NUMERICAL MODELLING OF MOIST CONVECTION IN TWO AND THREE SPACE-DIMENSIONS

by

Martin James Miller
Department of Meteorology
Imperial College of Science and Technology

A Thesis submitted for the
Degree of Doctor of Philosophy
in the University of London

1972

ABSTRACT

A primitive equation model for the numerical simulation of cumulonimbus convection is described in detail. The model is fully three-dimensional in space and is based on an (x,y,p) coordinate system. An analysis of the non-hydrostatic primitive equation set in this coordinate system is carried out and the implications of certain simplifications to the equations examined. In particular, sound waves are eliminated from the system while leaving the internal gravity waves undistorted.

The heights of pressure surfaces are computed from a form of 'balance'-equation. The use of gridpoints on constant-pressure surfaces eliminates a particular modelling problem which arises in the computation of condensation and evaporation processes when height is taken as vertical coordinate.

The model is formulated with open lateral boundaries allowing simulations in any specified initial wind field. Aspects of the lateral boundary conditions are therefore considered in detail particularly with respect to gravity wave propagation.

The cloud microphysical processes are incorporated in which the conversion of cloudwater to rainwater, accretion of raindrops and the evaporation of raindrops are parameterised in terms of the larger-scale variables.

Numerical results are presented and discussed. These results take the form of simulations of the lifecycle of a single cumulonimbus cloud in unsheared and sheared environments.

An encouraging feature is the sensitivity of the model to the simulated microphysics. Features of the sheared results indicate lines along which work with the model should be pursued both as severe storm studies and in connection with the parameterisation of deep convection in synoptic scale models.

LIST OF CONTENTS

page

CHAPTER I :- INTRODUCTION

1.1	Introduction	5
1.2	Theoretical models of convection	6
1.3	Numerical models	7
1.4	Vorticity and primitive equation models	8
1.5	The choice of vertical coordinate	9
1.6	Practical limitations	10
1.7	The background to the model	11

CHAPTER II :- THE BASIC EQUATIONS AND LINEARISED ANALYSIS

2.1	Model equations	14
2.2	Computational considerations	17
2.3	Analysis of motions of infinitesimally small amplitude in the primitive equation system used in the model	22
2.4	The implications of various approximations	28
2.5	Boundaries	29

CHAPTER III:- THE NON-HYDROSTATIC FORMULATION OF THE MODEL

3.1	Analysis of momentum equation	31
3.2	Importance of non-hydrostatic pressure	32
3.3	Vertical momentum equation	34
3.4	Computation of complete deviation height field h'	36
3.5	Balance equation for the height field	36
3.6	Numerical solution of the balance-type equation	39

CHAPTER IV :- BOUNDARY CONDITIONS

4.1	Boundary conditions	42
4.2	Boundary formulation	43
4.3	Specification of normal velocity v_n	44
4.4	Computation of the horizontal divergence D	45
4.5	Computation of boundary divergence	47
4.6	Computation of boundary velocity components	48
4.7	The computation of the isobaric heights h' on the boundary	49
4.8	Computational instability associated with the non-linear terms on the boundary	49
4.9	The computation of the temperature θ' on the lateral boundaries	49
4.10	An alternative boundary formulation	50
4.11	Initial tests of the model	51
4.12	The dry model	52
4.13	Energy transmission associated with gravity wave propagation	52
4.14	Closed boundaries	53
4.15	Open boundaries	56

page

CHAPTER V :- THERMODYNAMICS AND CLOUD PHYSICS

5.1 Condensation and evaporation 57
5.2 The adjustment procedure 58
5.3 Accuracy 59
5.4 Computation of $e_s(T)$ 61
5.5 Cloud physics modelling 63
5.6 Parameterisations 64
5.7 The production term PROD 65
5.8 Values of α, β, γ and l_{crit} 67
5.9 Rainfall 67
5.10 Summary of the cloud physics 68

