A NUMERICAL SIMULATION OF A COLD OROGRAPHIC CLOUD SYSTEM

by

R.G.Derickson
E.C.Nickerson
and
J.A.Peterka

Prepared Under

National Science Foundation
Grant Number GA-26580

Washington, D.C.

Fluid Dynamics and Diffusion Laboratory
College of Engineering
Colorado State University
Fort Collins, Colorado

February 1973

CER72-73RGD-ECN-JAP19

ABSTRACT

A NUMERICAL SIMULATION OF A COLD OROGRAPHIC CLOUD SYSTEM

A computer simulation of a two-dimensional, meso-scale, cold orographic cloud system, which represents the first stage of development of a comprehensive model, is presented. Simulation is achieved by numerically solving, in finite difference form, a set of time-dependent hydrodynamic and thermodynamic equations. The domain of solution is an 11 kilometer long and 3.5 kilometer deep rectangular box containing a triangular orographic barrier with an altitude of 1 kilometer and base of 3 kilometers. Grid spacing is constant at 100 meters. The equations of the model are based upon Ogura and Phillips (1962), Ogura (1963), and Orville (1965), with appropriate modifications in the energy and vorticity equations relevant to a cold cloud system. Ice microphysics is not included. The condensation-evaporation process is included by parameterization, but the precipitation mechanism is omitted. Condensation is continually driven by the forced lifting of upstream moisture over the orographic barrier and is influenced by the formation of a lee wave structure that evolves in time as the solution progresses from the initial state. The bulk thermal stratification of the model is stable, as governed by the upstream temperature sounding.

An expedient method of initialization, which minimizes the adjustment or "settling down" period associated with the degree of refinement of the initial state of a numerical solution, was developed. Special emphasis was given to the development of physically realistic boundary conditions that minimize artificialities inherent in numerical solutions as caused by wrongly posed numerical boundary conditions. A

significant "state of the art" achievement was realized in developing the appropriate boundary conditions.

Two basic cases were performed, corresponding to one elapsed hour of atmospheric time: one in which the top boundary was a rigid lid, and the other in which the boundary was flexible, allowing wave energy to pass through the boundary. These two cases utilized a "locally" constant eddy exchange coefficient i.e. the coefficient appears as a constant in the transport equations. In both cases a cap cloud formed over the orographic obstacle and a lenticular cloud formed downstream in the first lee wave crest. The clouds contain water only, no ice. A comparison of kinetic energy and cloud evolutions indicated that the flexible boundary is more appropriate than the rigid boundary. A third case was performed, simulating a shorter span of time than the other cases, using a non-linear, finite-differenced eddy exchange coefficient. The flexible boundary was employed in this case. Results favored using the non-linear coefficient over the "locally" constant coefficient of the other cases pending an improvement on the boundary condition for the eddy coefficient at the lower boundary.

ACKNOWLEDGMENTS

The research was made possible by the National Science Foundation, grant GA-26580. A portion of the computing was performed at the National Center for Atmospheric Research at Boulder, Colorado with resources provided by that agency.

TABLE OF CONTENTS

TABLE OF CONTENTS - continued

TABLE OF CONTENTS - continued

LIST OF FIGURES

LIST OF SYMBOLS

Symbol	Definition
Cp	Specific heat of dry air
$e_{s\ell}$	Saturation vapor pressure with respect to liquid
E	Ratio of the molecular weights of water to dry air
$f_1, f_2, f_3, f_4,$	Functions of height used in equation (17) to predict saturation
g	Acceleration due to gravity
g/Kg	Grams per kilogram
h	Grid spacing (100 meters), same as Δx and Δz
i,j	Subscripts in the horizontal and vertical directions, respectively ($1 \leq i \leq I$ and $1 \leq j \leq J$)
K	The eddy exchange coefficient
$\overline{K}$	The "locally" constant eddy exchange coefficient
K_m	Kilometer
$L_{i\ell}$	Latent heat of fusion
L_{iv}	Latent heat of sublimation
m/sec	Meters per second
n	Normal direction i.e. perpendicular to a slope, boundary, etc.
P	Reference pressure
Pr	Production of ice due to nucleation and diffusional growth
Q	Total moisture, including vapor, liquid, and ice phases
R	Gas constant for dry air
R_v	Gas constant for water vapor
t	Time

Symbol	Definition
T_o	Initial temperature as a function of height (^oC)
u,w	Horizontal and vertical components of velocity, respectively
U_{const}	Constant horizontal velocity component at the top boundary
x,z	Space coordinates in horizontal and vertical directions, respectively
α	Horizontal advective factor in the numerical scheme
β	Horizontal turbulent mixing factor in the numerical scheme
∇	Nabla operator $(\frac{\partial}{\partial x} i + \frac{\partial}{\partial z} j)$
∇^2	Laplacian operator $(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2})$
δ	Vertical advective factor in the numerical scheme
ε	Vertical turbulent mixing factor in the numerical scheme
η	Vorticity
θ	Potential temperature deviation from the base state
$\theta(z)$	Initial value of θ as a function of height
Θ	Potential temperature base state
λ	Mixing length constant
ω	Vapor plus liquid phases of water, mixing ratio
ω_i	Ice phase, mixing ratio
ω_ℓ	Liquid phase, mixing ratio
ω_v	Vapor phase, mixing ratio
ω_{vs}	Saturated vapor, mixing ratio
$\overline{\pi}$	Nondimensional pressure, function of height only
ϕ	Entropy deviation from a base state

Symbol	Definition
ψ	Stream function
Δt	Length of a time step
Δx, Δz	Grid spacing, same as h (100 meters)
^{o}K	Degrees Kelvin

1.0 INTRODUCTION

1.1 General Statement of the Research

This report represents completion of the initial phase in the development of a computer-simulated, two-dimensional, meso-scale, cold orographic cloud system. The method of simulation involves the numerical solution of a set of shallow convection hydrodynamic and thermodynamic equations similar to a set first derived by Ogura and Phillips (1962), extended by Ogura (1963), and further extended by Orville (1965). Finite difference analogs are applied to the equations in their analytical form in order to establish the numerical framework. The goal herein is to describe the method of simulation, discuss its merits and limitations, and to indicate both the successes and failures encountered during the development of the cloud model. The work up to the present time has established a firm basis for a more realistic simulation in the future development of the orographic cloud model with the knowledge gained pointing to those areas where more refinement is needed and suggesting future avenues of numerical experimentation. It is envisioned that the cloud model, coupled with actual field observations, will lend greater understanding to the physical processes of the real atmosphere, indicating the relative importance of the mechanisms involved in a cold orographic cloud system. A long range objective is to utilize the model, when it has been developed to a sufficient level of sophistication, to evaluate weather modification potential in a cold orographic system.

1.2 Background Information for the Present Stage of Development

Orography, or mountainous terrain, is known to have considerable influence on both synoptic (global) and meso-scale (local) circulations as well as the associated microclimate. Under varying conditions,

mountain surfaces act as elevated heat and moisture sources or sinks, as well as mechanical lifting devices to upstream heat, momentum, and moisture. The manner in which a mountain behaves is dependent on such interrelated conditions as surface roughness and slope, vegetation and snow covering, and the prevailing seasonal and daily cycles of solar insolation and radiative losses. The thermodynamic stratification of the ambient atmosphere is of prime importance to the general influence of an orographic obstacle on flow behavior. This list is by no means exhaustive and one can realize that orographic influence is quite complex indeed.

In the first stage of development in the model, it was not possible or realistic to incorporate all of the above physical mechanisms. The long range goal is to extend the structure of the model to include other phenomena as experience and knowledge are gained.

In the presented cold orographic cloud model, the major forcing mechanisms are considered to be the upstream temperature, moisture, and momentum. The orographic barrier is treated strictly as a mechanical lifting device which acts neither as a thermodynamic source nor sink. Radiation is therefore omitted at the mountain surface and in the overlying atmosphere. No surface roughness is characterized explicitly.

The horizontal scale of the modelled system is presently not large enough for significant ice growth (i.e. a parcel travels through the system in a relatively short time). Initial calculations of the ice nucleation and growth-by-diffusion processes indicated they were much too small to include in the model for the average residence time of a parcel. The potential exists for the inclusion of ice in the next phase of the model development at which time the size of the system will be

increased significantly. The model is presently 11 kilometers long and 3.5 kilometers deep. Condensation occurs in the model but precipitation of the condensation products does not occur at this time.

Field data have been used to help determine initial conditions, to serve as a guide for boundary conditions, and to act as a basis for comparison with the simulated results. Classical analytical solutions and classical field observations for airflow over mountains also have been consulted for added comparison and reinforcement.

1.3 Aspects of the Numerical Approach

The dependent variables appearing in the equations of the simulated cloud system interact in a complicated and nonlinear manner, making a closed-form analytical solution cumbersome, if not impossible. A numerical approach allows for solutions of the variables at discrete points in space for discrete steps in time. Although this technique does not make drastic linearizing assumptions necessary for an analytical solution (if one is even possible), it has many associated difficulties, which must be overcome.

At large or essentially infinite vertical and horizontal distances from an orographic barrier, definite statements about the magnitude and general behavior of the system variables can be cited. However, in the finite difference case, as opposed to the analytic case, it is not convenient to place the boundaries at large distances from the obstacle because this would increase the number of grid points to the extent that either computer memory is exceeded or the computational time becomes exorbitantly impractical. Therefore, an arbitrary, smaller region of the total atmosphere must be chosen. (The equations are based upon the Boussinesq assumption with regard to density, so the vertical extent of the solution is limited even in the analytical case.)

Boundary conditions must be applied to the resulting artificial boundaries such that a numerical solution is possible, but more importantly, such that physical reality is not hampered. Many numerical solutions appearing in the literature have employed boundary conditions that permit solutions yet distort the physics of the problem. Such results do not correlate with the real world or, at best, are no better than a linearized analytical solution.

In view of this, one of the major pursuits of the research has been to seek realistic boundary conditions that insure uniqueness to the equations in their analytical form, satisfy requirements for a numerical solution, and minimize any artificiality on the character of the solution. A significant degree of success has been realized in this pursuit after the expenditure of several efforts. These various efforts will be summarized in the text of this thesis, with the goal of illustrating the types of difficulties that arise in a numerical solution with respect to boundary conditions.

Another difficulty associated with numerical solutions involves the grid spacing or the distance between discrete points in the grid domain. Phenomena smaller in scale than the grid length, known as "sub-grid" phenomena, (e.g. turbulence, thermodynamic gradients, etc.) are lost to the system unless they are somehow included by a parameterization. Requirements for spatial resolution are most critical at specific regions in the grid network (e.g. where microphysical or dynamical events are most pronounced such as in a cloud or near the earth's surface) which suggests using a variable, or expanded, grid spacing in those areas. For convenience, the presented model has a constant grid spacing of 100 meters, which was chosen with regard to the total size of the grid network.

Turbulent mixing in the model is parameterized by an eddy exchange coefficient which varies in space.

Time resolution, above and beyond numerical stability criteria, generally does not present much of a problem in numerical solutions. Time steps are usually small in comparison to the physical events being modelled but one must proceed cautiously to be certain that this is true. For example, models including sound waves have a much more stringent requirement for temporal resolution than do incompressible models.

Lastly, phenomena larger in scale than the total grid network, known as "super-grid" phenomena, are also lost to the system unless parameterized. For instance, an upstream shear (which is the result of synoptic scale mechanisms) can only be parameterized in a meso-scale model, as was done in the presented cloud model. "Super-grid" phenomena involving interaction between the upper and lower portions of the atmosphere cannot be simulated in the model due to its shallow depth.

1.4 Brief Literature Review

The literature contains many mathematical models (both analytical and numerical) of cloud systems, but the majority of these models either do not consider orography explicitly, or do not consider it at all. However, there are several synoptic scale prognostic and general circulation models now in existence in which orography is parameterized on a subgrid basis. Fairly sophisticated models simulating orographically induced cumulus convection in meso-scale systems appear in the literature, most notably by Orville (1965, 1967, 1968, 1970). Cold orographic simulation, however, is in its infancy. Chappell (1970) and Grant et al (1971) have done extensive work in the area of modelling ice microphysics without modelling a general dynamic structure, but rather, considering a

mean distribution of vertical motion over an orographic obstacle. Their work has been related to the potential of weather modification. Willis (1970) presented a model of cold orographic precipitation which superimposed a parameterized microphysical process on a mean vertical velocity field over an orographic barrier. Because of the limitations inherent in these models, there is a great need for a cold orographic cloud model that is time dependent and which couples the microphysical and dynamical processes.

2.0 THE NUMERICAL MODEL

2.1 Basic Assumptions of the Model

Before proceeding to the equations which describe the simulated orographic cloud system, it is pertinent to enumerate the basic assumptions underlying the model.

These basic assumptions are:

1. The flow is two-dimensional.

2. The fluid is incompressible. Along with the first assumption, this allows definition of a stream function.

3. Eddy viscosity and eddy diffusivity are considered equal. A single eddy coefficient is used in all the transport equations.

4. The eddy exchange coefficient varies in space but the entire field is smoothed to produce a locally constant counterpart for each grid point in the model.

5. Radiation is excluded at the modelled ground surface and in the overlying simulated atmosphere itself.

6. Momentum on the upstream boundary is not affected by the presence of the orographic obstacle, i.e. upstream influence is precluded at the boundary.

7. The microphysics of ice is not included in the model. (The reasons for this exclusion are given in section 1.2 of the introduction. The expressions involving ice are included in the ensuing discussion for completeness, however.)

8. The precipitation process is omitted at this time. Condensation products are carried through the grid network by the flow. (The

precipitation process is relevant only if ice microphysics is included in the model.)

2.2 The Basic Equations

The system of equations that constitute the cloud model evolve from Ogura and Phillips (1962), Ogura (1963), and Orville (1965). In essence, the system represents deviation from a dry adiabatic atmosphere. Entropy is specified with respect to a base state of ice since the model is concerned with sub-freezing phenomena.

The basic transport equations are:

$$\partial\eta/\partial t = -u\,\partial\eta/\partial x - w\,\partial\eta/\partial z + \overline{K}\,\nabla^2\eta + g\,\frac{\partial}{\partial x}\left(\frac{\theta}{\Theta} + E\omega_v - \omega_\ell - \omega_i\right) \tag{1}$$

$$\partial\phi/\partial t = -u\,\partial\phi/\partial x - w\,\partial\phi/\partial z + \overline{K}\,\nabla^2\phi \tag{2}$$

$$\partial Q/\partial t = -u\,\partial Q/\partial x - w\,\partial Q/\partial z + \overline{K}\,\nabla^2 Q \tag{3}$$

$$\partial\omega/\partial t = -u\,\partial\omega/\partial x - w\,\partial\omega/\partial z + \overline{K}\,\nabla^2\omega - Pr \tag{4}$$

where η is vorticity; ϕ is the perturbation of entropy from a reference state; θ is the potential temperature deviation from an adiabatic atmosphere of temperature Θ; Q is the total moisture which includes the ice, liquid, and vapor phases; ω is total moisture minus the ice component; ω_v, ω_ℓ, and ω_i represent the vapor, liquid, and ice phases, respectively. All the moisture variables are dimensionless mixing ratios of moisture substance to dry air. Pr represents the production of ice by nucleation and the growth of existing ice substance by water vapor diffusion. E is the ratio of the molecular weights of water and dry air. The last term in equation (1) is the

"vorticity production" term due to horizontal gradients in the temperature and moisture variables.

$\overline{K}$ is a locally constant eddy exchange coefficient obtained by space averaging the actual K. The eddy coefficient, K, varies in space according to the equation:

$$K = (\Delta x)^3 \lambda |\nabla \eta| \tag{5}$$

as given by Leith (1969) and also discussed by Fox and Lilly (1972). The constant λ is 3.7, as given by Leith, and Δx is 100 meters, which is the grid spacing. The space averaging operator used to obtain the locally constant coefficient, $\overline{K}$, is given by

$$\overline{K}_{i,j} = 0.8\, K_{i,j} + 0.05\, (K_{i+1,j} + K_{i-1,j} + K_{i,j+1} + K_{i,j-1}) \tag{6}$$

where i and j denote subscripting in the horizontal and vertical directions, respectively.

By defining a locally constant eddy exchange coefficient, the equations are simplified to exclude terms containing derivatives of the coefficient. For example, equation (3) would have the following form if the actual K were used:

$$\partial Q/\partial t = -u\, \partial Q/\partial x - w\, \partial Q/\partial z + K\nabla^2 Q + \partial K/\partial x\, \partial Q/\partial x + \partial K/\partial z\, \partial Q/\partial z. \tag{3a}$$

Problems associated with using $\overline{K}$ instead of K are discussed later.

The assumptions of two-dimensionality and fluid incompressibility allow for a stream function which is defined by:

$$\nabla^2 \psi = -\eta \tag{7}$$

with the horizontal component of velocity, u, and the vertical component of velocity, w, given by:

$$u = \partial\psi/\partial z \tag{8}$$

$$w = -\partial\psi/\partial x \ . \tag{9}$$

Vorticity is related to the velocity components by:

$$\eta = \partial w/\partial x - \partial u/\partial z \ . \tag{10}$$

The moisture variables are related by:

$$Q = \omega + \omega_i \tag{11}$$

$$\omega = \omega_{vs} + \omega_\ell \qquad \text{for saturation} \tag{12}$$

$$\omega = \omega_v \qquad \text{for unsaturation} \tag{13}$$

with ω_{vs} representing saturated vapor.

The entropy perturbation is given by:

$$\phi = \frac{\theta}{\Theta} + \frac{L_{iv}\omega_{vs}}{Cp\ \Theta} + \frac{L_{i\ell}\omega_\ell}{Cp\ \Theta} \qquad \text{saturated} \tag{14}$$

$$\phi = \frac{\theta}{\Theta} + \frac{L_{iv}\omega}{Cp\ \Theta} \qquad \text{unsaturated} \tag{15}$$

with L_{iv} and $L_{i\ell}$ representing the latent heats of sublimation and fusion, respectively. Cp is the specific heat of dry air.

Equations (14) and (15) are used to determine ϕ initially and to compute the lower boundary values at the end of each time step as will be explained later.

In order to close the system, providing a link between all variables, it is necessary to have a criterion for determining saturation. Ogura (1963) expressed the saturation mixing ratio by a series expansion of an exponential:

$$\omega_{vs} = \frac{Cp\ \Theta}{L_{v\ell}}\ f_1\ \{1 + f_4\theta + \frac{1}{2}\ (f_4\theta)^2\} \tag{16}$$

where $f_4 = L_{v\ell}/R_v\Theta^2$, R_v is the gas constant of water vapor, and $L_{v\ell}$ is the latent heat of vaporization. Using this expression in equation (14) yields:

$$\theta_{sat} = f_3^{-1} \{f_2 + [f_2^2 + f_3(\phi - \frac{L_{i\ell}}{Cp\Theta} - f_1)]^{1/2}\} \qquad (17)$$

with

$$f_1 = \frac{f_4 e_{s\ell}(T_o)\Theta R}{Cp\ P\ \overline{\pi}\ 1/K}$$

$$f_2 = \frac{1}{2\Theta} + \frac{f_1 f_4}{2}$$

$$f_3 = \frac{f_1 f_4^2}{2}$$

$$e_{s\ell}(T_o) = 6.11\ \text{Exp}\ \{17.27\ [(T_o - 273.16)/(T_o - 35.86)]\}$$

$$T_o = \Theta - gz/Cp$$

$$\overline{\pi} = 1 - \frac{gz}{Cp}/(\Theta + \overline{\theta}\ (z))$$

$\overline{\theta}(z)$ = initial value of θ as a function of height, z .

Here $e_{s\ell}(T_o)$ is the saturation vapor pressure over liquid water; R is the gas constant of dry air; $1/K = Cp/R$; P is the reference pressure; $\overline{\pi}$ is a nondimensional pressure found in the reference by Ogura and Phillips.

A straightforward solution for θ in equation (15) gives:

$$\theta_{unsat} = \Theta\phi - \frac{L_{iv}\omega}{Cp} . \qquad (18)$$

Equations (17) and (18) serve to determine whether saturation has or has not occurred at each grid point in the model for each time step. To elucidate, we summarize:

$$\text{if } \theta_{sat} < \theta_{unsat}, \text{ then } \theta = \theta_{unsat}, \ \omega_v = \omega, \ \omega_\ell = 0 \qquad (19)$$

$$\text{if } \theta_{sat} > \theta_{unsat}, \text{ then } \theta = \theta_{sat}, \ \omega_v = \omega_{vs}, \ \omega_\ell = \omega - \omega_v \qquad (20)$$

In either case if $Q > \omega$, then $\omega_i = Q - \omega$. If $Q \leq \omega$, then $\omega_i = 0$ and $\omega = Q$. Here it will be noted that ω_i is found as a residual dependent upon the two transport equations (3) and (4).

The production term which appears in equation (4) contains both nucleation and diffusional growth of ice crystals. As has been mentioned, ice has been excluded from the model at present. Therefore, equation (4) is not used and Q is substituted in all expressions where ω appears. The potential exists for inclusion of the ice process in the model, pending an increase in the size of the model.

2.3 The Grid Network

The equations of the cloud model are solved at discrete points in a grid network which represents a two-dimensional orographic cloud system eleven kilometers long and three and one-half kilometers deep. Grid spacing is constant at one hundred (100) meters, creating an 110 by 35 basic rectangular shape. The orographic barrier, which is one kilometer high, interrupts this basic shape beginning three kilometers from the upwind face and has windward and leeward slopes of 1/2 and -1, respectively. (See figure 2.1.) The downstream boundary is five kilometers from the downstream edge of the mountain. The windward slope of 1/2 was chosen as more realistic than 1 whereas the leeward slope was taken for numerical convenience and expediency. Since the windward slope is believed to be more important than the leeward slope in cold orographic systems, this unequal treatment of the slopes is justified.

Ten numerically cumbersome points result from the choice of the windward slope. (See figure 2.2.) Finite difference representations of vertical derivatives at these points have truncation errors an order of magnitude greater than the resulting truncating errors at the other points in the grid. It was found that a simple arithmetic average of the two horizontally adjacent points results in an error that is consistent with the rest of the domain. This fact is readily shown by a Taylor series argument.

The lowest grid points, on the slopes and level portions of the grid, do not correspond to the earth's surface but rather, correspond to some distance above the surface. This condition is reflected in the tangential velocity component (slip) and the eddy exchange coefficient at the lower boundary. The other variables are affected by these two parameters to produce the effect that the lower boundary does not coincide with the earth's surface. More detailed discussions concerning the treatment of the slip and the eddy exchange coefficient at the lower boundary are given in section 4.2.2 and section 4.3.3, respectively.

2.4 The Numerical Scheme

2.4.1 Overview of the scheme - The numerical solution is obtained by marching forward in time increments that are limited in size by numerical stability requirements. These time increments are relatively short, ranging from about six to nine seconds in length such that high frequency mechanisms, except for acoustic waves, are properly included with regard to the hundred meter grid spacing. Acoustic waves are prevented, or "filtered", by the incompressible nature of the fluid as governed by the basic equations of the model. Spatial and

temporal derivatives in all equations are replaced by finite difference analogs.

A rather novel time-splitting scheme (see the next section) is used to solve the basic transport equations i.e. equations (1), (2), (3), and (4). (It will be remembered that equation (4) does not appear in the model at present because of the exclusion of ice. Equation (11) is omitted for the same reason.) This scheme is used to determine values of the transported variables at each grid point in the grid network at each time step.

The stream function is obtained each time step by solving equation (7), Poisson's equation, utilizing a successive over-relaxation method (see section 2.4.3). The velocity components are found, once the stream function is known, by solving equations (8) and (9) in centered-difference format.

The eddy exchange coefficient, K, is found at each grid point per given time step by solving equation (5) with centered-difference analogs. Once this is accomplished for the entire grid network, a locally constant coefficient, $\overline{K}$, is established at each grid point by means of equation (6). (See section 2.2.)

The remaining equations of the model, except equation (10), involve no finite differencing; they are solved in a systematic manner based on the variables gotten by finite differencing. Equation (10) is used only to explain the method of initializing the model. (This explanation will be given in section 3.0.) The unnumbered expressions which lie between equations (17) and (18) in section 2.2 are functions of height only; they are computed prior to initiation of the model for use in equation (17).

The order in which the variables are solved each time step is important and will now be given in closing this section. The order of solution is: Q, ϕ, θ, ω_v, ω_ℓ, η, ψ, u, w, K, $\overline{K}$. Appendix A contains both a flow chart and further explanation concerning the numerical scheme.

2.4.2 Time-splitting scheme - The variables Q, ϕ, and η (also ω when ω_i is eventually included) are predicted by a two-step, numerically explicit method involving an intermediate time $t + 1/2$. Crowley (1970) employed this method which utilizes a forward time difference concept and centered difference analogs for the spatial derivatives.

We have, using a representative variable ϕ :

$$\phi_{i,j}^{t+1/2}=\phi_{i,j}^{t}-\frac{1}{2}\alpha^{t}(\phi_{i+1,j}^{t}-\phi_{i-1,j}^{t})+(\frac{1}{2}(\alpha^{t})^{2}+\beta^{t})(\phi_{i+1,j}^{t}-2\phi_{i,j}^{t}+\phi_{i-1,j}^{t}) \tag{21a}$$

$$\phi_{i,j}^{t+1}=\phi_{i,j}^{t+1/2}-\frac{1}{2}\delta^{t}(\phi_{i,j+1}^{t+1/2}-\phi_{i,j-1}^{t+1/2})+(\frac{1}{2}(\delta^{t})^{2}+\varepsilon^{t})(\phi_{i,j+1}^{t+1/2}-2\phi_{i,j}^{t+1/2}+\phi_{i,j-1}^{t+1/2}) \tag{21b}$$

where

$$\alpha^{t} = u^{t}\,\Delta t/\Delta x$$

$$\delta^{t} = w^{t}\,\Delta t/\Delta z$$

$$\beta^{t} = \overline{K}^{t}\,\Delta t/(\Delta x)^{2}$$

$$\varepsilon^{t} = \overline{K}^{t}\,\Delta t/(\Delta z)^{2}$$

Δt = length of a time step .

Superscripts refer to time, and subscripts refer to space. Also $\beta^{t} = \varepsilon^{t}$ since $\Delta x = \Delta z = 100$ meters. Computational stability requires that

$$\alpha^{2} + 2\beta < 1$$

$$\delta^{2} + 2\varepsilon < 1 .$$

The first step, equation (21a), involves x-derivatives only and the second step, equation (21b), involves z-derivatives only. The first step is performed at each grid point of the total grid network before proceeding to the second step. The "vorticity production" term, i.e. last term, of equation (1) is computed by the usual central differencing operator and is added during the first step in the above scheme. The ice production term of equation (4) will be treated in the same manner when it is eventually included in the model.

Equations (21a) and (21b) can be considered as coupled operators, a reasoning which is applied to the slopes where the tangential and normal directions correspond to the first and second steps of the total operation, respectively. This approach maintains consistency in the numerical scheme as a whole.

Another unique feature of the time-splitting scheme, in addition to the coupled operator and non-centered time differencing features, is a purposely built-in damping mechanism. This mechanism, which is proportional to the square of the velocity components, is embodied in the terms $\frac{1}{2}(\alpha^t)^2$ and $\frac{1}{2}(\delta^t)^2$. These additional terms are grouped with the actual damping terms β^t and ε^t, respectively. Since center-differenced advective terms, as is the case in equations (21a) and (21b), are unconditionally unstable if there is no damping, it is essential to include such a mechanism in some way. If the velocities at any point in the model are large and the corresponding $\overline{K}$ is relatively small, this built-in feature insures numerical stability. Results of the model indicate that $\overline{K}$ is much larger than the added factor except near the grid top where turbulent exchange is small and advection is large. Magnitudes of the various terms show that the

built-in damping effect does not alter the character of the solution but only insures numerical stability.

Non-centered differencing in the advective terms leads to a "psuedo-viscosity" or diffusivity (Molenkamp, 1968) which often exceeds the magnitude of the eddy exchange term itself in many numerical applications. This "psuedo" effect is latent and in many cases it is diffucult or impossible to quantify this embedded mechanism. The advantage of the scheme used in this model is that the eddy exchange effects are totally apparent i.e. there are no hidden effects.

2.4.3 Relaxation method to solve for the stream function - The extrapolated Liebmann, or successive overrelaxation method (S.O.R.), is used to solve equation (7), Poisson's equation, for the stream function each time step. Most of the more sophisticated methods available for solving Poisson's equation cannot be used in the model due to the presence of the mountain which disrupts the basic rectangular domain. (What results is a nonconvex region in which not all two points can be connected by a single straight line.) The advanced methods that might conceivably work for a non-rectangular domain, such as the fast fourier technique or various factorization algorithms, place undesirable restrictions on the nature of the boundary conditions i.e. boundary conditions must be periodic, reflective, etc.

The best method for obtaining the optimum relaxation factor necessary for convergence of the S.O.R. seems to be experimentation. For Dirichlet boundary conditions, the presence of the obstacle does not cause any perceptible deviation from the optimum factor given analytically for a pure rectangular domain of the same overall dimensions. This interesting result is due to the fact that the nature of

the associated iterative matrix is not altered by the obstacle (Varga, 1962) and, additionally, the number of grid points occupied by the obstacle is small compared to the total number of grid points in the domain. (In general, the shape of a rectangle and the number of points it comprises influences the value of the optimum relaxation factor.) Boundary conditions (Neumann, extrapolative, etc.) have a profound influence on the optimum relaxation factor for a given rectangle, however, because they change the nature of the iterative matrix. Analytical methods to determine the optimum factor for boundary conditions other than Dirichlet do not seem to be available. However, by using the optimum factor given for a rectangle (with Dirichlet boundary conditions) as a basis, it is possible to experimentally obtain the optimum relaxation factor with the inclusion of an obstacle within the basic rectangular shape and with the desired boundary conditions. This is done by systematically trying values greater and lesser than the optimum factor for the rectangle (with Dirichlet conditions) until the number of iterations, including the obstacle and the chosen boundary conditions, is minimized in solving Poisson's equation. In short, the original relaxation factor serves only to expedite the trial and error procedure by providing an initial "ball park" value.

The relaxation technique requires an initial estimate of the final solution values of the stream function. Using the solution from the previous time step would seem appropriate for this initial estimate. A better method, found by experimentation and comparison, is to perform a second degree extrapolation in time of the stream function. This technique requires storage of the stream function from the three most

recent time steps, but offers no difficulty because of the peripheral memory devices available on most computers.

The ten "odd" grid points discussed in section 2.3 cause no difficulty in the solution of the stream function. A lower order numerical representation was used at these points. When compared to higher order schemes, no detectable difference appeared in the final solution or number of iterations required for convergence. The nature of Poisson's equation which is elliptical, seems to "dilute" the errors of the lower order scheme employed at these ten points which are few in number compared to the total grid network.

2.5 Initial Conditions

This section will summarize the initial conditions of the simulated cloud model. A detailed description concerning the method of initializing the model will be given in section 3.0 since initialization, especially with respect to the flow field, is a major aspect of the research presented in this report.

The variables θ and Q are given initially throughout the entire grid as functions of height only, based upon the initial upwind soundings. The lapse rate for θ is $3.22^{o}K/Km$ with θ equal to $271.3^{o}K$. Total moisture, Q , varies linearly from 3.5 g/Kg at the lowest altitude to 0.5 g/Kg at the grid top. Liquid moisture, ω_{ℓ} , is everywhere zero. The reference pressure, P , is 750 millibars.

The initial stream function is obtained by solving equation (7) with a constant vorticity (i.e. space independent) which is determined from a linear upstream shear. This linear profile is zero at the bottom of the grid and reaches a maximum of 10m/sec at the grid top. The

velocity components are initialized by equations (8) and (9) once the initial stream function is found.

These initial conditions stem from field data by Balick and Rasmussen (1972). The linear shear is only a rough approximation to actual observations and can be easily refined at a later date in both magnitude and shape. The same type of refinement is also possible for the moisture. The lapse rate for temperature is very realistic since it is essentially unchanged from the actual data; it yields a rather stable environment.

2.6 Concept of a Virtual Point

A knowledge of the "virtual" point concept is essential for understanding the next section which is concerned with the boundary conditions of the model. The use of an external or "virtual" point is a powerful method to handle spatial boundary conditions in numerical solutions. This method depends upon the vanishing of either the first or second derivative, but not both, at a boundary. Centered-difference formulae are considered here since they are the type used in the model; centered differences are employed because they cause the least difficulty in numerical solutions. The finite difference analogs to the first and second normal derivatives in centered-difference form are:

$$\partial\phi/\partial n = \frac{\phi_1 - \phi_{-1}}{2h}$$

$$\partial^2\phi/\partial n^2 = \frac{\phi_1 - 2\phi_o + \phi_{-1}}{h^2} \quad .$$

Here ϕ is any variable, n denotes the normal direction, and h is the grid spacing. ϕ_1 lies interior to the boundary, ϕ_o lies on the boundary, and ϕ_{-1} is the "virtual" point which lies outside the

boundary. Now, if one of these derivatives is small compared to the other, or indeed vanishes, we may proceed. If both vanish, the problem is trival.

Consider that the first derivative vanishes, but not the second. Then $\phi_{-1} = \phi_1$. substituting this into the second derivative analog yields: $\partial^2\phi/\partial n^2 = 2(\phi_1 - \phi_0)/h^2$. Now on the other hand, if the second derivative vanishes and not the first, we have: $\phi_{-1} = 2\phi_0 - \phi_1$. This is eliminated in the first derivative analog to give: $\partial\phi/\partial n = (\phi_1 - \phi_0)/h$.

It will be noted that in both cases the remaining derivative analog has the appearance of noncentered differencing. But it is important to realize that this result develops from centered-difference analogs. If both derivatives are non-zero and this representation is attempted for both, computation problems would be certain to appear. If, and only if, one derivative is zero and the other is not, will this technique work.

It is interesting to note that a "virtual" point may "reside" in a different medium if the boundary is an interface between two different substances. But this is of no consequence since the "virtual" point is only a concept based upon physical reasoning at the boundary.

2.7 Boundary Conditions

2.7.1 Introduction to boundary conditions - The boundary conditions for the variables of the simulated cloud system are enumerated in this section with brief accompanying explanations. A later section (section 4.0) will be devoted to an in depth discussion concerning the development of these boundary conditions since this development, along with the method of initializing the model, represents a significant

portion of the research endeavor. Appendix B contains a summary of the boundary conditions and additional explanations in regard to finite difference applications.

2.7.2 Top boundary - For ϕ and Q, it is assumed that vertical mixing is negligible and that the variables continue to vary linearly in the neighborhood of the top boundary for all time. A "virtual" point (see section 2.6) is defined in each case by setting the second derivative analog equal to zero. The "virtual" point is then eliminated in the first derivative analog. The resulting simplified equations at the top boundary are solved using the same time-splitting scheme (section 2.4.2) applied to the interior portion of the grid. In essence, the boundary conditions are $\partial^2\phi/\partial z^2 = 0$ and $\partial^2 Q/\partial z^2 = 0$.

Vorticity is set equal to the value one grid point below the boundary all along the top. This is done at the end of each step in the time-splitting scheme. The boundary condition is essentially $\partial\eta/\partial z = 0$.

A Neumann condition is applied to the stream function in the form $\partial\psi/\partial z = U$, where U is a constant horizontal velocity component along the top and is invariant in time.

The eddy exchange coefficient (i.e. the locally constant coefficient) is treated similarly to vorticity with the difference being that there is no transport equation involved, so the condition is applied only once, at the end of each time step. The boundary condition corresponds to $\partial\overline{K}/\partial z = 0$.

2.7.3 Side boundaries - Linear extrapolation from interior values is applied to ϕ and Q at the end of each time step (after the second step of the time-splitting scheme, not at the end of each step

of the time-splitting scheme) for both lateral boundaries. This corresponds to $\partial^2\phi/\partial x^2 = \partial^2 Q/\partial x^2 = 0$.

Vorticity and the eddy coefficient are set equal to their first interior values at the end of each time step. This is done for both lateral boundaries. Since the side values of vorticity are used only in the first step of the time-splitting scheme, it is redundant to apply the boundary condition twice as for the top boundary. The eddy coefficient has no transport equation so the time-splitting operator is not relevant. For both variables, $\partial/\partial x = 0$ at the lateral boundaries.

The stream function is specified as a function of height at the upstream boundary and does not change with time. At the downstream face, the values from the previous time step are maintained during convergence of the iterative routine (see section 2.4.3) then reset by second degree extrapolation from interior values. In essence $\partial^3\psi/\partial x^3 = 0$ at the downstream face.

2.7.4 Lower boundary - Vorticity and the stream function are zero along the lower boundary, both on the level portions and the mountain slopes. The normal component of velocity is zero and the tangential component is obtained by noncentered differencing at the lower boundary, slopes and level portions.

Q is found at the lower boundary by a similar method as for the top boundary except that different terms are neglected. Here the normal flux of moisture is zero which requires the first derivative with respect to the normal direction to vanish on the slopes and levels, i.e. $\partial Q/\partial n = 0$. In addition, the second derivative is assumed zero, i.e. $\partial^2 Q/\partial n^2 = 0$, lest the lower boundary acts as a sink to moisture.

θ is specified as a function of slope height at the lower boundary and acts as a constraint on the solution. In general, radiative gains and losses at the earth's surface produce a sinusoidal time variation of temperature superimposed on the height dependency. ϕ is then found as a function of θ and Q on the slopes and level portions by using either equation (14) if saturation has occurred or equation (15) if not. A comparison of Q and ω_{VS} as obtained by equation (16) determines the saturation criterion.

For the purpose of the eddy exchange coefficient, the numerical lower boundary is considered ten meters above the physical boundary i.e. the earth's surface. (The tangential velocity component is treated somewhat differently in regard to the distance above the physical boundary. This is more fully explained in section 4.2.2.) The value of the eddy exchange coefficient lies linearly between the value of the grid point immediately above in the normal direction and the value below which must be zero. (Eddy exchange vanishes at the ground.) On the slopes the whole procedure is done in the normal direction.

3.0 METHOD OF INITIALIZING THE MODEL

It is necessary to specify some initial state of the model, that is, specify values for the system variables corresponding to an initial time. Once this initialization is completed, the solution may progress via the time-dependent structure of the basic equations. Undoubtedly there are several ways in which the thermodynamic and dynamic fields may be initialized, since any irregularities that arise from either a crudely or well specified initial state would adjust, i.e. smooth out, as the solution progresses in time. The ultimate solution can be expected to be independent of the starting values but it is desirable from the standpoint of economy to set forth a refined set of initial conditions that requires the least adjustment time possible. The solution becomes valuable only when the initial adjustment, or "settling down" period, is completed. The dynamic variables are perhaps the most difficult to initialize; the thermodynamic variables, which are transported by the dynamic field, offer less difficulty to initialization. This section of the thesis presents a method that seems quite efficient in initializing the model, the technique for initializing the dynamic field being the most significant contribution to the total initialization.

Since the forcing mechanism of the cloud system is the upstream sounding of the dependent variables, it seemed appropriate to initialize all fields as functions of height only, based on the upwind values. The exceptions to this treatment are the stream function and vorticity fields.

Far upstream from the orographic obstacle the only component of velocity that would exist is the horizontal one. Here $w = \frac{\partial w}{\partial x} = 0$ such that equation (10) reduces to $\eta = -\frac{\partial u}{\partial z}$. By specifying a linear upstream shear the vorticity is then constant at locations far upstream. (For a shear that is not linear, but higher degree, vorticity becomes a function of height.) As a first approximation, one can say that this constant vorticity applies to the entire grid domain of the model. This is not to say that vertical motion is nonexistent everywhere in the grid domain (the mountain boundary which causes the streamlines to deviate from a horizontal alignment prevents vertical motion from vanishing), but that $\frac{\partial u}{\partial z}$ and $\frac{\partial w}{\partial x}$ adjust to maintain a constant vorticity i.e. $\frac{\partial w}{\partial x} - \frac{\partial u}{\partial z}$ = constant. Solving equation (7) with the constant vorticity then yields an initial stream field and, hence, an initial velocity field.

