)=RHOC(I)*CC(I,J)
          DU8(I)=ADCO(J)*RHOC(I)*(DU8(I)-FLOAT(J-1)*ARB
     1         *CSTOR(I))
          DU8(I)=DU3(I)*DWPDZ(I,J)-FLOAT(J-1)*ADCO(3)
     1         *(DU3(I)-CENT(I))*DWPDZ(I,1)
```

```
105   CONTINUE
      DO 110 I=1,NLVLS
            DU0(I)=DU5(I)+DU8(I)+DU9(I)
110   CONTINUE
C
      CALL SOURCES(NLVLS,DT,8,UNO,DU0,0.0,0.0)
      CTB=CC(1,J)/CBAR(1)
      CALL ETBFCT(DU2,DU1,NLVLS,CTB,0.0)
      DO 115 I=1,NLVLS
            CC(I,J)=DU1(I)*RHOCI(I)*(1.-FCON(I))+FCON(I)*FARR(I)
115   CONTINUE
      RETURN
      END
```

QUATERNARY RESEARCH CENTER
UNIVERSITY OF WASHINGTON
SEATTLE, WASHINGTON
98195

8705 102

QUATERNARY LIBRARY