CHAPTER VI :- INITIALISATION AND DISCUSSION OF SIMULATION RESULTS

6.1 Initialisation 69
6.2 Interpretation of results 74
6.3 Diagrams (explanation of) 74
6.4 Simulation runs 75
6.5 Description of RUN 1 76
6.6 Importance of cloud physics parameterisation 77
6.7 RUN 2 78
6.8 RUN 3 79
6.9 RUN 4 83

DIAGRAMS FOR RUN 1 84

DIAGRAMS FOR RUN 2 103

DIAGRAMS FOR RUN 3 118

CHAPTER VII:- SUMMARY AND CONCLUDING REMARKS 135

APPENDIX 1 :- THE NUMERICAL COMPUTATION SCHEME 139

APPENDIX 2 :- LISTING OF COMPUTER PROGRAM 144

REFERENCES 193

ACKNOWLEDGEMENTS 196

CHAPTER I

1.1 INTRODUCTION

The observed structure of the atmosphere is determined by the solar radiation input and the earth's rotation; exchanges of heat, moisture and momentum occur on all space-scales from the global to the molecular. When these exchanges involve motions arising directly from the conversion of available potential energy then the motion is termed convection. These convective motions occur on all scales and a detailed knowledge of the structure and processes involved in motions on any particular scale is necessary for the full understanding of all other scales of motion with which it interacts.

The rapid development of large, high-speed digital computers has enabled numerical models to be developed in which the meso, synoptic and global-scale features of the atmosphere are simulated. Scales of motion smaller than meso-scale, the so-called sub-grid scales have to be parameterised in terms of the larger-scale variables; this parameterisation of smaller scales is necessary if the maintenance and development of the larger-scale systems is to be properly represented and requires knowledge of the structure and physical characteristics of these smaller scales.

A principal sub-grid scale phenomenon which requires parameterising is that of cumulonimbus or 'deep' convection; cb convection is the predominant weather regime in tropical latitudes in particular and represents exchanges of mass, heat, moisture and momentum from the surface layers to the vicinity of the tropopause.

Although qualitative studies of cumulonimbus clouds were pieced together from ground observations prior to World War II,

it was the advent of more aircraft and sensors such as radar that enabled more quantitative studies to be made. Systematic studies of thunderstorms were made by Byers and Braham (Thunderstorm Project, 1949), Newton (1959) and Ludlam and Browning (1962). These workers all formulated models in which the concepts of time and space evolution and of multi-cell structure were introduced; they also stressed the importance of organised downdrafts in determining the structure of the storm. Propagation speeds and directions were also examined. This first appreciation of the variety of structural detail in convection led to more detailed quantitative observations of cumulonimbus and cumulus clouds.

1.2 THEORETICAL MODELS OF CONVECTION

There are many practical limitations to detailed observation of cumulus and cumulonimbus convection. As a complementary approach, mathematical models of these phenomena have been constructed, one of the first to be published being that of Malkus and Witt (1959). These models have been both analytical and numerical. The complexity of the problem makes the analytic approach very difficult, and substantial assumptions need to be made in order to progress; on the other hand, the solution of the numerical initial-value problem also has its computational difficulties resulting from numerical approximation and solution.

With the assumption of steady, two-dimensional inviscid flow, (of Richardson number $\sim$ unity), finite amplitude convective overturning in shear consistent with Ludlam and Browning's (1962) observational model, was studied by Green and Pearce (1962) and more recently by Moncrieff (1972). Green and Pearce examined the dynamical feasibility of Ludlam and Browning's model

under the assumptions given. Moncrieff extended the work to derive transfer properties of this model in terms of large-scale parameters as is required for convective parameterisations in large-scale models. This type of study is also applicable to some frontal motions but its assumptions make the immediate application difficult and a more general approach is desirable.

1.3 NUMERICAL MODELS

Many numerical models of thermal convection have been developed since the first experiments of Malkus and Witt (1959). The earlier models were shallow and two-dimensional, either of dry thermals in a neutral environment or of moist convection in which only the latent heat release associated with the condensation process was modelled; notable amongst these are the models of Ogura (1962, 1963), Lilly (1962), Murray and Anderson (1965), Nickerson (1965) and Orville (1965). These early models were either in rectangular Cartesian coordinates (x,z) —(i.e. with no y-dependence) or axially-symmetric cylindrical coordinates (r,z) —(i.e. with no azimuthal dependence). Ogura endeavoured to compare models in both coordinate systems. Murray (1970) has done a similar, more thorough, comparison with interesting results particularly with respect to the intensities of the convection in the two coordinate systems.