It is reasonable to assume an initially constant vorticity everywhere in the model for the purposes of initializing the stream function because the coupling of equations (1) and (7) insures an adjustment influenced by the presence of the mountain as the solution progresses in time. The greatest adjustment is felt in the vicinity of the mountain as would be expected.

Another way to conceptualize this method of initializing the stream function is to consider the vorticity in the cloud system as the sum of a mean quantity governed by the upstream shear and a perturbation quantity (not necessarily small) that expresses the deviation from the mean due to the mountain i.e. $\eta(x,z,t) = \overline{\eta} + \eta'(x,z,t)$. In essence, we can assume $\eta'(x,z,t)$ to be zero at the start of the model for the sake of initialization.

Orville (1967), for convective flow above a two-dimensional triangular mountain, superimposed the potential flow around a circular cylinder, the radius of which equals the mountain height, upon a linear shear flow that begins at a height corresponding to the mountain top and extends upward. From these superimposed flows he obtained initial velocity and vorticity fields. These conditions, along with the initial thermodynamic fields, required lengthy integration on the computer before kinetic energy stabilized and the model could be started. (Orville's forcing mechanisms are heat and moisture on the mountain surface.)

The adjustment time in the proposed model was found to be quite short with kinetic energy changing remarkably little from the start. By comparison, Orville's kinetic energy changed in excess of 30% over an adjustment period of 30 minutes of computer time (then he added heating and moisture at the mountain surface), whereas the kinetic energy of the presented model changed by less than 6% over an adjustment period of approximately 12 minutes, on the same computer.

As an experiment, further improvement was sought by causing the thermodynamic variables to conform to the initial streamlines rather than to horizontal lines based on the upstream soundings. This technique resulted in large horizontal gradients in moisture and temperature near the mountain, hence large vorticity production in that region. (Consider the last term in equation (1).) The adjustment period increased, not decreased.

Initializing the thermodynamic variables as functions of height only and the stream function by a constant vorticity based on the upstream shear allows the equations to adjust the variables smoothly. In short, the upwind values which determine the mean characteristics

of the model are perturbed by the mountain shape in a manner that maintains compatibility between the various parameters as time progresses from the initial state.

In closing this discussion, it is pertinent to mention that the prescribed method of initialization, which seems to minimize the adjustment period of the model, is quite convenient because several starting conditions can be tried without excessive expense on the computer.

4.0 THE NUMERICAL BOUNDARY CONDITIONS AND RELATED DISCUSSION

4.1 Preface

The appropriate boundary conditions were determined by a combination of physical reasoning and numerical experimentation. The "vorticity production" term (last term of equation (1)) and eddy exchange coefficient were invaluable instruments in developing the final boundary conditions because they display a great deal of sensitivity to changes in the other variables. If a change near the boundaries occurred in either one or both of these parameters which was inconsistent with the interior, it was an indication that the boundary conditions were wrongly posed. The other variables were much less sensitive to inconsistencies, requiring more computational time before it became apparent that something was awry.

The discussion to follow will elucidate the successes and failures encountered while trying to achieve reasonable boundary conditions. The importance of boundary conditions cannot be overemphasized in the realm of numerical solutions where a condition may lead to answers which may not be physically meaningful. Also, a boundary condition which may be suitable for one circumstance, such as a neutral environment with no gravity waves, may be completely inadequate for another such as a stably stratified atmosphere which has gravity waves.

4.2 The Stream Function

4.2.1 Inflow boundary - Incoming momentum is one of the primary driving mechanisms in a cold orographic cloud system. Moist air at the lower altitudes is lifted over the orographic barrier to the condensation level, thus supplying cloud water for the nucleation and

growth of ice crystals. Chappell (1970) and Grant et al (1971) discuss the intricacies of cold orographic cloud systems.

It seemed appropriate to specify the horizontal component of velocity at the upstream face as a function of altitude and invariant in time. Specifying the horizontal velocity component specifies the stream function itself, but vertical motion develops as part of the solution. In other words, the positions of the streamlines at the upstream boundary are invariant, but the streamline slopes may change with time. A linear profile was chosen to approximate field observations. It would be a simple matter to experiment with other profiles as the level of sophistication in the model increases.

Kao (1965), Benjamin (1970), and Wong (1970) have shown that stratification which produces gravity waves also causes upstream influence in the form of blocking (stagnation) and other associated effects. Upstream influence depends strongly upon the shape of the barrier and extends far upstream. Lilly (1969) cites field observations which indicate blocking up to the ridge top on the windward side. Evidence seems to indicate a wedge-shaped stagnation zone as shown in figure 4.1. There is also manifestation of local overturning near the slope surface within the blocked region.

One of the weaknesses of the present model is that all the upwind air is forced over the barrier to the exclusion of any blocking or other upstream influence. To properly include upstream influence, it would probably be necessary to perform a synoptic scale balance, i.e. consider mechanisms larger than the present model, because the forcing dynamics on a meso-scale system result from the synoptic scale geostrophic balance. This might be prohibitive in a strictly

two-dimensional model because the synoptic balance is three-dimensional due to coriolis influences. (One alternative may be to place the upstream boundary beyond the stagnation wedge which is not known a priori.)

4.2.2 Lower boundary - As was done by Orville (1967), the stream function is zero along the lower boundary for all time. This type of treatment extends back to the work of Malkus and Witt (1959) who assumed this condition for numerical expediency. They found that the lower boundary was not critical in their particular investigation. The same reasoning was applied to this model, but the validity is open to debate as will be discussed later.

Slip occurs at the model surface (slopes and levels) which is not coincident with the ground, but lies some unspecified distance above it. (This distance is considered to be ten meters for the purpose of establishing a value for the eddy coefficient at the lower boundary.) It is essential to have motion at the lowest level due to the spatial resolution of the model. Otherwise, the thermodynamic variables would have an unreasonable effect on the solution in the vicinity of the lower boundary. (Orville, 1964).

Slip is allowed by computing the tangential velocity using noncentered differencing of the stream function in the normal direction on the slopes and level portions. (This differencing more correctly corresponds to setting the velocity at the bottom equal to the velocity halfway between the lowest grid point and the grid point above. A Taylor series truncation argument which appears in Appendix A verifies this claim.) The normal component of velocity is zero at the lower

boundary which lies within the atmospheric surface layer where vertical motion, or motion normal to a slope, is vanishingly small.

At least two shortcomings are apparent in the present treatment of the lower boundary for the stream function. One shortcoming is that the flow in the region which lies between the numerical boundary and the physical ground is mathematically construed to be invariant in time and space. This arbitrary treatment precludes such behavior as Katabatic winds, separation, etc., which may have profound influence on the dynamics of the whole system. The second shortcoming, which is coupled to the first, is that a viable surface layer cannot be parameterized with the present lack of spacial resolution at the lower boundary. An expanding grid spacing in the proximity of the lower boundary could remedy this lack of resolution.

4.2.3 Top boundary - Since gravity waves occur in the model, it is undesirable to specify a constant stream function at the top boundary as has been used in many numerical solutions appearing in the literature. Reflection of waves off a rigid lid affects the dynamics of the entire system in an unrealistic manner. The only time a rigid lid occurs in nature is the infrequent case when a strong inversion layer is present far above the earth's surface.

A less serious constraint is to specify a constant horizontal component of velocity at the top boundary. Roache (1970) suggests $(\frac{\partial \psi}{\partial z})_{top} = U_{const}$, which is a Neumann boundary condition. Horizontal motion remains invariant whereas vertical motion develops as part of the solution. In the Dirichlet, or rigid lid case, vertical motion vanishes at the top boundary.

Some discussion on the Neumann boundary condition is pertinent. In the two-dimensional linear theory of lee waves, steady state analytical solutions stem from perturbation methods. The mean horizontal component of velocity is constant or a function of height only, whereas the mean vertical component is zero since it corresponds to an undisturbed environment. The perturbation of the horizontal component, which is a function of both space variables, is considered small compared to the mean value and hence is neglected. The perturbation of the vertical component due to the presence of an obstacle is sought as the solution.

The same concept is applied in justifying the top boundary condition for the stream function in this cloud model. At the top, it is assumed that any deviation from some constant horizontal velocity component is small compared to that constant value. Vertical motion everywhere in the system and horizontal motion everywhere except near the top boundary become part of the solution.

Lilly (1971) attests that variation in the horizontal component of the wind can be significant in the presence of large amplitude waves. Generally, wave amplitude increases inversely with the square root of the mean density. The implications are that wave amplitude increases up into the upper troposphere or lower stratosphere where eventual turbulent break-down takes place. The upper boundary of the model is 3.5 kilometers above the earth's surface, well within the troposphere. Very large amplitudes should not occur at this level, but may occur in the immediate vicinity of the mountain lee. Therefore, the upper boundary condition may not be too restrictive nor cause an unwanted wave reflection. A possible shortcoming of the model, however, is that the

top boundary lies too shallow in the atmosphere to the exclusion of any interaction between the stratosphere and troposphere. Lilly's (1972) description of wave momentum flux implies that the shallowness of the model may have serious dynamical consequences.

Two cases were performed on the top boundary for comparative purposes. One with the Neumann condition and the other with the Dirichlet condition. These cases will be discussed in section 5.0.

4.2.4 Downstream boundary - The downstream boundary condition proved to be the most difficult to resolve. Originally it was hoped that a downstream condition given by Roache (1970) could be utilized. He states that for some numerical flow problems it might be possible to neglect horizontal variation in vertical motion at the downstream face i.e. $\frac{\partial w}{\partial x} = \frac{\partial^2 \psi}{\partial x^2} = 0$. This simplifies Poisson's equation, equation (7), to include only vertical derivatives i.e. $\frac{\partial^2 \psi}{\partial z^2} = -\eta$. This last equation is solved at the downstream boundary as a two-point boundary value problem with the stream function zero at the bottom and with a Neumann condition $(\partial\psi/\partial z = U_{const})$ at the top. However, it is solved prior to solving Poisson's equation for the entire domain so that the downstream boundary condition is essentially Dirichlet in nature.

Roache's condition was used in the first runs of the computer model. It was not until other aspects of the model were resolved, and more time steps were taken, that it became apparent that the condition was inadequate. The first thought was that the downstream boundary was not far enough from the mountain for the condition to work properly. Extending the boundary did not solve the problem. Figures 4.2 and 4.3 show the results of these failures. The failures result from the fact that Roache's boundary condition allows no curvature in the streamlines

in the neighborhood of the downstream boundary. When a wave originating at the mountain reaches the downstream face it must conform to this constraint of no curvature. The character of the wave is exaggerated and eventual computational disaster results as shown in the figures.

It is conceivable that this particular boundary condition would work if the numerical boundary were far enough downstream where the wave amplitudes had decayed appreciably. But this would greatly increase computational time. Roache's condition seems to work for the case of a homogeneous atmosphere which has no gravity waves. (Figure 4.4)

The boundary condition finally used was obtained by equating the finite difference analogs of the horizontal second derivative at the first and second interior grid points i.e. $(\frac{\partial^2\psi}{\partial x^2})_{I-1} = (\frac{\partial^2\psi}{\partial x^2})_{I-2}$ and solving for the stream function at the boundary. (See Figure 4.5) This maintains the trend of concavity in the function at the points $i = I-1$ and $i = I-2$ but does not require equality of the curvatures at these two points. What results is a second degree extrapolation from interior points i.e. $\psi_I = 3\psi_{I-1} - 3\psi_{I-2} + \psi_{I-3}$. Since lee waves are considerably longer than the grid spacing a higher degree curve should not be necessary. A numerical interpretation is that the third derivative vanishes one and one-half grid points from the boundary i.e. $(\frac{\partial^3\psi}{\partial x^3})_{I-1\ 1/2} = 0$.

The problem remaining, however, is how to apply this boundary condition and whether it satisfies uniqueness of solution. Should it be applied after every iteration of the relaxation scheme or after convergence of the scheme? If applied every iteration the condition truly represents a vanishing third derivative, but if it is applied after convergence, it is merely an "after the fact" extrapolation.

An heuristic proof of uniqueness was found numerically for the case of application every iteration by converging to the same solution (to within a small tolerance) independent of several initial estimates in the relaxation scheme. A more rigorous proof is shown in Appendix C.

When the condition was employed after every iteration of the relaxation scheme in longer runs of the model, computational time was too great. For Roache's downstream condition the number of iterations averaged about thirty per time step. The new condition (applied every iteration) increased this average to an excess of five hundred iterations per time step. The next best approach was to apply the new condition at the end of convergence, using the boundary values of the stream function from the previous time step during the process of convergence. (The boundary condition is then Dirichlet as far as Poisson's equation is concerned, followed by a spatial extrapolation.) Since the time steps are less than ten seconds, the character of the solution should not be destroyed. Only the occurrence of an unsteady phenomenon that is transient over a very short period would upset the solution. This possibility seems unlikely.

4.3 The Variables ϕ, Q, η, θ, and K

4.3.1 Top boundary - At the top, one can reason that turbulent mixing is negligibly small compared to advection. This assumption has a priori justification because the bulk Richardson number associated with the upwind sounding is large enough to preclude turbulence. Turbulence should occur only in regions of the model where the local Richardson number can be less than one (e.g., near large wave inflection). The top boundary is far enough above the mountain to be governed by the

bulk stability of the model. The boundary is also far enough below the portion of the atmosphere where turbulent breakdown occurs associated with the breaking of large amplitude waves to be isolated from that effect (Lilly, 1972). Therefore, the mixing terms may be neglected in the transport equations at the top boundary.

Neglecting the vertical mixing term and assuming linear variation of a variable at the upper boundary amounts to setting the second derivative equal to zero, i.e., $\frac{\partial^2 \phi}{\partial z^2} = 0$ where ϕ represents any variable in this case. The "virtual" point concept is used to transform the first derivative analog at the top boundary. The horizontal mixing term is retained to lend damping to the numerical scheme. (It offers no difficulty anyway since finite differencing is performed along the top boundary, not across it as for the vertical derivatives.) In this manner, a transformed version of the transport equation appears at the top which is solved along with the equation for the interior, using the same time-splitting operator for consistency.

The a priori reasoning concerning turbulence at the top of the model was reinforced by locally computed Richardson numbers in the model itself. The bulk value is 14.2. The values along the lateral and top boundaries do not deviate significantly from this bulk value. However there is considerable variation near the mountain slopes and in the lee waves as had been expected.

The method of assuming negligible mixing and linear variation worked well for both ϕ and Q but not for η. The eddy coefficient, which is coupled to vorticity, showed a sudden large increase near the top boundary and the vorticity itself changed inconsistently.

The variables ϕ and Q have a natural tendency to vary linearly with height in the model, while vorticity is governed by the shear and

local gradients in the thermodynamic variables. After careful consideration it was felt that the "vorticity production" term should not change significantly with height, nor should the shear, in the vicinity of the top boundary. (This argument applies to each point at a given altitude. The above mentioned parameters would experience horizontal variation from point to point at a given altitude.) This means that the vorticity at the top boundary equals the vorticity one grid point below. Consequently, setting the top value equal to the value immediately below at the end of each step of the time-splitting scheme would seem reasonable. The success in consistency of the vorticity and eddy coefficient that resulted when this technique was employed seems to warrant its usage. The effective boundary condition is $\frac{\partial \eta}{\partial z} = 0$.

θ is a function of ϕ and Q at the top. The moisture variables ω_v and ω_ℓ are functions of θ and Q .

The eddy exchange coefficient is set equal to the value one grid point below, as in the case of vorticity, but this is done at the end of a time step since there is no transport equation involved. This treatment of the eddy coefficient is consistent due to coupling with the vorticity.

During the first runs of the model, extrapolation techniques were performed on the various functions to obtain boundary values. This met with failure. Extrapolations display an erratic behavior since they are hypersensitive to disturbances that are continuously propogating through the system. They are also sensitive to gradients in the variables themselves.

4.3.2 Lateral boundaries - At the beginning it was felt that the thermodynamic variables should be specified at the upstream face (as functions of altitude) since they are driving parameters along with upstream momentum. Figure 4.6 shows the results of that attempt. Disturbances propogating upstream reflect off the boundary causing an erroneous production of vorticity which upsets the entire solution. Matsuno (1966) discusses reflective phenomena of numerical boundaries in detail.

Nitta (1962) suggests various outflow conditions for numerical solutions, one of which is to apply a linear extrapolation from interior values at the end of a time step. (This has the appearance of a vanishing second derivative in the horizontal direction.) Yamada (1971) has enjoyed success with this method at both inflow and outflow boundaries. This technique was finally used in the cloud model for ϕ and Q.

Before resorting to this extrapolative technique, however, a more elegant application of the vanishing second derivative was sought for ϕ and Q. As in the case of the top boundary, the extrapolation implies negligible mixing and linear variation such that the "virtual" point concept might be utilized again. But this met with failure. The solution began to break down in the lower right hand corner of the grid, where velocities are small and mixing becomes significantly greater. However, many things remain unclear in this regard and seem to warrant further investigation.

Vorticity is set equal to the first interior value at the completion of a time step at both lateral boundaries. This method uses the the same reasoning that was applied to the top boundary except that the

horizontal direction is considered. Also, Roache (1970) indicates that numerical drifting errors may result unless vorticity is handled in this manner. Lastly, this particular boundary condition on vorticity is consistent with the condition applied to the thermodynamic variables which are coupled to the vorticity.

The eddy exchange coefficient is handled the same as vorticity in the interest of consistency. The model results justify this decision. Mixing characteristics have small space-wise variation at distances far from the obstacle. Even downstream, where waves produce a periodic variation in mixing, the variation of mixing is very slight over the span of one grid length near the boundary.

4.3.3 Lower boundary - In general, the surface temperature observed in nature is a periodic function of time owing to the daily cycles of solar insolation and radiative losses. In addition, the surface temperature is also a function of slope height. Geiger (1965) explains slope behavior elaborately.

At the present stage of the model development, the span of time over which the calculations are made justify omission of the temporal periodicity. Slope temperature becomes an unchanging function of height only.

If the lower boundary value of Q is known, it is then a straightforward task to determine whether saturation has occurred by comparing the value of Q to the value of ω_{vs} from equation (16). ϕ is then found by using either equation (14) or (15). The problem, of course, is finding the boundary value of Q.

One of the suppositions in the present model is that the mountain surface neither subtracts nor adds moisture to the cloud

system, which is fairly well justified since a snow covered ground does not exhibit the vast evapotranspiration mechanism that occurs on a vegetated surface in the warm seasons. The only sink effect, excluding precipitation, is the nocturnal deposition of frost which is a negligible quantity. In short, the model surface can be treated as an inert entity to moisture.

Therefore, no moisture flux takes place in the air layer next to the surface. This holds for the lowest grid points in the model which lie some small distance above the physical surface. In this region $K\,\partial Q/\partial n = 0$ (where n denotes the normal direction) on the levels and slopes of the model. Furthermore one can argue that moisture is well enough mixed in the neighborhood of the lower boundary such that there is no variation of Q in the normal direction within this neighborhood. Hence all normal derivatives vanish in that region. (The normal direction is considered because it is appropriate for both slopes and levels.)

As a consequence, the transport equation for Q simplifies to include only tangential derivatives. The modified equation is valid only in the vicinity of the lower boundary. It is solved using the coupled operator, equations (21), as for the interior of the grid network.

Somewhere between the lowest grid point and the point which lies above, the flux of moisture in the normal direction is no longer zero. At this higher level it is necessary to retain the terms containing normal derivatives, which means that the lower point becomes involved in the finite differencing for the point above it (in the normal direction). In essence, the lower boundary is independent of the

overlying values for Q but not the converse. This kind of approach stems from the fact that the grid spacing does not properly resolve the moisture behavior near the bottom boundary.

An attempt to use the vanishing first derivative to define a "virtual" point for the second derivative resulted in a persistent sink effect at the bottom boundary. Thus the second derivative was set equal to zero also but with the justification already mentioned. Better resolution using smaller grid spacing near the lower boundary, where things change rapidly with height, would improve the solution in regard to the behavior of moisture.

To apply the same method to ϕ as to Q on the lower boundary, i.e. solve the associated transport equation in reduced form, then solve for θ as a function of ϕ and Q would be inappropriate. For one thing, it is not as easy to determine a viable boundary condition for ϕ at the lower boundary as it is for Q. In addition, entropy is a function of temperature in the strict sense, not vice versa. The behavior at the surface would be erratic if θ were given as a function of ϕ and Q (supposing that a reasonable lower boundary condition were available for ϕ) because this would make θ a transported variable at the surface, whereas the radiation process which controls surface temperature is not a transport phenomenon in the advective and turbulent mixing senses. The method used, that of specifying the temperature at the lower boundary, acts as a constraint on the model. In essence it is assumed that the radiation balance maintains time independent temperatures at the lower boundary.

A value for the eddy exchange coefficient is necessary at the lower boundary since it is used in the reduced equation for Q . Lumley and

Panofsky (1964) state that the eddy exchange coefficients (Viscosity, diffusivity) are zero at the earth's surface. For a neutrally stratified environment the coefficients vary linearly with height within the surface layer, which is anywhere from 20 to 200 meters deep. For the purposes of the eddy coefficient, it is assumed that the first interior grid points and the lower boundary points themselves both lie within the surface layer. The lapse rate in the model is close enough to neutral to assume a linear variation of the eddy coefficient within this layer. Assuming the bottom boundary to be ten meters above the earth's surface, the eddy coefficient is interpolated between zero and the value one grid point above (in the normal direction).

The vorticity is maintained at zero along the lower boundary on both the slopes and level portions. This specification follows Orville (1965) who borrowed from Malkus and Witt (1959). The latter had found that the bottom boundary condition for vorticity had no influence on the general character of their solution. (See sec. 4.2.2)

There is evidence that the coupling, via equation (5), of vorticity and eddy exchange in the cloud model renders the present treatment of vorticity at the lower boundary inappropriate. Gradients in vorticity are artifically large near the bottom, causing the eddy exchange coefficient to behave erratically in that region. The difficulty is not apparent when the space-averaged coefficient is used [equation (6)], but disporportionately large gradients in the coefficient appear if the transport equations are solved in the form of equation (3a) in which the eddy coefficient is finite differenced. The boundary condition for the eddy exchange coefficient itself aggravates

this situation. A more detailed discussion will be given later. (sec. 5.4)

4.4 The Velocity Components

4.4.1 Top boundary - Both velocity components are needed at the top boundary in order to compute ϕ and Q . The horizontal component, u , is constant for all time as has been discussed in section (4.2.3). The vertical component, w , is found in the usual way via equation (9).

4.4.2 Lateral boundaries - Although the velocity components are not needed at the inflow and outflow boundaries they are calculated for inspection. U is calculated in the usual way, via equation (8), but w is gotten by non-centered differencing.

4.4.3 Lower boundary - The normal component of velocity is considered vanishingly small near the lower boundary and, hence, set to zero.

The tangential component, which is needed in the solution of Q at the lower boundary is computed by non-centered differencing. (See the discussion in section (4.2.2).) This manner in which slip is permitted at the lower boundary is somewhat lacking in rigor and may have serious dynamical implications on the model.

5.0 MODEL RESULTS AND DISCUSSION

5.1 General Introduction

The solution of the cloud model is time dependent, with each transport equation containing a temporal term. The equations of the model incorporate time variation from an initial state as a means of convergence to a possible steady state condition, although steady state is not required by the model. Recent evidence by Balick and Rasmussen (1972) suggests that a steady state orographic cloud may not occur in nature. In reality, then, one might expect the model to yield an initially quasi-steady state structure until which time cloud activity becomes pronounced. The length of simulation time that was performed with the model is insufficient to make a final judgment concerning stationarity of the solution.

Two basic cases were performed: one in which the model top is rigid with a Dirichlet boundary condition on the stream function and the other in which a Neumann condition is applied as discussed in section 4.2.3. All other things are the same in these two cases, with both simulating slightly over an hour of real atmospheric time. The Neumann case, which represents the culmination of all the efforts of development in the model thus far, was performed first. Afterwards, it was decided that a rigid lid case would serve as an interesting and perhaps enlightening basis for comparison.

The essence of the two cases appears in Figures 5.1 through 5.10 via the variables Q, θ, ω_ℓ, η and ψ. The figures are drafted duplications of the actual microfilm outputs. Each variable is depicted in six frames, the first corresponding to the initial state with each

ensuing frame representing a hundred (100) time step increment from the previous one. The only exception to this is the last frame which corresponds to the 480 time step or a little over an hour of elapsed atmospheric time.

A third case was explored in which the eddy exchange coefficient was finite differenced in lieu of the space-averaged approach taken in the other two cases. (See the first part of section 2.2) Various subcases evolved which suggested improvements in the lower boundary conditions for both the vorticity and the eddy exchange coefficient. This last case was not run as long as the other two cases because of difficulties that developed at the lower boundary.

5.2 The Neumann Case

In the Neumann case, the evolution of the stream function (Figure 5.5) seems to indicate no sign of wave reflection off the top boundary. However, the wave structure of the model as a whole implies a transient nature, even in the last frame.

A stationary gravity wave system, which is expected in the model, exhibits an upwind tilt alignment according to linear theory and field observation (Lilly, 1969, 1971, 1972). The model, however, shows a very definite downstream tilt in the short, large amplitude wave immediately downstream of the mountain. This tendency lessens as time progresses but continues to persist. In addition, no periodic array of waves is established in the wake of the mountain although the incipience of a periodicity is apparent. Nothing so definite can be said about the tilt of the waves above the mountain top, but it is obvious that the wave amplitudes are not growing with height as occurs in the prototype atmosphere. This lack of amplitude growth is due to the fact that the

model's air density does not decrease with altitude, which is an attendent mechanism in the real atmosphere.

The apparent transience of the solution, even by the 480 time step, may be real, but possibly misleading. Perhaps it is because enough time has not been modelled to achieve a steady state (or quasi-steady state) condition. Another reason may be that the lateral boundaries are not far enough from the obstacle, perhaps permitting transient reflections. A third possibility may be the steepness of the leeward slope which acts to supply a large amount of momentum to the surface under the restoring action of gravity. This momentum should be counteracted by the opposing pressure gradient that develops to the lee of a mountain (Scorer, 1955), but since there is no surface friction in the model, some of the momentum may be reflecting upward, back into the solution.

One thing that seems to justify the solution stationarity, however, is that the lenticular cloud (Figure 5.3) which forms in the wave crests downstream of the mountain remains in one place. As it grows, it does not shift position or advect out of the model as would a non-stationary cloud. (Cold orographic clouds are somewhat stationary in behavior.)

The behavior of the cap cloud forming over the mountain crest, however, seems to indicate a shortcoming in the model. As the solution progresses from the starting conditions the isohumes of Q begin to pack together toward the center of the upstream boundary. (It will be remembered that the upstream boundary is allowed to flux by linear extrapolation from interior values at the end of each time step.) The upstream values of Q lose their linear variation with height. Larger

values near the bottom shift upward and lesser values at the top move downward. The resulting effect is to cause the cloud to form as far upstream as the boundary itself. The whole process seems to be a manifestation of upstream influence, but it cannot be stated with certainty whether the disturbances which propogate upstream to alter the moisture profile are of a real or numeric mode.

The same packing behavior occurs to ϕ and hence θ since the latter is a function of ϕ and Q except on the lower boundary. (See section 2.7.4 and section 4.3.3.) It is likely that the proximity of the upstream boundary to the mountain barrier is partly or entirely responsible for the packing phenomenon.

Since the upstream values of moisture and entropy are altered, this changes the forcing function of the modelled cloud system. The upstream alteration is unsteady so this would surely have some transient effect on the entire system.

The cusp of the mountain appears to have an anomalous effect on the solution at the lower boundary, most visibly on the variable Q (Figure 5.1). The anomoly is manifested in the break that occurs in the lowest isohume which then advects downstream, yet does not reconnect. This peculiar result originates at the cusp where condensation first takes place rather artificially. (The first runs of the model had less upstream moisture such that no condensation took place. The anomaly did not occur in those runs.) Naturally, all the variables receive unrealistic treatment at the mountain cusp which would indeed be a singular point in an analytical solution, and certainly does not have adequate spacial resolution in the numerical model.

The space-averaged eddy coefficient, $\overline{K}$, does not strictly conserve the variables Q, ϕ, and η which have associated transport equations. In the real atmosphere, what is lost due to mixing at one point is gained at another. In the model, however, local source or sink effects arise depending on the sign and magnitude of $\frac{\partial \overline{K}}{\partial x}$ and $\frac{\partial \overline{K}}{\partial z}$. These two factors are omitted as discussed in section 2.2 and if they are small or indeed vanish at each point in the model, the above variables are conserved in the numerical scheme. Calculations indicate that the greatest discrepancies occur near the mountain boundary, but this does not seem to destroy the general structure of the solution.

The relative importance of eddy exchange in a cold orographic cloud system is hard to evaluate in the actual atmosphere. This is one of the things that might be learned from a numerical model. For an environment whose bulk stratification is stable, turbulence can exist only at isolated localities in the cloud system. The overall effect of this local turbulence may not be very important, but this consideration is open for debate.

The downstream boundary conditions appear adequate for all the variables. However, there is distance enough for only one or two waves to form downstream of the mountain. It would be desirable to extend the boundary far enough downstream to allow several waves to develop because this would certainly lend more validity to the solution.

The variables seem well-behaved at the top boundary, where vertical mixing is considered negligible.

The method of handling the ten "odd" points (see sec. 2.3) causes an unreasonable effect on the vorticity. This is most likely due to

the arbitrary zero specification of vorticity at the lower boundary which results in large gradients of vorticity near the boundary. The other variables show no difficulty in this regard. Also the degree of irregularity in the vorticity at these points seems to diminish with time (Figure 5.4).

5.3 The Dirichlet Case

The stream function in the Dirichlet case (Figure 5.10) shows a more exaggerated downstream tilt and greater wave amplitude leeward of the obstacle. The waves at the higher altitudes seem to be longer and certainly have less amplitude than the waves in the Neumann case. Wave energy cannot pass through the top as in the Neumann case. Also, there is evidence of the beginnings of blocking near the surface upwind of the mountain which does not happen in the Neumann case.

The lenticaular cloud in the Dirichlet case (Figure 5.8) displays a highly transient behavior, shifting position toward the downstream face then retreating as time progresses. This serves to indicate the greater degree of non-steady behavior in the Dirichlet case as opposed to the Neumann case.

The packing of the isohumes observed in the Neumann case also occurs in the Dirichlet case, but to a lesser extent. As a consequence, the cap cloud develops more realistically.

The fact that the packing of ϕ and Q is more pronounced in the Neumann case offers a possible explanation for this phenomenon. Wong (1970) shows that the existence of lee waves has an upstream influence on upwind momentum in addition to the effect of blocking. This accompanying influence is dependent on the amplitude of the lee waves downstream and acts to distort the upstream wind profile at

altitudes above the obstacle. In the Dirichlet case the amplitudes of the waves which lie higher than the mountain are less than for the Neumann case. Now although the upstream momentum is fixed at the upstream boundary in both cases, the larger amplitude lee waves in the Neumann case send a stronger "signal" upstream. This "signal" is manifested in ϕ and Q since the boundary values of these variables are allowed to flux each time step. Placing the upstream boundary further from the obstacle would most likely alleviate or certainly decrease the packing effect. One other remedy might be to allow the stream function itself to change upstream along with the other variables.

The anomalous effect of the cusp on the variable Q is less extreme in the Dirichlet case than in the Neumann case and it seems that the effect is only a by-product of the initial "settling down" period in the model. In time, the Neumann case would probably adjust this anomaly also.

The same discussion of section 5.2 regarding the space-averaged eddy coefficient applies to the Dirichlet case also.

A comparison of the kinetic energies for the Neumann and Dirichlet cases shows that the Neumann case remains somewhat more stable after an initial hump in the energy curve. The energy in the Dirichlet case does not show as great a variation at first, but then displays a continuously increasing trend that becomes more extreme toward the end of the simulation time. (See Figure 5.11)

5.4 The Finite-Differenced Case for the Eddy Exchange Coefficient, K

A third case was performed in which the eddy exchange coefficient was finite differenced. This approach is more rigorous, mathematically speaking, than the space-averaged method. The Neumann condition

was placed on the stream function at the top boundary for this last case.

In the first attempt the vorticity was maintained zero at the lower boundary as for the other cases. After only thirty time steps the solution demonstrated an unusual character near the cusp on the leeward side (Figure 5.12a). The situation grew worse with time. A detailed analysis of all the variables led to the suspicion that the problem resulted from the rather arbitrary specification of the bottom vorticity, which is linked to the eddy coefficient via equation (5).

In the next attempt the condition $\partial^2\eta/\partial n^2 = 0$ was placed at the lower boundary (n denotes the normal direction). A "virtual" point was used to modify the analog of the first derivative, $\partial\eta/\partial n$. Since motion in the normal direction becomes vanishingly small near the lower boundary, this condition can be shown (using equation (8) in a rotated form) to give second degree freedom to the tangential velocity at the lower boundary. In this new effort, the solution was much better behaved as seen in Figure 5.12b which corresponds to the hundredth time step. However, the solution is beginning to show signs of trouble at this point and eventually "blows up."

The difficulty stems from the lower boundary condition on K. (It will be remembered from section 4.3.3 that K at the boundary is found by linear interpolation in the normal direction between the value above and the zero value which lies ten meters below at the physical ground.) This condition causes excessively large gradients in the value of K near the lower boundary. In that region, the derivatives $\partial K/\partial x$ and $\partial K/\partial z$ which can be grouped with $-u$ and $-w$

by rearranging equation (3a), complete with these velocity components in their magnitude to the point of disrupting the solution. [Equation (3a) is used only to represent the differencing of K ; the other variables, of course, have corresponding terms in their associated equation.] One possible improvement would be to incorporate the transformed analog of $\partial\eta/\partial n$ into equation (5) to obtain K at the boundary. This does not seem inconsistent with the present resolution of the lower boundary, which lies some unspecified distance above the physical surface. (See section 4.2.2 in regard to slip at the bottom boundary.) The best answer probably lies in better spatial resolution near the lower boundary.

If a variable eddy exchange coefficient is to be considered, it is far more appropriate to finite difference the coefficient than to use the space-averaging technique. Otherwise, the transported variables are not properly conserved in the numerical scheme and physical significance is lost.

6.0 SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

6.1 Summary and Conclusions

The knowledge gained during the development of the cold orographic cloud model offers hope and direction for an improved simulation. The realization of a viable dynamic system upon which to superimpose a microphysical ice process appears close at hand.

The present technique for condensation, which approximates the actual microphysics by a parameterization, seems adequate. In general, condensation strongly depends on nucleating particles. Water vapor cools to a supercooled state without condensing if enough of these particles are not present in the actual atmosphere, whereas the model assumes condensation based only on the temperature criterion of the Clausius Clapeyron relationship. Cloud water appears at reasonable locations in the model even though this parameterization is used.

By comparison, the Neumann boundary condition for the stream function at the upper boundary is superior to the Dirichlet condition, pending an improvement on the upstream boundary to alleviate the packing phenomenon of ϕ and Q. The basic philosophies behind all the other boundary conditions appear sound and seem reinforced by the results of the model.

The numerical scheme, which enjoys a noncentered time-splitting process and fully displays all damping characteristics (there is no hidden or "psuedo" damping) has proved quite successful. Although the scheme is designed for equations with a constant turbulent mixing coefficient, it can also be used for the case of a nonconstant

coefficient by grouping the factors $\partial K/\partial x$ and $\partial K/\partial z$ into the advective terms i.e. $(-u + \partial K/\partial x)$ and $(-w + \partial K/\partial z)$.

The method of initialization, which minimizes the adjustment period of the initial state, is one of the major accomplishments of the research. A variety of starting conditions and boundary condition experimentations can be attempted at reasonable computer expense. This offers an excellent benefit in that a series of boundary experimentations and initial states would possibly lend increased understanding to the physical processes of a cold orographic cloud system.

6.2 Recommendations

One of the most obvious lessons gained from the model is that the boundaries should be extended for dynamical and microphysical reasons. This extension most certainly should be done for the lateral boundaries in order to lessen the upstream influence, to include several wave lengths downstream, and to allow a parcel enough time for significant ice growth. Perhaps the depth of the model can be left unaltered so that the equations themselves, which assume shallow convection, can be retained in their present form. But maintaining the shallowness of the model may possibly preclude an important interplay between the upper and lower portions of the atmosphere, to the detriment of realism in the dynamic structure.

The mountain shape itself is quite unrealistic in addition to the numerical difficulties it creates at the cusp and at the ten "odd" points (sec. 2.3). If the lateral boundaries were extended, more realistic mountain slopes could be depicted. Transforming the chosen shape into a straight line by one of many techniques found in the literature would solve many of the present numerical difficulties at

the lower boundary. Since a transformation would produce a rectangular grid domain, the solution of equation (7) for the stream function could be expedited by more sophisticated methods available for a rectangular shaped grid network. The transformation would also eliminate any "odd" points associated with the slopes.

Better spatial resolution at the lower boundary would improve the solution. This refinement could be accomplished by an expanding grid spacing near the bottom of the grid network. Some mechanism of surface drag should also be included in lieu of the present method of slip at the lower boundary (see sec. 4.2.2). (At present, the numerical boundary is some unspecified distance above a non-slip surface.)

The space-averaged turbulent mixing coefficient should certainly be replaced by the actual coefficient. Because of the bulk stability of the model (Richardson number of 14.2), turbulent mixing is significant only near the mountain boundary, exactly where the present treatment of mixing is least suitable. Using the actual coefficient implies using either $\partial\eta/\partial n = 0$ or $\partial^2\eta/\partial n^2 = 0$ at the lower boundary (see sec. 5.4). The better of the two conditions could be determined by experimentation.

The turbulent mixing mechanism (equation (5)) which is coupled to vorticity is perhaps unsuitable for the cloud model, which is a meso-scale simulation. The assumption of two-dimensional turbulence, upon which the present scheme is based, applies more rigorously to a synoptic scale volume in which the horizontal dimensions are much greater than the vertical dimension. A more appropriate characterization for meso-scale phenomena is suggested by Drake (1972). This method depends

upon the local stabilities and the local dynamic deformations within the cloud system.

The model now assumes equality of eddy viscosity and eddy diffusivity. It could be important to distinguish the two processes for increased realism in the solution, since the modelled system simulates a stably stratified environment in which eddy viscosity should dominate eddy diffusivity.

One method to allow upstream influence may be to place the upstream boundary beyond the stagnation wedge which lies on the windward side of the mountain, while maintaining the present boundary conditions. This method could possibly prevent having to perform a synoptic scale balance for achieving the same result.

Radiation should eventually be included in the model. A first step would be to do this implicitly rather than to invoke a complicated mathematical treatment. For the mountain surface, this could be done by superimposing a sinusoidal time variation of temperature to duplicate the diurnal cycles. Field data by Balick and Rasmussen (1972) implies an analogous time variation in the body of air above the atmospheric surface. Observations show that the lapse rate remains nearly constant but shifts toward colder temperatures toward the evening hours. This type behavior could easily be parameterized in the model. However, refinement of the radiation processes should be made in the final analysis.

Several experiments with surface roughness, type of snow covering, vegetation covering, etc. should eventually be made for the sake of gaining knowledge about their relative importance to the cloud system as a whole.

Inclusion of the precipitation process would be essential to complete the integrity of the model.

REFERENCES

REFERENCES

Balick, L. K. and Rasmussen, J. L., 1972; A case study of the water budget of an orographic cloud. Atmospheric Science Technical Paper 187, Department of Atmospheric Science, Colorado State University.

Benjamin, T. B., 1970; Upstream influence. J. Fluid Mech., Vol. 40, 49-79.

Bergen, J. D., 1969; Cold air drainage on a forested mountain slope. J. Appl. Met., Vol. 8, No. 6, 884-895.

Chappell, C. F., 1970; Modification of cold orographic clouds. Ph.D. Thesis, Colorado State University.