Also among the earlier models was one of a squall-line, the work of Ogura and Charney (1962). This was unique in its use of the primitive equations rather than a vorticity equation.

Many problems of a computational nature were encountered in solving the time-dependent p.d.e's of these early models although problems connected with lateral boundaries were for the most part avoided by the "closed box" assumption, i.e. no

flow was allowed through the boundaries. This of course precluded the use of non-zero ambient windfields.

The last few years have seen research work directed mostly towards the sophistication of existing models, and their extension to the full depth of the troposphere with cloud physics parameterisations; but it is felt by the author that the basic inadequacies of the existing models have made some of the more complex additions redundant. Reference to these most recent models is made in Chapter V.

1.4 VORTICITY AND PRIMITIVE EQUATION MODELS

The basis of a numerical model of the whole atmosphere or of any individual atmospheric phenomenon is the system of equations describing the physics. The choice of equations and the simplifications used are governed in part by knowledge of observed characteristics of the phenomenon under investigation. Further restrictions are sometimes imposed by the modeller's ability to obtain solutions within the desired conditions.

The basic equations are those defining conservation of mass, momentum, energy and water substance. Convective models have so far all been limited to two-dimensions, they also have in common the use of the Boussinesq or anelastic equation set, (in which density changes only appear in association with gravity). A vorticity equation can readily be derived, and, with the assumption of two-dimensionality, a stream-function (ψ) can be defined; this together with the vorticity and other conservation equations forms a comparatively simple modelling system in two dimensions. Using this system a Poisson-type equation is solved for ψ. If the pressure field is required explicitly, a second Poisson-type equation then has to be solved.

[The pressure could alternatively be obtained by integrating a Bernouilli equation along each streamline using Lagrangian techniques.]

The extension of such a vorticity system into three-dimensions with reasonable vertical resolution becomes an almost impossible task and the alternative approach using a model based on the primitive equations must be pursued.

Vorticity models were at first more extensively developed than primitive equation models although such primitive systems were considered [see Charney (1955) and Hinklemann (1959), models developed for the general circulation e.g. Smagorinsky (1958, 1963) and synoptic forecast models e.g. Shuman (1962, 1968), Bushby and Timpson (1967)].

The use of primitive equations on smaller convective scales is minimal, the previously-mentioned model of Ogura and Charney being noteworthy. They carried out a frequency analysis of the perturbation equations and considered ways of simplifying the set in a practical and consistent fashion. This analysis was done in height coordinates although the primitive equations are more often used in pressure coordinate form on the synoptic-scale.

1.5 THE CHOICE OF VERTICAL COORDINATE

The choice of height or pressure as a vertical coordinate is determined in part by the nature of the study being undertaken. The standard use of pressure coordinates (pressure surface representation) on the large-scale is prompted to some extent by the neatness of handling the air density in the momentum and continuity equations, (although the availability of data at pressure levels is the main reason).

For theoretical work height coordinates are used almost exclusively, forming a true Cartesian set; by suitable manipulation they can be used to handle the density conveniently also. The mathematical imperfections of pressure coordinates are however small and rarely introduce approximations that are significant. (More is said about this in Chapter II, §2.3.)

Convection models have, without exception, been based on a height coordinate system to allow the stream function/vorticity equation formulation, (although a stream function formulation can be used in pressure coordinates with approximations).

This use of gridpoints fixed in space introduces one substantial disadvantage when modelling deep convection. Serious inconsistencies occur in computing the condensation rate of vapour to liquid water if the departures of pressure (both hydrostatic and non-hydrostatic) are not taken into account; however to do so introduces highly complex, implicit relationships which have not been adequately handled by modellers to date. The use of gridpoints in pressure-surfaces circumvents this problem very conveniently. This is discussed in Chapter V, §5.4.

Pressure is used as vertical coordinate in the model described here. The novelty of the use of p-coordinates in a non-hydrostatic system however is reflected in the need to derive consistent vertical momentum and balance equations, and this forms the basis of Chapter III. Also a frequency analysis has been carried out on the pressure coordinate system (see Chapter II, §2.3).

1.6 PRACTICAL LIMITATIONS

An inescapable modelling problem arises from the limitations of available computers; the detail of the physical processes,

the overall dimensions of the space-grid and the resolution and accuracy are all subject to this restriction. At the time of writing it is not practicable to model cloud scale and microscale processes in detail over a three-dimensional grid with high resolution. This statement requires some qualification, for by programming a model to process a limited number of gridpoints in the central memory with the aid of backing storage it might possibly be feasible albeit very time-consuming.

The problems of presenting the results of three-dimensional models such as the one forming the basis of this thesis are considerable, and while only modest efforts have been made here, the future must lie in digigraphic presentation and 3-D perspective plotting methods.

1.7 THE BACKGROUND TO THE MODEL

The aim of the project described here was to design a fully three-dimensional model to simulate 'moist overturning flows'. An individual cumulonimbus cell is therefore simulated including moist processes.

The model must be large enough to contain all stages of the cloud's life-cycle i.e. boundary effects such as impingement of the cloud on an upper boundary must not interfere. As a result, the model was designed to span a depth of atmosphere greater than the depth of the tropical troposphere — up to 100 mb. In view of the computer limitations referred to earlier, the horizontal area was limited to 15 km x 15 km. While this is sufficient to contain all but the very largest clouds it certainly will not contain all the accompanying motion field. This areal limitation was with a horizontal gridlength of one kilometre, a rather coarse grid. Because of this limitation on horizontal

area, the model was designed with open lateral boundaries and the boundary conditions formulated to model in a simple manner that part of the environment outside the main updraft and the dynamic parts of the downdraft.

The vertical gridspacing was irregular, defined by seven pressure surfaces [1000, 850, 700, 500, 300, 200 and 150 mb] with the upper boundary condition applied at 100 mb.

The physical variables were computed on a staggered grid in space but not in time. (Full details are given in the Appendices.)

The deficiences introduced by the coarseness of the grid (such as large implicit smoothings due to truncations) are, however, technological in so far as they can be minimised by finer grids and improved finite-differences, and it is to the development and functioning of the model itself that the main effort has been directed.

SUMMARY OF THESIS CHAPTERS

The various modelling problems encountered and the methods by which these were handled in the model form the bulk of this thesis although the penultimate chapter demonstrates the model's ability to fulfil its ultimate purpose.

This first chapter has attempted to give a background to the model including its uniqueness within the convective simulation field of research.

Chapter II lays down the basis of the model and, in view of the novel use of pressure coordinates, a linearised frequency analysis of the equations of motion in pressure coordinates is performed and used as a guide to a consistent,

simplified modelling system.

Chapter III examines the non-hydrostatic formulation in this coordinate system and derives a form of balance equation for the height field computation.

Chapter IV contains a discussion of lateral boundary conditions including the model formulation and examines the behaviour of internal gravity oscillations in the model and their propagation of energy.

Chapter V gives details of all the moist processes modelled i.e. latent heating and cooling and the cloud physics parameterisations

Chapter VI contains a discussion of some preliminary results obtained with the model. These are for convection in both unsheared and sheared environments.

A closing summary, comments on future work and appendices containing details of the computational scheme and the computer program (for reference purposes) complete the thesis.

CHAPTER II

THE BASIC EQUATIONS AND LINEARISED ANALYSIS

This chapter lists the basic equations and contains a linearised analysis of these equations in pressure coordinate form. Aspects of the computational problem are considered particularly with respect to the finite-difference scheme used and explicit and implicit smoothing techniques. Details of the finite-differencing methods used are given in the Appendix.