Crowley, W. P., 1970; A numerical model for viscous, free-surface, barotropic wind driven ocean circulations. J. Comp. Phys., Vol. 5, 139-168.

Drake, R. L., 1972; Interacting convective cells in a two-dimensional anelastic system, dry convection. Nation Center of Atmospheric Research preprint, Boulder, Colorado.

Drake, R. L., M. B. Ellingson, D. P. Anderson, P. D. Coyle, 1972; Interacting dry line thermals, numerical experiments. National Center for Atmospheric Research preprint, Boulder, Colorado.

Egger, J., 1970; On the simulation of subgrid orographic effects in numerical forecasting. J. Atmos. Sci., Vol. 27, 896-902.

Fox, D. G. and Lilly, D. K., 1972; Numerical simulation of turbulent flows. Reviews of Geo. and Space Phys., Vol. 10, No. 1, 51-72.

Garabedian, P. R., 1964; Partial Differential Equations, John Wiley and Sons, Inc.

Geiger, R., 1965; The Climate Near the Ground, Harvard University Press.

Grant, L. O., C. F. Chappell, P. W. Mielke, Jr., 1971; The climax experiment for seeding cold orographic clouds. Proc. First Int. Conf. in Cloud Models, Camberra, Australia.

Kao, T. W., 1965; The phenomenon of blocking in stratified flows. J. Geo. Res., Vol. 70, No. 4, 815-822.

Krishnamurti, T. N., 1964; Theory of two-dimensional mountain waves. Reviews of Geo., Vol. 2, No. 4, 593-624.

Leith, C. E., 1969; "Numerical simulation of turbulent flow," in Properties of Matter Under Unusual Conditions (H. Mark and S. Fernbach, Eds.), Interscience Publishers Inc., New York.

Lilly, D. K., 1969; The Colorado lee wave program. Clear Air Turbulence and Its Detection, Plenum Press, 232-245.

Lilly, D. K., 1971; Brief reports, observations of mountain-induced turbulence. J. Geo. Res., Vol. 76, No. 27, 6585-6588.

Lilly, D. K., 1972; Wave momentum flux - a GARP problem. Bull. Amer. Met. Soc., Vol. 53, No. 1, 17-23.

Lin, J. T. and C. J. Apelt, 1970; Stratified flow over an obstacle, a numerical experiment. Project THEMIS, Tech. Rep. No. 4, CER69-70JTL2, Fluid Dynamics and Diffusion Laboratory, Colorado State University.

Lumley, J. L. and H. A. Panofsky, 1964; The Structure of Atmospheric Turbulence, Interscience Publishers Inc., New York.

Malkus, J. S., and G. Witt, 1959; The evolution of a convective element: A numerical calculation. The Atmosphere and the Sea in Motion, Rockefeller Institute Press, 425-439.

Matsuno, T., 1966; False reflection of waves at the boundary due to the use of finite differences. J. Met. Soc. Japan, Vol. 44, No. 2, 145-157.

Molenkamp, C. R., 1968; Accuracy of finite-difference methods applied to the advection equation. J. Appl. Met., Vol. 7, 160-167.

Nickerson, E. C., 1965; A numerical experiment in buoyant convection involving the use of a heat source. J. Atmos. Sci., Vol. 22, 412-418.

Nitta, T., 1962; The outflow boundary condition in numerical time integration of advective equations. J. Met. Soc. Japan, Vol. 40, No. 1, 13-24.

Ogura, Y., 1963; The evolution of a moist convective element in a shallow, conditionally unstable atmosphere: A numerical calculation. J. Atmos. Sci., Vol. 20, 407-424.

Ogura, Y. and N. A. Phillips, 1962; Scale analysis of deep and shallow convection in the atmosphere. J. Atmos. Sci., Vol. 19, 173-179.

Orville, H. D., 1964; On mountain upslope winds. J. Atmos. Sci., Vol. 21, No. 6, 622-633.

Orville, H. D., 1965; A numerical study of the initiation of cumulus clouds over mountainous terrain. J. Atmos. Sci., Vol. 24, 684-699.

Orville, H. D., 1968; Ambient wind effects on the initiation of cumulus clouds over mountains. J. Atmos. Sci., Vol. 25, 385-403.

Orville, H. D., 1970; A numerical simulation of the life history of a rainstorm. J. Atmos. Sci., Vol. 27, 1148-1159.

Queney, P., ed., 1960; The airflow over mountains. W.M.O. Tech. Note No. 34.

Roache, P. J., 1970; Sufficiency conditions for a commonly used downstream boundary condition on stream function. J. Comp. Phys., Vol. 6, 317-321.

Scorer, R. S., 1953; Theory of airflow over mountains: II - the flow over a ridge. Quart. J. Roy. Met. Soc., Vol. 79, 70-83.

Scorer, R. S., 1955; Theory of airflow over mountains: IV - separation of flow from the surface. Quart. J. Roy. Met. Soc., Vol. 81, 340-350.

Takeda, T., 1971; Numerical simulation of a precipitating convective cloud: the formation of a "long-lasting" cloud. J. Atmos. Sci., Vol. 28, 350-376.

Taylor, P. A. and Y. Delage, 1971; A note on finite-difference schemes for the surface and planetary boundary layers. Boundary Layer Met., Vol. 2, 108-121.

Varga, R. S., 1962; Matrix Iterative Analysis, Prentice-Hall, Englewood Cliffs, N.J.

Willis, P. T., 1970; A parameterized numerical model of orographic precipitation. EG&G Inc., Boulder, Colorado.

Wong, K. K., 1970; Stratified flow over extended obstacles and its application to topographical effect on vertical wind shear. J. Atmos. Sci., Vol. 27, 884-889.

Yamada, T., 1971; Numerical and wind tunnel simulation of response of stratified shear layers to nonhomogeneous surface features. Ph.D. Thesis, Colorado State University.

APPENDICES

APPENDIX A

A FLOWCHART OF THE MODEL AND FURTHER EXPLANATION CONCERNING THE NUMERICAL SCHEME

APPENDIX A

A FLOWCHART OF THE MODEL AND FURTHER EXPLANATION CONCERNING THE NUMERICAL SCHEME

A flowchart depicting the method of initializing the model (described in section 3.0) is shown in Figure A1. The layout of the model itself (described in section 2.0) is given in the flowchart of Figure A2. Both flowcharts serve to supplement and reinforce their related discussions in the main text of the thesis.

Section 4.2.2 explains how slip is allowed at the lower boundary and makes the claim that the uncentered differencing of the stream function more correctly corresponds to the tangential velocity component half a grid spacing above the lower boundary rather than at the boundary itself. Consider first a single Taylor's series expansion (in the vertical direction) at the lower boundary with j increasing upward (see Figure A3 as a guide)

$$\psi_{j+1} = \psi_j + h\left(\frac{\partial\psi}{\partial z}\right)_j + o(h^2) \tag{A1}$$

from which we obtain

$$\left(\frac{\partial\psi}{\partial z}\right)_j = U_j = \frac{\psi_{j+1} - \psi_j}{h} + o(h) \tag{A2}$$

(remembering that $\frac{\partial\psi}{\partial z} = u$ as given by equation (8)) where U_j is the tangential velocity component at the lower boundary and h is the grid spacing. Now consider the two expansions

$$\psi_{j+1} = \psi_{j+1/2} + h/2\left(\frac{\partial\psi}{\partial z}\right)_{j+1/2} + \frac{(h/2)^2}{2}\left(\frac{\partial^2\psi}{\partial z^2}\right)_{j+1/2} + \frac{(h/2)^3}{6}\left(\frac{\partial^3\psi}{\partial z^3}\right)_{j+1/2} + o(h^4) \tag{A3}$$

$$\psi_j = \psi_{j+1/2} - h/2\left(\frac{\partial\psi}{\partial z}\right)_{j+1/2} + \frac{(h/2)^2}{2}\left(\frac{\partial^2\psi}{\partial z^2}\right)_{j+1/2} - \frac{(h/2)^3}{6}\left(\frac{\partial^3\psi}{\partial z^3}\right)_{j+1/2} + o(h^4) \tag{A4}$$

Subtracting equation (A4) from equation (A3) yields after rearranging

$$\left(\frac{\partial\psi}{\partial z}\right)_{j+1/2} = U_{j+1/2} = \frac{\psi_{j+1} - \psi_j}{h} + o(h^2) \tag{A5}$$

where $U_{j+1/2}$ is the tangential velocity component one half a grid spacing above the lower boundary. One can see that equations (A2) and (A5) differ on their right hand sides only in the order of truncation error, equation (A5) having less error. Therefore the differencing

$$\frac{\psi_{j+1} - \psi_j}{h}$$

more correctly represents the tangential velocity component $U_{j+1/2}$ than U_j . (For this discussion, the series expansions were performed on the level portion of the lower boundary for convenience. The same argument holds for the slopes.)

Flowchart of Initialization of the Model

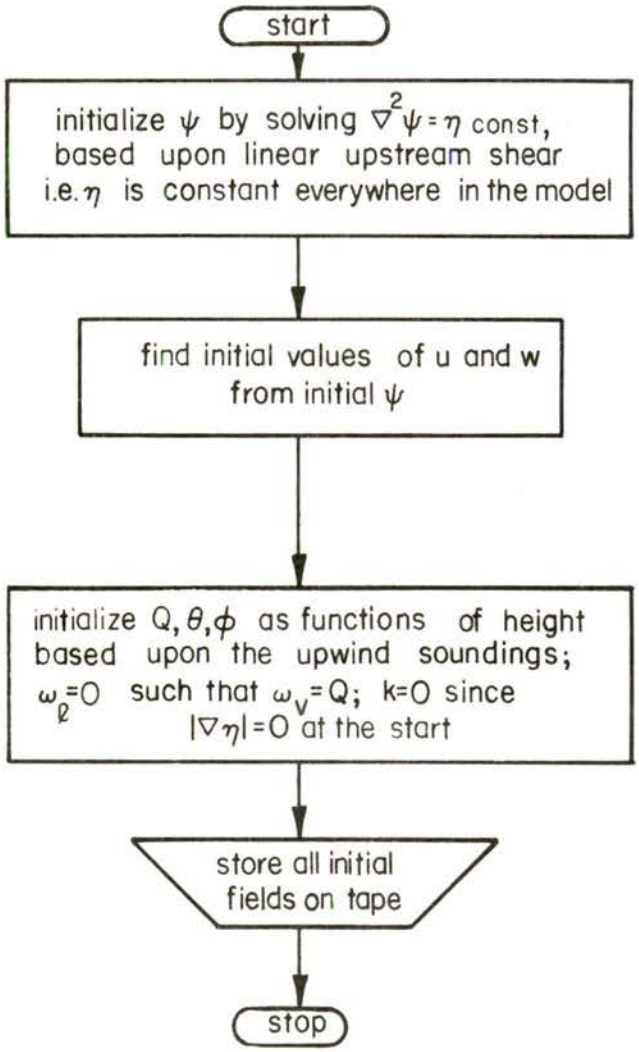


Figure A1.

Flowchart of the Model

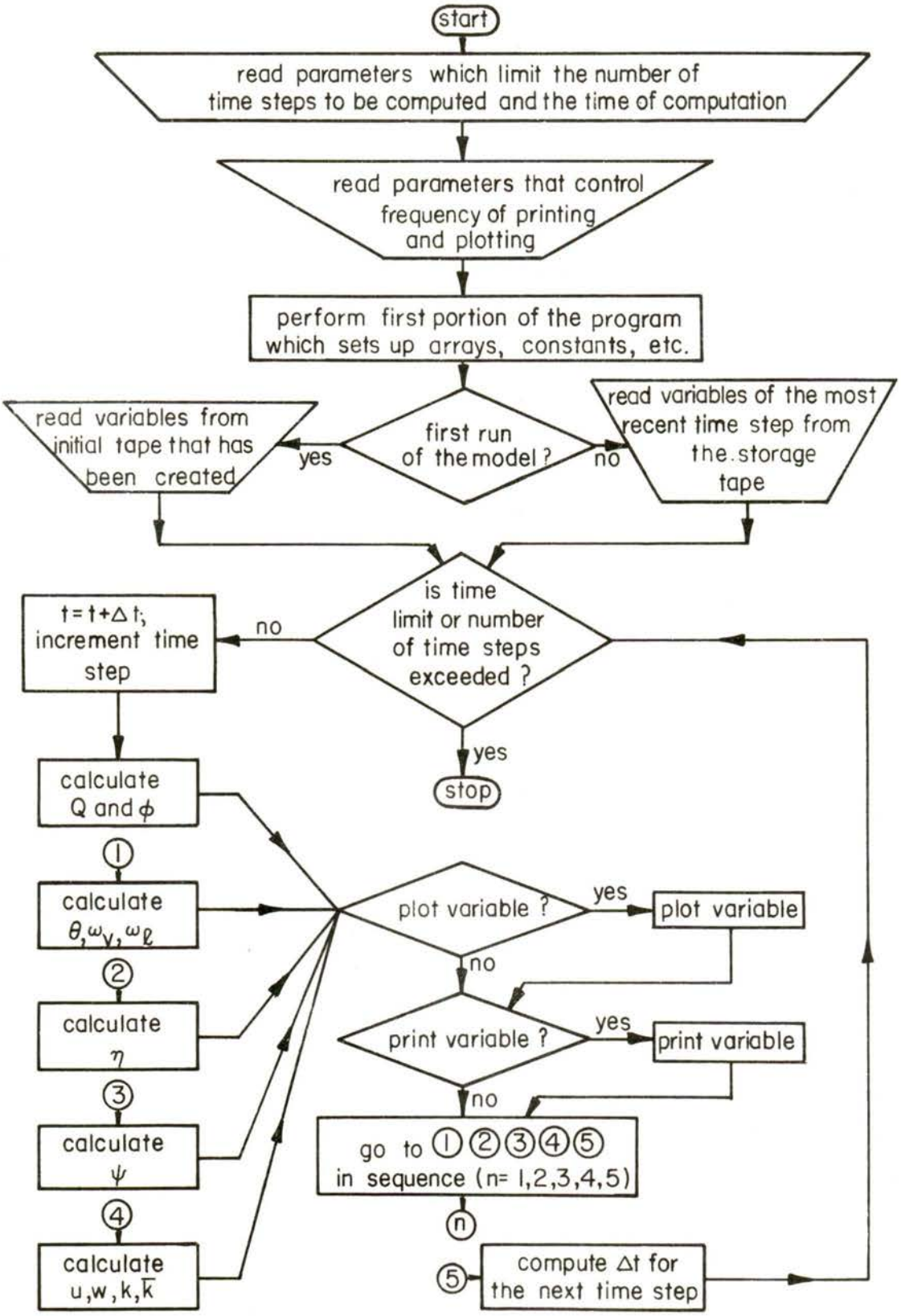


Figure A2.

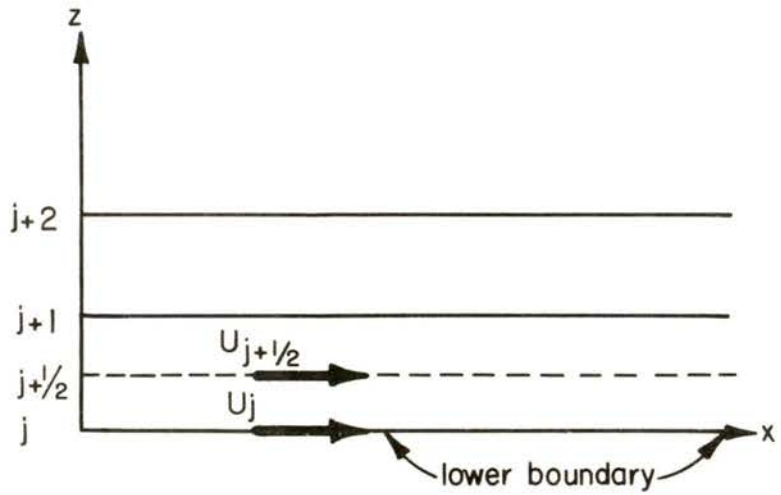


Figure A3.

APPENDIX B

A SUMMARY OF THE BOUNDARY CONDITIONS WITH ADDITIONAL EXPLANATIONS

APPENDIX B

A SUMMARY OF THE BOUNDARY CONDITIONS WITH ADDITIONAL EXPLANATIONS

This appendix presents a discussion concerning the application of boundary conditions in the model including a summary of the conditions used at each boundary. There are different ways an analytical boundary condition may be transformed into a numerical counterpart. Two methods are employed in the model: the "virtual" point method and the extrapolation method. For explanatory purposes we need consider only one boundary; the top boundary is chosen for convenience. At the top boundary $1 \leq i \leq I$ and $j = J$ where i and j denote subscripting in the x and z (horizontal and vertical) directions, respectively.

Consider any variable ϕ and its associated transport equation

$$\partial\phi/\partial t = -u\ \partial\phi/\partial x - w\ \partial\phi/\partial z + \overline{K}\ \partial^2\phi/\partial x^2 + \overline{K}\ \partial^2\phi/\partial z^2 \qquad (B1)$$

If we place the boundary condition $\partial\phi/\partial z = 0$ at the top boundary, the "virtual" point method transforms the finite difference analog of $\partial^2\phi/\partial z^2$ (see section 2.6 which describes the "virtual" point concept) so that we may re-represent equation (B1) as

$$\partial\phi/\partial t = -\ u\ \partial\phi/\partial x + \overline{K}\ \partial^2\phi/\partial x^2 + \overline{K}(\partial^2\phi/\partial z^2)_{\text{transformed}} \qquad (B2)$$

which is solved at the boundary along with equation (B1) for the interior of the model. The extrapolation method, on the other hand, does not involve the transport equation at all. A finite difference representation of $\partial\phi/\partial z$ at the top boundary may be given by

$$\partial\phi/\partial z = \frac{\phi_J - \phi_{J-1}}{h} \qquad \text{(B3)}$$

where h is the grid spacing. Setting this expression equal to zero we obtain

$$\phi_J = \phi_{J-1} \cdot \qquad \text{(B4)}$$

Equation (B4) is applied at the end of a time step to obtain the boundary values from interior values. Now if we consider the boundary condition $\partial^2\phi/\partial z^2 = 0$ at the top boundary, the "virtual" point method yields

$$\partial\phi/\partial t = -u\ \partial\phi/\partial x - w(\partial\phi/\partial z)_{\text{transformed}} + \overline{K}\ \partial^2\phi/\partial x^2 \qquad \text{(B5)}$$

and the extrapolation method yields

$$\phi_J = 2\phi_{J-1} - \phi_{J-2} \cdot \qquad \text{(B6)}$$

In summary, the "virtual" point method involves solving a transformed transport equation at the boundary itself and the extrapolation method involves interior values at the end of a time step.

The boundary condition summary of Figure B1 shows the analytical form of the boundary conditions for the variables of the model and the corresponding numerical form of these conditions.

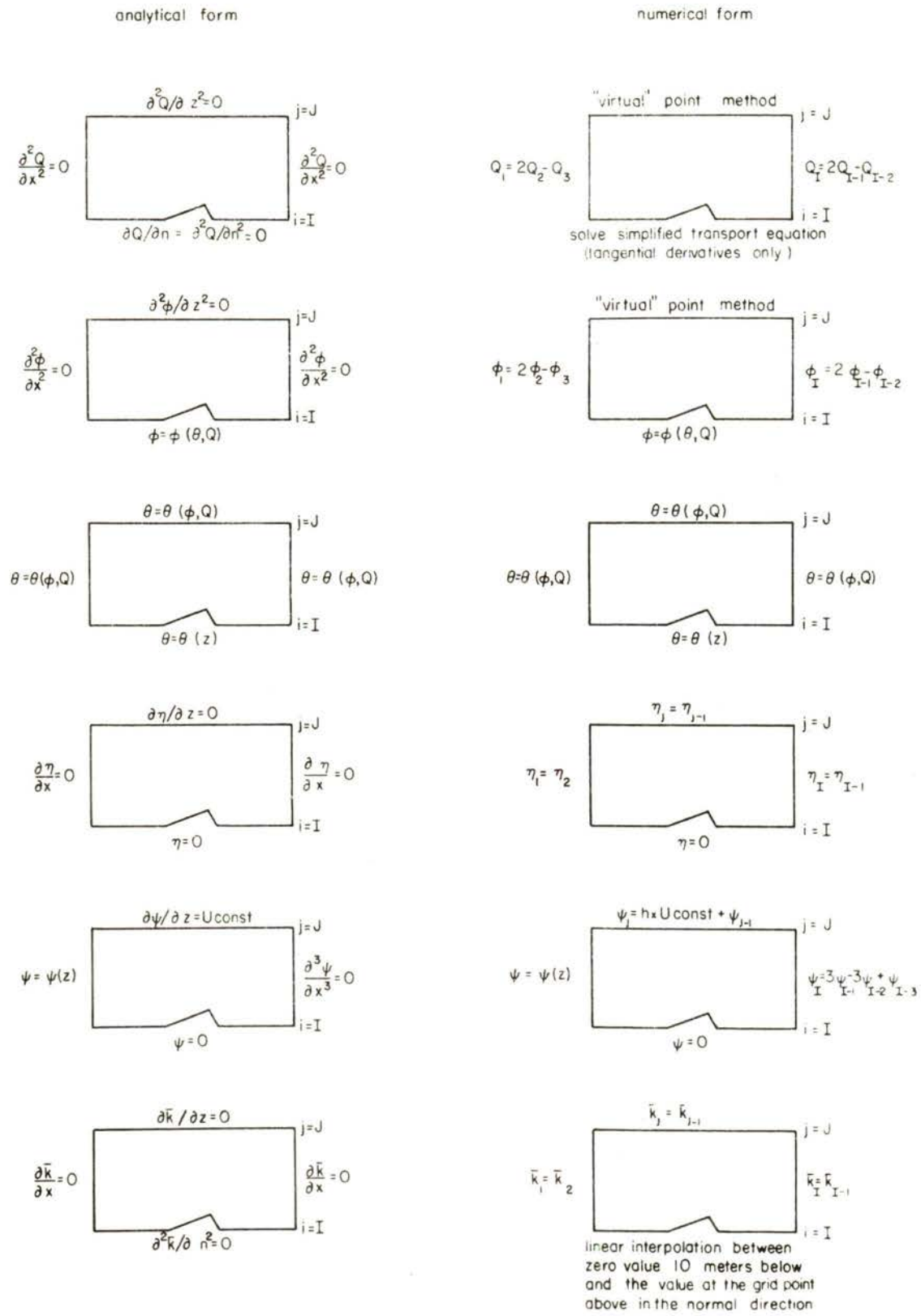


Figure B1. Schematic summary of the boundary conditions in the model.

APPENDIX C

UNIQUENESS PROOF FOR POISSON'S EQUATION OF THE STREAM FUNCTION WITH THE IMPOSED BOUNDARY CONDITIONS OF THE CLOUD MODEL

APPENDIX C

UNIQUENESS PROOF FOR POISSON'S EQUATION OF THE STREAM FUNCTION WITH THE IMPOSED BOUNDARY CONDITIONS OF THE CLOUD MODEL

This appendix presents a proof of uniqueness of the equation

$$\nabla^2\psi = -\eta \tag{C1}$$

for the imposed boundary conditions of the cloud model. The classical method of energy conservation (Garabedian (1964)) is employed.

Assuming two solutions, ψ_1 , and ψ_2 , which both satisfy equation (C1), we may define

$$\phi = \psi_2 - \psi_1$$

such that

$$\nabla^2\psi_2 - \nabla^2\psi_1 = \nabla^2\phi = 0.$$

Then we may also state that

$$\phi\nabla^2\phi = 0$$

which may be reexpressed as

$$\phi\nabla^2\phi = \nabla \cdot \phi\nabla\phi - \nabla\phi\cdot\nabla\phi = \nabla\cdot\phi\nabla\phi - |\nabla\phi|^2 = 0 \tag{C2}$$

by a well known vector identity. Equation (C2) is integrated over the entire area, A , of the model to give

$$\int_A\int \phi\nabla^2\phi \; dA = \int_A\int \nabla\cdot\phi\nabla\phi \; dA - \int_A\int |\nabla\phi|^2 \; dA. \tag{C3}$$

Gauss's divergence theorem is then used to transform the first integral on the right hand side of equation (C3) to yield

$$\int\int_A \nabla\cdot\phi\nabla\phi \, dA = \oint_s \phi\nabla\phi\cdot\vec{n} \, ds$$

where $\vec{n}$ is the outward normal unit vector of the closed curve encompassing the cloud model and s is the arc length of that curve. Equation (C3) may be rewritten as

$$\int\int_A \phi\nabla^2\phi \, dA = \oint_s \phi\nabla\phi\cdot\vec{n} ds - \int\int_A |\nabla\phi|^2 \, dA = 0 . \qquad (C4)$$

Equation (C4) is used to construct the proof of uniqueness.

Consider the four boundaries of the model as shown in Figure C1:

(a) left boundary $\psi_2 = \psi_1 = \psi(z)$; therefore $\phi = \psi_2 - \psi_1 = 0$

(b) right boundary $\dfrac{\partial^3\psi_2}{\partial x^3} = \dfrac{\partial^3\psi_1}{\partial x^3} = 0$; therefore $\dfrac{\partial^3\phi}{\partial x^3} = 0$

(c) top boundary $\dfrac{\partial\psi_2}{\partial z} = \dfrac{\partial\psi_1}{\partial z} = U_{const}$; therefore $\dfrac{\partial\phi}{\partial z} = 0$

(d) lower boundary $\psi_2 = \psi_1 = 0$; therefore $\phi = 0$.

Using Figure C1 as a guide we have at the top boundary

$$\left(\frac{\partial\phi}{\partial z}\right)_{x,H} = 0 \text{ for } 0 \leq x \leq L$$

from which it follows that

$$\left(\frac{\partial}{\partial x}\frac{\partial\phi}{\partial z}\right)_{x,H} = \left(\frac{\partial^2}{\partial x^2}\frac{\partial\phi}{\partial z}\right)_{x,H} = 0 \text{ , etc. for } 0 \leq x \leq L \qquad (C5)$$

i.e. all derivatives in the x-direction of $\dfrac{\partial\phi}{\partial z}$ vanish at the top.

Since $\nabla^2\phi = 0$ we may differentiate with respect to x to obtain

$$\frac{\partial}{\partial x}\nabla^2\phi = \frac{\partial^3\phi}{\partial x^3} + \frac{\partial}{\partial x}\frac{\partial^2\phi}{\partial z^2} = 0 \qquad \text{(C6)}$$

which holds everywhere in the closed region and on the boundaries of the model. But the right boundary condition of the model is

$$\left(\frac{\partial^3\phi}{\partial x^3}\right)_{L,z} = 0$$

such that equation (C6), as applied to the right boundary, becomes

$$\left(\frac{\partial}{\partial x}\frac{\partial^2\phi}{\partial z^2}\right)_{L,z} = \frac{\partial}{\partial z}\left(\frac{\partial^2\phi}{\partial x \partial z}\right)_{L,z} = 0 \ .$$

Therefore

$$\left(\frac{\partial^2\phi}{\partial x \partial z}\right)_{L,z} = \text{constant, } C_o \qquad \text{(C7)}$$

but equation (C5) can be applied to the upper right hand corner of the model to yield

$$\left(\frac{\partial^2\phi}{\partial x \partial z}\right)_{L,H} = 0$$

which shows that the constant C_o must be zero. Then equation (C7) is written as

$$\left(\frac{\partial^2\phi}{\partial x \partial z}\right)_{L,z} = \frac{\partial}{\partial z}\left(\frac{\partial\phi}{\partial x}\right)_{L,z} = 0$$

such that

$$\left(\frac{\partial\phi}{\partial x}\right)_{L,z} = \text{constant, } C_1 \ .$$

However, $\phi = \frac{\partial\phi}{\partial x} = \frac{\partial^2\phi}{\partial x^2} = 0$ at $x = L$ and $z = 0$ by the same reasoning used to obtain equation (C5) such that the constant C_1 also must vanish. We then have

$$\left(\frac{\partial \phi}{\partial x}\right)_{L,z} = 0$$

instead of $\left(\frac{\partial^3 \phi}{\partial x^3}\right)_{L,z} = 0$ at the right boundary of the model, which is a key result of the uniqueness proof. This result is based on corner arguments. Figure C2 summarizes the final boundary conditions on ϕ, which differ from the initial boundary conditions only at the right boundary.

The line integral of equation (C4) has four contributions, one for each boundary of the model. That is,

$$\int_S \phi \nabla \phi \cdot \vec{n} ds = \underbrace{\int_0^H \phi\left(-\frac{\partial \phi}{\partial x}\right) dz}_{\text{Left}} + \underbrace{\int_0^L \phi \frac{\partial \phi}{\partial z} dx}_{\text{Top}} + \underbrace{\int_0^H \phi \frac{\partial \phi}{\partial x} dz}_{\text{Right}} + \underbrace{\int_0^L \phi\left(-\frac{\partial \phi}{\partial z}\right) dx}_{\text{Bottom}} .$$

But each contribution is zero because either $\phi = 0$ or $\frac{\partial \phi}{\partial n} = 0$ at each boundary. Then equation (C4) reduces to

$$\int_A \int \phi \nabla^2 \phi \, dA = - \int_A \int |\nabla \phi|^2 \, dA = 0 . \qquad \text{(C8)}$$

This equation requires that

$$\left(\frac{\partial \phi}{\partial x}\right)^2 = \left(\frac{\partial^2 \phi}{\partial z}\right)^2 = 0$$

because all the terms in the equation are non-negative. Therefore

$$\frac{\partial \phi}{\partial x} = \frac{\partial \phi}{\partial z} = 0 \quad \text{or} \quad \phi = \text{constant}, \; C_3 ,$$

but the constant C_3 must vanish because ϕ is zero at the left and lower boundaries.

We may now state that the solution of $\nabla^2 \psi = - \eta$ is unique, with no additive constant, for the given boundary conditions since $\phi = 0$ such that $\psi_2 = \psi_1$.

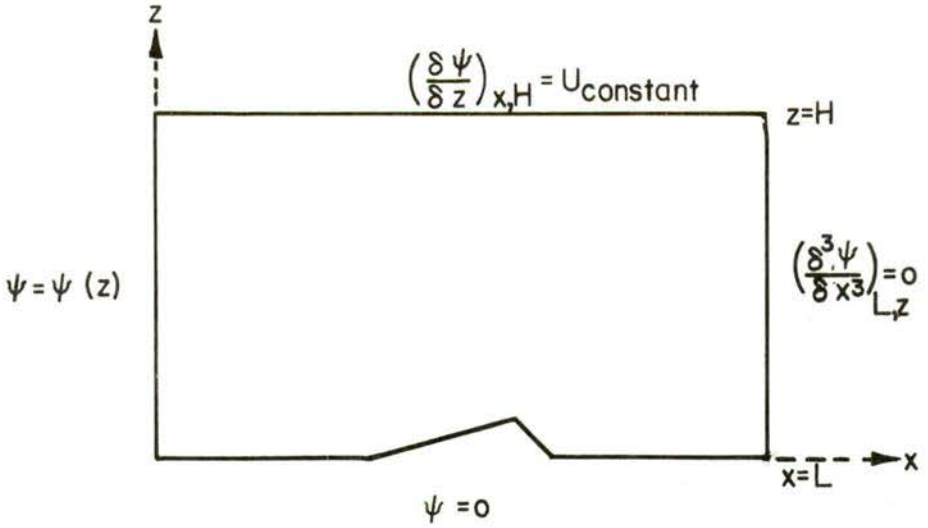


Figure C1.

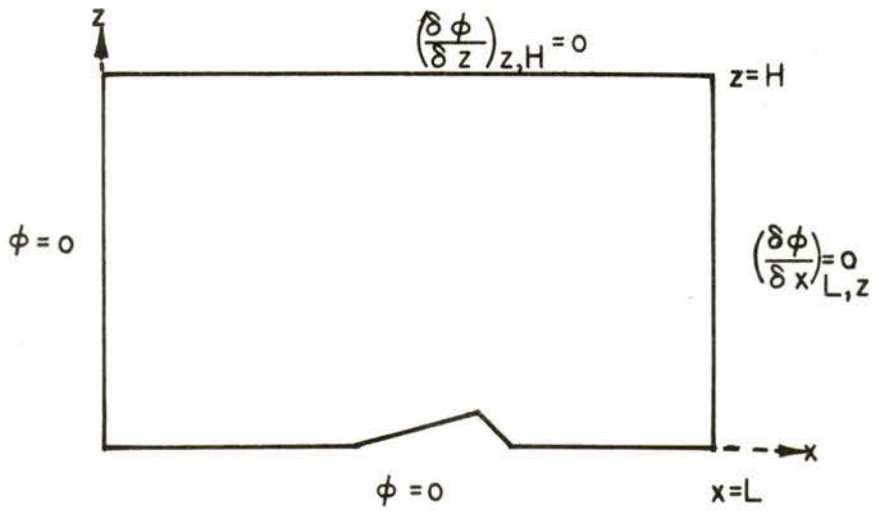


Figure C2.

APPENDIX D

LISTINGS OF THE COMPUTER PROGRAMS FOR THE INITIALIZATION PACKAGE, THE CLOUD MODEL PACKAGE, AND THE PLOTTING PACKAGE

INITIALIZATION PACKAGE

```
*JOB,5228,V0424010,DERICKSON
*LIMIT,T=30S,PR=60
*ASSIGN,AR2=7
*FORTRAN
      PROGRAM BETSY
      DIMENSION TK(36),TPP(36),TP(36),W(36),TMP(36),PIBK(36),PHIPR(36),
     1PRES(36),DELC(36),ESLTK(36),ESITK(36),TR(36)
      DIMENSION WSITERM(36),DIFU(36),GPRTERM(36),PIB(36),F22(36),F1CONT0
     1(36),RHOA(36),R124(36),FLLI(36),F1(36),F2(36),F3(36)
      DIMENSION OS(36),WVS(36),WSI(36),RHOV(36),SPR(36),AVAL(36)
      DIMENSION AVOR(36)
      DIMENSION D(111,36),E(111,36)
      COMMON S(111,36)
      COMMON/BLOCK4/DELT,TIME,NSTEPS,ITER,IVAR,JFORM
      COMMON/BLOCK15/MLT
C INITIALIZE TIME
      DELT=9.76
      TIME=0.0
      NSTEPS=0
      ITER=0
C
      LML=31
      MLT=25
      LR=110
      LR1=111
      MLB=36
      EPS=0.005
      V=10.
      JM=10
      DELXZ=100.
      NUMITER=600
      RELF=0.485
C
      DELXZ2=2.*DELXZ
      DELXZSQ=DELXZ**2
      C2=2./3.
      WBASE=0.0035
      DWDZ=-0.857143E-06
      TREF=271.35
      THETREF=TREF
      DKDZ=0.00522
      DTDZ=-0.00976
      D0=0.249
      P0=1000.
      T0=288.0
      FF1=1.7
      FF2=1.05
      CONA=1.E-05
      DELTA=0.
      CONB=0.7
      FLVL=604.2
      FLIV=677.5
      FLLI1=79.7
      SLPLLI=0.574
      FLLIBOT=78.7
      CP=0.240
      PREF=750.
      OLR=0.00685
      XPO=5.000*.00685/OLR
      ALR=0.00976
      EXK=1./0.286
      R=0.068557
      RV=0.110226
      F4=FLVL/(RV*THETREF**2)
      C1=1.-(2.*THETREF)
      C2=F4*2.
      C3=(F4**2)*2.
      C4=F4*THETREF*R*(CP*PREF)
      CON1=R*(RV*PREF)
      CON2=FLIV*(RV*THETREF**2)
      CON3=CON2**2*2.
      FN0=0.1
      RHO1=0.917
      CON4=FN0*RHO1*3.14159
      THERM=5.59E-05
      CON5=FLIV**2*(RV*THERM)
      CON7=61.E+05*(1.-CON4)
      CON8=FLIV/CP
      CON8A=CON8*THETREF
      CON9=FLLI1/CP
      CON10=CP*THETREF/FLVL
      CON11=CP*THETREF
      WRITE(6,4) F4,CON1,CON2,CON3,CON4,CON5
      WRITE(6,5) CON7,CON8,CON8A,CON9,CON10,CON11
4 FORMAT(1H0,6(1X,E13.7))
5 FORMAT(1H0,6(1X,E13.7)/1X,6(1X,E13.7))
      WRITE(6,7)
7 FORMAT(1H1)
      WRITE(6,5)
5 FORMAT(1H0,'   TR     DELC     W        TP       TPP       PIB       PIBK
     1    PHIPR     PRES,MB         F1         F2         F3      ESLTK      E
     2SITK      ZM'/1X)
      JJ=MLB
      DO 1 J=1,MLB
      JJ=JJ-1
      FLLI(J)=FLLIBOT-SLPLLI*JJ
      W(J)=WBASE+DWDZ*100.*JJ
      TR(J)=TREF-OLR*100.*JJ
      TK(J)=TREF+DTDZ*100.*JJ
      DELC(J)=TR(J)-273.16
      TPP(J)=DKDZ*100.*JJ
      PHIPR(J)=(TPP(J)+CONB*W(J))*THETREF
      TP(J)=TPP(J)+THETREF
      PIB(J)=1.-ALR*100.*JJ/(THETREF+TPP(J))
      PIBK(J)=PIB(J)**EXK
      ZM=2456.5+100.*JJ
      PRES(J)=1013.25*((288.16-OLR*ZM)/288.16)**XPO
      RHOA(J)=0.00034838*PRES(J)/TR(J)
      R124(J)=((CON4/RHOA(J))**0.5)*FF1*FF2
      DIFU(J)=(D0*(TR(J)/T0)**1.81)*(P0/PRES(J))
      GPRTERM(J)=DIFU(J)*CON5/(TR(J)**2)
      ARG1=17.27*(TK(J)-273.16)/(TK(J)-35.86)
      ESLTK(J)=6.11*EXP(ARG1)
      ARG2=21.87*(TK(J)-273.16)/(TK(J)-7.66)
      ESITK(J)=6.11*EXP(ARG2)
      WSITERM(J)=CON1*ESITK(J)/PIBK(J)
      F1(J)=C4*ESLTK(J)/PIBK(J)
      F2(J)=C1+C2*F1(J)
      F22(J)=F2(J)**2
      F3(J)=C3*F1(J)
      F1CONT0(J)=F1(J)*CON10
      OS(J)=(PHIPR(J)-FLLI(J)+W(J)*CON11-F1(J))*F3(J)+F22(J)
      IF(OS(J).LE.0.) OS(J)=0.0
      OS(J)=(SQRT(OS(J))-F2(J))/F3(J)
      SAT=F4*TPP(J)
      WVS(J)=F1CONT0(J)*(1.+SAT+0.5*SAT**2)
      SAT=CON2*TPP(J)
      WSI(J)=WSITERM(J)*(1.+SAT+0.5*SAT**2)
```

```
      RHOV(J)=HW(J)*RHOA(J)
      GPR(J)=(DIFU(J)*RHOV(J)-RHO(J)*(1.+GPRTERM(J)*RHO(J))
      AVAL(J)=4.*R124(J)*GPR(J)
      TMP(J)=PIB(J)*TP(J)
      WRITE(6,2) TR(J),DELC(J),HW(J),TP(J),TPP(J),PIB(J),PIBK(J),PHIPR(J
     1),PRES(J), F1(J), F2(J), F3(J),ESLTK(J),ESITK(J),ZM
    2 FORMAT(1H0,2(1X,F6.1),1X,F6.5,2(1X,F6.1),9(1X,F9.5),1X,F9.5)
    1 CONTINUE
      WRITE(6,11)
   11 FORMAT(/////)
      WRITE(6,101)
  101 FORMAT(1H0,*   WSITERM          DIFU          GPRTERM          F22
     1      FICON10          RHOA          R124          FLL(*,1))
      DO 14 J=1,MLB
      WRITE(6,12) WSITERM(J),DIFU(J),GPRTERM(J),F22(J),FICON10(J),RHOA(J
     1),R124(J),FLL(J)
   12 FORMAT(1H ,8(1X,E13.5))
   14 CONTINUE
      WRITE(6,11)
      WRITE(6,102)
  102 FORMAT(1H0,*       OS            WVS            WSI            RHOV
     1      GPR            AVAL            TEMP*,/)
      DO 15 J=1,MLB
      WRITE(6,66) OS(J),WVS(J),WSI(J),RHOV(J),GPR(J),AVAL(J),TMP(J)
   66 FORMAT(1H ,7(1X,E13.5))
   15 CONTINUE
C
C WRITE SINGLE SUBSCRIPTED VARIABLES ON TAPE
      BUFFER OUT(7,1) (WSITERM(1),F3(36))
    6 IF(UNIT,7) 6,9,7
    7 WRITE(6,8)
    8 FORMAT(1H0,*END OF TAPE ERROR*,/)
    9 CONTINUE
      ENDFILE 7
      IDIM=LR1
      JDIM=MLB
C
C
      DO 70 J=1,MLB
      DO 70 I=1,LR1
      D(I,J)=0.0
      E(I,J)=0.0
   70 S(I,J)=0.0
C
C
C  DOUBLE SUBSCRIPTED ARRAYS NOW WRITTEN ON TAPE
C
C
C
C  PHI
      CALL STD(S,PHIPR,IDIM,JDIM,LML)
      IVAR=1
      CALL WRTAPE(S,LR1,MLB)
C
C  Q,TOTAL MOISTURE
      CALL STD(S,HW,IDIM,JDIM,LML)
      IVAR=2
      CALL WRTAPE(S,LR1,MLB)
C
C  THETA PRIME
      CALL STD(S,TPP,IDIM,JDIM,LML)
      IVAR=10
      CALL WRTAPE(S,LR1,MLB)
C
C  W,VAPOR
      CALL STD(S,HW,IDIM,JDIM,LML)
      IVAR=9
      CALL WRTAPE(S,LR1,MLB)
C
C  WL,LIQUID
      DO 61 J=1,MLB
      DO 61 I=1,LR1
   61 S(I,J)=0.0
      IVAR=8
      CALL WRTAPE(S,LR1,MLB)
C
C  WI,ICE
      DO 65 J=1,MLB
      DO 65 I=1,LR1
   65 S(I,J)=0.0
      IVAR=7
      CALL WRTAPE(S,LR1,MLB)
C
      VOR=-V/(MLT+JN)
      DO 23 J=1,MLB
   23 AVOR(J)=VOR/100.
C  VORTICITY
      CALL STD(S,AVOR,IDIM,JDIM,LML)
      IVAR=12
      CALL WRTAPE(S,LR1,MLB)
C
      CALL MARGO(V,EPS,RELF,LML,MLT,JN,LR,NUMITER,VOR)
      JT=MLT+1
      JB=MLT+JN
      MLB=JB+1
      WRITE(6,24)
      WRITE(6,103)
  103 FORMAT(1H0,57X,*STREAM FUNCTION*,/)
   24 FORMAT(//////////)
      DO 25 J=1,MLB
      WRITE(6,36) (S(I,J),I=1,LR1)
   36 FORMAT(12(1X,F9.2))
      WRITE(6,20)
   20 FORMAT(/)
   25 CONTINUE
C
C  STREAM FUNCTION
      IVAR=13
      CALL WRTAPE(S,LR1,MLB)
C
      WRITE(6,24)
C
C U-COMP AND W-COMP
      J=1
      DO 16 I=2,LR
      E(I,J)=-(S(I+1,J)-S(I-1,J))/DELXZ2
   16 D(I,J)=(S(I,J)-S(I,J+1))/DELXZ
      DO 18 I=2,LR
      DO 18 J=2,MLT
      E(I,J)=-(S(I+1,J)-S(I-1,J))/DELXZ2
   18 D(I,J)=(S(I,J-1)-S(I,J+1))/DELXZ2
      LIMX=LML+20
      DO 26 J=JT,JB
      LIMX=LIMX-2
      DO 26 I=2,LIMX
      E(I,J)=-(S(I+1,J)-S(I-1,J))/DELXZ2
   26 D(I,J)=(S(I,J-1)-S(I,J+1))/DELXZ2
      IL=LML+20
      DO 27 J=JT,JB
```

```
      IL=IL+1
      DO 27 I=IL,LR
      E(I,J)=-(S(I+1,J)-S(I-1,J))/DELX22
   27 D(I,J)=(S(I,J-1)-S(I,J+1))/DELX22
      II=LML-1
      J1=MLB
      I2=LML+30
      DO 28 K=1,10
      I1=I1+2
      I2=I2-1
      J1=J1-1
      D(I2,J1)=(S(I2+1,J1)+S(I2,J1-1))/DELX22
      E(I2,J1)=-D(I2,J1)
      D(I1+1,J1)=(S(I1+1,J1-1)+S(I1,J1-1))/(2.8*DELX2)
      E(I1+1,J1)=D(I1+1,J1)/2.
      E(I1,J1)=S(I1-1,J1)/DELX22
   28 D(I1,J1)=(S(I1,J1-1)+S(I1,J1))/DELX22
C CUSP
      D(LML+20,JT)=S(LML+20,MLT)/DELX2
      E(LML+20,JT)=0.0
      DO 37 I=1,LML
      E(I,MLB)=0.0
   37 D(I,MLB)=S(I,JB)/DELX2
      LMLP30=LML+30
      DO 38 I=LMLP30,LR1
      E(I,MLB)=0.0
   38 D(I,MLB)=S(I,JB)/DELX2
      DO 17 J=1,JB
      E(1,J)=E(2,J)
      E(LR1,J)=E(LR,J)
      D(1,J)=D(2,J)
   17 D(LR1,J)=D(LR,J)
      WRITE(6,104)
  104 FORMAT(1H0,62X,*U-COMP*/)
      DO 29 J=1,MLB
      WRITE(6,36) (D(I,J),I=1,LR1)
      WRITE(6,20)
   29 CONTINUE
C
C  U-COMP
      IVAR=15
      CALL WRTAPE(D,LR1,MLB)
C
      WRITE(6,24)
      WRITE(6,105)
  105 FORMAT(1H0,62X,*W-COMP*/)
      DO 55 J=1,MLB
      WRITE(6,36) (E(I,J),I=1,LR1)
      WRITE(6,20)
   55 CONTINUE
C
C  W-COMP
      IVAR=16
      CALL WRTAPE(E,LR1,MLB)
C EDDY VISCOSITY
      DO 200 J=1,MLB
      DO 200 I=1,LR1
  200 S(I,J)=0.0
      IVAR=17
      CALL WRTAPE(S,LR1,MLB)
C
      END
      SUBROUTINE WRTAPE(A,LR1,MLB)
      DIMENSION A(LR1,MLB)
      COMMON/BLOCK4/DELT,TIME,NSTEPS,ITER,IVAR,JFORM
      BUFFER OUT(7,1) (DELT,IVAR)
    6 IF(UNIT,7) 6,8,7
    7 WRITE(6,8)
    8 FORMAT(1H0,*END OF TAPE ERROR IN WRTAPE*/)
    5 BUFFER OUT(7,1) (A(1,1),A(LR1,MLB))
    9 IF(UNIT,7) 9,16,10
   10 WRITE(6,8)
   16 CONTINUE
      IF(IVAR.EQ.17) ENDFILE 7
      RETURN
      END
      SUBROUTINE STD(A,B,IDIM,JDIM,LML)
      DIMENSION A(IDIM,JDIM),B(1)
      COMMON/BLOCK15/MLT
      JLIM=JDIM-10
      JLIM1=JLIM+1
      DO 1 J=1,JLIM
      DO 1 I=1,IDIM
    1 A(I,J)=B(J)
      ILIM=LML+20
      DO 2 J=JLIM1,JDIM
      ILIM=ILIM-2
      DO 2 I=1,ILIM
    2 A(I,J)=B(J)
      ILIM=LML+20
      DO 3 J=JLIM1,JDIM
      ILIM=ILIM+1
      DO 3 I=ILIM,IDIM
    3 A(I,J)=B(J)
      RETURN
      END
      SUBROUTINE MARGO(V,EPS,RELF,LML,MLT,JN,LR,NUMITER,VOR)
      DIMENSION RES(600)
      COMMON S(111,36)
      COMMON/BLOCK4/DELT,TIME,NSTEPS,ITER,IVAR,JFORM
      JT=MLT+1
      JB=MLT+JN
      MLB=JB+1
      JT1=JT+1
      LR1=LR+1
      LML1=LML+1
      LML19=LML+19
      X1=V/((MLB-1)**2)
      DO 2 J=1,MLB
      X2=(MLB-J)**2
      DO 2 I=1,LR1
      S(I,J)=X1*X2*100.
    2 CONTINUE
      LL=LML+20
      S(LL,JT)=0.0
      IRL=LL
      DO 5 J=JT1,JB
      LL=LL-2
      IRL=IRL+1
      DO 5 I=LL,IRL
      S(I,J)=0.0
    5 CONTINUE
      C1=4./5.
      C2=2./5.
      VOR=VOR*100.
      FJB=JB
      RFS=(FJB-0.5)*V/FJB
      DO 20 I=1,NUMITER
```

```
      DIFF=-1.E+10
      EMAX=DIFF
      ITER=1
      DO 10 J=2,MLT
      DO 10 I=2,LR
      R=(S(I-1,J)+S(I+1,J)+S(I,J-1)+S(I,J+1)-4.*S(I,J)+VOR)*RELF
      EMAX=AMAX1(EMAX,R)
      S(I,J)=S(I,J)+R
      R=R/S(I,J)
      R=ABS(R)
      DIFF=AMAX1(DIFF,R)
   10 CONTINUE
      LIMX=LML+20
      DO 15 J=JT,JB
      LIMX=LIMX-2
      DO 11 I=2,LIMX
      R=(S(I-1,J)+S(I+1,J)+S(I,J-1)+S(I,J+1)-4.*S(I,J)+VOR)*RELF
      EMAX=AMAX1(EMAX,R)
      S(I,J)=S(I,J)+R
      R=R/S(I,J)
      R=ABS(R)
      DIFF=AMAX1(DIFF,R)
   11 CONTINUE
      I=LIMX+1
      R=(S(I-1,J)+4.*S(I,J-1)-5.-6.*S(I,J)+VOR)*(4.*RELF-6.)
      EMAX=AMAX1(EMAX,R)
      S(I,J)=S(I,J)+R
      R=R/S(I,J)
      R=ABS(R)
      DIFF=AMAX1(DIFF,R)
   15 CONTINUE
      IL=LML+20
      DO 12 J=JT,JB
      IL=IL+1
      DO 12 I=IL,LR
      R=(S(I-1,J)+S(I+1,J)+S(I,J-1)+S(I,J+1)-4.*S(I,J)+VOR)*RELF
      EMAX=AMAX1(EMAX,R)
      S(I,J)=S(I,J)+R
      R=R/S(I,J)
      R=ABS(R)
      DIFF=AMAX1(DIFF,R)
   12 CONTINUE
      RES(ITER)=DIFF
      DO 9 I=2,LR
    9 S(I,1)=UFS*100.*S(I,2)
      IF(DIFF-EPS)21,21,20
   20 CONTINUE
   21 CONTINUE
      DO 19 J=1,JB
   19 S(LR1,J)=3.*S(LR,J)-3.*S(LR-1,J)+S(LR-2,J)
      WRITE(6,14)
      WRITE(6,22) ITER,RELF
   22 FORMAT(1X,'NUMBER OF ITERATIONS=',I4,2X,'RELAXATION FACTOR=',F9.6)
      WRITE(6,14)
   14 FORMAT(////)
      WRITE(6,26)
   26 FORMAT('1H0,50X,'RESIDUALS IN PERCENTAGE CHANGE')
      WRITE(6,25) (RES(I),I=1,ITER)
   25 FORMAT(10(1X,F9.5))
      RETURN
      END
*S,
*E,L
```