2.1 MODEL EQUATIONS

The primitive equation set can be written in the following form:

Horizontal momentum equations:

$$\frac{\partial u}{\partial t} = -\frac{\partial \psi}{\partial x} + v\zeta - \omega\frac{\partial u}{\partial p} \qquad \text{——(2.1.1)}$$

$$\frac{\partial v}{\partial t} = -\frac{\partial \psi}{\partial y} - u\zeta - \omega\frac{\partial v}{\partial p} \qquad \text{——(2.1.2)}$$

[from which the effects of small-scale momentum transfers, both internal and at the lower boundary, are omitted];

(u,v) are the Cartesian components of the wind field,

$$\omega = \frac{Dp}{Dt}\ ; \qquad \psi = gh' + \frac{1}{2}(u^2 + v^2)\ ; \qquad \zeta = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} + f\ ;$$

h' is the deviation of the height of a pressure surface from that in a horizontally stratified, unperturbed atmosphere.

The vertical momentum equation:

This equation takes a somewhat complex form in pressure coordinates and this form and its use are discussed in detail in Chapter III but given here (as incorporated in the balance equation) for convenience:

$$\frac{1}{g}\cdot\frac{D}{Dt}\left(\frac{\omega}{\rho_s}\right) + g^{*2}\rho\frac{\partial h}{\partial p} + g^* = 0$$

where $g^* = g(1 + l)$ and l is the liquid water concentration in gm/gm.

Mass continuity equation

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial \omega}{\partial p} = 0 \qquad \text{——(2.1.3)}$$

This is an almost exact equation, the only simplifications in making the transformation from the exact form

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) + \frac{\partial}{\partial y}(\rho v) + \frac{\partial}{\partial z}(\rho w) = 0$$

are the omission of small non-hydrostatic terms (see Eq.(2.11)).

Adiabatic equation

The heat conservation equation (written in flux-form) is

$$\frac{\partial \theta}{\partial t} = -\frac{\partial}{\partial x}(u\theta) - \frac{\partial}{\partial y}(v\theta) - \frac{\partial}{\partial p}(\omega\theta) + \frac{\theta}{T}\left(\frac{L}{c_p}Q + F\right) \qquad \text{——(2.1.4)}$$

where θ is potential temperature, $\frac{L}{c_p}Q$ is a term representing latent heat sources and sinks and F, a non-adiabatic heating function is connected solely with the initiation of the convective circulation — radiation is neglected.

Continuity of water substance

For a model in which the water substance can have different phases we require several relevant equations and these are discussed later.

We do however have the simple conservation of water vapour equation

$$\frac{\partial q}{\partial t} = -\frac{\partial}{\partial x}(uq) - \frac{\partial}{\partial y}(vq) - \frac{\partial}{\partial p}(\omega q) - Q \qquad \text{——(2.1.5)}$$

where q is the specific humidity.

Boundary conditions

Boundary conditions are required for the integration of Eq.(2.1.3)

and an equation for the height field, these are provided by:

i) $\omega = 0$ at $p = 100$ mb for all time

ii) $w = \frac{Dh}{Dt} = 0$ at $p = 1000$ mb for all time

Condition i) is the free-surface condition; condition ii) is the kinematic condition appropriate for a horizontal lower boundary and is integrated in time to give the value of $h_{p=1000}$ at each timestep.

Lateral boundary conditions are discussed in Chapter IV.

Hydrostatic and non-hydrostatic modelling

If the hydrostatic assumption is made, as is the case in all known synoptic and general circulation models, the hydrostatic equation together with Eqs.(2.1.1)-(2.1.5) form a hydrostatic primitive set of equations which can be integrated forward in time over a finite space-grid. This is the basis of the synoptic-scale models of Bushby and Timpson (1967) and Pearce (1972).

When the acceleration terms are retained in the vertical momentum equation, the height field is no longer just a function of temperature and surface pressure but has to be computed from a form of balance equation (see Chapter III).

The cumulonimbus model

For the model described in this thesis, the non-hydrostatic system is used. The essential latent heating/cooling processes and the microphysics complicate the computation of Eqs.(2.1.4) and (2.1.5). The extra equations for the inclusion of cloud and rainwater are discussed in Chapter V.