CLOUD MODEL PACKAGE

```
*,IMIT,T=10,PR=295,P=12
*ASSIGN,A,AR=8
*I,ATRAY.
      PROGRAM MODEL
      DIMENSION FKE(400)
      DIMENSION TDF(21)
      COMMON BLOCK1 WSITERM(36),DIFL(36),SPRTERM(36),PIB(36),F22(36),F1
     *0,T0(36),RHOA(36),R124(36),FLL1(36),F1(36),F2(36),F3(36)
      COMMON BLOCK2 MLT,LR,LML,DELTZ,RELF,RHOI,THETREF,DP,D1,D2,D3,D4,EP
     *S,FLAM,CON22,DHK,WRDS,WMITER,PWIS,DELTZ
      COMMON BLOCK3 JB,MLB,LP1,DELTZ2,DELTSQ,D1D2,D2D3,D3D4,D1D4,D1D2
     *,F4,D
      COMMON BLOCK4 DELT,TIME,NSTEPS,ITER,IMAP,KPRM
      COMMON BLOCK5 A(111,36),B(111,36),C(111,36),D(111,36),E(111,36),F
     *(111,36),G(111,36),H(111,36)
      COMMON BLOCK7 NM(25),IDENT(5)
      COMMON BLOCK8 Z(111,36)
      COMMON BLOCK10 ITARRAY(2,111),IFORMAT(2,5)
      COMMON BLOCK11 IPLOT,IPRNT
      COMMON BLOCK12 TIMECHK(22)
      COMMON BLOCK14 TITLE,FKMEM,CIRCR,CIRCT,CIRCIC,CIRCB,CIRC
      COMMON BLOCK15 QLT,QRT,QTP,QINIT,QINT2,QRATIO,QCHFF,QLTM,QRTM,QTPM
      COMMON BLOCK18 JSKIP
C TIME LIMIT IN SECONDS AS SUPPLIED BY USER IS LESS THAN CP LIMIT TARG
      TLIMIT=710
      LIMSTEP=250
C
C
C NUMBER OF FILES TO SKIP TO GET TO NEWEST TIME STEP ON TAPE
      NFILES=13
C
C PARAMETERS TO DETERMINE HOW MANY TIME STEPS BEFORE PLOTTING,PRINTING,ETC
      IPLOTHK=10
      IPRNTHK=50
C
C NUMBER OF LINES TO SKIP IN PRINTING VARIABLES TO TOP OF MOUNTAIN.
C (ALL LINES ARE PRINTED STARTING FROM TOP OF MOUNTAIN DOWNWARD)
      JSKIP=9
C
C
C
C
C ARRAYS PERTINENT TO MASS STORAGE AND PLOTTING AND PRINTING
      NM(1)=3HPH
      NM(2)=3HMG
      NM(3)=6HDEFA
      NM(4)=6HDEFB
      NM(5)=6HDEFA2
      NM(6)=6HDEFB2
      NM(7)=2HU
      NM(8)=2HL
      NM(9)=2HV
      NM(10)=5HTHETA
      NM(11)=3HM
      NM(12)=3HETA
      NM(13)=3HPS
      NM(14)=4HTEMP
      NM(15)=6HLOGMP
      NM(16)=6HLOGMP
      NM(17)=4HEDDY
      NM(18)=4HPS2
      NM(19)=4HPS3
      NM(20)=4HPS4
      ITARRAY(1,1)=10H
      ITARRAY(1,2)=10H  G,TOTAL
      ITARRAY(2,2)=10HMOISTURE
      ITARRAY(1,3)=10HLOGARITHM
      ITARRAY(2,3)=10HPRODUCTION
      ITARRAY(1,7)=10H        W,
      ITARRAY(2,7)=10HICE
      ITARRAY(1,8)=10H      WL,L
      ITARRAY(2,8)=10HQUID
      ITARRAY(1,9)=10H       W,V
      ITARRAY(2,9)=10HAPOR
      ITARRAY(1,10)=10H          THE
      ITARRAY(2,10)=10HTA
      ITARRAY(1,11)=10H4,SAT VAPO
      ITARRAY(2,11)=10HR AND LIQ
      ITARRAY(1,12)=10H          VOR
      ITARRAY(2,12)=10HTICITY
      ITARRAY(1,13)=10H    STREAM
      ITARRAY(2,13)=10HFUNCTION
      ITARRAY(1,14)=10H          TE
      ITARRAY(2,14)=10HMP
      ITARRAY(1,15)=10H       W-CO
      ITARRAY(2,15)=10HMP
      ITARRAY(1,16)=10H       W-CO
      ITARRAY(2,16)=10HMP
      ITARRAY(1,17)=10H    EDDY VI
      ITARRAY(2,17)=10HSCOSITY
      IFORMAT(1,1)=10H(12(1X,F9.
      IFORMAT(2,1)=3H2)
      IFORMAT(1,2)=10H(12(4X,F6.
      IFORMAT(2,2)=3H2)
      IFORMAT(1,3)=10H(12(1X,F9.
      IFORMAT(2,3)=3H7)
C
C SPECIFIED CONSTANTS
      MLT=25
      LR=110
      LML=31
      DELTZ=100.
      RELF=0.485
      RHOI=0.917
      THETREF=273.3
      DP=0.240
      D1=4. 5.
      D2=2. 3.
      D3=4. 6.
      D4=1. 5.
      EPS=0.005
      FLAM=3.7
      CON22=0.9
      DHK=1.E-05
      WRDS=3696
      WMITER=300
      PWIS=150.
      DELTZ=685.7
C CONSTANTS NOT IN COMMON,BUT NEEDED FOR COMPUTED CONSTANTS
      P=0.110226
      FLL1=677.5
      FLL2=604.2

C COMPUTED CONSTANTS FROM SPECIFIED CONSTANTS
      BMLT=MLT
      MLBMLT+1
      LP1=LR+
```

```
      DELXZ2=2.*DELXZ
      DELXZSQ=DELXZ**2
      CON2=FLIV*(RV*THETREF**2)
      CON5=(CON2**2)/2.
      CON11=CP*THETREF
      CON8=FLIV*CP
      CON21=FLAM*DELXZ**3
      F4=FLVL*(RV*(THETREF**2))
      JT=MLT+1
C
C READING IN SINGLY SUBSCRIPTED VARIABLES
      BUFFER IN(8,1) (WSITERM(1),F3(36))
   26 IF(UNIT,8) 26,29,27
   27 WRITE(6,28)
   28 FORMAT(1H0,*EOF OR PARITY ERROR IN MODEL* )
   29 CONTINUE
      BUFFER IN(8,1) (VEOF,VEOF)
   80 IF(UNIT,8) 80,82,84,82
   82 WRITE(6,83)
      GO TO 31
   84 CONTINUE
C
      IF(NFILES.EQ.0) GO TO 89
      DO 88 K=1,NFILES
   88 CALL SKFL(8)
   89 CONTINUE
C
C
C READING PHI,Q,THETA,WV,WL,WI,VORTICITY,PSI,U-COMP,W-COMP,EDDY VISCOSITY
C
C PHI,Q,THETA,WV,WL,WI,VORTICITY
      CALL RDATA(A,LR1,MLB)
      CALL RDATA(E,LR1,MLB)
      CALL RDATA(C,LR1,MLB)
      CALL RDATA(D,LR1,MLB)
      CALL RDATA(F,LR1,MLB)
      CALL RDATA(G,LR1,MLB)
      CALL RDATA(B,LR1,MLB)
C
C
C INITIALIZE TOTAL MOISTURE SCHEME
      QINT1=0.0
      DO 53 J=1,MLB
      DO 53 I=1,LR1
   53 QINT1=QINT1+E(I,J)
      QINT1=QINT1*DELXZSQ
      WRITE(6,54) QINT1
   54 FORMAT(1H0,*TOTAL MOISTURE AT INITIATION=*,F9.1 )
C
C
C STORING VARIABLES JUST READ IN
C
      CALL BRANWT(NM(1),A(1,1),NWRDS)
      CALL BRANCK(NM(1))
      CALL BRANWT(NM(2),E(1,1),NWRDS)
      CALL BRANCK(NM(2))
      CALL BRANWT(NM(10),C(1,1),NWRDS)
      CALL BRANCK(NM(10))
      CALL BRANWT(NM(9),D(1,1),NWRDS)
      CALL BRANCK(NM(9))
      CALL BRANWT(NM(8),F(1,1),NWRDS)
      CALL BRANCK(NM(8))
      CALL BRANWT(NM(7),G(1,1),NWRDS)
      CALL BRANCK(NM(7))
      CALL BRANWT(NM(12),B(1,1),NWRDS)
      CALL BRANCK(NM(12))
C
C PSI,U-COMP,W-COMP,EDDY VISCOSITY
      CALL RDATA(H,LR1,MLB)
      CALL RDATA(C,LR1,MLB)
      CALL RDATA(D,LR1,MLB)
      CALL RDATA(G,LR1,MLB)
C STORING U-COMP,W-COMP,AND EDDY VISCOSITY
      CALL BRANWT(NM(15),C(1,1),NWRDS)
      CALL BRANCK(NM(15))
      CALL BRANWT(NM(16),D(1,1),NWRDS)
      CALL BRANCK(NM(16))
      CALL BRANWT(NM(17),G(1,1),NWRDS)
      CALL BRANCK(NM(17))
C CHECK FOR END OF FILE
      BUFFER IN(8,1) (VEOF,VEOF)
   60 IF(UNIT,8) 60,62,64,62
   62 WRITE(6,63)
   63 FORMAT(1H0,*PARITY ERROR IN MODEL* )
      GO TO 31
   64 CONTINUE
C
C STORING PSI AND INITIALIZING EXTRAP SCHEME FOR SOR
C
      CALL BRANWT(NM(13),H(1,1),NWRDS)
      CALL BRANCK(NM(13))
      CALL BRANWT(NM(18),H(1,1),NWRDS)
      CALL BRANCK(NM(18))
      CALL BRANWT(NM(19),H(1,1),NWRDS)
      CALL BRANCK(NM(19))
      CALL BRANWT(NM(20),H(1,1),NWRDS)
      CALL BRANCK(NM(20))
C
C INITIALIZE LAGINT ROUTINE FOR TIME INCREMENTS
      T(1)=DELT
      DO 22 K=2,4
   22 T(K)=T(K-1)-T(1)
C STARTING TIME STEP
      NUMSTP=0
C
C
C
C NOW WE ARE READY TO GO
C
   30 CONTINUE
      TCHK=TIMEF(ARG)/1000.
      IF(TCHK.GE.TLIMIT) GO TO 31
      DELA=DELT
      TIME=TIME+DELT
      NSTEPS=NSTEPS+1
      IF(NSTEPS.GT.LIMSTEP) GO TO 31
      IPLOT=MOD(NSTEPS,IPLCHK)
      IPRNT=MOD(NSTEPS,IPRCHK)
      CALL DIANE
      NUMSTP=NUMSTP+1
      FKE(NUMSTP)=FKINEN
      IF(ITER.GE.NUMITER) GO TO 31
      WRITE(6,40) NSTEPS
   40 FORMAT(1H1,//,1H ,*THIS IS FOR TIME STEP*,I4 )
      WRITE(6,46) DELA,TIME,ITER,FKINEN,CIRCR
   46 FORMAT(1HO,*DELT=*,F6.2,*   TIME=*,F8.2,*   ITER=*,I4,*   KINETIC
     *ENERGY=*,E13.6,*   CIRCULATION=*,F8.3)
      WRITE(6,47) CIRC,CIRCT,CIRC10,CIRCB
```

```
47 FORMAT(1H ,*BY AREA=*,F10.1,* TOP=*,F10.1,* SIDES=*,F10.1,* BOTTOM
  1=*,F10.1 )
   WRITE(6,48) QLT,QLTM,QRT,QRTM,QTP,QTPM,QDIFF,QRATIO
48 FORMAT(1H ,*Q,INFLOW=*,F9.5,* M:XING=*,F9.5,*   Q,OUTFLOW=*,F9.5,*
  1 M:XING=*,F9.5,*   Q,TOP=*,F9.5,* M:XING=*,F9.5 TH ,*Q,NET CHANGE
  2FOR AREA=*,F9.5,*  RATIO,AREA TO LINE INTEGRAL=*,F7.5)
   WRITE(6,49) QINT1,QINT2
49 FORMAT(1H , *Q,TOTAL AT PREVIOUS TIME STEP=*,F9.1,* Q,TOTAL AT THI
  1S TIME STEP=*,F9.1)
   QINT1=QINT2
   TOTIME=TIMECHK(22)-TIMECHK(1)
   WRITE(6,44) TOTIME
44 FORMAT(1H0,*TOTAL TIME FOR TIME STEP=*,F8.3 )
   WRITE(6,41) (TIMECHK(I),I=1,22)
41 FORMAT(10(1X,F8.3))
   DO 45 K=1,21
45 TDF(K)=TIMECHK(K+1)-TIMECHK(K)
   WRITE(6,41) (TDF(K),K=1,21)
   GO TO 30
31 CONTINUE
   WRITE(6,32)
32 FORMAT(1H0,*KINETIC ENERGY FOR ALL TIME STEPS* )
   WRITE(6,33) (FKE(K),K=1,NUMSTP)
33 FORMAT(10(1X,E12.5))
   WRITE(6,34) TLIMIT,LIMSTEP,NUMITER
34 FORMAT(1H0,*TIME LIMIT=*,F6.1,* TIME STEP LIMIT=*,I3,* ITERATION L
  1IMIT=*,I3)
   TCHK=TIMEF(ARG)/1000.
   WRITE(6,35) TCHK,NSTEPS,ITER
35 FORMAT(1H ,*TIME=*,F6.1,* TIME STEP=*,I3,* ITERATIONS=*,I3 )
   END
   SUBROUTINE RDATA(A,LR1,MLB)
   DIMENSION A(LR1,MLB)
   COMMON/BLOCK4/DELT,TIME,NSTEPS,ITER,IVAR,JFORM
   BUFFER IN(8,1) (DELT,IVAR)
 1 IF(UNIT,8) 1,4,2
 2 WRITE(6,3)
 3 FORMAT(1H0,*PARITY ERROR IN RDATA OR EOF*.)
 4 BUFFER IN(8,1) (A(1,1),A(LR1,MLB))
11 IF(UNIT,8) 11,14,12
12 WRITE(6,3)
14 CONTINUE
   RETURN
   END
C
C
   SUBROUTINE DIANE
   COMMON/BLOCK1/WSITERM(36),DIFU(36),GPRTERM(36),PIB(36),F22(36),F1C
  1ON10(36),RHOA(36),R124(36),FLLI(36),F1(36),F2(36),F3(36)
   COMMON/BLOCK2/MLT,LR,LML,DELXZ,RELF,RHO1,THETREF,CP,C1,C2,C3,C4,EP
  1S,FLAM,CON22,CHK,NWRDS,NUMITER,RVIS,UDELXZ
   COMMON/BLOCK3/JB,MLB,LR1,DELXZ2,DELXZSQ,CON2,CON3,CON11,CON8,CON21
  1,F4,JT
   COMMON/BLOCK4/DELT,TIME,NSTEPS,ITER,IVAR,JFORM
   COMMON/BLOCK5/A(111,36),B(111,36),C(111,36),D(111,36),E(111,36),F(
  1111,36),G(111,36),H(111,36)
   COMMON/BLOCK7/IBM(25),INDEX(51)
   COMMON/BLOCK8/Z(111,36)
   COMMON/BLOCK10/ITARRAY(2,17),IFORMAT(2,3)
   COMMON/BLOCK11/ IPLOT,IPRNT
   COMMON/BLOCK12/TIMECHK(22)
   COMMON/BLOCK14/T(10),FKINEN,CIRCR,CIRCT,CIRCTO,CIRCB,CIRC
   COMMON/BLOCK15/QLT,QRT,QTP,QINT1,QINT2,QRATIO,QDIFF,QLTM,QRTM,QTPM
   COMMON/BLOCK16/JSKIP
C   ***********************************************************************
C   ***********************************************************************
C   ***********************************************************************
C   ***********************************************************************
C   ***********************************************************************
C  THIS IS THE START OF A NEW TIME STEP
      TIMECHK(1)=TIMEF(ARG)/1000.
C
C READ IN OLD VALUE OF PHI AND Q
      CALL BRANRD(NM(1),A(1,1),NWRDS)
      CALL BRANCK(NM(1))
      CALL BRANRD(NM(2),E(1,1),NWRDS)
      CALL BRANCK(NM(2))
C  U IS IN C, W IN D, EDDY VISCOSITY IN G
C
C SET H=PHI TO INITIALIZE DIFFERENCE SCHEME
      CALL BRANRD(NM(1),H(1,1),NWRDS)
      CALL BRANCK(NM(1))
      A0=DELT/DELXZ
      A1=A0/DELXZ
C
C
      TIMECHK(2)=TIMEF(ARG)/1000.
C
C COMPUTATION OF Q AND PHI(Q IN E, PHI IN A)
C FIRST HALF OF TIME STEP FOR TOP PORTION
      DO 1 J=2,MLT
      DO 1 I=2,LR
      COEFA1=((-G(I+1,J)+G(I-1,J))/DELXZ2+C(I,J))*A0
      COEFB1=COEFA1*COEFA1/2.+G(I,J)*A1
      COEFA1=COEFA1/2.
      H(I,J)=A(I,J)-COEFA1*(A(I+1,J)-A(I-1,J))+COEFB1*(A(I+1,J)-2.*A(I,J
     1)+A(I-1,J))
      Z(I,J)=E(I,J)-COEFA1*(E(I+1,J)-E(I-1,J))+COEFB1*(E(I+1,J)-2.*E(I,J
     1)+E(I-1,J))
    1 CONTINUE
C
C  FIRST HALF OF TIME STEP FOR THE LEFT PORTION
      LIMX=LML+20
      DO 3 J=JT,JB
      LIMX=LIMX-2
      DO 3 I=2,LIMX
      COEFA1=((-G(I+1,J)+G(I-1,J))/DELXZ2+C(I,J))*A0
      COEFB1=COEFA1*COEFA1/2.+G(I,J)*A1
      COEFA1=COEFA1/2.
      H(I,J)=A(I,J)-COEFA1*(A(I+1,J)-A(I-1,J))+COEFB1*(A(I+1,J)-2.*A(I,J
     1)+A(I-1,J))
      Z(I,J)=E(I,J)-COEFA1*(E(I+1,J)-E(I-1,J))+COEFB1*(E(I+1,J)-2.*E(I,J
     1)+E(I-1,J))
    3 CONTINUE
C
C FIRST HALF OF TIME STEP FOR RIGHT PORTION
      IL=LML+20
      DO 5 J=JT,JB
      IL=IL+1
      DO 5 I=IL,LR
      COEFA1=((-G(I+1,J)+G(I-1,J))/DELXZ2+C(I,J))*A0
      IF(I.EQ.IL) COEFA1=((-G(I+1,J)+G(I,J)+G(I-1,J-1)-G(I,J-1))/DELXZ+C
     1(I,J))*A0
      COEFB1=COEFA1*COEFA1/2.+G(I,J)*A1
      COEFA1=COEFA1/2.
      H(I,J)=A(I,J)-COEFA1*(A(I+1,J)-A(I-1,J))+COEFB1*(A(I+1,J)-2.*A(I,J
     1)+A(I-1,J))
      Z(I,J)=E(I,J)-COEFA1*(E(I+1,J)-E(I-1,J))+COEFB1*(E(I+1,J)-2.*E(I,J
```

```
     1)+E(I-1,J))
    5 CONTINUE
C
C
      CALL FIRST(A0,A1)
C
C
C  ODD POINTS,END OF FIRST HALF OF TIME STEP
      CALL ODDPTS(H,LR1,MLB)
      CALL ODDPTS(Z,LR1,MLB)
C
      QINT2=0.0
C SECOND HALF FOR TOP
      FKSAVE=G(LML+20,JT)
      G(LML+20,JT)=(G(LML+21,JT)+G(LML+19,JT))/2.
      DO 2 J=2,MLT
      DO 2 I=2,LR
      COEFA2=((-G(I,J-1)+G(I,J+1))/DELXZ2+D(I,J))+A0
      COEFB2=COEFA2+COEFA2/2.+G(I,J)*A1
      COEFA2=COEFA2/2.
      A(I,J)=H(I,J)-COEFA2*(H(I,J-1)-H(I,J+1))+COEFB2*(H(I,J+1)-2.*H(I,J
     1)+H(I,J-1))
      E(I,J)=Z(I,J)-COEFA2*(Z(I,J-1)-Z(I,J+1))+COEFB2*(Z(I,J+1)-2.*Z(I,J
     1)+Z(I,J-1))
      QINT2=QINT2+E(I,J)
    2 CONTINUE
      G(LML+20,JT)=FKSAVE
C
C SECOND HALF FOR LEFT
      LIMX=LML+20
      DO 4 J=JT,JB
      LIMX=LIMX-2
      DO 4 I=2,LIMX
      COEFA2=((-G(I,J-1)+G(I,J+1))/DELXZ2+D(I,J))+A0
      IF(J.EQ.JB) COEFA2=((-G(I,J-1)+G(I,J))/DELXZ+D(I,J))+A0
      IF(I.EQ.LIMX) COEFA2=((-G(I,J-1)+G(I,J)+G(I-1,J)-G(I-1,J-1))/DELXZ
     1+D(I,J))+A0
      COEFB2=COEFA2+COEFA2/2.+G(I,J)*A1
      COEFA2=COEFA2/2.
      A(I,J)=H(I,J)-COEFA2*(H(I,J-1)-H(I,J+1))+COEFB2*(H(I,J+1)-2.*H(I,J
     1)+H(I,J-1))
      E(I,J)=Z(I,J)-COEFA2*(Z(I,J-1)-Z(I,J+1))+COEFB2*(Z(I,J+1)-2.*Z(I,J
     1)+Z(I,J-1))
      QINT2=QINT2+E(I,J)
    4 CONTINUE
C
C SECOND HALF FOR RIGHT
      IL=LML+20
      DO 6 J=JT,JB
      IL=IL+1
      DO 6 I=IL,LR
      COEFA2=((-G(I,J-1)+G(I,J+1))/DELXZ2+D(I,J))+A0
      IF(J.EQ.JB) COEFA2=((-G(I,J-1)+G(I,J))/DELXZ+D(I,J))+A0
      IF(I.EQ.IL) COEFA2=((-G(I,J-1)+G(I,J)+G(I+1,J-1)-G(I+1,J))/DELXZ+D
     1(I,J))+A0
      COEFB2=COEFA2+COEFA2/2.+G(I,J)*A1
      COEFA2=COEFA2/2.
      A(I,J)=H(I,J)-COEFA2*(H(I,J-1)-H(I,J+1))+COEFB2*(H(I,J+1)-2.*H(I,J
     1)+H(I,J-1))
      E(I,J)=Z(I,J)-COEFA2*(Z(I,J-1)-Z(I,J+1))+COEFB2*(Z(I,J+1)-2.*Z(I,J
     1)+Z(I,J-1))
      QINT2=QINT2+E(I,J)
    6 CONTINUE
      TIMECHK(5)=TIMEF(ARG)/1000.
C
      CALL SCND(A0,A1)
C
C
C INFLOW AND OUTFLOW BOUNDARIES
      CALL INOUT(E,LR1,MLB)
      CALL INOUT(A,LR1,MLB)
C
C B.C. FOR Q AND PHI AT ODD POINTS,END OF SECOND HALF OF TIME STEP
      CALL ODDPTS(E,LR1,MLB)
      CALL ODDPTS(A,LR1,MLB)
      TIMECHK(4)=TIMEF(ARG)/1000.
C
C
C
C  MOISTURE BALANCE THROUGH BOUNDARIES
      QLT=0.0
      QRT=0.0
      QLTM=0.0
      QRTM=0.0
C (INFLOW AND OUTFLOW)
      DO 11 J=2,JB
      QLT=QLT+C(1,J)*E(1,J)
      QRT=QRT-C(LR1,J)*E(LR1,J)
      QLTM=QLTM+G(1,J)*(E(2,J)-E(1,J))
      QRTM=QRTM+G(LR1,J)*(E(LR1,J)-E(LR,J))
      QINT2=QINT2+E(1,J)/2.
      QINT2=QINT2+E(LR1,J)/2.
   11 CONTINUE
      QLT=QLT+C(1,1)*E(1,1)/2.
      QLT=QLT+C(1,MLB)*E(1,MLB)/2.
      QLT=QLT*DELT*DELXZ
      QRT=QRT-C(LR1,1)*E(LR1,1)/2.
      QRT=QRT-C(LR1,MLB)*E(LR1,MLB)/2.
      QRT=QRT*DELT*DELXZ
      QLTM=QLTM*DELT
      QRTM=QRTM*DELT
      QTP=0.0
      QTPM=0.0
C (TOP)
      DO 12 I=2,LR
      QINT2=QINT2+E(I,1)/2.
      QTPM=QTPM+G(I,1)*(E(I,1)-E(I,2))
   12 QTP=QTP-D(I,1)*E(I,1)*A0
      QTP=QTP-D(1,1)*E(1,1)*(2.*A0)
      QTP=QTP-D(LR1,1)*E(LR1,1)*(2.*A0)
      QTP=QTP*DELT*DELXZ
      QTPM=QTPM*DELT
      QINT2=QINT2*DELXZSQ
      QDIFF=QINT2-QINT1
      QDIV=QLT+QRT+QTP+QLTM+QRTM+QTPM
      QABS=ABS(QDIV)
      IF(QABS.LT..001) QDIV=0.001
      QRATIO=QDIFF/QDIV
C
      TIMECHK(5)=TIMEF(ARG)/1000.
C
C
C THIS WILL BE REPLACED WITH THE PRODUCTION TERM(PR=W)
C  'A' HAS AND W=0 FOR NOW
      DO 10 J=1,MLB
      DO 10 I=1,LR1
      F(I,J)=E(I,J)
```

```
   10 H(I,J)=0.0
C
C
C NOW COMPUTE NEW THETA,WL,AND WV
C  WI IS ALREADY IN H (WI=PR)
C  COMPUTE WL IN G, WV IN C, THETA IN D
C  W IS IN F ALREADY,Q IN E, WI IN H, PHI IN A
C
      TIMECHK(6)=TIMEF(ARG)/1000.
C
C  READ IN OLD VALUES OF THETA
C (THIS IS TO GET LOWER BOUNDARY)
      CALL BRANRD(NM(10),D(1,1),N4RDS)
      CALL BRANCK(NM(10))
C
C
C TOP PORTION
      DO 207 J=1,MLT
      DO 207 I=1,LR1
      D(I,J)=THETREF+A(I,J)-CON8*F(I,J)
      OS=(A(I,J)-FLL1(J)*F(I,J)/CON11-F1(J))*F3(J)+F22(J)
      IF(OS) 15,15,16
   15 OS=0.0
   16 OS=(SQRT(OS)-F2(J))/F3(J)
      IF(OS-D(I,J)) 26,26,27
   26 G(I,J)=0.0
      C(I,J)=F(I,J)
      GO TO 207
   27 D(I,J)=OS
      SAT=F4*D(I,J)
      C(I,J)=F1CON10(J)*(1.+SAT+0.5*SAT**2)
      G(I,J)=F(I,J)-C(I,J)
      IF(G(I,J).LT.0.0) C(I,J)=F(I,J)
  207 CONTINUE
C LEFT PORTION
      LIMX=LML+21
      DO 208 J=JT,JB
      LIMX=LIMX-2
      DO 205 I=1,LIMX
      D(I,J)=THETREF+A(I,J)-CON8*F(I,J)
      OS=(A(I,J)-FLL1(J)*F(I,J)/CON11-F1(J))*F3(J)+F22(J)
      IF(OS) 19,19,20
   19 OS=0.0
   20 OS=(SQRT(OS)-F2(J))/F3(J)
      IF(OS-D(I,J)) 31,31,32
   31 G(I,J)=0.0
      C(I,J)=F(I,J)
      GO TO 205
   32 D(I,J)=OS
      SAT=F4*D(I,J)
      C(I,J)=F1CON10(J)*(1.+SAT+0.5*SAT**2)
      G(I,J)=F(I,J)-C(I,J)
      IF(G(I,J).LT.0.0) C(I,J)=F(I,J)
  205 CONTINUE
      I1=LIMX+1
      JT=J
      CALL THETA(I1,JT)
  208 CONTINUE
      DO 203 I=1,LML
  203 CALL THETA(I,MLB)
C RIGHT PORTION
      IL=LML+20
      DO 209 J=JT,JB
      IL=IL+1
      I1=IL-1
      IF(J.GT.JT) CALL THETA(I1,J)
      DO 209 I=IL,LR1
      D(I,J)=THETREF+A(I,J)-CON8*F(I,J)
      OS=(A(I,J)-FLL1(J)*F(I,J)/CON11-F1(J))*F3(J)+F22(J)
      IF(OS) 23,23,24
   23 OS=0.0
   24 OS=(SQRT(OS)-F2(J))/F3(J)
      IF(OS-D(I,J)) 36,36,37
   36 G(I,J)=0.0
      C(I,J)=F(I,J)
      GO TO 209
   37 D(I,J)=OS
      SAT=F4*D(I,J)
      C(I,J)=F1CON10(J)*(1.+SAT+0.5*SAT**2)
      G(I,J)=F(I,J)-C(I,J)
      IF(G(I,J).LT.0.0) C(I,J)=F(I,J)
  209 CONTINUE
      LMLP30=LML+30
      DO 204 I=LMLP30,LR1
  204 CALL THETA(I,MLB)
C
      TIMECHK(7)=TIMEF(ARG)/1000.
C
C
C PLOTTING AND PRINTING PHI AND Q
      IF(IPRNT)301,300,301
  300 IVAR=1
      JFORM=3
      CALL MODPRNT(A,LR1,MLB)
      IVAR=2
      CALL MODPRNT(E,LR1,MLB)
  301 IF(IPLOT) 303,302,303
  302 IVAR=1
      CALL PREPLOT(A,LR1,MLB)
      IVAR=2
      CALL PREPLOT(E,LR1,MLB)
  303 CONTINUE
      TIMECHK(8)=TIMEF(ARG)/1000.
      IF(IPLOT.NE.0) TIMECHK(9)=TIMECHK(8)
C
C RE-STORE PHI AND Q (IN A AND E RESPECTIVELY)
      CALL BRANWT(NM(1),A(1,1),N4RDS)
      CALL BRANCK(NM(1))
      CALL BRANWT(NM(2),E(1,1),N4RDS)
      CALL BRANCK(NM(2))
C
C
C  PLOTTING AND PRINTING THETA, WV, WL, WI
C (THETA IN D,WV IN C, WL IN G, WI IN H )
      IF(IPRNT) 501,500,501
  500 IVAR=10
      JFORM=2
      CALL MODPRNT(D,LR1,MLB)
      JFORM=3
      IVAR=9
      CALL MODPRNT(C,LR1,MLB)
      JSKPSV=JSKIP
      JSKIP=1
      IVAR=8
      CALL MODPRNT(G,LR1,MLB)
      JSKIP=JSKPSV
  501 IF(IPLOT)503,502,503
  502 IVAR=10
```

```
      TIMECHK(9)=TIMEF(ARG)/1000.
C
      CALL PREPLOT(D,LR1,MLB)
      IVAR=9
      CALL PREPLOT(C,LR1,MLB)
      IVAR=8
      CALL PREPLOT(G,LR1,MLB)
      IVAR=7
      CALL PREPLOT(H,LR1,MLB)
  505 CONTINUE
C
C
C
C  COMPUTATION OF GRAVITY TERM FOR THE VORTICITY (IN E)
      TIMECHK(10)=TIMEF(ARG)/1000.
C
      VFT=9.8*DELT/DELXZ2
      VFT1=VFT+2.
C  TOP PORTION
      DO 45 J=1,MLT
      DO 45 I=2,LR
      IF(G(I+1,J).LT.0.0) G(I+1,J)=0.0
      IF(G(I-1,J).LT.0.0) G(I-1,J)=0.0
      E(I,J)=VFT*((D(I+1,J)-D(I-1,J))/THETREF+.608*(C(I+1,J)-C(I-1,J))-G
     1(I+1,J)+G(I-1,J)-H(I+1,J)+H(I-1,J))
   45 CONTINUE
C  LEFT PORTION
      LIMX=LML+21
      DO 76 J=JT,MLB
      LIMX=LIMX-2
      DO 46 I=2,LIMX
      IF(G(I+1,J).LT.0.0) G(I+1,J)=0.0
      IF(G(I-1,J).LT.0.0) G(I-1,J)=0.0
      E(I,J)=VFT*((D(I+1,J)-D(I-1,J))/THETREF+.608*(C(I+1,J)-C(I-1,J))-G
     1(I+1,J)+G(I-1,J)-H(I+1,J)+H(I-1,J))
   46 CONTINUE
      E(LIMX+1,J)=VFT1*((D(LIMX+1,J)-D(LIMX,J))/THETREF+0.608*(C(LIMX+1,
     1J)-C(LIMX,J))-G(LIMX+1,J)+G(LIMX,J)-H(LIMX+1,J)+H(LIMX,J))
   76 CONTINUE
C  RIGHT PORTION
      IL=LML+20
      DO 47 J=JT,MLB
      E(IL,J)=VFT1*((D(IL+1,J)-D(IL,J))/THETREF+0.608*(C(IL+1,J)-C(IL,J)
     1)-G(IL+1,J)+G(IL,J)-H(IL+1,J)+H(IL,J))
      IL=IL+1
      DO 47 I=IL,LR
      IF(G(I+1,J).LT.0.0) G(I+1,J)=0.0
      IF(G(I-1,J).LT.0.0) G(I-1,J)=0.0
      E(I,J)=VFT*((D(I+1,J)-D(I-1,J))/THETREF+.608*(C(I+1,J)-C(I-1,J))-G
     1(I+1,J)+G(I-1,J)-H(I+1,J)+H(I-1,J))
   47 CONTINUE
C  (INFLOW AND OUTFLOW)
      DO 40 J=1,MLB
      E(1,J)=VFT1*((D(2,J)-D(1,J))/THETREF+.608*(C(2,J)-C(1,J))-G(2,J)+G
     1(1,J)-H(2,J)+H(1,J))
      E(LR1,J)=VFT1*((D(LR1,J)-D(LR,J))/THETREF+.608*(C(LR1,J)-C(LR,J))-
     1G(LR1,J)+G(LR,J)-H(LR1,J)+H(LR,J))
   40 CONTINUE
C
      IF(IPRNT) 901,900,901
  900 IVAR=5
      JSKIP=5
      CALL MODPRNT(E,LR1,MLB)
  901 CONTINUE
C
      TIMECHK(11)=TIMEF(ARG)/1000.
C
C  RE-STORE THETA
      CALL BRANWT(NM(10),D(1,1),N4RDS)
      CALL BRANCK(NM(10))
C
C
C
C READ IN PSI,EDDY VISCOSITY,U-COMP,AND W-COMP
      CALL BRANRD(NM(13),A(1,1),N4RDS)
      CALL BRANCK(NM(13))
      CALL BRANRD(NM(17),H(1,1),N4RDS)
      CALL BRANCK(NM(17))
      CALL BRANRD(NM(15),C(1,1),N4RDS)
      CALL BRANCK(NM(15))
      CALL BRANRD(NM(16),F(1,1),N4RDS)
      CALL BRANCK(NM(16))
C  ALSO READ IN VORTICITY FROM PREVIOUS TIME STEP
      CALL BRANRD(NM(12),B(1,1),N4RDS)
      CALL BRANCK(NM(12))
C SET Z=ETA TO INITIALIZE DIFFERENCE SCHEME
      CALL BRANRD(NM(12),Z(1,1),N4RDS)
      CALL BRANCK(NM(12))
C
      TIMECHK(12)=TIMEF(ARG)/1000.
C
C COMPUTE ETAX AND ETAZ(IN D AND G RESPECTIVELY)
C
C TOP
      FKSAVE=H(LML+20,JT)
      H(LML+20,JT)=(H(LML+21,JT)+H(LML+19,JT))/2.
      DO 97 J=2,MLT
      DO 97 I=2,LR
      D(I,J)=A1*(A(I+1,J)-2.*A(I,J)+A(I-1,J))+(H(I+1,J)-2.*H(I,J)+H(I-1,
     1J))/DELXZSQ
      G(I,J)=A1*(A(I,J-1)-2.*A(I,J)+A(I,J+1))+(H(I,J-1)-2.*H(I,J)+H(I,J+
     11))/DELXZSQ+2.*A1*(A(I+1,J-1)-A(I+1,J+1)-A(I-1,J-1)+A(I-1,J+1))*(H
     2(I+1,J-1)-H(I+1,J+1)-H(I-1,J-1)+H(I-1,J+1))/(16.*DELXZSQ)
   97 CONTINUE
C
C LEFT
      H(LML+20,JT)=FKSAVE
      FKSAVE=H(LML-1,MLB)
      H(LML-1,MLB)=2.*H(LML-1,JB)-H(LML-1,JB-1)
      LIMX=LML+19
      DO 99 J=JT,JB
      LIMX=LIMX-2
      DO 98 I=2,LIMX
      D(I,J)=A1*(A(I+1,J)-2.*A(I,J)+A(I-1,J))+(H(I+1,J)-2.*H(I,J)+H(I-1,
     1J))/DELXZSQ
      IF(J-JB) 192,190,190
  190 G(I,J)=A1*(A(I+1,J-1)-A(I-1,J-1))*(H(I+1,J-1)-H(I-1,J-1)+H(I-1,J-
     1H(I+1,J))/(4.*DELXZSQ)
      GO TO 98
  192 CONTINUE
      G(I,J)=A1*(A(I,J-1)-2.*A(I,J)+A(I,J+1))+(H(I,J-1)-2.*H(I,J)+H(I,J+
     11))/DELXZSQ+2.*A1*(A(I+1,J-1)-A(I+1,J+1)-A(I-1,J-1)+A(I-1,J+1))*(H
     2(I+1,J-1)-H(I+1,J+1)-H(I-1,J-1)+H(I-1,J+1))/(16.*DELXZSQ)
   98 CONTINUE
      I=LIMX+1
      D(I,J)=A1*(A(I+1,J)-2.*A(I,J)+A(I-1,J))+(H(I+1,J)-2.*H(I,J)+H(I-1,
     1J))/DELXZSQ
```

```
      G(I,J)=A1*(A(I,J-1)-2.*A(I,J))+(H(I,J-1)-H(I,J)+H(I-1,J)-H(I-1,J-1
     1))/DELXZSQ-2.*A1*(-2.*A(I,J)-A(I-1,J+1)-A(I+1,J-1)+A(I-1,J)+A(I+1,
     2J)+A(I,J-1))+(-H(I,J)-H(I-1,J+1)-H(I+1,J-1)+2.*H(I-1,J)+H(I+1,J)+H
     3(I,J-1)-H(I-1,J-1))/(4.*DELXZSQ)
   99 CONTINUE
      H(LML-1,MLB)=FKSAVE
C
C RIGHT
      IL=LML+21
      DO 100 J=JT,JB
      I=IL
      D(I,J)=A1*(A(I+1,J)-2.*A(I,J))+(H(I+1,J)-H(I,J)+H(I-1,J-1)-H(I,J-1
     1))/DELXZSQ
      G(I,J)=A1*(A(I,J-1)-2.*A(I,J))+(H(I,J-1)-H(I,J)+H(I+1,J+1)-H(I+1,J
     1))/DELXZSQ
      IL=IL+1
      DO 100 I=IL,LR
      D(I,J)=A1*(A(I+1,J)-2.*A(I,J)+A(I-1,J))+(H(I+1,J)-2.*H(I,J)+H(I-1,
     1J))/DELXZSQ
      IF(J-JB) 196,194,194
  194 G(I,J)=A1*(A(I+1,J-1)-A(I-1,J-1))+(H(I+1,J-1)-H(I-1,J-1)+H(I-1,J-
     1H(I+1,J))/(4.*DELXZSQ)
      GO TO 100
  196 CONTINUE
      G(I,J)=A1*(A(I,J-1)-2.*A(I,J)+A(I,J+1))+(H(I,J-1)-2.*H(I,J)+H(I,J+
     11))/DELXZSQ+2.*A1*(A(I+1,J-1)-A(I+1,J+1)-A(I-1,J-1)+A(I-1,J+1))+(H
     2(I+1,J-1)-H(I+1,J+1)-H(I-1,J-1)+H(I-1,J+1))/(16.*DELXZSQ)
  100 CONTINUE
C
C  NOW COMPUTE VORTICITY FOR THIS TIME STEP
C (ADD GRAVITY TERM IN 1ST HALF OF TIME STEP)
      CIRC=0.0
C
C  FIRST HALF OF TIME STEP FOR TOP PORTION
      DO 48 J=2,MLT
      DO 48 I=2,LR
      COEFA1=(2.*(-H(I+1,J)+H(I-1,J))/DELXZ2+C(I,J))*A0
      COEFB1=COEFA1+COEFA1/2.+H(I,J)*A1
      COEFA1=COEFA1/2.
      Z(I,J)=B(I,J)-COEFA1*(B(I+1,J)-B(I-1,J))+COEFB1*(B(I+1,J)-2.*B(I,J
     1)+B(I-1,J))+E(I,J)-D(I,J)
   48 CONTINUE
C
C  FIRST HALF FOR LEFT
      LIMX=LML+20
      DO 50 J=JT,JB
      LIMX=LIMX-2
      DO 50 I=2,LIMX
      COEFA1=(2.*(-H(I+1,J)+H(I-1,J))/DELXZ2+C(I,J))*A0
      COEFB1=COEFA1+COEFA1/2.+H(I,J)*A1
      COEFA1=COEFA1/2.
      Z(I,J)=B(I,J)-COEFA1*(B(I+1,J)-B(I-1,J))+COEFB1*(B(I+1,J)-2.*B(I,J
     1)+B(I-1,J))+E(I,J)-D(I,J)
   50 CONTINUE
C
C  FIRST HALF FOR RIGHT
      IL=LML+20
      DO 52 J=JT,JB
      IL=IL+1
      DO 52 I=IL,LR
      COEFA1=(2.*(-H(I+1,J)+H(I-1,J))/DELXZ2+C(I,J))*A0
      IF(I.EQ.IL)COEFA1=((-H(I+1,J)+H(I,J)+H(I-1,J-1)-H(I,J-1))/DELXZ+C(I
     1),J))*A0
      COEFB1=COEFA1+COEFA1/2.+H(I,J)*A1
      COEFA1=COEFA1/2.
      Z(I,J)=B(I,J)-COEFA1*(B(I+1,J)-B(I-1,J))+COEFB1*(B(I+1,J)-2.*B(I,J
     1)+B(I-1,J))+E(I,J)-D(I,J)
   52 CONTINUE
C
C (TOP LINE)
      DO 54 I=2,LR
      Z(I,1)=Z(I,2)
   54 CONTINUE
      CALL ETA1(A0,A1)
C
C  ODD POINTS,END OF FIRST HALF OF TIME STEP
      CALL ODDPTS(Z,LR1,MLB)
C
C SECOND HALF FOR TOP
      FKSAVE=H(LML+20,JT)
      H(LML+20,JT)=(H(LML+21,JT)+H(LML+19,JT))/2.
      DO 49 J=2,MLT
      DO 49 I=2,LR
      COEFA2=(2.*(-H(I,J-1)+H(I,J+1))/DELXZ2+F(I,J))*A0
      COEFB2=COEFA2+COEFA2/2.+H(I,J)*A1
      COEFA2=COEFA2/2.
      B(I,J)=Z(I,J)-COEFA2*(Z(I,J-1)-Z(I,J+1))+COEFB2*(Z(I,J+1)-2.*Z(I,J
     1)+Z(I,J-1))-G(I,J)
      CIRC=CIRC+DELXZSQ*B(I,J)
   49 CONTINUE
      H(LML+20,JT)=FKSAVE
C
C SECOND HALF FOR LEFT
      LIMX=LML+20
      DO 51 J=JT,JB
      LIMX=LIMX-2
      DO 51 I=2,LIMX
      COEFA2=(2.*(-H(I,J-1)+H(I,J+1))/DELXZ2+F(I,J))*A0
      IF(J.EQ.JB) COEFA2=((-H(I,J-1)+H(I,J))/DELXZ+F(I,J))*A0
      IF(I.EQ.LIMX) COEFA2=((-H(I,J-1)+H(I,J)+H(I-1,J)-H(I-1,J-1))/DELXZ
     1+F(I,J))*A0
      COEFB2=COEFA2+COEFA2/2.+H(I,J)*A1
      COEFA2=COEFA2/2.
      B(I,J)=Z(I,J)-COEFA2*(Z(I,J-1)-Z(I,J+1))+COEFB2*(Z(I,J+1)-2.*Z(I,J
     1)+Z(I,J-1))-G(I,J)
      CIRC=CIRC+DELXZSQ*B(I,J)
   51 CONTINUE
C
C SECOND HALF FOR RIGHT
      IL=LML+20
      DO 53 J=JT,JB
      IL=IL+1
      DO 53 I=IL,LR
      COEFA2=(2.*(-H(I,J-1)+H(I,J+1))/DELXZ2+F(I,J))*A0
      IF(J.EQ.JB) COEFA2=((-H(I,J-1)+H(I,J))/DELXZ+F(I,J))*A0
      IF(I.EQ.IL) COEFA2=((-H(I,J-1)+H(I,J)+H(I+1,J+1)-H(I+1,J))/DELXZ+F
     1(I,J))*A0
      COEFB2=COEFA2+COEFA2/2.+H(I,J)*A1
      COEFA2=COEFA2/2.
      B(I,J)=Z(I,J)-COEFA2*(Z(I,J-1)-Z(I,J+1))+COEFB2*(Z(I,J+1)-2.*Z(I,J
     1)+Z(I,J-1))-G(I,J)
      CIRC=CIRC+DELXZSQ*B(I,J)
   53 CONTINUE
C
C (TOP LINE)
      DO 56 I=2,LR
      B(I,1)=B(I,2)
      CIRC=CIRC+DELXZSQ*B(I,1)/2.
```

```
   56 CONTINUE
      CALL ETA2(A0,A1)
C  (INFLOW AND OUTFLOW)
      DO 57 J=1,MLB
      B(1,J)=B(2,J)
      CIRC=CIRC+DELXZSQ*B(1,J)/2.
      B(LR1,J)=B(LR,J)
      CIRC=CIRC+DELXZSQ*B(LR1,J)/2.
   57 CONTINUE
C
C  ODD POINTS,END OF SECOND HALF OF TIME STEP
      CALL ODDPTS(B,LR1,MLB)
C
      TIMECHK(13)=TIMEF(ARG)/1000.
C
C  READ IN OLD VALUES OF THE STREAM FUNCTION
      CALL BRANRD(NM(13),A(1,1),N4ROS)
      CALL BRANCK(NM(13))
C
C
C
C
      DO 58 K=1,3
      L=K
      IF(K.EQ.3) L=7
      NAME1=NM(20-L)
      CALL BRANRD(NAME1,C(1,1),N4ROS)
      CALL BRANCK(NAME1)
      NAME2=NM(21-K)
      CALL BRAN4T(NAME2,C(1,1),N4ROS)
      CALL BRANCK(NAME2)
      T(5-K)=T(4-K)
   58 CONTINUE
      T(1)=T(1)+DELT
C
C
C FIRST GUESS FOR PSI IN SOR(2ND DEG EXTRAPOLATION IN TIME)
      CALL LAGINT
C
C FINITE DIFFERENCE CONSTANTS CAN BE DESTROYED (C,D,F,G)
C
C
C
C
C
C  NOW THE SOR ROUTINE TO SOLVE FOR THE STREAM FUNCTION
C
C
      TIMECHK(14)=TIMEF(ARG)/1000.
C
C
C  START OF ITERATION  * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
      DO 70 IJ=1,NUMITER
      DIFF=-1.E+10
      ITER=IJ
C  TOP PORTION
      DO 64 J=2,MLT
      DO 64 I=2,LR
      R=(A(I-1,J)+A(I+1,J)+A(I,J-1)+A(I,J+1)-4.*A(I,J)+DELXZSQ*B(I,J))*R
     1ELF
      A(I,J)=A(I,J)+R
      R=R/A(I,J)
      R=ABS(R)
      DIFF=AMAX1(DIFF,R)
   64 CONTINUE
C
C  LEFT PORTION
      LIMX=LML+20
      DO 66 J=JT,JB
      LIMX=LIMX-2
      DO 65 I=2,LIMX
      R=(A(I-1,J)+A(I+1,J)+A(I,J-1)+A(I,J+1)-4.*A(I,J)+DELXZSQ*B(I,J))*R
     1ELF
      A(I,J)=A(I,J)+R
      R=R/A(I,J)
      R=ABS(R)
      DIFF=AMAX1(DIFF,R)
   65 CONTINUE
      I=LIMX+1
      R=(A(I-1,J)+C1+A(I,J-1)-6.*A(I,J)+DELXZSQ*B(I,J))+C5*RELF
      A(I,J)=A(I,J)+R
      R=R/A(I,J)
      R=ABS(R)
      DIFF=AMAX1(DIFF,R)
   66 CONTINUE
C
C  RIGHT PORTION
      IL=LML+20
      DO 67 J=JT,JB
      IL=IL+1
      DO 67 I=IL,LR
      R=(A(I-1,J)+A(I+1,J)+A(I,J-1)+A(I,J+1)-4.*A(I,J)+DELXZSQ*B(I,J))*R
     1ELF
      A(I,J)=A(I,J)+R
      R=R/A(I,J)
      R=ABS(R)
      DIFF=AMAX1(DIFF,R)
   67 CONTINUE
C
C TOP BOUNDARY FOR STREAM FUNCTION (NEUMANN CONDITION)
      DO 63 I=2,LR
   63 A(I,1)=UDELXZ+A(I,2)
C
      IF(DIFF-EPS)71,71,70
   70 CONTINUE
C  END OF ITERATION * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
   71 CONTINUE
C DOWNSTREAM B.C. FOR PSI
      DO 60 J=1,MLB
   60 A(LR1,J)=3.*A(LR,J)-3.*A(LR-1,J)+A(LR-2,J)
      TIMECHK(15)=TIMEF(ARG)/1000.
C  PLOTTING AND PRINTING VORTICITY
      IVAR=12
      IF(IPRNT) 601,600,601
  600 JFORM=3
      CALL MODPRNT(B,LR1,MLB)
  601 CONTINUE
      TIMECHK(16)=TIMEF(ARG)/1000.
      IF(IPLOT)605,602,605
  602 CALL PREPLOT(B,LR1,MLB)
  605 CONTINUE
C
      TIMECHK(17)=TIMEF(ARG)/1000.
C  PLOTTING AND PRINTING STREAM FUNCTION
      IVAR=15
      IF(IPRNT) 701,700,701
  700 JFORM=1
      CALL MODPRNT(A,LR1,MLB)
```

```
  701 IF(IPLOT) 703,702,703
  702 CALL PREPLOT(A,LR1,MLB)
  703 CONTINUE
C
      TIMECHK(18)=TIMEF(ARG)/1000.
C  NOW COMPUTE U-COMP,W-COMP,AND EDDY VISCOSITY
C
C
      FKINEN=0.0
      UMAX=-1.E+10
      ANUMAX=UMAX
C  TOP PORTION
      DO 90 J=2,MLT
      DO 90 I=2,LR
      C(I,J)=(A(I,J-1)-A(I,J+1))/DELXZ2
      FKINEN=FKINEN+C(I,J)**2
      UABS=ABS(C(I,J))
      UMAX=AMAX1(UMAX,UABS)
      D(I,J)=(A(I-1,J)-A(I+1,J))/DELXZ2
      FKINEN=FKINEN+D(I,J)**2
      G(I,J)=((B(I+1,J)-B(I-1,J))/DELXZ2)**2+((B(I,J+1)-B(I,J-1))/DELXZ2
     1)**2
      G(I,J)=CON21*SQRT(G(I,J))
      IF(G(I,J).GT.RVIS) G(I,J)=RVIS
      ANUMAX=AMAX1(ANUMAX,G(I,J))
   90 CONTINUE
C  SET EDDY VISCOSITY AT TOP EQUAL TO FIRST INTERIOR GRID VALUE
      DO 85 I=2,LR
   85 G(I,1)=G(I,2)
C
C  LEFT PORTION
      LIMX=LML+20
      DO 91 J=JT,JB
      LIMX=LIMX-2
      DO 89 I=2,LIMX
      C(I,J)=(A(I,J-1)-A(I,J+1))/DELXZ2
      FKINEN=FKINEN+C(I,J)**2
      UABS=ABS(C(I,J))
      UMAX=AMAX1(UMAX,UABS)
      D(I,J)=(A(I-1,J)-A(I+1,J))/DELXZ2
      FKINEN=FKINEN+D(I,J)**2
      G(I,J)=((B(I+1,J)-B(I-1,J))/DELXZ2)**2+((B(I,J+1)-B(I,J-1))/DELXZ2
     1)**2
      G(I,J)=CON21*SQRT(G(I,J))
      IF(G(I,J).GT.RVIS) G(I,J)=RVIS
      ANUMAX=AMAX1(ANUMAX,G(I,J))
   89 CONTINUE
C  EDDY VISCOSITY ON SLOPES,LINEAR VARIATION IN THE NORMAL DIRECTION
C  ODD POINTS ARE GOTTEN BY SIMILAR SCHEME
      II=LIMX+2
      G(II,J)=10.*(G(II,J-1)+G(II-1,J-1))/(2.256*DELXZ+20.)
      G(II-1,J)=(G(II-2,J)+G(II,J-1))/4.
   91 CONTINUE
C
C  ODD POINTS AND SLOPES FOR VELOCITIES
      I=LML-1
      I2=LML+30
      J=MLB
      DO 92 K=1,9
      I=I+2
      I2=I2-1
      J=J-1
      C(I,J)=(A(I,J-1)+A(I,J))/DELXZ2
      FKINEN=FKINEN+C(I,J)**2
      D(I,J)=A(I-1,J)/DELXZ2
      FKINEN=FKINEN+D(I,J)**2
      C(I+1,J)=(A(I+1,J-1)+A(I,J-1))/(2.8*DELXZ)
      C(I2,J)=(A(I2+1,J)+A(I2,J-1))/DELXZ2
      D(I+1,J)=C(I+1,J)/2.
      D(I2,J)=-C(I2,J)
      CIRCB=CIRCB+A(I+1,J-1)+A(I,J-1)
      CIRCB=CIRCB+A(I2+1,J)+A(I2,J-1)
   92 CONTINUE
C  (TOP ODD POINT)
      C(I+2,J-1)=(A(I+2,J-2)+A(I+2,J-1))/DELXZ2
      D(I+2,J-1)=A(I+1,J-1)/DELXZ2
C  RIGHT PORTION
      IL=LML+20
      DO 96 J=JT,JB
      IL=IL+1
      DO 93 I=IL,LR
      C(I,J)=(A(I,J-1)-A(I,J+1))/DELXZ2
      FKINEN=FKINEN+C(I,J)**2
      UABS=ABS(C(I,J))
      UMAX=AMAX1(UMAX,UABS)
      D(I,J)=(A(I-1,J)-A(I+1,J))/DELXZ2
      FKINEN=FKINEN+D(I,J)**2
      G(I,J)=((B(I+1,J)-B(I-1,J))/DELXZ2)**2+((B(I,J+1)-B(I,J-1))/DELXZ2
     1)**2
      G(I,J)=CON21*SQRT(G(I,J))
      IF(G(I,J).GT.RVIS) G(I,J)=RVIS
      ANUMAX=AMAX1(ANUMAX,G(I,J))
   93 CONTINUE
      G(IL-1,J)=10.*(G(IL-1,J-1)+G(IL,J))/(1.414*DELXZ+20.)
   96 CONTINUE
C  (CUSP)
      C(LML+20,JT)=(C(LML+19,JT)+C(LML+21,JT))/2.
      D(LML+20,JT)=0.0
      G(LML+20,JT)=10.*G(LML+20,MLT)/(DELXZ+10.)
C
      TIMECHK(19)=TIMEF(ARG)/1000.
C  BOUNDARY VALUES FOR U-COMP,W-COMP,AND EDDY VISCOSITY
C     (TOP BOUNDARY FOR U AND W)
      CIRCT=0.0
      DO 94 I=2,LR
      C(I,1)=(A(I,1)-A(I,2))/DELXZ
      CIRCT=CIRCT-C(I,1)*DELXZ
      FKINEN=FKINEN+C(I,J)**2
      D(I,1)=(A(I-1,1)-A(I+1,1))/DELXZ2
      FKINEN=FKINEN+D(I,J)**2
   94 CONTINUE
C
C  LEFT LEVEL PORTION
      CIRCB=0.0
      DO 86 I=2,LML
      C(I,MLB)=A(I,JB)/DELXZ
      CIRCB=CIRCB+C(I,MLB)*DELXZ
      D(I,MLB)=0.0
   86 G(I,MLB)=10.*G(I,JB)/(DELXZ+10.)
C  RIGHT LEVEL PORTION
      LMLP30=LML+30
      DO 87 I=LMLP30,LR
      C(I,MLB)=A(I,JB)/DELXZ
      CIRCB=CIRCB+C(I,MLB)*DELXZ
      D(I,MLB)=0.0
   87 G(I,MLB)=10.*G(I,JB)/(DELXZ+10.)
C
C  INFLOW AND OUTFLOW
```

```
      CIRCIO=0.0
      DO 95 J=1,MLB
      C(1,J)=C(2,J)
      C(LR1,J)=C(LR,J)
      D(1,J)=D(2,J)
      CIRCIO=CIRCIO-D(1,J)*DELXZ
      D(LR1,J)=D(LR,J)
      CIRCIO=CIRCIO+D(LR1,J)*DELXZ
      G(1,J)=G(2,J)
      G(LR1,J)=G(LR,J)
   95 CONTINUE
C
      CIRCR=CIRC/(CIRCT+CIRCIO+CIRCB)
C
C
C SMOOTHING OPERATOR ON EDDY COEFFICIENT (5-POINT)
      DO 150 J=2,MLT
      DO 150 I=2,LR
      E(I,J)=0.8*G(I,J)+.05*(G(I+1,J)+G(I-1,J)+G(I,J+1)+G(I,J-1))
  150 CONTINUE
      LIMX=LML+20
      DO 152 J=JT,JB
      LIMX=LIMX-2
      DO 151 I=2,LIMX
      E(I,J)=0.8*G(I,J)+.05*(G(I+1,J)+G(I-1,J)+G(I,J+1)+G(I,J-1))
  151 CONTINUE
      E(LIMX+2,J)=G(LIMX+2,J)
      E(LIMX+1,J)=G(LIMX+1,J)
  152 CONTINUE
      IL=LML+20
      DO 153 J=JT,JB
      E(IL,J)=G(IL,J)
      IL=IL+1
      DO 153 I=IL,LR
      E(I,J)=0.8*G(I,J)+.05*(G(I+1,J)+G(I-1,J)+G(I,J+1)+G(I,J-1))
  153 CONTINUE
      DO 154 I=2,LR
      E(I,1)=E(I,2)
  154 E(I,MLB)=G(I,MLB)
      DO 155 J=1,MLB
      E(1,J)=E(2,J)
  155 E(LR1,J)=E(LR,J)
C
C STORING U-COMP,W-COMP,EDDY VISCOSITY
      CALL BRANWT(NM(15),C(1,1),NWRDS)
      CALL BRANCK(NM(15))
      CALL BRANWT(NM(16),D(1,1),NWRDS)
      CALL BRANCK(NM(16))
      CALL BRANWT(NM(17),E(1,1),NWRDS)
      CALL BRANCK(NM(17))
C
      TIMECHK(20)=TIMEF(ARG)/1000.
C
C  PLOTTING AND PRINTING U-COMP, W-COMP, AND EDDY COEF
      IF(IPRINT) 801,800,801
  800 IVAR=15
      JFORM=2
      CALL MODPRNT(C,LR1,MLB)
      IVAR=16
      CALL MODPRNT(D,LR1,MLB)
      IVAR=17
      JFORM=1
      CALL MODPRNT(E,LR1,MLB)
  801 CONTINUE
      TIMECHK(21)=TIMEF(ARG)/1000.
      IF(IPLOT)803,802,803
  802 IVAR=15
      CALL PREPLOT(C,LR1,MLB)
      IVAR=16
      CALL PREPLOT(D,LR1,MLB)
      IVAR=17
      CALL PREPLOT(E,LR1,MLB)
  803 CONTINUE
C COMPUTATION OF TIME INCREMENT FOR NEXT TIME STEP
      DELT=ANUMAX**2+CON22*(DELXZ**2)*(UMAX**2)
      DELT=(-ANUMAX+SQRT(DELT))/(UMAX**2)
C
C  STORE STREAM FUNCTION AND VORTICITY
      CALL BRANWT(NM(13),A(1,1),NWRDS)
      CALL BRANCK(NM(13))
      CALL BRANWT(NM(12),B(1,1),NWRDS)
      CALL BRANCK(NM(12))
C
C
C
      TIMECHK(22)=TIMEF(ARG)/1000.
C  END OF TIME STEP* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C ****************************************************************************
C ****************************************************************************
C ****************************************************************************
C ****************************************************************************
      RETURN
      END
      SUBROUTINE THETA(I1,J1)
      COMMON/BLOCK1/WSITERM(36),DIFU(36),GPRTERM(36),PIB(36),F22(36),F1C
     1ON10(36),RHOA(36),R124(36),FLL1(36),F1(36),F2(36),F3(36)
      COMMON/BLOCK2/MLT,LR,LML,DELXZ,RELF,RHO1,THETREF,CP,C1,C2,C3,C4,EP
     1S,FLAM,CON22,CHK,NWRDS,NUMITER,RVIS,UDELXZ
      COMMON/BLOCK3/JB,MLB,LR1,DELXZ2,DELXZSQ,CON2,CON3,CON11,CON8,CON21
     1,F4,JT
      COMMON/BLOCK5/A(111,36),B(111,36),C(111,36),D(111,36),E(111,36),F(
     1111,36),G(111,36),H(111,36)
C THETA IN D,WL IN G,WV IN C,W IN F,WI IN H,PHI IN A,Q IN E
      I=I1
      J=J1
C W=Q  AND  WI=0
      F(I,J)=E(I,J)
      H(I,J)=0.0
      SAT=F4*D(I,J)
      WVS=F1CON10(J)*(1.+SAT+0.5*SAT**2)
      IF(F(I,J)-WVS) 53,54,54
   53 C(I,J)=F(I,J)
      G(I,J)=0.0
      A(I,J)=(D(I,J)+CON8*C(I,J))/THETREF
      GO TO 55
   54 C(I,J)=WVS
      G(I,J)=F(I,J)-C(I,J)
      WL=G(I,J)
      IF(WL) 56,57,57
   56 WL=0.0
      C(I,J)=F(I,J)
   57 CONTINUE
      A(I,J)=(D(I,J)+CON8*C(I,J)+FLL1(J)*WL/CP)/THETREF
   55 CONTINUE
      RETURN
      END
      SUBROUTINE OOOPTS(A,LR1,MLB)
      DIMENSION A(LR1,MLB)
```

```
      COMMON BLOCK2 MLT,LR,LML,DELXZ,RELF,RHO1,THETREF,CP,C1,C2,C3,C4,EP
     1S,FLAM,CON22,CHK,N&RDS,NUMITER,RVIS,UDELXZ
      I=LML-1
      J=MLB
      DO 299 K=1,10
      I=I+2
      J=J-1
      A(I,J)=(A(I+1,J)+A(I-1,J)).2.
  299 CONTINUE
      RETURN
      END
      SUBROUTINE INOUT(A,LR1,MLB)
      DIMENSION A(LR1,MLB)
      LB=LR1-1
      DO 1 J=1,MLB
      A(1,J)=2.*A(2,J)-A(3,J)
      A(LR1,J)=2.*A(LB,J)-A(LB-1,J)
    1 CONTINUE
      RETURN
      END
      SUBROUTINE PREPLOT(A,LR1,MLB)
      DIMENSION A(LR1,MLB)
      COMMON/BLOCK4/DELT,TIME,NSTEPS,ITER,IVAR,JFORM
      BUFFER OUT(8,1) (DELT,IVAR)
    5 IF(UNIT,8) 5,8,6
    6 WRITE(6,7)
    7 FORMAT(1H0,*END OF TAPE ERROR IN PREPLOT*/)
    8 BUFFER OUT(8,1) (A(1,1),A(LR1,MLB))
    9 IF(UNIT,8) 9,12,10
   10 WRITE(6,7)
   12 CONTINUE
      IF(IVAR,EQ,17) ENDFILE 8
      RETURN
      END
      SUBROUTINE MODPRNT(A,LR1,MLB)
      DIMENSION A(LR1,MLB)
      DIMENSION IFORM(2)
      COMMON/BLOCK2/MLT,LR,LML,DELXZ,RELF,RHO1,THETREF,CP,C1,C2,C3,C4,EP
     1S,FLAM,CON22,CHK,N&RDS,NUMITER,RVIS,UDELXZ
      COMMON/BLOCK4/DELT,TIME,NSTEPS,ITER,IVAR,JFORM
      COMMON/BLOCK10/ITARRAY(2,17),IFORMAT(2,3)
      COMMON/BLOCK18/JSKIP
      IFORM(1)=IFORMAT(1,JFORM)
      IFORM(2)=IFORMAT(2,JFORM)
      JT=MLT+1
C
C FINDING MAXIMUM AND MINIMUM VALUES OF FUNCTION
      VMAX=-1.E+10
      VMIN=-VMAX
      DO 15 J=1,MLT
      DO 15 I=1,LR1
      VMAX=AMAX1(A(I,J),VMAX)
      VMIN=AMIN1(A(I,J),VMIN)
   15 CONTINUE
      LIMX=LML+22
      DO 16 J=JT,MLB
      LIMX=LIMX-2
      DO 16 I=1,LIMX
      VMAX=AMAX1(A(I,J),VMAX)
      VMIN=AMIN1(A(I,J),VMIN)
   16 CONTINUE
      IL=LML+19
      DO 17 J=JT,MLB
      IL=IL+1
      DO 17 I=IL,LR1
      VMAX=AMAX1(A(I,J),VMAX)
      VMIN=AMIN1(A(I,J),VMIN)
   17 CONTINUE
C
      WRITE(6,1)
    1 FORMAT(1H1)
      WRITE(6,2) ITARRAY(1,IVAR),ITARRAY(2,IVAR)
    2 FORMAT(1H ,50X,2A10)
      WRITE(6,3) NSTEPS,DELT,TIME,VMIN,VMAX
    3 FORMAT(1H ,*TIME STEP=*,I4,* DELT=*,F5.2,* TIME=*,F7.2,45*,*MIN=*,
     1E13.6,*      MAX=*,E13.6/)
      IF(IVAR-13) 8,7,8
    7 WRITE(6,4) ITER
    4 FORMAT(1H ,*NUMBER OF ITERATIONS=*,I4)
    8 CONTINUE
      DO 10 J=1,MLT,JSKIP
      JJ=MLB+1-J
      WRITE(6,9) JJ
    9 FORMAT(1H ,49X,*J=*,I2)
      WRITE(6,IFORM) (A(I,J),I=1,LR1)
      WRITE(6,20)
   10 CONTINUE
      DO 11 J=JT,MLB
      JJ=MLB+1-J
      WRITE(6,9) JJ
      WRITE(6,IFORM) (A(I,J),I=1,LR1)
      WRITE(6,20)
   11 CONTINUE
   20 FORMAT(/)
      RETURN
      END
      SUBROUTINE LAGINT
      COMMON/BLOCK2/MLT,LR,LML,DELXZ,RELF,RHO1,THETREF,CP,C1,C2,C3,C4,EP
     1S,FLAM,CON22,CHK,N&RDS,NUMITER,RVIS,UDELXZ
      COMMON/BLOCK3/JB,MLB,LR1,DELXZ2,DELXZSQ,CON2,CON3,CON11,CON8,CON21
     1,F4,JT
      COMMON/BLOCK5/A(111,36),B(111,36),C(111,36),D(111,36),E(111,36),F(
     1111,36),G(111,36),H(111,36)
      COMMON/BLOCK7/NM(25),INDEX(51)
      COMMON/BLOCK14/T(10),FKINEN,CIRCR,CIRCT,CIRIO,CIRCB,CIRC
      DO 1 J=2,MLT
      DO 1 I=2,LR
    1 A(I,J)=0.0
      LIMX=LML+21
      DO 2 J=JT,JB
      LIMX=LIMX-2
      DO 2 I=2,LIMX
    2 A(I,J)=0.0
      IL=LML+20
      DO 3 J=JT,JB
      IL=IL+1
      DO 3 I=IL,LR
    3 A(I,J)=0.0
      MS1=17
      DO 6 L=2,4
      MS1=MS1+1
      CALL BRANCH(NM(MS1),C(1,1),N&RDS)
      CALL BRANCH(NM(MS1))
      TOP=1
      BOT=1
      DO 5 K=2,4
      IF(K,EQ,L) GO TO 5
      TOP=TOP*(T(1)-T(K))
```

```
      BOT=BOT+(T(L)-T(K))
    5 CONTINUE
      DO 12 J=2,MLT
      DO 12 I=2,LR
      A(I,J)=A(I,J)+(TOP/BOT)*C(I,J)
   12 CONTINUE
      LIMx=LML+21
      DO 14 J=JT,JB
      LIMx=LIMx-2
      DO 14 I=2,LIMx
      A(I,J)=A(I,J)+(TOP/BOT)*C(I,J)
   14 CONTINUE
      IL=LML+20
      DO 16 J=JT,JB
      IL=IL+1
      DO 16 I=IL,LR
      A(I,J)=A(I,J)+(TOP/BOT)*C(I,J)
   16 CONTINUE
    6 CONTINUE
      RETURN
      END
      SUBROUTINE FRST(A0,A1)
      COMMON/BLOCK2/MLT,LR,LML,DELXZ,RELF,RHO1,THETREF,CP,C1,C2,C3,C4,EP
     1S,FLAM,CON22,CHK,N&RDS,NUMITER,RV1S,UDELXZ
      COMMON/BLOCK3/JB,MLB,LR1,DELXZ2,DELXZSQ,CON2,CON3,CON11,CON8,CON21
     1,F4,JT
      COMMON/BLOCK5/A(111,36),B(111,36),C(111,36),D(111,36),E(111,36),F(
     1111,36),G(111,36),H(111,36)
      COMMON/BLOCK8/Z(111,36)
C (TOP BOUNDARY)
      DO 101 I=2,LR
      C(I,1)=((-G(I+1,1)+G(I-1,1))/DELXZ2+C(I,1))*A0
      B(I,1)=C(I,1)+C(I,1)/2.+G(I,1)*A1
      C(I,1)=C(I,1)/2.
      H(I,1)=A(I,1)-C(I,1)*(A(I+1,1)-A(I-1,1))+B(I,1)*(A(I+1,1)-2.*A(I,1
     1)+A(I-1,1))
      Z(I,1)=E(I,1)-C(I,1)*(E(I+1,1)-E(I-1,1))+B(I,1)*(E(I+1,1)-2.*E(I,1
     1)+E(I-1,1))
  101 CONTINUE
C (LEFT,LEVEL)
      LMLM1=LML-1
      DO 102 I=2,LMLM1
      C(I,MLB)=((-G(I+1,MLB)+G(I-1,MLB))/DELXZ2+C(I,MLB))*A0
      B(I,MLB)=C(I,MLB)+C(I,MLB)/2.+G(I,MLB)*A1
      C(I,MLB)=C(I,MLB)/2.
      Z(I,MLB)=E(I,MLB)-C(I,MLB)*(E(I+1,MLB)-E(I-1,MLB))+B(I,MLB)*(E(I+1
     1,MLB)-2.*E(I,MLB)+E(I-1,MLB))
  102 CONTINUE
C (RIGHT,LEVEL)
      LMLP31=LML+31
      DO 105 I=LMLP31,LR
      C(I,MLB)=((-G(I+1,MLB)+G(I-1,MLB))/DELXZ2+C(I,MLB))*A0
      B(I,MLB)=C(I,MLB)+C(I,MLB)/2.+G(I,MLB)*A1
      C(I,MLB)=C(I,MLB)/2.
      Z(I,MLB)=E(I,MLB)-C(I,MLB)*(E(I+1,MLB)-E(I-1,MLB))+B(I,MLB)*(E(I+1
     1,MLB)-2.*E(I,MLB)+E(I-1,MLB))
  105 CONTINUE
C (SLOPES)
      J=MLB
      I1=LML
      I2=LML+30
      DO 104 K=1,9
      J=J-1
      I1=I1+2
      I2=I2-1
      C(I1,J)=((-G(I1+2,J-1)+G(I1-2,J+1))/(8.*DELXZ)+C(I1,J))*A0
      B(I1,J)=C(I1,J)+C(I1,J)/4.+G(I1,J)*A1/8.
      C(I1,J)=C(I1,J)/4.
      Z(I1,J)=E(I1,J)-C(I1,J)*(E(I1+2,J-1)-E(I1-2,J+1))+B(I1,J)*(E(I1+2,
     1J-1)-2.*E(I1,J)+E(I1-2,J+1))
      C(I2,J)=((-G(I2+1,J+1)+G(I2-1,J-1))/(4.*DELXZ)+C(I2,J))*A0
      B(I2,J)=C(I2,J)+C(I2,J)/2.+G(I2,J)*A1/2.
      C(I2,J)=C(I2,J)/2.
      Z(I2,J)=E(I2,J)-C(I2,J)*(E(I2+1,J+1)-E(I2-1,J-1))+B(I2,J)*(E(I2+1,
     1J+1)-2.*E(I2,J)+E(I2-1,J-1))
  104 CONTINUE
C (LEFT EDGE OF MOUNTAIN)
      C(LML,MLB)=C(LML,MLB)*A0
      Z(LML,MLB)=E(LML,MLB)-C(LML,MLB)*(E(LML,MLB)-E(LML-1,MLB))
      IF(C(LML,MLB).LT.0.0) Z(LML,MLB)=E(LML,MLB)-C(LML,MLB)*(E(LML+2,JB
     1)-E(LML,MLB))/2.236
C (RIGHT EDGE OF MOUNTAIN)
      I=LML+30
      J=MLB
      C(I,J)=C(I,J)*A0/1.414
      Z(I,J)=E(I,J)-C(I,J)*(E(I,J)-E(I-1,J-1))
      IF(C(I,J).LT.0.0) Z(I,J)=E(I,J)-C(I,J)*(E(I+1,J)-E(I,J))*1.414
C (CUSP)
      I=LML+20
      J=JT
      C(I,J)=C(I,J)*A0/2.236
      Z(I,J)=E(I,J)-C(I,J)*(E(I,J)-E(I-2,J+1))
      IF(C(I,J).LT.0.0) Z(I,J)=E(I,J)-2.236*C(I,J)*(E(I+1,J+1)-E(I,J))/1
     1.414
      RETURN
      END
      SUBROUTINE SCND(A0,A1)
      COMMON/BLOCK2/MLT,LR,LML,DELXZ,RELF,RHO1,THETREF,CP,C1,C2,C3,C4,EP
     1S,FLAM,CON22,CHK,N&RDS,NUMITER,RV1S,UDELXZ
      COMMON/BLOCK3/JB,MLB,LR1,DELXZ2,DELXZSQ,CON2,CON3,CON11,CON8,CON21
     1,F4,JT
      COMMON/BLOCK5/A(111,36),B(111,36),C(111,36),D(111,36),E(111,36),F(
     1111,36),G(111,36),H(111,36)
      COMMON/BLOCK8/Z(111,36)
      COMMON/BLOCK15/QLT,QRT,QTP,QINT1,QINT2,QRATIO,QDIFF,QLTH,QRTH,QTPH
C (TOP BOUNDARY)
      DO 111 I=2,LR
      D(I,1)=D(I,1)*A0
      A(I,1)=H(I,1)-D(I,1)*(H(I,1)-H(I,2))
      E(I,1)=Z(I,1)-D(I,1)*(Z(I,1)-Z(I,2))
  111 CONTINUE
C (LEFT,LEVEL)
      LMLM1=LML-1
      DO 112 I=2,LMLM1
      E(I,MLB)=Z(I,MLB)
      QINT2=QINT2+E(I,J)/2.
  112 CONTINUE
C (RIGHT,LEVEL)
      LMLP31=LML+31
      DO 115 I=LMLP31,LR
      E(I,MLB)=Z(I,MLB)
      QINT2=QINT2+E(I,J)/2.
  115 CONTINUE
C (SLOPES)
      J=MLB
      I1=LML
      I2=LML+30
      DO 114 K=1,9
```

```
      J=J-1
      I1=I1+2
      I2=I2-1
      E(I1,J)=Z(I1,J)
      QINT2=QINT2+E(I1,J)/2.
      QINT2=QINT2+E(I1-1,J)
      E(I2,J)=Z(I2,J)
      QINT2=QINT2+E(I2,J)/2.
  114 CONTINUE
C (LEFT EDGE OF MOUNTAIN)
      E(LML,MLB)=Z(LML,MLB)
C (RIGHT EDGE OF MOUNTAIN)
      E(LML+30,MLB)=Z(LML+30,MLB)
C (CUSP)
      E(LML+20,JT)=Z(LML+20,JT)
      RETURN
      END
      SUBROUTINE ETA1(A0,A1)
      COMMON/BLOCK2/MLT,LR,LML,DELXZ,RELF,RHO1,THETREF,CP,C1,C2,C3,C4,EP
     1S,FLAM,CON22,CHK,N&RDS,NUMITER,RVIS,UDELXZ
      COMMON/BLOCK3/JB,MLB,LR1,DELXZ2,DELXZSQ,CON2,CON3,CON11,CON8,CON21
     1,F4,JT
      COMMON/BLOCK5/A(111,36),B(111,36),C(111,36),D(111,36),E(111,36),F(
     1111,36),G(111,36),H(111,36)
      COMMON/BLOCK8/Z(111,36)
C (LEFT,LEVEL)
      LMLM1=LML-1
      DO 102 I=2,LMLM1
      COEFA1=  ((-H(I+1,MLB)+H(I-1,MLB))/DELXZ+C(I,MLB))*A0
      COEFB1=COEFA1*COEFA1/2.+H(I,MLB)*A1
      COEFA1=COEFA1/2.
      Z(I,MLB)=B(I,MLB)-  COEFA1*(B(I+1,MLB)-B(I-1,MLB))+  COEFB1*(B(I+1
     1,MLB)-2.*B(I,MLB)+B(I-1,MLB))+E(I,MLB)
  102 CONTINUE
C (RIGHT,LEVEL)
      LMLP31=LML+31
      DO 103 I=LMLP31,LR
      COEFA1=  ((-H(I+1,MLB)+H(I-1,MLB))/DELXZ+C(I,MLB))*A0
      COEFB1=COEFA1*COEFA1/2.+H(I,MLB)*A1
      COEFA1=COEFA1/2.
      Z(I,MLB)=B(I,MLB)-  COEFA1*(B(I+1,MLB)-B(I-1,MLB))+  COEFB1*(B(I+1
     1,MLB)-2.*B(I,MLB)+B(I-1,MLB))+E(I,MLB)
  103 CONTINUE
C (SLOPES)
      J=MLB
      I1=LML
      I2=LML+30
      DO 104 K=1,9
      J=J-1
      I1=I1+2
      I2=I2-1
      COEFA1=((-H(I1+2,J-1)+H(I1-2,J+1))/(2.5*DELXZ)+C(I1,J))*A0
      COEFB1=COEFA1*COEFA1/4.+H(I1,J)*A1/5.
      COEFA1=COEFA1/4.
      Z(I1,J)=B(I1,J)-COEFA1*(B(I1+2,J-1)-B(I1-2,J+1))+COEFB1*(B(I1+2,J-
     11)-2.*B(I1,J)+B(I1-2,J+1))+E(I1,J)
      COEFA1=((-H(I2+1,J+1)+H(I2-1,J-1))/(2.*DELXZ)+C(I2,J))*A0
      COEFB1=COEFA1*COEFA1/2.+H(I2,J)*A1/2.
      COEFA1=COEFA1/2.
      Z(I2,J)=B(I2,J)-COEFA1*(B(I2+1,J+1)-B(I2-1,J-1))+COEFB1*(B(I2+1,J+
     11)-2.*B(I2,J)+B(I2-1,J-1))+E(I2,J)
  104 CONTINUE
C (LEFT EDGE OF MOUNTAIN)
      C(LML,MLB)=C(LML,MLB)+A0
      Z(LML,MLB)=B(LML,MLB)-C(LML,MLB)*(B(LML,MLB)-B(LML-1,MLB))
      IF(C(LML,MLB).LT.0.0) Z(LML,MLB)=B(LML,MLB)-C(LML,MLB)*(B(LML+2,JB
     1)-B(LML,MLB))/2.256
C (RIGHT EDGE OF MOUNTAIN)
      I=LML+30
      J=MLB
      C(I,J)=C(I,J)+A0/1.414
      Z(I,J)=B(I,J)-C(I,J)*(B(I,J)-B(I-1,J-1))
      IF(C(I,J).LT.0.0) Z(I,J)=B(I,J)-C(I,J)*(B(I+1,J)-B(I,J))*1.41
C (CUSP)
      I=LML+20
      J=JT
      C(I,J)=C(I,J)+A0/2.256
      Z(I,J)=B(I,J)-C(I,J)*(B(I,J)-B(I-2,J+1))
      IF(C(I,J).LT.0.0) Z(I,J)=B(I,J)-2.256*C(I,J)*(B(I+1,J+1)-B(I,J))/1
     1.414
      RETURN
      END
      SUBROUTINE ETA2(A0,A1)
      COMMON/BLOCK2/MLT,LR,LML,DELXZ,RELF,RHO1,THETREF,CP,C1,C2,C3,C4,EP
     1S,FLAM,CON22,CHK,N&RDS,NUMITER,RVIS,UDELXZ
      COMMON/BLOCK3/JB,MLB,LR1,DELXZ2,DELXZSQ,CON2,CON3,CON11,CON8,CON21
     1,F4,JT
      COMMON/BLOCK5/A(111,36),B(111,36),C(111,36),D(111,36),E(111,36),F(
     1111,36),G(111,36),H(111,36)
      COMMON/BLOCK8/Z(111,36)
C (LEFT,LEVEL)
      LMLM1=LML-1
      DO 112 I=2,LMLM1
      B(I,MLB)=Z(I,MLB)+2.*(H(I,JB)-H(I,MLB))*(Z(I,JB)-Z(I,MLB))*A1-(H(I
     1+1,JB)-H(I-1,JB)+H(I-1,MLB)-H(I+1,MLB))*(C(I+1,MLB)-C(I-1,MLB))*A1
     2/DELXZ
  112 CONTINUE
C (RIGHT,LEVEL)
      LMLP31=LML+31
      DO 113 I=LMLP31,LR
      B(I,MLB)=Z(I,MLB)+2.*(H(I,JB)-H(I,MLB))*(Z(I,JB)-Z(I,MLB))*A1-(H(I
     1+1,JB)-H(I-1,JB)+H(I-1,MLB)-H(I+1,MLB))*(C(I+1,MLB)-C(I-1,MLB))*A1
     2/DELXZ
  113 CONTINUE
C (SLOPES)
      J=MLB
      I1=LML
      I2=LML+30
      DO 114 K=1,9
      J=J-1
      I1=I1+2
      I2=I2-1
      I=I1
      B(I1,J)=Z(I1,J)+(H(I,J-1)+H(I-1,J-1)-2.*H(I,J))*(Z(I,J-1)+Z(I-1,J-
     11)-2.*Z(I,J))*A1/2.5-(H(I+1,J-2)+H(I,J-1)-H(I-1,J-1)-H(I-2,J)+H(I
     2+2,J-1)+H(I-2,J-1))*(C(I+2,J-1)-C(I-2,J+1))*A1/(10.*DELXZ)
      I=I2
      B(I2,J)=Z(I2,J)+(H(I+1,J)+H(I,J-1)-2.*H(I,J))*(Z(I+1,J)+Z(I,J-1)-2
     1.*Z(I,J))*A1-(2.*H(I+1,J)-2.*H(I,J-1)+H(I-1,J-1)-H(I+1,J+1))*(C(I+
     21,J+1)-C(I-1,J-1))*A1/(4.*DELXZ)
  114 CONTINUE
C (LEFT EDGE OF MOUNTAIN)
      B(LML,MLB)=Z(LML,MLB)
C (RIGHT EDGE OF MOUNTAIN)
      B(LML+30,MLB)=Z(LML+30,MLB)
C (CUSP)
      B(LML+20,JT)=Z(LML+20,JT)
      RETURN
```

```
      END
*ASCENT
          ENTRY     SKFL
 SKFL     CON       0
          SA1       SKFL
          LX1       30
          SA1       X1+1
          SA1       X1
          BX6       X1
          SA6       UN
          RJ        2
          JP        *+3
          EQ        400001B
          EQ        0
          EQ        UN
          EQ        ST
          EQ        SKFL
 UN       CON       3
 ST       CON       0
          END
*RUN
*END
```

PLOTTING PACKAGE

```
*JOB,5228,V0424010,DERICKSON
*LIMIT,T=7,PR=100,PT=9,DDB0=500
*ASSIGN,AR1=8,R
*FORTRAN
      PROGRAM DOTTIE
      COMMON/BLOCK1/WSITERM(36),DIFU(36),GPRTERM(36),PIB(36),F22(36),F1C
     1ON10(36),RHOA(36),R124(36),FLL1(36),F1(36),F2(36),F3(36)
      COMMON/BLOCK2/MLT,LR,LML,DELXZ,RELF,RHO1,THETREF,CP,C1,C2,C3,C4,EP
     1S,FLAM,CON22,CHK,N&RDS,NUMITER
      COMMON/BLOCK4/DELT,TIME,NSTEPS,ITER,IVAR,JFORM
      COMMON/BLOCK8/Z(111,36)
      COMMON/BLOCK14/AFLO(17),AFHI(17),AFINC(17),ITARRAY(2,17)
      COMMON/BLOCK15/LR1,MLB
      COMMON/BLOCK16/MX1,MX2,MY2,MX3,NY1,NY2
C
C NUMBER OF FILES TO BE SKIPPED
      NFILES=0
C
C(SKIP SINGLE SUBSCRIPTED VARIABLES)
      CALL SKFL(8)
C
      ITARRAY(1,1)=10H          P
      ITARRAY(2,1)=10HMI
      ITARRAY(1,2)=10H  Q,TOTAL
      ITARRAY(2,2)=10HMOISTURE
      ITARRAY(1,3)=10HVORTICITY
      ITARRAY(2,3)=10HPRODUCTION
      ITARRAY(1,7)=10H         WI,
      ITARRAY(2,7)=10HICE
      ITARRAY(1,8)=10H      WL,LI
      ITARRAY(2,8)=10HQUID
      ITARRAY(1,9)=10H        WV,V
      ITARRAY(2,9)=10HAPOR
      ITARRAY(1,10)=10H          THE
      ITARRAY(2,10)=10HTA
      ITARRAY(1,11)=10HW,SAT VAPO
      ITARRAY(2,11)=10HR AND LIQ
      ITARRAY(1,12)=10H          VOR
      ITARRAY(2,12)=10HTICITY
      ITARRAY(1,13)=10H    STREAM
      ITARRAY(2,13)=10HFUNCTION
      ITARRAY(1,14)=10H           TE
      ITARRAY(2,14)=10HMP
      ITARRAY(1,15)=10H        U-CO
      ITARRAY(2,15)=10HMP
      ITARRAY(1,16)=10H        W-CO
      ITARRAY(2,16)=10HMP
      ITARRAY(1,17)=10H    EDDY VI
      ITARRAY(2,17)=10HSCOSITY
      AFLO(1)=0.055
      AFINC(1)=0.0025
      AFHI(1)=0.065
      AFLO(2)=0.0005
      AFINC(2)=0.00025
      AFHI(2)=0.0035
      AFLO(7)=0.0
      AFINC(7)=0.0
      AFHI(7)=0.0
      AFLO(8)=0.00005
      AFINC(8)=0.00005
      AFHI(8)=0.00065
      AFLO(9)=0.0005
      AFINC(9)=0.00025
      AFHI(9)=0.0035
      AFLO(10)=1.0
      AFINC(10)=2.
      AFHI(10)=15.
      AFLO(12)=-0.010
      AFINC(12)=0.005
      AFHI(12)=0.010
      AFLO(13)=100.
      AFINC(13)=100.
      AFHI(13)=1000.
      AFLO(15)=2.0
      AFINC(15)=2.
      AFHI(15)=10.
      AFLO(16)=0.0
      AFINC(16)=0.0
      AFHI(16)=0.0
      AFLO(17)=50.
      AFINC(17)=50.
      AFHI(17)=250.
      LML=51
      LR1=111
      MLT=25
      MLB=36
      JB=MLB-1
      LR=LR1-1
C RASTOR POINTS FOR MOUNTAIN SHAPE AND TITLE POSITIONS
      FLML=LML
      FJB=JB
      FLR=LR
      FMLT=MLT
      MX1=51.+(FLML-1.)*922./FLR
      MX2=51.+(FLML+19.)*922./FLR
      MY2=51.+9220./FLR
      MX3=51.+(FLML+29.)*922./FLR
      NY1=110.+FJB*922./FLR
      NY2=NY1+80
C
C
C SKIP FILES AS SPECIFIED BY NFILES
C
      IF(NFILES.EQ.0) GO TO 9
      DO 8 N=1,NFILES
      CALL SKFL(8)
    8 CONTINUE
    9 CONTINUE
C
      DO 31 IT=1,26
      WRITE(6,35)
   35 FORMAT(/////)
      DO 10 K=1,5
      CALL RDVAR
      CALL MODPLOT
   10 CONTINUE
      CALL RDVAR
      DO 12 K=1,2
      CALL RDVAR
      CALL MODPLOT
   12 CONTINUE
C
C U-COMP
      CALL RDVAR
      CALL MODPLOT
C W-COMP
      CALL RDVAR
C EDDY COEFFICIENT
```

```
      CALL RDVAR
      CALL MODPLOT
C
      WRITE(6,36) NSTEPS,TIME
   36 FORMAT(1H ,*THIS IS FOR TIME STEP*,I4,*  TOTAL REAL TIME=*,F8.2/)
      BUFFER IN(8,1) (VEOF,VEOF)
   15 IF(UNIT,8) 15,16,28,16
   16 WRITE(6,17)
   17 FORMAT(1H0,*NO EOF OR PARITY ERROR IN DOTTIE*/)
   28 WRITE(6,29)
   29 FORMAT(1H0,*EOF ENCOUNTERED*/)
   30 CONTINUE
   31 CONTINUE
      END
      SUBROUTINE RDVAR
      DIMENSION A(111,36)
      COMMON/BLOCK4/DELT,TIME,NSTEPS,ITER,IVAR,JFORM
      COMMON/BLOCK8/Z(111,36)
      COMMON/BLOCK15/LR1,MLB
      BUFFER IN(8,1) (DELT,IVAR)
    5 IF(UNIT,8) 5,8,6
    6 WRITE(6,7)
    7 FORMAT(1H0,*EOF OR PARITY ERROR IN RDVAR*/)
    8 BUFFER IN(8,1) (A(1,1),A(LR1,MLB))
    9 IF(UNIT,8) 9,12,10
   10 WRITE(6,7)
   12 CONTINUE
      MLBP1=MLB+1
      DO 1 J=1,MLB
      DO 1 I=1,LR1
    1 Z(I,J)=A(I,MLBP1-J)
      RETURN
      END
      SUBROUTINE MODPLOT
      DIMENSION ITITLE(2),ISTITLE(2)
      COMMON/BLOCK4/DELT,TIME,NSTEPS,ITER,IVAR,JFORM
      COMMON/BLOCK8/Z(111,36)
      COMMON/BLOCK14/AFLO(17),AFHI(17),AFINC(17),ITARRAY(2,17)
      COMMON/BLOCK15/LR1,MLB
      COMMON/BLOCK16/MX1,MX2,MY2,MX3,NY1,NY2
      ITITLE(1)=ITARRAY(1,IVAR)
      ITITLE(2)=ITARRAY(2,IVAR)
      CALL OPTION(0,1,0,0,1)
      CALL FRSTPT(MX1,51)
      CALL VECTOR(MX2,MY2)
      CALL VECTOR(MX3,51)
      ENCODE(14,100,ISTITLE)NSTEPS
  100 FORMAT(*TIME STEP=*I4)
      CALL PWRT(286,NY1,ISTITLE,14,1,0)
      ENCODE(10,101,ISTITLE)DELT
  101 FORMAT(*DELT=*F5.2)
      CALL PWRT(476,NY1,ISTITLE,10,1,0)
      ENCODE(12,102,ISTITLE)TIME
  102 FORMAT(*TIME=*F7.2)
      CALL PWRT(619,NY1,ISTITLE,12,1,0)
      IF(IVAR-13) 10,9,10
    9 ENCODE(9,103,ISTITLE)ITER
  103 FORMAT(*ITER=*I4)
      CALL PWRT(772,NY1,ISTITLE,9,1,0)
   10 CONTINUE
      CALL PWRT(952,NY2,ITITLE,20,2,0)
      FLO=AFLO(IVAR)
      FHI=AFHI(IVAR)
      FINC=AFINC(IVAR)
      TCHK=TIMEF(ARG)/1000.
      WRITE(6,14) ITITLE,TCHK
   14 FORMAT(1H ,*STARTING TIME FOR*,1X,2A10,5X,F7.3)
      IF(IVAR-15) 25,24,25
   24 CALL UCOMP
      GO TO 26
   25 CONTINUE
      CALL CALCNT(Z,LR1,MLB,FLO,FHI,FINC,0,-1,0)
   26 CONTINUE
      IF(IVAR.EQ.8) CALL CLOUD
      TCHK=TIMEF(ARG)/1000.
      WRITE(6,15) ITITLE,TCHK
   15 FORMAT(1H ,*FINISHING TIME FOR*,2A10,5X,F7.3/)
      IF(IVAR-13) 20,16,20
   16 FLO=5000.
      FHI=17000.
      FINC=4000.
      CALL CALCNT(Z,LR1,MLB,FLO,FHI,FINC,2,-1,0)
      TCHK=TIMEF(ARG)/1000.
      WRITE(6,15) ITITLE,TCHK
   20 CONTINUE
      CALL FRAME
      END
      SUBROUTINE CLOUD
      COMMON/BLOCK2/MLT,LR,LML,DELXZ,RELF,RHO1,THETREF,CP,C1,C2,C3,C4,EP
     1S,FLAM,CON22,CHK,NWRDS,NUMITER
      COMMON/BLOCK8/Z(111,36)
      COMMON/BLOCK15/LR1,MLB
C TOP
      DO 10 J=11,MLB
      DO 10 I=1,LR1
      IF(Z(I,J)-0.000020) 10,8,8
    8 X=I
      Y=J
      CALL PSYM(X,Y,1H.,0,0,1)
   10 CONTINUE
C LEFT
      LIMX=LML-2
      DO 20 J=1,11
      LIMX=LIMX+2
      DO 20 I=1,LIMX
      IF(Z(I,J)-0.000020) 20,18,18
   18 X=I
      Y=J
      CALL PSYM(X,Y,1H.,0,0,1)
   20 CONTINUE
C RIGHT
      IL=LML+31
      DO 30 J=1,11
      IL=IL-1
      DO 30 I=IL,LR1
      IF(Z(I,J)-0.000020) 30,28,28
   28 X=I
      Y=J
      CALL PSYM(X,Y,1H.,0,0,1)
   30 CONTINUE
      RETURN
      END
      SUBROUTINE UCOMP
      COMMON/BLOCK2/MLT,LR,LML,DELXZ,RELF,RHO1,THETREF,CP,C1,C2,C3,C4,EP
     1S,FLAM,CON22,CHK,NWRDS,NUMITER
      COMMON/BLOCK8/Z(111,36)
      COMMON/BLOCK15/LR1,MLB
      JB=MLB-1
```

```
      X4=LR1
      Y4=MLB
      Y2=Y4+0.9*X4+0.05
      LML20=LML+20
      LML30=LML+30
      CALL SET(0.05,0.95,0.05,Y2,-12.0,12.0,1.,Y4,1)
      CALL PERIM(LR,0,JB,0)
      AB=LR
      DO 100 I=1,LR1,10
      AT=I-1
      A1=AT/AB
      XCENTER=0.9*A1+0.05
      X1=XCENTER-0.035
      X2=XCENTER+0.035
      JBOT=1
      IF(I.GT.LML.AND.I.LE.LML20) JBOT=(I-LML)*2+1
      IF(I.GT.LML20.AND.I.LT.LML30) JBOT=11-I+LML20
      CALL SET(X1,X2,0.05,Y2,-12.0,12.0,1.,Y4,1)
      Y0=JBOT
      CALL FRSTPT(0.0,Y0)
      IDASH=1777B
      CALL DASHLN(IDASH)
      DO 50 J=JBOT,MLB
      X=Z(I,J)
      Y=J
      CALL VECTOR(X,Y)
   50 CONTINUE
      CALL FRSTPT(0.0,Y4)
      IDASH=1430B
      CALL DASHLN(IDASH)
      CALL VECTOR(0.0,Y0)
  100 CONTINUE
      RETURN
      END
      SUBROUTINE CALCNT    (AM,MX,NY,FLO,HI,FINC,NSET,NHI,NDOT)
      DIMENSION AM(MX,NY)
      COMMON/CONT/MT,NT,IX,IY,IDX,IDY,ISS,NP,CV,NNT,ASH,INX(8),INY(8),
     1IPT(3,3),LEGEND(11),REC(800),NQ
      COMMON/BLOCK2/MLT,LR,LML,DELXZ,RELF,RHO1,THETREF,CP,C1,C2,C3,C4,EP
     1S,FLAM,CON22,CHK,NORDS,NUMITER
      COMMON/BLOCK4/DELT,TIME,NSTEPS,ITER,IVAR,JFORM
      COMMON/CALIN/XLML,XLML30
C     INITIALIZE
      XLML=LML
      XLML30=LML+30
      IF(NSET)80,81
   80 NQ=IABS(NSET)
      GO TO 82
   81 NQ=1
   82 CONTINUE
      IF(IVAR.EQ.13) GO TO 11
      IL=LML-1
      IR=LML+30
      DO 10 J=1,10
      IL=IL+2
      IR=IR-1
      AM(IL,J)=3.*AM(IL-1,J)-3.*AM(IL-2,J)+AM(IL-3,J)
      AM(IR,J)=3.*AM(IR+1,J)-3.*AM(IR+2,J)+AM(IR+3,J)
      IF(J.EQ.10) GO TO 10
      IL1=IL+1
      IR1=IR-1
      DIFF=AM(IR,J)-AM(IL,J)
      DIV=IR-IL
      DO 9 I=IL1,IR1
      AM(I,J)=AM(IL,J)+DIFF*(I-IL)/DIV
   10 CONTINUE
   11 CONTINUE
C
      MT=MX
      NT=NY
      CRIT=16.
      IF (FINC.LT.0.) CRIT=-FINC
      FANC=FINC
      GLO=FLO
      HA=HI
      NNT=2
      IDASH=NDOT
      IF(NDOT.EQ.0) IDASH=1777B
      IPT(1,1)=8
      IPT(1,2)=1                                                    SCAN0320
      IPT(1,3)=2                                                    SCAN0330
      IPT(2,1)=7                                                    SCAN0340
      IPT(2,3)=3                                                    SCAN0350
      IPT(3,1)=6                                                    SCAN0360
      IPT(3,2)=5                                                    SCAN0370
      IPT(3,3)=4                                                    SCAN0380
      INX(1)=-1                                                     SCAN0390
      INX(2)=-1                                                     SCAN0400
      INX(3)=0                                                      SCAN0410
      INX(4)=1                                                      SCAN0420
      INX(5)=1                                                      SCAN0430
      INX(6)=1                                                      SCAN0440
      INX(7)=0                                                      SCAN0450
      INX(8)=-1                                                     SCAN0460
      INY(1)=0                                                      SCAN0470
      INY(2)=1                                                      SCAN0480
      INY(3)=+1
      INY(4)=+1                                                     SCAN0500
      INY(5)=0                                                      SCAN0510
      INY(6)=-1                                                     SCAN0520
      INY(7)=-1                                                     SCAN0530
      INY(8)=-1                                                     SCAN0540
      IF ((GLO.EQ.0.).AND.(HA.EQ.0.))50,52
   50 GLO=1.E100
      HA=-GLO
      DO 55 K=1,NT
      DO 55 KK=1,MT
      GLO=MIN1 (AM(KK,K),GLO)
      HA=MAX1 (AM(KK,K),HA)
   55 CONTINUE
   52 IF (FANC.LE.0.)  FANC = (HA-GLO)/CRIT
C
   51 CONTINUE
      IF(NHI)55,53,54
   54 AMA=HA
      IF (ABS (GLO).GT.ABS (HA))AMA=GLO
      ENCODE(8,100,ISH)AMA
  100 FORMAT(E8.1)
      DECODE(8,101,ISH) IH
  101 FORMAT(5X,I3)
      IEXP=2-IH
      ASH=10.**IEXP
   55 IF(NSET.NE.0) GO TO 24
      Y4=MT
      Y4=NT
      IF(MT-NT)21,21,22
   21 Y2 = .95
      Y2 = (Y4/Y4)*.90+.05
```

```
      GO TO 23
   22 X2 = .95
      Y2 = (Y4/X4)*.90+.05
   23 CALL SET (.05,X2,.05,Y2,1.,X4,1.,Y4,1)
      CALL PERIM(MT-1,0,NT-1,0)
   24 CONTINUE
      IF(NSET.EQ.2) GO TO 25
      ENCODE (108,201,LEGEND) GLO,HA,FANC,ASH,AM(2,2)
  201 FORMAT(15HCONTOUR FROM E11.4,4H TO E11.4,18H CONTOUR INTERVAL E11.
     14,12H    SCALING=E8.1,9H PT(2,2)=E11.3)
      CALL OPTION( 0,1,1,0)
      MM=96
      MMM=108
      IF(NSET.LT.0) MM=22
      CALL PWRT(50,MM,LEGEND,MMM,0,0)
   25 CONTINUE
      CALL OPTION(0,1,0,0)
C   DETERMINE CURRENT LEVEL TO BE CONTOURED
C
      CV=GLO-FANC
  120 CV=CV+FANC
C     CALL DASHLN(IDASH)
C
      CALL SCAN(AM,MT,NT)
C
C  TEST FOR ALL LEVELS CONTOURED, INCREMENT IF NOT
C
      IF (HA-CV) 150,150, 120
  150 CONTINUE
C
C     CALL DASHLN(1777B)
      IF(NH)31,32,33
   32 CALL HILO (AM,MT,NT)
   31 RETURN
   33 CALL PTVALU(AM,MT,NT)
      RETURN
      END                                                          DCNT0510
      SUBROUTINE SCAN(AM,M,N)
      DIMENSION AM(M,N)
      COMMON/CONT/MT,NT,IX,IY,IDX,IDY,ISS,NP,CV,NNT,ASH,INX(8),INY(8),
     1IPT(3,3),LEGEND(11),REC(800),NQ
      NP=0                                                         SCAN0230
      DO 58 J=1,800                                                SCAN0570
   58 REC(J)=0                                                     SCAN0580
      ISS=0                                                        SCAN0590
    2 MT1=MT-1                                                     SCAN0600
      DO 110 I=1,MT1                                               SCAN0610
      IF (AM(I,1)-CV) 55,110,110                                   SCAN0620
   55 IF (AM(I+1,1)-CV) 110,57,57                                  SCAN0630
   57 IX=I+1                                                       SCAN0640
      IY=1                                                         SCAN0650
      IDX=-1                                                       SCAN0660
      IDY=0                                                        SCAN0670
      CALL LINEAR (AM, MT,NT)
  110 CONTINUE                                                     SCAN0690
      NT1=NT-1                                                     SCAN0700
      DO 20 I=1,NT1                                                SCAN0710
      IF (AM(MT,I)-CV) 15,20,20                                    SCAN0720
   15 IF (AM(MT,I+1)-CV)  20,17,17                                 SCAN0730
   17 IX=MT                                                        SCAN0740
      IY=I+1                                                       SCAN0750
      IDX=0                                                        SCAN0760
      IDY=-1                                                       SCAN0770
      CALL LINEAR (AM, MT,NT)
   20 CONTINUE                                                     SCAN0790
   22 DO 30 I=1,MT1                                                SCAN0800
      MT2=MT+1-I                                                   SCAN0810
      IF (AM(MT2,NT)-CV) 25,30,30                                  SCAN0820
   25 IF (AM(MT2-1,NT)-CV)  30,27,27                               SCAN0830
   27 IX=MT2-1                                                     SCAN0840
      IY=NT                                                        SCAN0850
      IDX=1                                                        SCAN0860
      IDY=0                                                        SCAN0870
      CALL LINEAR (AM, MT,NT)
   30 CONTINUE                                                     SCAN0890
      DO 40 I=1,NT1                                                SCAN0900
      NT2=NT+1-I                                                   SCAN0910
      IF (AM(1,NT2)-CV)  35,40,40                                  SCAN0920
   35 IF (AM(1,NT2-1)-CV)   40,37,37                               SCAN0930
   37 IX=1                                                         SCAN0940
      IY=NT2-1                                                     SCAN0950
      IDX=0                                                        SCAN0960
      IDY=1                                                        SCAN0970
      CALL LINEAR (AM, MT,NT)
   40 CONTINUE                                                     SCAN0990
      ISS=1                                                        SCAN1000
      NT1=NT-1                                                     SCAN1010
      MT1=MT-1                                                     SCAN1020
      DO 10 J=2,NT1                                                SCAN1030
      DO 10 I=1,MT1                                                SCAN1040
      IF (AM(I,J)-CV)  5,10,10                                     SCAN1050
    5 IF (AM(I+1,J)-CV)  10,7,7                                    SCAN1060
    7 COM   =100*(I+1)+J
      IF (NP) 12,11,12                                             SCAN1080
   12 DO 9 ID= 1,NP                                                SCAN1090
      IF (REC(ID)-COM   ) 9,10,9
    9 CONTINUE                                                     SCAN1110
   11 IX= I+1                                                      SCAN1120
      IY=J                                                         SCAN1130
      IDX=-1                                                       SCAN1140
      IDY=0                                                        SCAN1150
      CALL LINEAR (AM, MT,NT)
   10 CONTINUE                                                     SCAN1170
      RETURN                                                       SCAN1180
      END                                                          SCAN1190
      SUBROUTINE LINEAR(AM,IDIM,JDIM)
      COMMON/CONT/MT,NT,IX,IY,IDX,IDY,ISS,NP,CV,NNT,ASH,INX(8),INY(8),
     1IPT(3,3),LEGEND(11),REC(800),NQ
      COMMON/CALIN/XLML,XLML30
      DIMENSION AM(IDIM,JDIM)
      N=0                                                          TRAC0180
      IX0=IX                                                       TRAC0190
      IY0=IY                                                       TRAC0200
      ISX=IDX+2                                                    TRAC0210
      ISY=IDY+2                                                    TRAC0220
      IS=IPT(ISX,ISY)                                              TRAC0230
      IS0=IS                                                       TRAC0250
      IF(IS0-8)1,1,17
   17 IS0=IS0-8                                                    TRAC0270
    1 IF (IDX) 10,2,10                                             CALC0150
    2 X=IX
      Z=IY                                                         CALC0170
      IY2=IY+IDY                                                   CALC0180
      DY=IDY                                                       CALC0190
      Y    = ((AM(IX,IY)-CV)/(AM(IX,IY)-AM(IX,IY2)))*DY + Z        C
      GO TO 54
   10 Y=IY
      W=IX                                                         CALC0230
```

```
      DX=IDX                                                        CALC0240
      IX2=IX+IDX                                                    CALC0250
      X    = ((AM(IX,IY)-CV)/(AM(IX,IY)-AM(IX2,IY)))*DX +W
   54 IF (IS.EQ.1)   306,49
  306 NP=NP+1                                                       TRAC0810
      REC(NP)=100*IX+IY                                             TRAC0820
   49 IS=IS+1                                                       TRAC0390
    9 IF (IS-9)  8, 7,7                                             TRAC0410
    7 IS=IS-8                                                       TRAC0420
    8 IDX=INX(IS)                                                   TRAC0430
      IDY=INY(IS)                                                   TRAC0440
      IX2=IX+IDX                                                    TRAC0450
      IY2=IY+IDY                                                    TRAC0460
      JTB=JTB+1                                                     TRAC0470
      IF(N)73,67
   67 CONTINUE
      CALL FRSTPT(X,Y)
      X1=X
      Y1=Y
      N=1
      GO TO 51
   73 CONTINUE
      IF(Y1.GE.11..AND.Y.GE.11.) GO TO 98
      XLFT1=XLML+2.*(Y1-1.)
      XLFT=XLML+2.*(Y-1.)
      XRHT1=XLML30-(Y1-1.)
      XRHT=XLML30-(Y-1.)
      IF(X1.LE.XLFT1.AND.X.LE.XLFT) GO TO 98
      IF(X1.GE.XRHT1.AND.X.GE.XRHT) GO TO 98
      IF(X1.LT.XLFT1.AND.X.GT.XLFT) GO TO 82
      IF(X1.GT.XRHT1.AND.X.LT.XRHT) GO TO 85
   80 CONTINUE
      IF(X1.LT.XRHT1.AND.X.GT.XRHT) GO TO 84
   81 CONTINUE
      IF(X1.GT.XLFT1.AND.X.LT.XLFT) GO TO 85
      GO TO 87
   82 CALL LINEINT(X,Y,X1,Y1,X0,Y0,1)
      CALL VECTOR(X0,Y0)
      GO TO 80
   83 CALL LINEINT(X,Y,X1,Y1,X0,Y0,2)
      CALL VECTOR(X0,Y0)
      GO TO 81
   84 CALL LINEINT(X,Y,X1,Y1,X0,Y0,2)
      CALL FRSTPT(X0,Y0)
      GO TO 98
   85 CALL LINEINT(X,Y,X1,Y1,X0,Y0,1)
      CALL FRSTPT(X0,Y0)
      GO TO 98
   87 CALL FRSTPT(X,Y)
      GO TO 99
   98 CONTINUE
      CALL VECTOR(X,Y)
   99 CONTINUE
      X1=X
      Y1=Y
   51 IF(ISS)20,50
   20 IF(IX-IX0)  12,21,12                                          TRAC0550
   21 IF(IY-IY0)  12,22,12                                          TRAC0560
   22 IF(IS-IS0)  12,23,12                                          TRAC0570
   23 IF (IS.EQ.1) 307,14
  307 NP=NP+1                                                       TRAC0890
      REC(NP)=100*IX+IY                                             TRAC0900
   14 IF (IDX) 92,55,92
   55 X=IX
      Z=IY                                                          CALC0170
      IY2=IY+IDY                                                    CALC0180
      DY=IDY                                                        CALC0190
      Y    = ((AM(IX,IY)-CV)/(AM(IX,IY)-AM(IX,IY2)))*DY + Z
      IF(Y1.GE.11..AND.Y.GE.11.) GO TO 98
      XLFT1=XLML+2.*(Y1-1.)
      XLFT=XLML+2.*(Y-1.)
      XRHT1=XLML30-(Y1-1.)
      XRHT=XLML30-(Y-1.)
      IF(X1.LE.XLFT1.AND.X.LE.XLFT) GO TO 98
      IF(X1.GE.XRHT1.AND.X.GE.XRHT) GO TO 98
      IF(X1.LT.XLFT1.AND.X.GT.XLFT) GO TO 92
      IF(X1.GT.XRHT1.AND.X.LT.XRHT) GO TO 95
   90 CONTINUE
      IF(X1.LT.XRHT1.AND.X.GT.XRHT) GO TO 94
   91 CONTINUE
      IF(X1.GT.XLFT1.AND.X.LT.XLFT) GO TO 95
      GO TO 99
   92 CALL LINEINT(X,Y,X1,Y1,X0,Y0,1)
      CALL VECTOR(X0,Y0)
      GO TO 90
   93 CALL LINEINT(X,Y,X1,Y1,X0,Y0,2)
      CALL VECTOR(X0,Y0)
      GO TO 91
   94 CALL LINEINT(X,Y,X1,Y1,X0,Y0,2)
      CALL FRSTPT(X0,Y0)
      GO TO 98
   95 CALL LINEINT(X,Y,X1,Y1,X0,Y0,1)
      CALL FRSTPT(X0,Y0)
   98 CONTINUE
      CALL VECTOR(X,Y)
   99 CONTINUE
   74 RETURN
   52 Y=IY
      W=IX                                                          CALC0230
      DX=IDX                                                        CALC0240
      IX2=IX+IDX                                                    CALC0250
      X    = ((AM(IX,IY)-CV)/(AM(IX,IY)-AM(IX2,IY)))*DX +W
      IF(Y1.GE.11..AND.Y.GE.11.) GO TO 108
      XLFT1=XLML+2.*(Y1-1.)
      XLFT=XLML+2.*(Y-1.)
      XRHT1=XLML30-(Y1-1.)
      XRHT=XLML30-(Y-1.)
      IF(X1.LE.XLFT1.AND.X.LE.XLFT) GO TO 108
      IF(X1.GE.XRHT1.AND.X.GE.XRHT) GO TO 108
      IF(X1.LT.XLFT1.AND.X.GT.XLFT) GO TO 102
      IF(X1.GT.XRHT1.AND.X.LT.XRHT) GO TO 105
  100 CONTINUE
      IF(X1.LT.XRHT1.AND.X.GT.XRHT) GO TO 104
  101 CONTINUE
      IF(X1.GT.XLFT1.AND.X.LT.XLFT) GO TO 105
      GO TO 109
  102 CALL LINEINT(X,Y,X1,Y1,X0,Y0,1)
      CALL VECTOR(X0,Y0)
      GO TO 100
  103 CALL LINEINT(X,Y,X1,Y1,X0,Y0,2)
      CALL VECTOR(X0,Y0)
      GO TO 101
  104 CALL LINEINT(X,Y,X1,Y1,X0,Y0,2)
      CALL FRSTPT(X0,Y0)
      GO TO 108
  105 CALL LINEINT(X,Y,X1,Y1,X0,Y0,1)
      CALL FRSTPT(X0,Y0)
  108 CONTINUE
```

```
      CALL VECTOR(X,Y)
  109 CONTINUE
      RETURN
   88 IF(IX2) 13,74,13
   13 IF (IX2-MT) 19,19,74
   19 IF (IY2)  11,74,11
   11 IF (IY2-NT)  12,12,74
   12 IF (CV-AM(IX2,IY2))  16,16,88
   88 ISTE=(IS/2)*2
      IF (ISTE.EQ.IS) 49,1
   16 IS=IS+8                                                    TRAC0830
      IX=IX2                                                     TIIC0840
      IY=IY2                                                     TRAC0850
      GO TO 9                                                    TRAC0860
      END                                                        TRAC0960
      SUBROUTINE HILO(AM,M,N)
      COMMON/CONT/MT,NT,IX,IY,IDX,IDY,ISS,NP,CV,NNT,ASH,INX(8),INY(8),
     1IPT(3,3),LEGEND(11),REC(800),NQ
      DIMENSION AM(M,N),JSIGN(2)
      DATA(JSIGN=1HL,1HH)
  100 FORMAT (E10.3)
      NMT=NNT+1
      NN=NT-NNT
      MM=MT-NNT
      DO 10 J=NMT,NN
      DO 10 I=NMT,MM
      II=I-NNT
   11 IF (AM(I,J)-AM(II,J))12,10,13
   12 KS=1
      DO 40 K=1,NNT
      DO 40 KK=1,7
      IX2=I+K*INX(KK)
      IY2=J+K*INY(KK)
      IF(AM(I,J)-AM(IX2,IY2))40,10,10
   40 CONTINUE
      GO TO 50
   13 KS=2
      DO 50 K=1,NNT
      DO 50 KK=1,7
      IX2=I+K*INX(KK)
      IY2=J+K*INY(KK)
      IF(AM(I,J)-AM(IX2,IY2))10,10,50
   50 CONTINUE
   50 ENCODE (10,100,LEGEND) AM(I,J)
      XPLT = I
      YPLT = J
      CALL PSYM(XPLT,YPLT,JSIGN(KS),0,0,1)
      CALL PWRT(XPLT-1.,YPLT-1.,LEGEND,10,0,0)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE PTVALU (AM,M,N)
      COMMON/CONT/MT,NT,IX,IY,IDX,IDY,ISS,NP,CV,NNT,ASH,INX(8),INY(8),
     1IPT(3,3),LEGEND(11),REC(800),NQ
      DIMENSION AM(M,N)
   12 MSKP=(MT+23)/24
      NSKP=(NT+47)/48
      DO 10 J=NQ,NT,NSKP
      DO 10 I=NQ,MT,MSKP
      AMN=AM(I,J)+ASH
      ENCODE(5,100,IWD)AMN
  100 FORMAT (F5.0)
      IWD=IWD.AND.47777777777777777777B
      XPLT=I
      YPLT=J
      CALL PWRT(XPLT,YPLT,IWD,4, 0,0)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE LINEINT(X,Y,X1,Y1,X0,Y0,LINE)
      COMMON/BLOCK2/MLT,LR,LML,DELXZ,RELF,RHO1,THETREF,CP,C1,C2,C3,C4,EP
     1S,FLAM,CON22,CHK,N&RDS,NUMITER
      JFLAG=0
      IFLAG=0
      FLML2=2-LML
      FLML31=LML+31
      TEST=1.E-02
      DIFF=ABS(X1-X)
      IF(DIFF-TEST) 5,6,6
    5 X0=X
      JFLAG=1
      GO TO 7
    6 SLOPE=(Y1-Y)/(X1-X)
      B=Y-SLOPE*X
    7 IF(LINE-2) 10,20,20
   10 IF(IFLAG-1) 11,11,40
   11 IF(JFLAG)14,12,14
   12 X0=(FLML2-2.*B)/(2.*SLOPE-1.)
   14 Y0=(X0+FLML2)/2.
      IFLAG=IFLAG+1
      IF(Y0-11.) 40,40,20
   20 IF(IFLAG-1) 21,21,40
   21 IF(JFLAG)24,22,24
   22 X0=(FLML31-B)/(SLOPE+1.)
   24 Y0=-X0+FLML31
      IFLAG=IFLAG+1
      IF(Y0-11.) 40,40,10
   40 RETURN
      END
*ASCENT
            ENTRY   SKFL
SKFL        CON     0
            SA1     SKFL
            LX1     30
            SA1     X1+1
            SA1     X1
            BX6     X1
            SA6     UN
            RJ      2
            JP      *+3
            EQ      400001B
            EQ      0
            EQ      UN
            EQ      ST
            EQ      SKFL
UN          CON     5
ST          CON     0
            END
*RJ1
*END
```