2.2 COMPUTATIONAL CONSIDERATIONS

When solving the time dependent partial differential equations of the model it is important to know what forms of motion are contained in the equation system being numerically integrated in time and space. The fastest-moving wave in the system limits the integration timestep used for a given finite-difference grid representation (the C.F.L. condition). This in turn determines the magnitude of the time-truncation error of a particular integration scheme for the system, and also the length of computing time required for simulating a given time-period of the system. Section 2.3 analyses the basic equations and boundary conditions for the motions they define.

The external gravity-wave (free-surface gravity wave) propagating with a horizontal phase-speed $c^* \sim \sqrt{gH}$ (see §2.5) corresponding to a velocity ~ 300 m.s^{-1} gives the limiting condition on the timestep. This fast-moving wave puts a rather severe limitation on the magnitude of timestep used i.e. a 1 km gridlength requires a timestep ~ 3 secs. [The accuracy of the latent heating computation is however a function of the timestep.] It might be possible to avoid the CFL restriction by the use of a semi-implicit integration scheme such as has been developed by Kwizak and Robert (1971) in which the fast-moving gravitational modes are effectively treated by an implicit time-scheme while the meteorological modes are treated by explicit time-integration. The scheme does not however demand a time-consuming fully implicit means of solution. This 'implicit' handling of the external gravity wave would allow a timestep several times larger to be used.

This fast-moving wave can be eliminated in a height coordinate model by the use of rigid upper and lower boundaries

but this does not seem to be possible using pressure coordinates.

SMOOTHING: EXPLICIT AND IMPLICIT

As noted in §2.1, no explicit Fickian-type diffusion terms are included in the prognostic equations. The inclusion and form of such terms explicitly presents difficult problems especially in a coarse-grid model and although the characteristics of the updraft produced in the model will be dependent to some extent on the mixing processes within the system, these processes are not modelled explicitly in this particular model.

Implicit 'pseudo-diffusion' is inevitable due to finite-differencing in space and time; the word 'diffusion' in this context is usually reserved for first-order accuracy schemes, i.e. ones whose errors take the equivalent form of second derivatives, but even for higher order schemes a smoothing effect is still present albeit a smaller one.

The widely-used upstream (backward) difference method for the advective terms (with forward timesteps) is an example of a scheme with an implied 'pseudo-diffusion' coefficient (Molenkamp, 1968). This coefficient can be minimised by choosing the largest possible timestep for stability. This implicit diffusion appears not to have been fully appreciated by some modellers, a recent example being the work of Takeda (1971) who uses explicit diffusion coefficients equal to 500 $m^2 s^{-1}$ while the scheme implies values several times larger. There are of course differences in effect due to the uniformity in space and time of the explicit coefficient as compared to the implicit one.

In a model whose numerical stability is controlled not by the maximum advection velocity and internal gravity wave propagation but by a much faster-moving wave, the use of a

backward-difference scheme would imply a large implicit smoothing. Backward differences best represent the continuous system when the backward gradient over a gridlength represents the mean value for a timestep; this is not the case for the stability criterion used here with fluid requiring several tens of timesteps to travel one gridlength.

Smoothing in the model

The model described here uses centred-differences for the space derivatives; this is of second-order accuracy and any inclusion of explicit second-order diffusion terms effectively reduces the accuracy of the scheme.

Analysis of the integration scheme used (Pearce, 1972) however shows the integration in time of the surface height field to be unstable without some explicit smoothing and the addition of explicit smoothing of this field with a coefficient $\sim 10^4 \, m^2 \, s^{-1}$ was necessary to control it.

The use of centred space differences presented only minor stability problems until the model was run with a non-zero ambient wind; the increased magnitudes of the advection terms then allowed non-linear instability to become significant and smoothing was introduced to suppress the growth of this numerical phenomenon. It is very necessary to do this in a convective model as "roughness" in the fields soon leads to spurious dynamical developments through interactions with the cloud physics. Although the smoothing is introduced as a computational device, its presence in the dynamical equations requires physical interpretation. The magnitude of smoothing used to control instability is a function of the gridlength used and for the coarse grid used in this model the eddy viscosity or diffusion coefficient needed to control stability is comparable to observed

turbulence measurements in clouds $\sim 10^2$-10^3 $m^2 s^{-1}$. The application of smoothing was necessary in the horizontal only and applied to the momentum equation as $K\nabla_h^2 \underset{\sim}{v}$ with $K = 500$ $m^2 s^{-1}$.