FIGURES

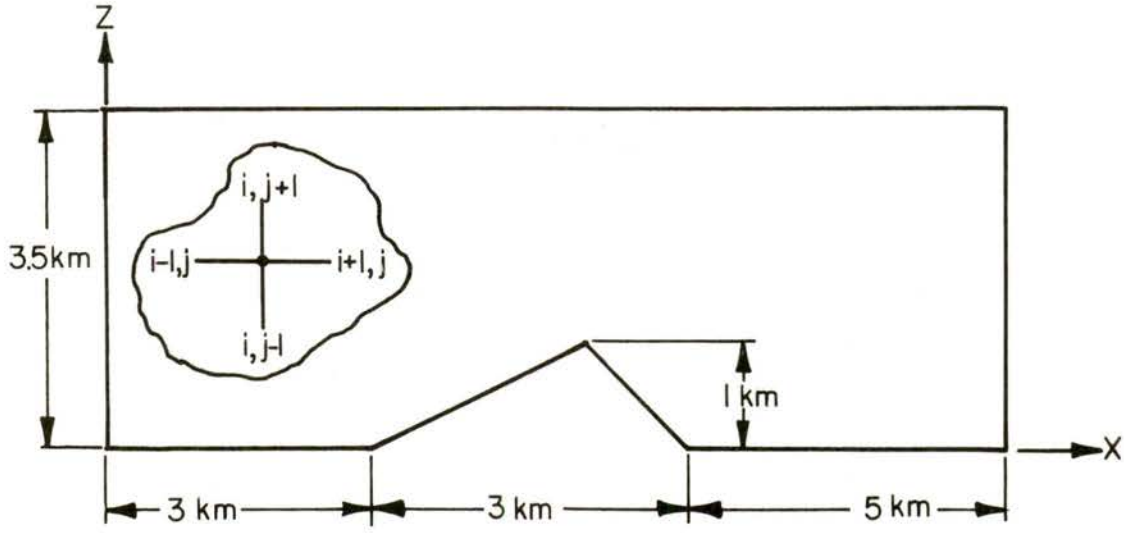


Figure 2.1 The grid network for the cloud model.

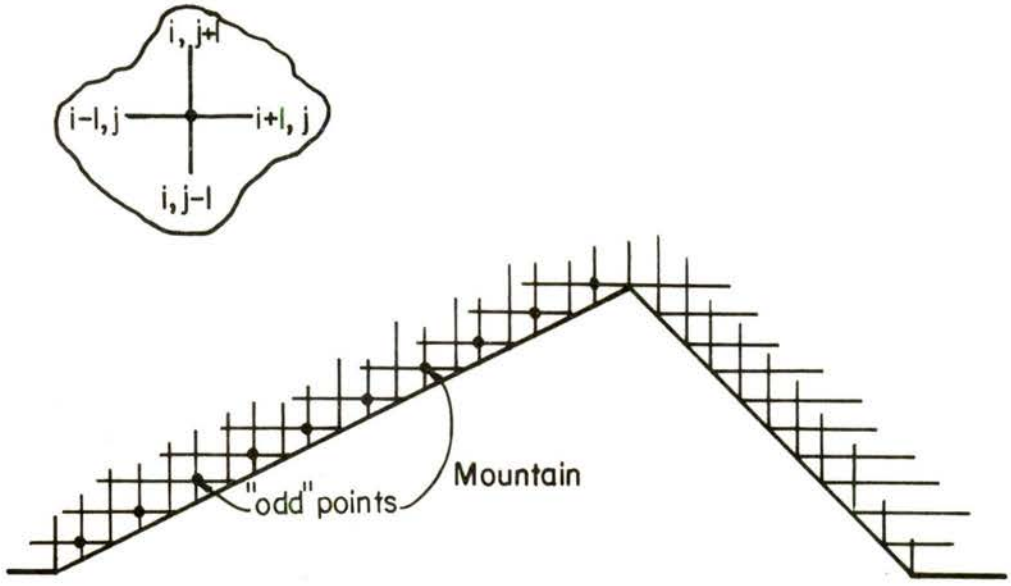


Figure 2.2 The ten "odd" points on the windward slope.

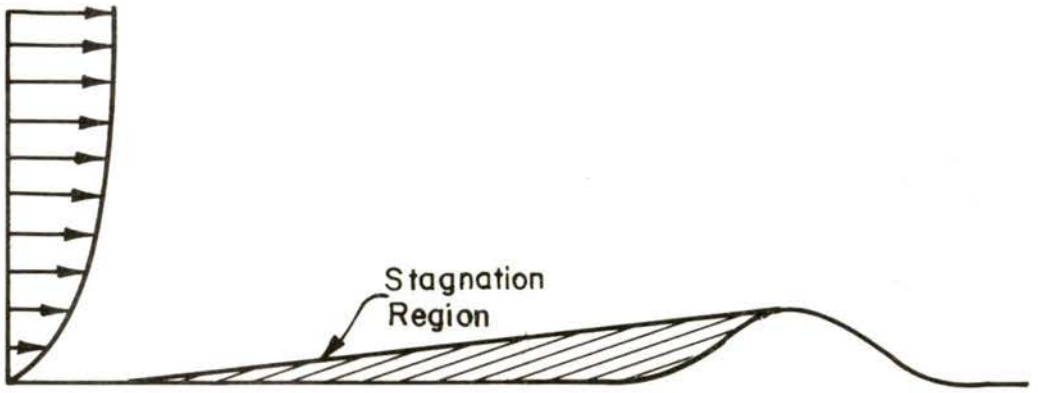
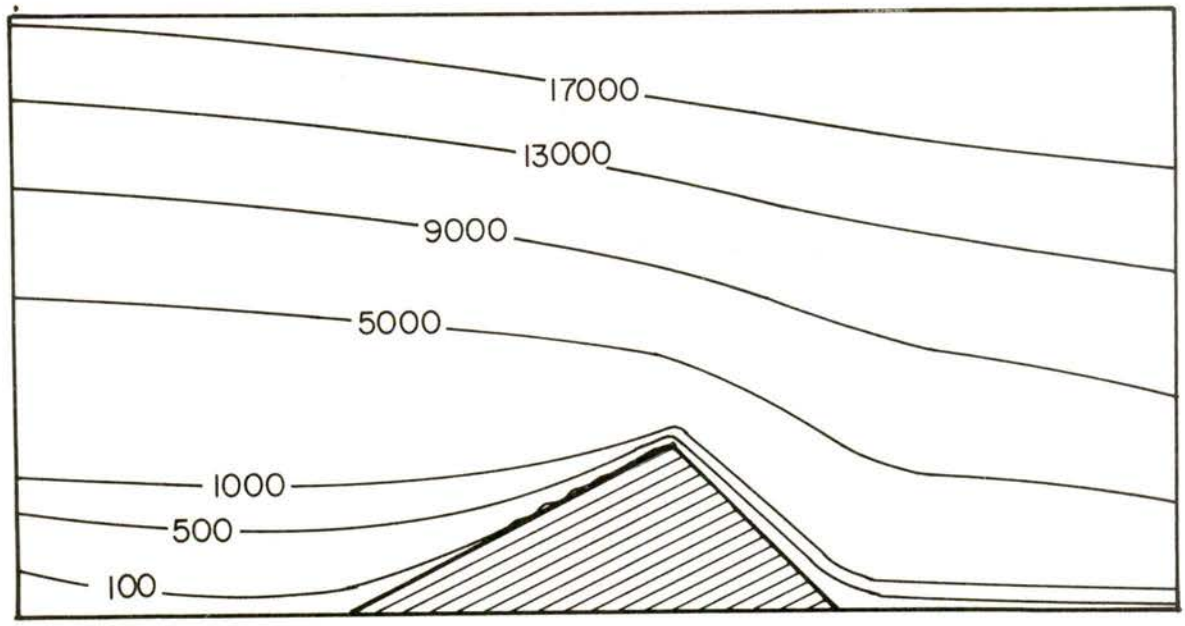
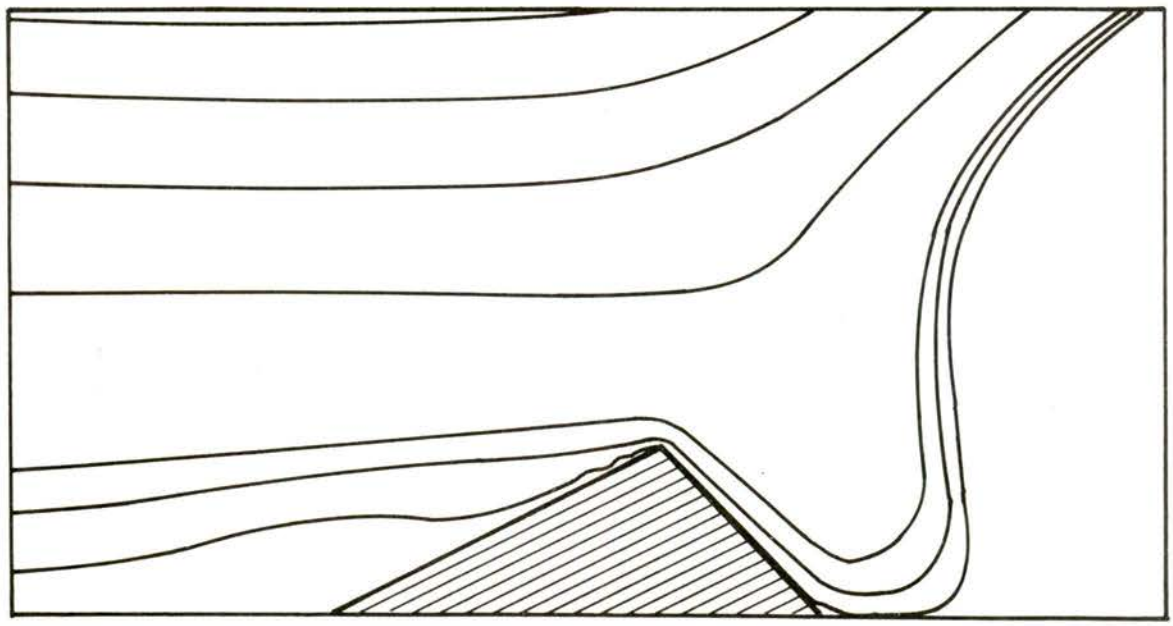


Figure 4.1 Stagnation region on windward side of a mountain obstacle for a stably stratified atmosphere.

Figure 4.2 The stream function using the downstream condition $\frac{\partial^2 \psi}{\partial x^2} = 0$; the horizontal extent is 7 kilometers which was then increased to 11 kilometers, the depth is 3.5 kilometers.

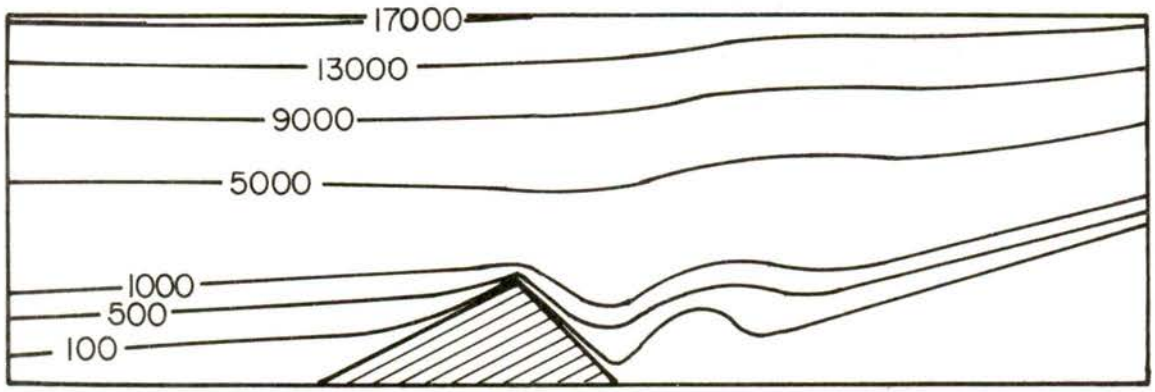


After 250 time steps.



After 330 time steps.

After 250 time steps.



After 290 time steps.

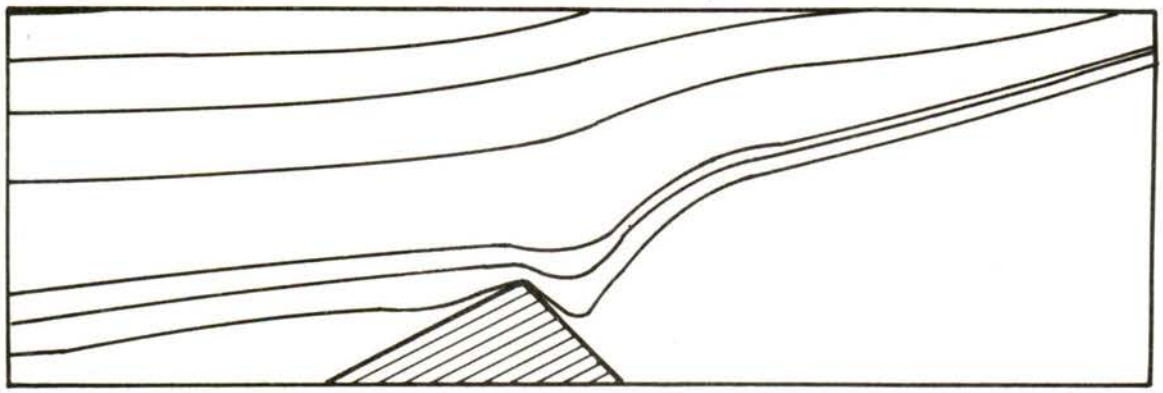


Figure 4.3 The stream function using the downstream condition $\frac{\partial^2 \psi}{\partial x^2} = 0$; the horizontal extent has been increased to 11 kilometers, depth is 3.5 kilometers.

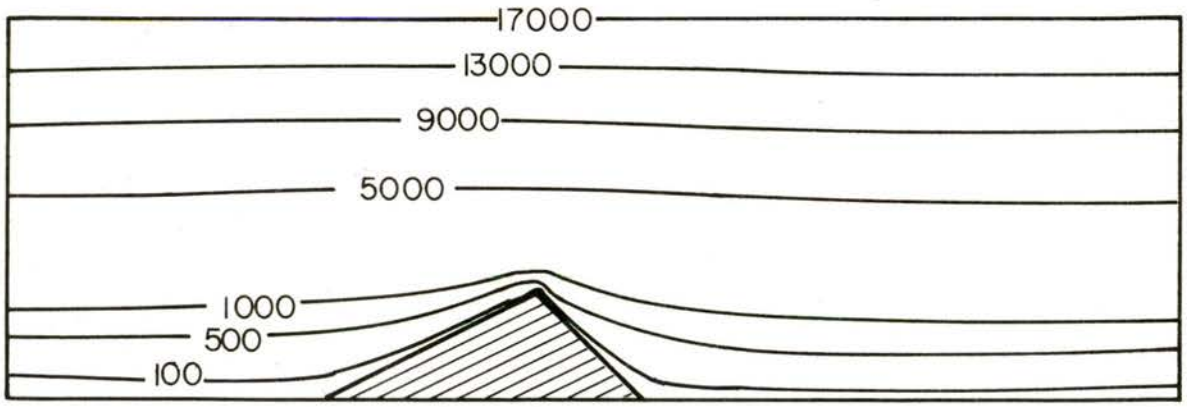


Figure 4.4 The stream function for a homogeneous atmosphere using Roache's downstream boundary condition of $\frac{\partial^2 \psi}{\partial x^2} = 0$; after 300 time steps.

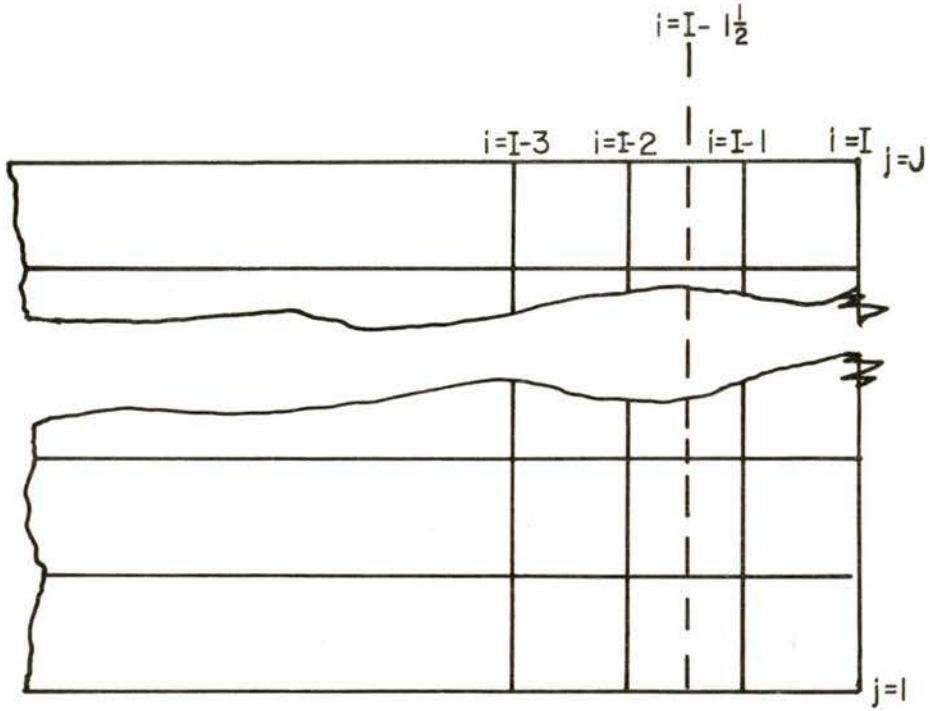


Figure 4.5 Detail of downstream boundary for explaining boundary condition on ψ ; $\left(\frac{\partial^3 \psi}{\partial x^3}\right)_{I-1\ 1/2} = 0$.

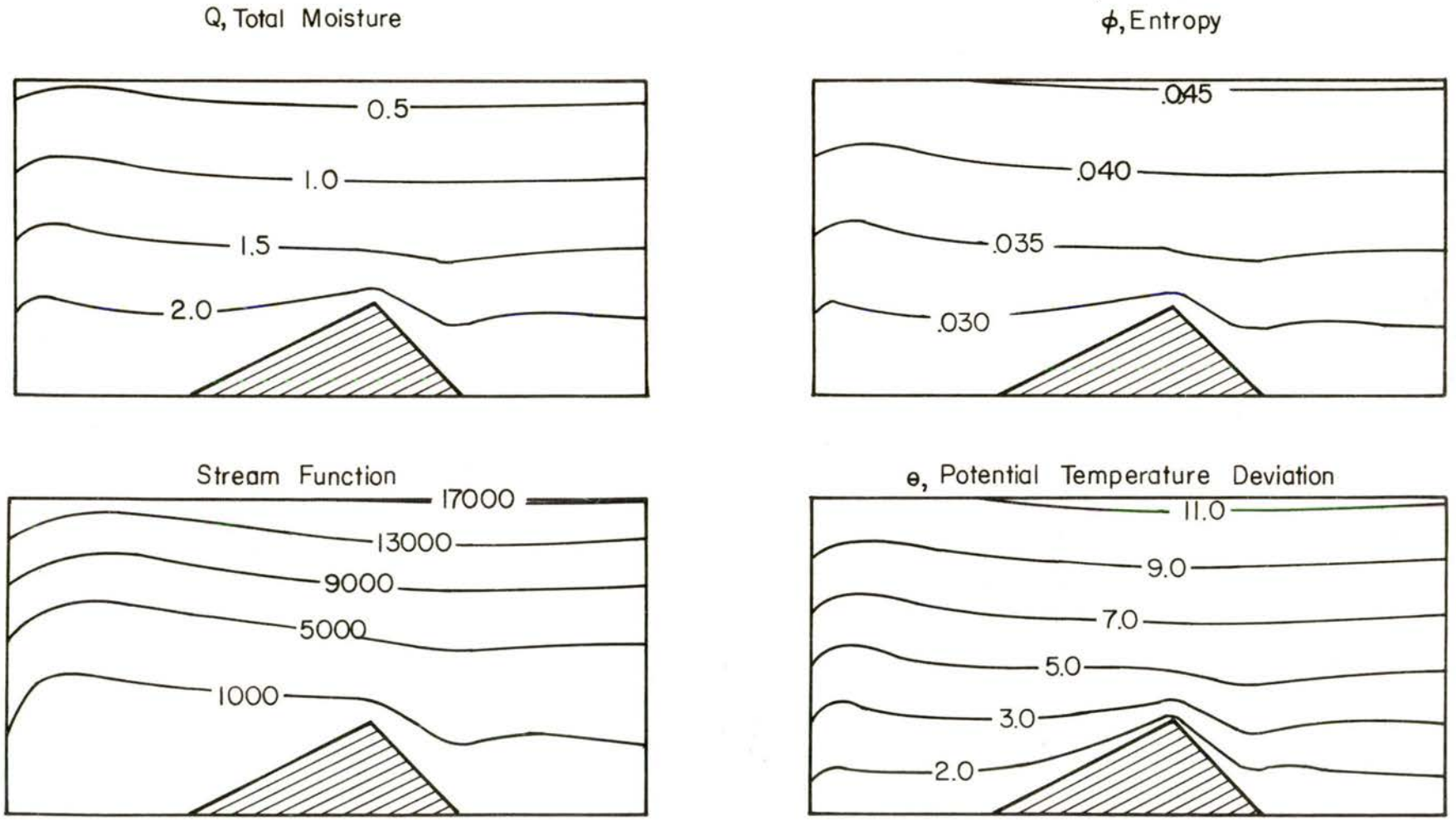


Figure 4.6 The results of specifying the thermodynamic variables as invariant in time at the inflow boundary; the horizontal extent is 7 kilometers, depth is 3.5 kilometers; represents 100 time steps for all variables.

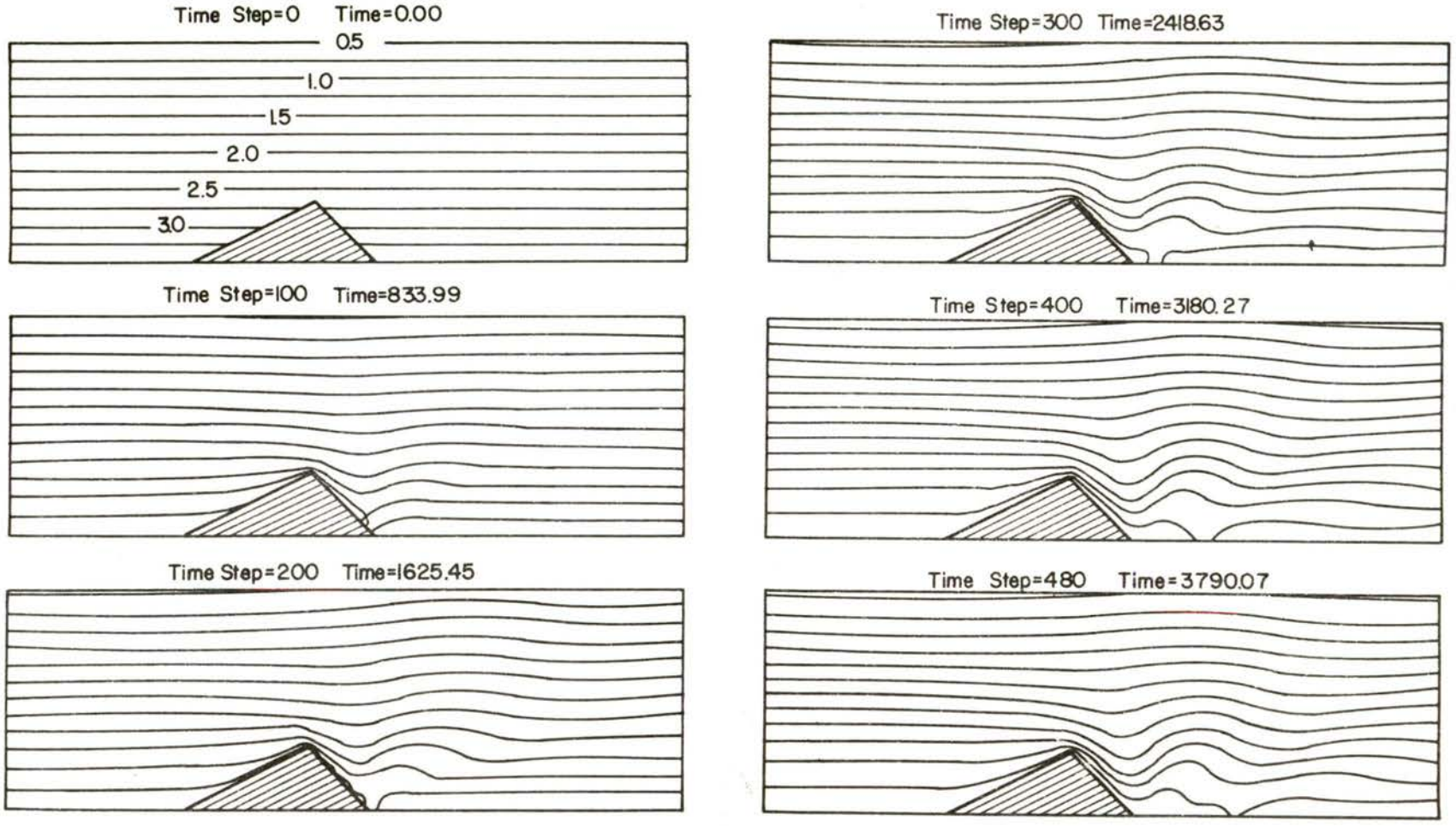


Figure 5.1 Neumann case; the evolution of the total moisture field, Q . The isohumes represent grams per kilogram, moisture to dry air. Time in real seconds.

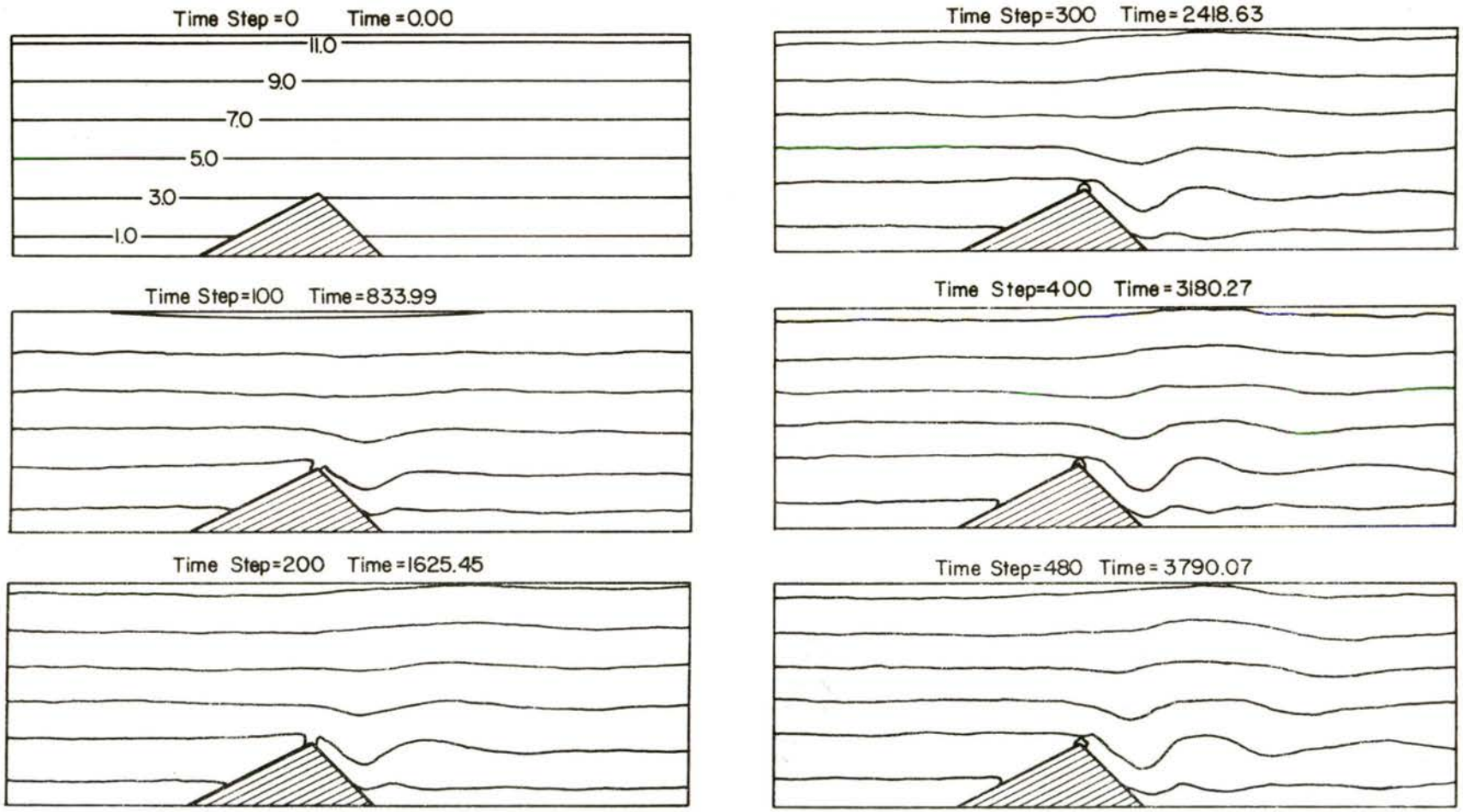


Figure 5.2 Neumann case; the evolution of the potential temperature deviation, θ . Isotherms correspond to degrees Kelvin. Time in real seconds.

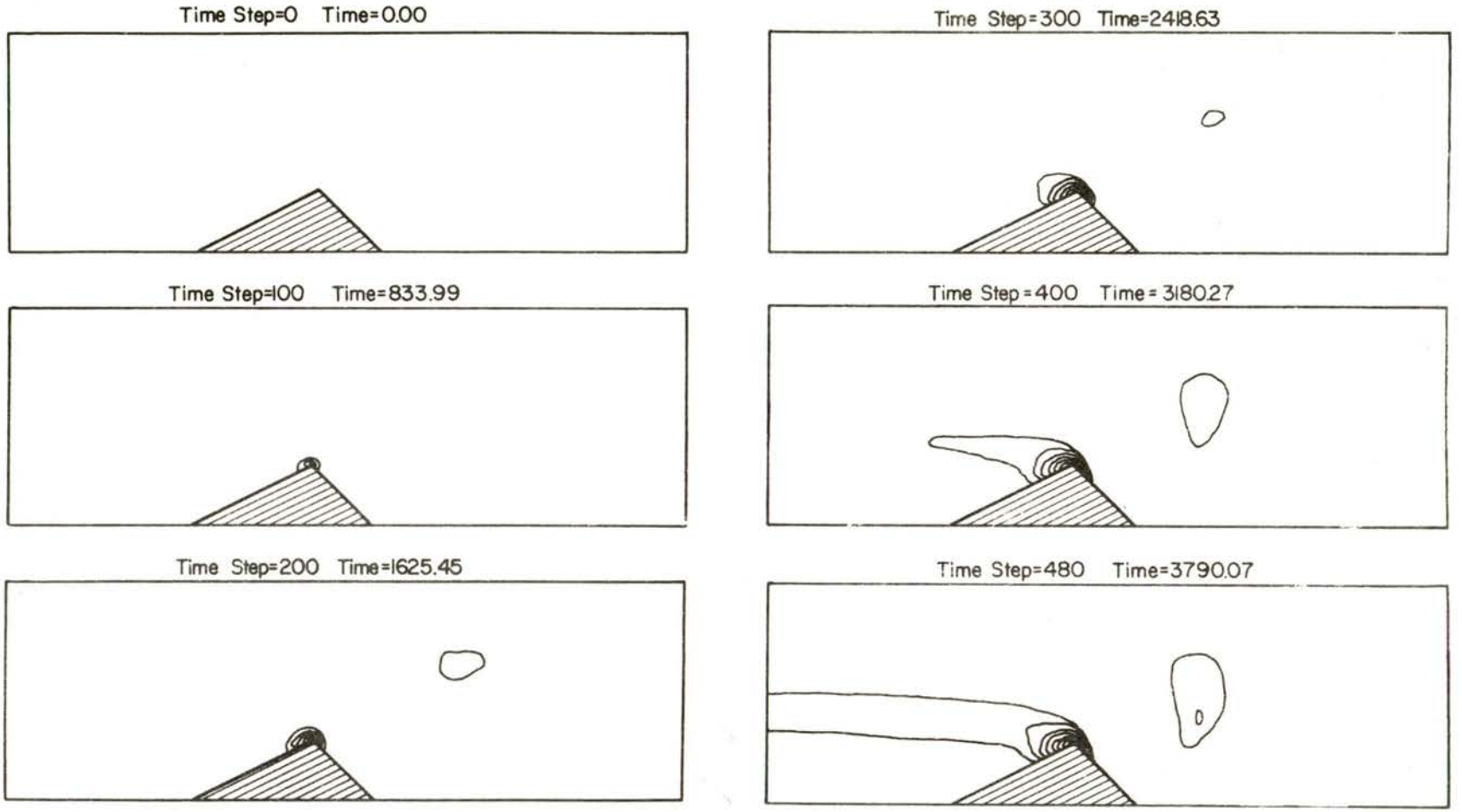


Figure 5.3 Neumann case; the evolution of liquid (cloud) water, ω_ℓ . The threshold value for plotting is 0.05 grams per kilogram, moisture to dry air; the maximum value of ω_ℓ reaches 0.468 grams per kilogram. Time in real seconds.

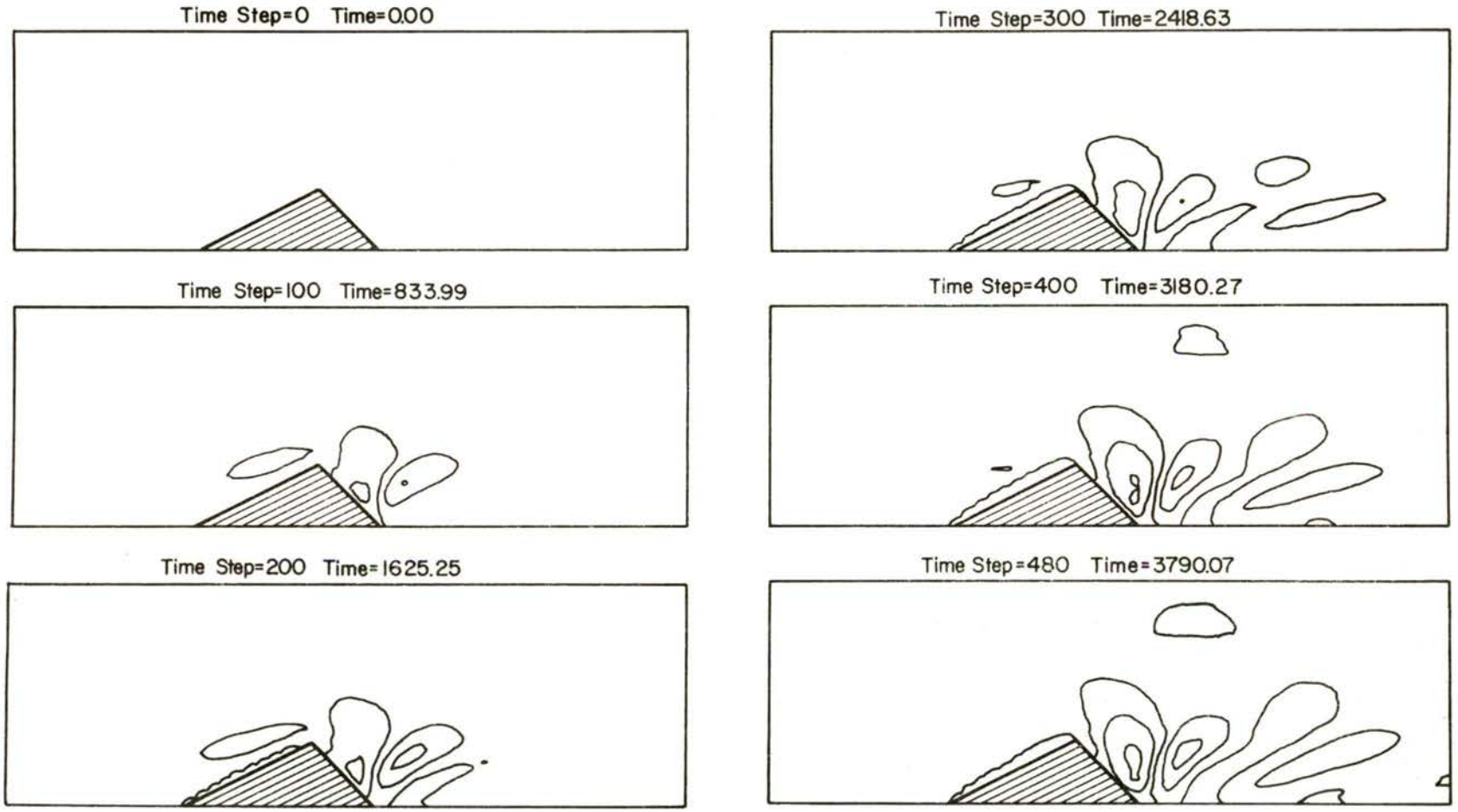


Figure 5.4 Neumann case; the evolution of the vorticity field, η. Values range from $-0.0200\ \text{sec}^{-1}$ to $0.0107\ \text{sec}^{-1}$. Time in real seconds.

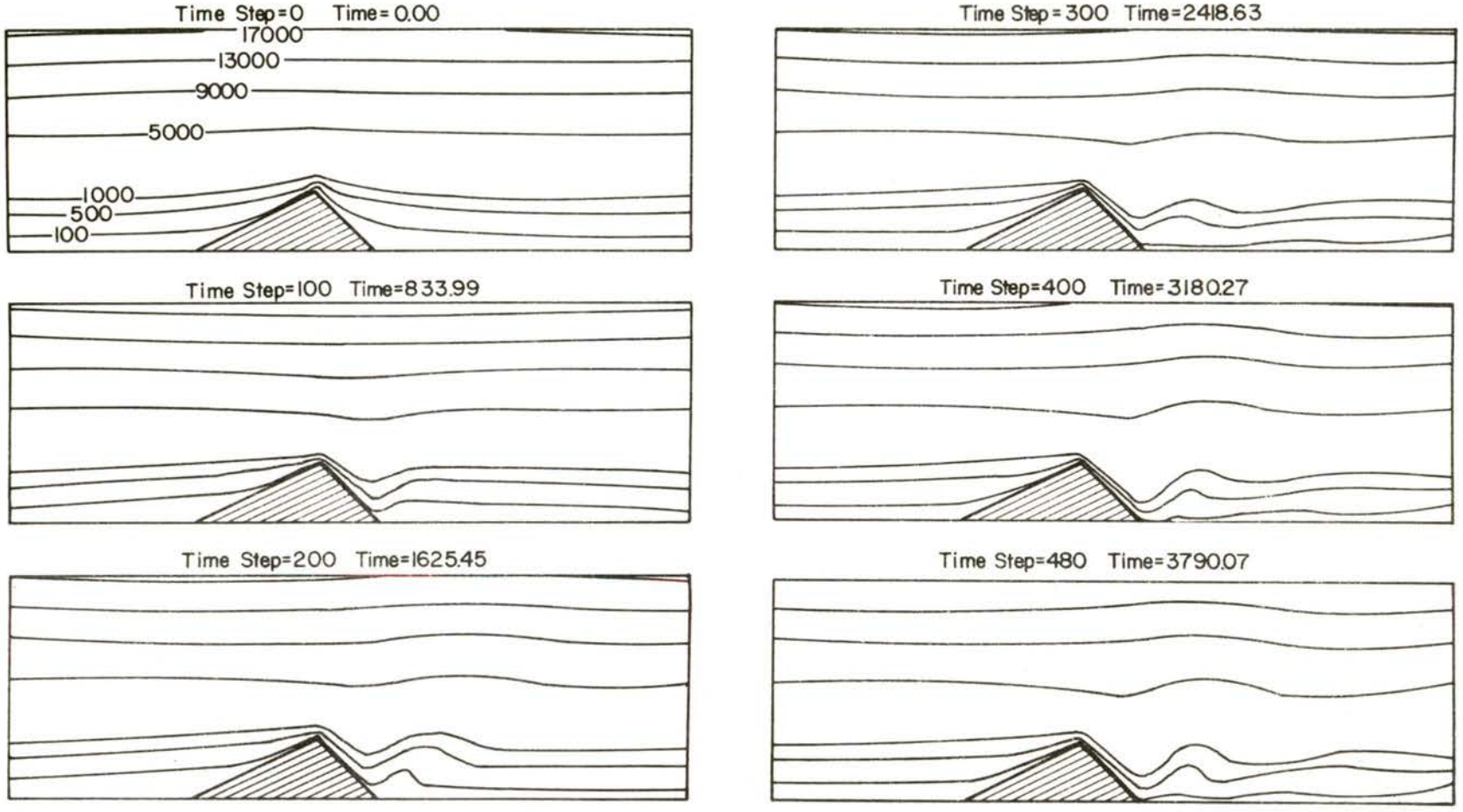


Figure 5.5 Neumann case; the evolution of the stream function, ψ . Dimensions are $m^2\ sec^{-1}$. Time in real seconds.

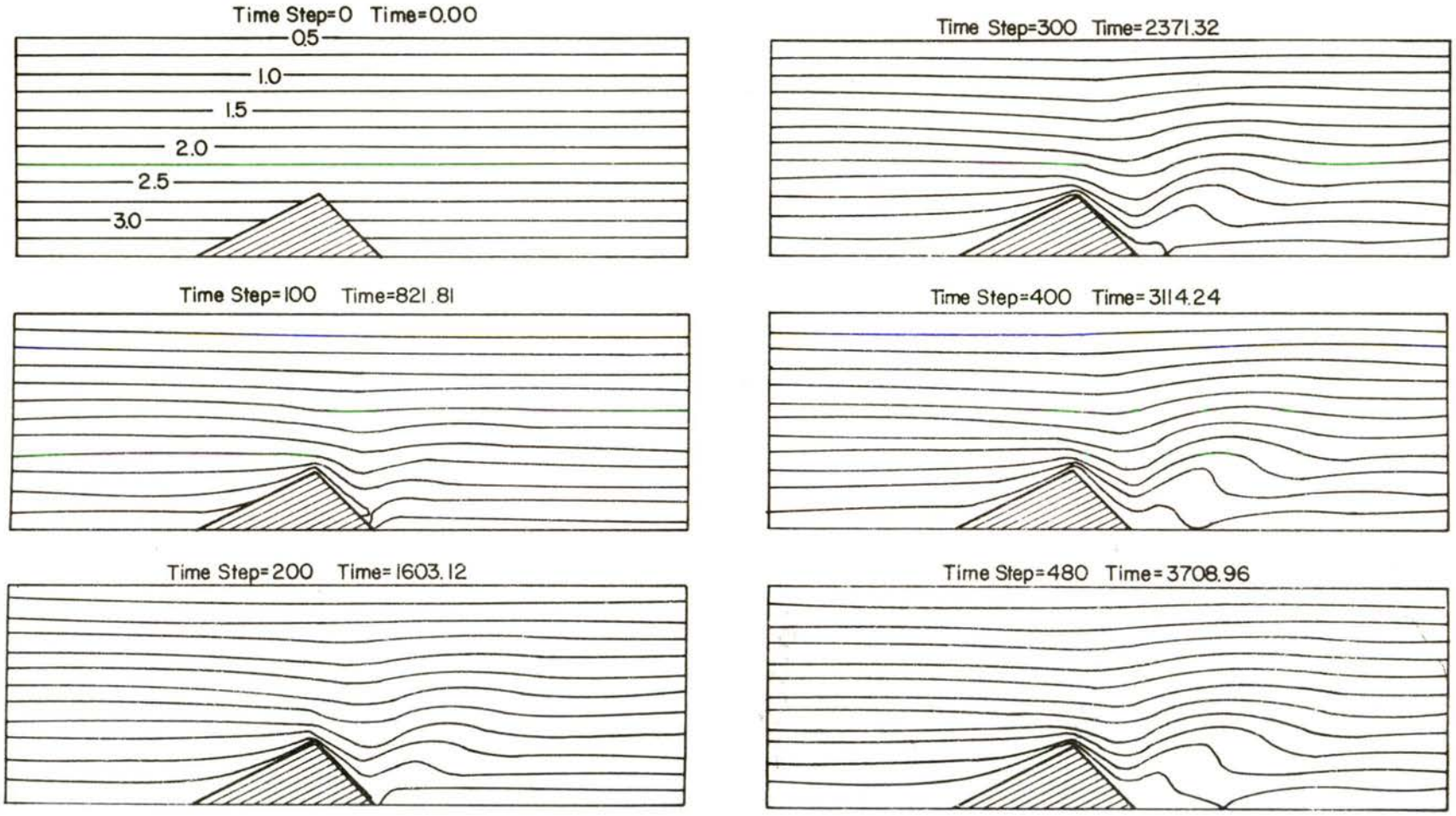


Figure 5.6 Dirichlet case; the evolution of the total moisture field, Q . The isohumes represent grams per kilogram, moisture to dry air. Time in real seconds.

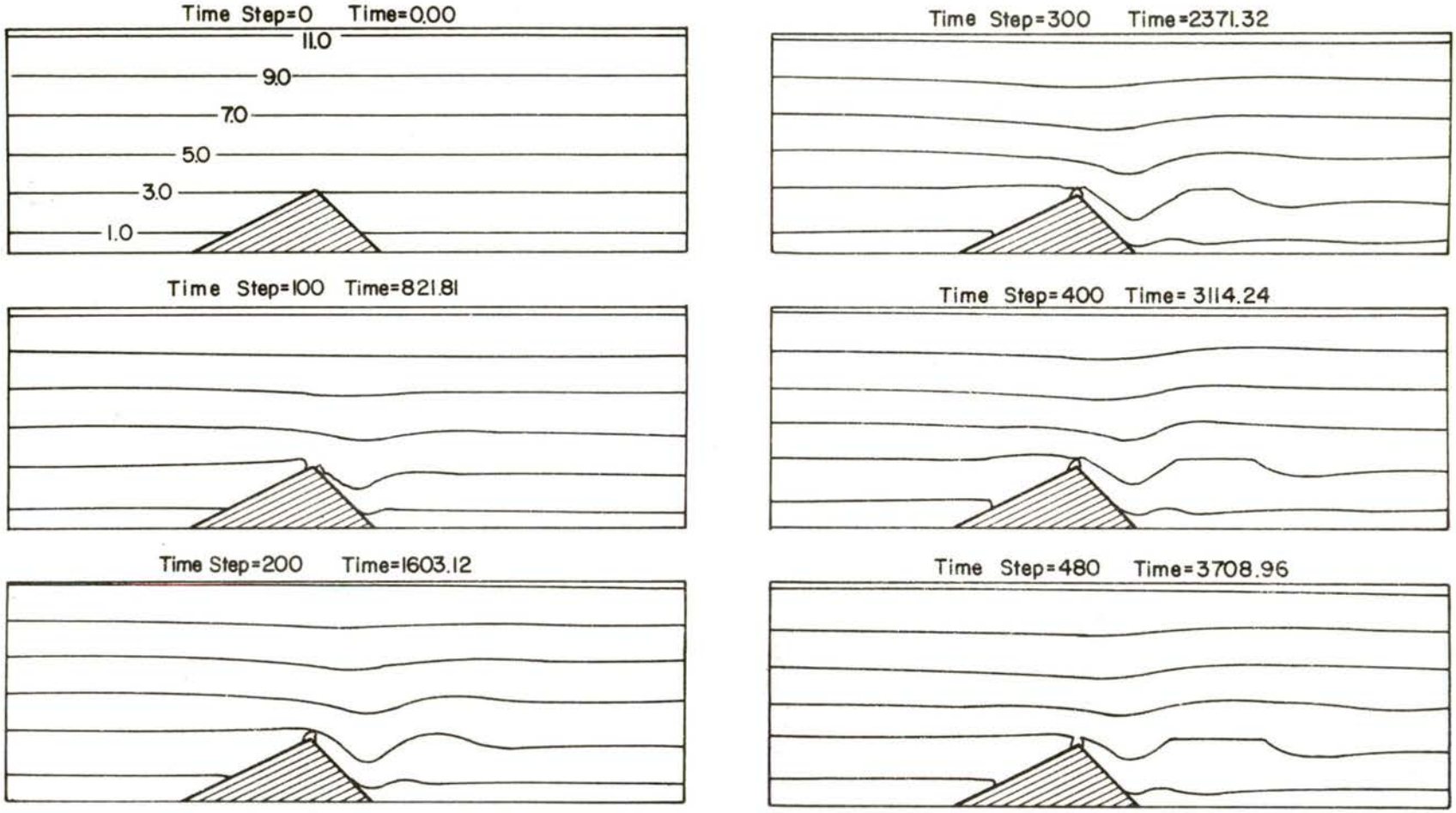


Figure 5.7 Dirichlet case; the evolution of the potential temperature deviation, θ . Isotherms correspond to degrees Kelvin. Time in real seconds.

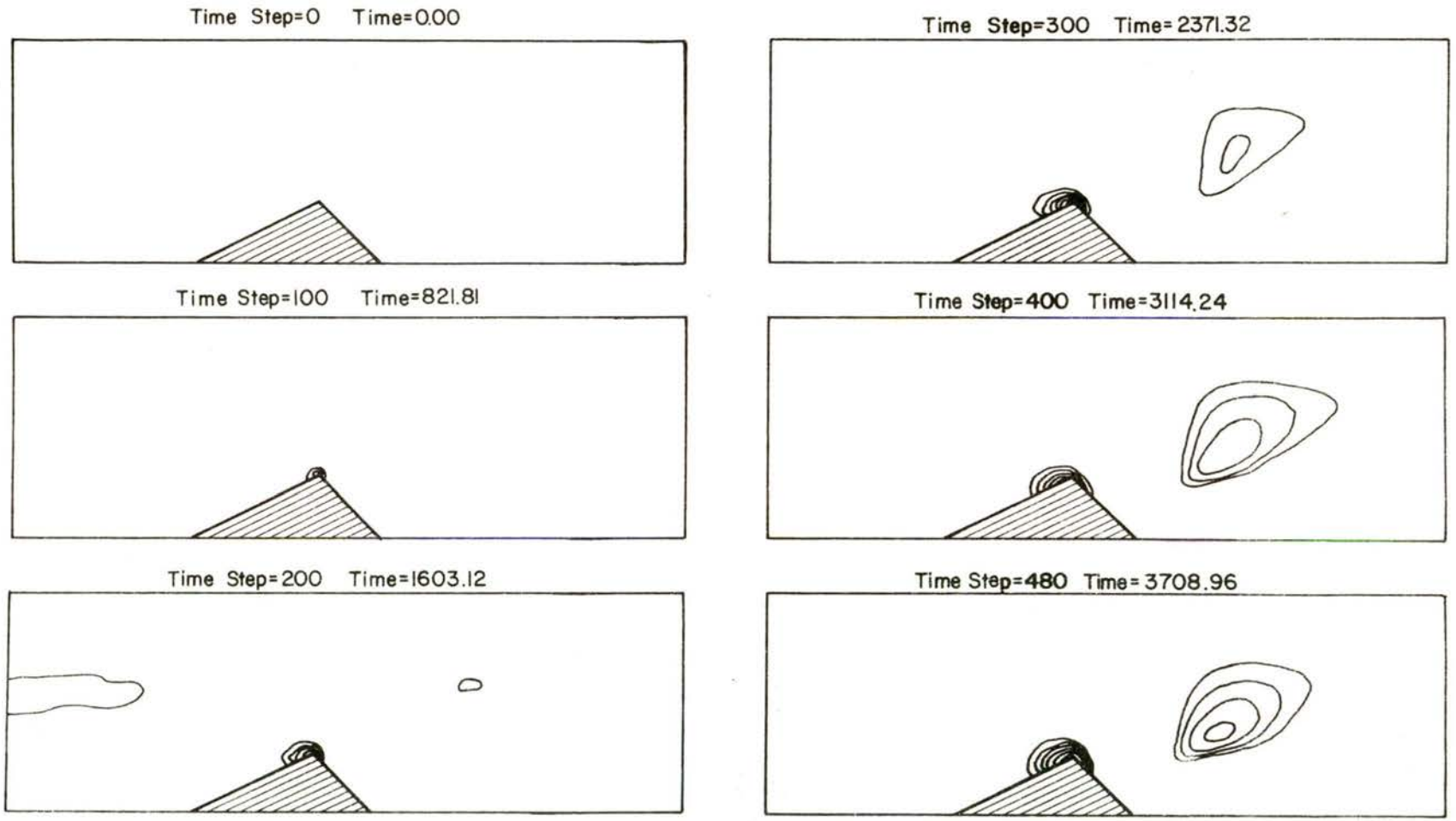


Figure 5.8 Dirichlet case; the evolution of liquid (cloud) water, ω_ℓ . The threshold value for plotting is 0.05 grams per kilogram, moisture to dry air; the maximum value of ω_ℓ reaches 0.474 grams per kilogram. Time in real seconds.

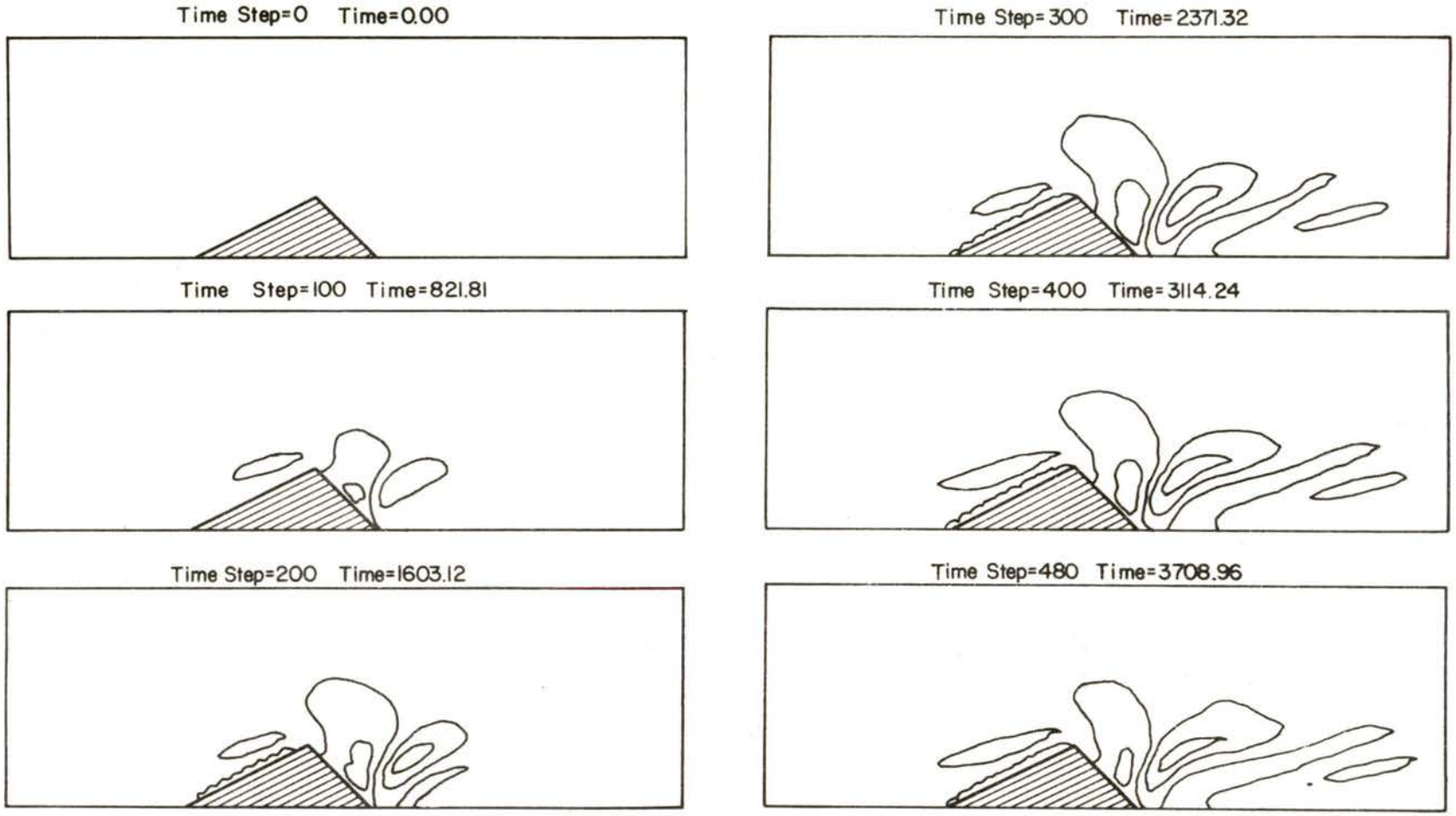


Figure 5.9 Dirichlet case; the evolution of the vorticity field, η. Values range from $-0.0127\ \text{sec}^{-1}$ to $0.0075\ \text{sec}^{-1}$. Time in real seconds.

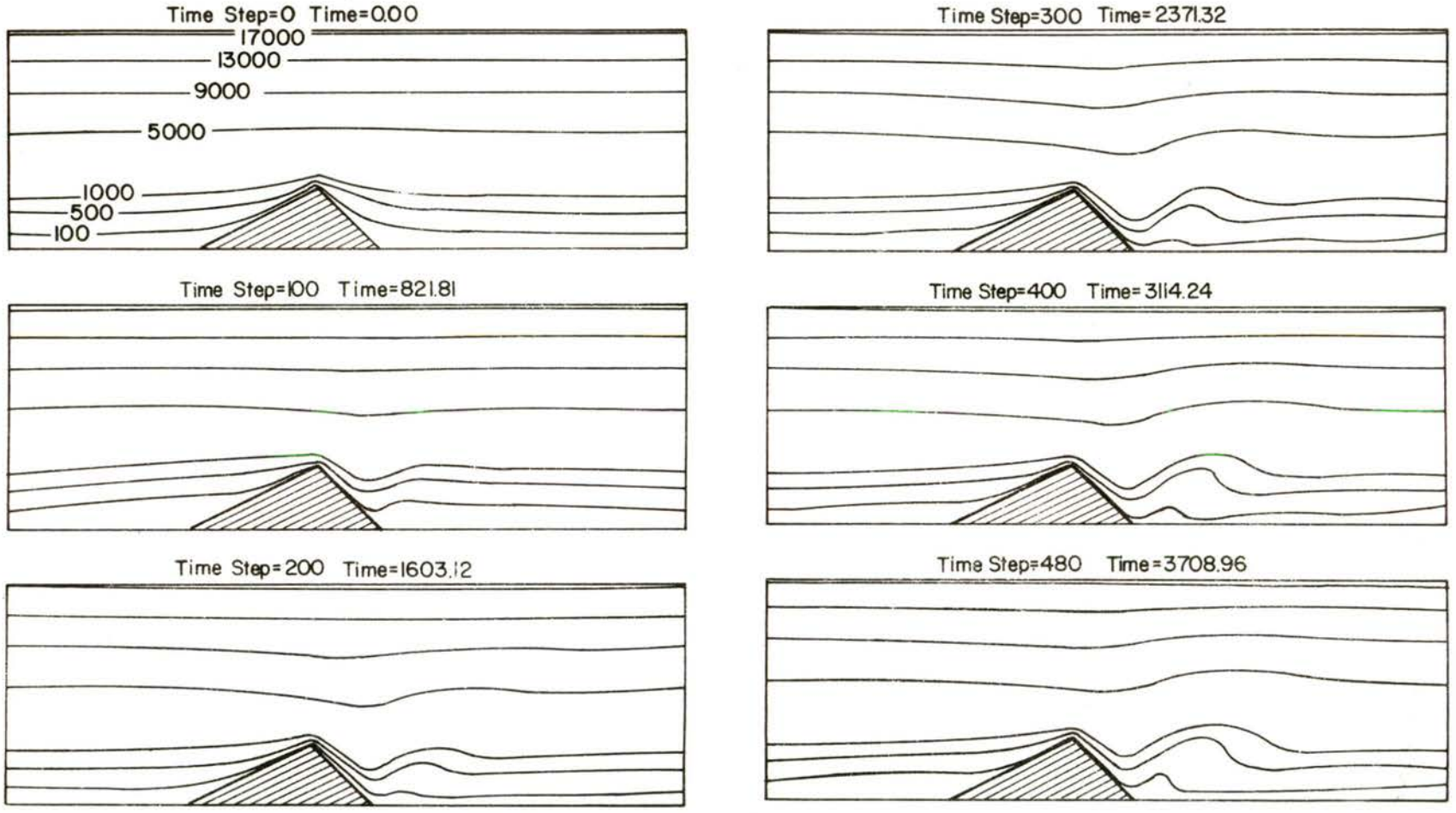


Figure 5.10 Dirichlet case; the evolution of the stream function, ψ . Dimensions are $m^2\ sec^{-1}$. Time in real seconds.

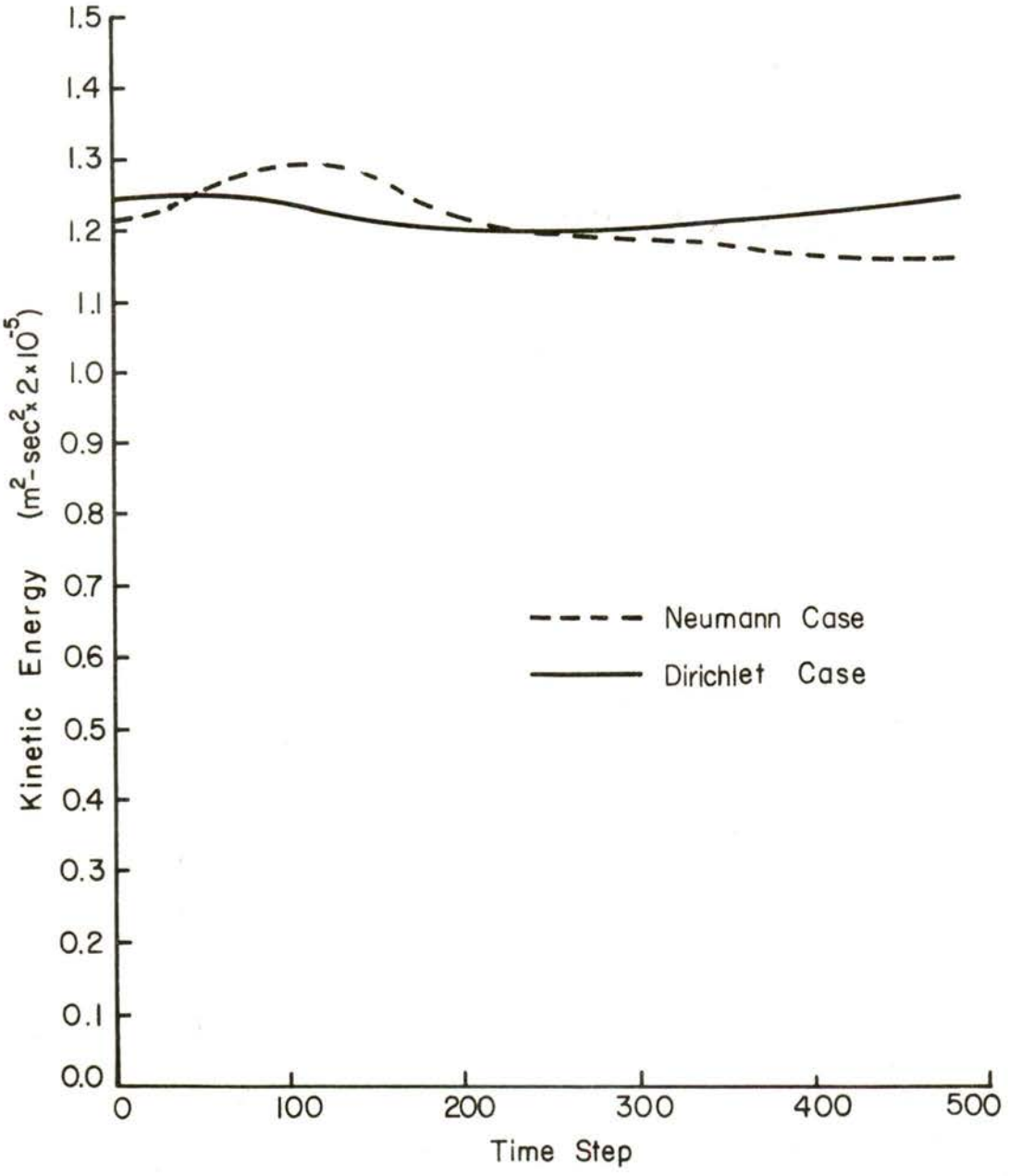


Figure 5.11 Comparison of kinetic energies for the Neumann and Dirichlet cases.

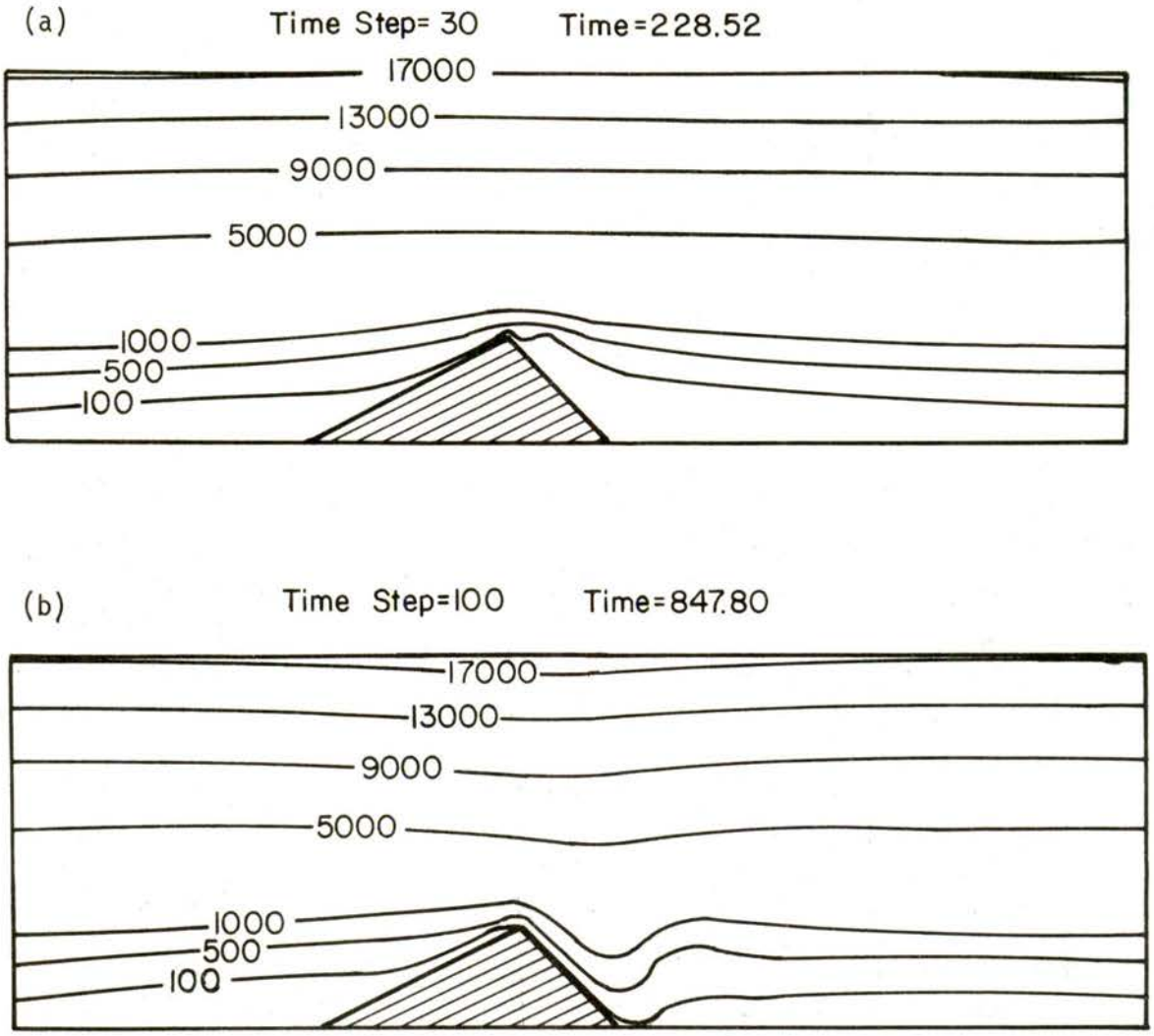


Figure 5.12 The stream function for the case where K is finite differenced. (a) after 30 time steps with $\eta = 0$ at lower boundary; (b) after 100 time steps with $\frac{\partial^2 \eta}{\partial n^2} = 0$ at the lower boundary. Time in real seconds.