The effect on the model of including similar terms on the potential temperature and moisture fields was substantial, and when applied from the early stages of the development of the convection —suppressed the convection almost completely. The use of a K variable in space and time was considered and results with a K proportional to $|\nabla_h^2|$ on the large-scale have indicated that effectively smaller values of K (averaged in space) would control instabilities.

This formulation, taking the form $K'_\theta |\nabla_h^2 \theta| \nabla_h^2 \theta$, and $K'_q |\nabla_h^2 q| \nabla_h^2 q$, was used in the model. Over localised regions in and around the cloud the effective K attained values ~ 500 $m^2 s^{-1}$ but in general was an order of magnitude smaller. This allowed the physical effects of the smoothing to be reduced while still suppressing instability growth. The smoothing was only introduced at a point in time when roughnesses first became apparent, and details of the application of smoothing are given in the discussion of results later.

The iterative solution of the finite-difference form of balance equation every timestep introduced a smoothing into the system through the pressure-gradient terms. This smoothing could in the future be minimised by the use of a finer grid and an 'exact' solution method.

FILTERING

When the model was initially perturbed such that an internal gravity oscillation was excited with no condensation, the fields showed signs of a vertical two-gridlength standing wave, with

convergence and divergence alternating along the vertical axis of symmetry; the wave was not apparent in the evolving, moist models. This gridlength oscillation, occurring under special circumstances, could be removed by applying a three-point linear filter of the form $F'_K = F_K + \alpha(F_{K+1} + F_{K-1} - 2F_K)$ (where F denotes the field to be filtered and suffix K is a gridpoint identifier). Filtering has been used in large-scale models to filter out short-wavelength numerical oscillations and also as a scale-dependent parameterisation of atmospheric diffusion (Schapiro, 1971).

The value $\alpha = \frac{1}{4}$ in the form given above removes two-gridlength oscillations completely; further combinations of filters can then be used to restore all other gridlength waves to their full values (Schapiro, 1970).

The behaviour of the dry gravity oscillation is discussed in Chapter IV and the effect of applying a filter (as outlined above) is also mentioned.

The use of such a filter (with $\alpha = \frac{1}{4}$) represents a large smoothing; the equivalent diffusion coefficient for equal gridlengths Δ and timestep ΔT would be $K = \Delta^2/4\Delta T$ i.e. $K \sim 10^5\ m^2\ s^{-1}$ for this particular model.

Since Δ and ΔT are constrained by the stability criterion the magnitude of the filtering is of the order of $|v_{max}|\Delta$ i.e. is governed by the grid spacing and the fastest moving wave in the system. Filtering needs therefore to be carried out sparingly on models containing free-surface gravity waves.

2.3 ANALYSIS OF MOTIONS OF INFINITESIMALLY SMALL AMPLITUDE IN THE PRIMITIVE EQUATION SYSTEM USED IN THE MODEL

Knowledge of the types of motion contained in a given equation set is essential for numerical purposes as well as for physical interpretation of the results of numerical experiments. An analysis is therefore carried out here on the 'complete' non-hydrostatic primitive equations in pressure-coordinates and the knowledge gained used to aid the formulation of the equation set given in §2.1.

The equations of motion can be expressed (after judicious use of the hydrostatic assumption - though obviously not on the vertical component of the momentum equation) in the following form:

$$\frac{Du}{Dt} + g\frac{\partial h}{\partial x} = 0 \qquad \text{——(2.1)}$$

$$-\frac{1}{\rho}\frac{\partial p}{\partial z} - g = \frac{Dw}{Dt} \qquad \text{——(2.2)}$$

$$\frac{D\Phi}{Dt} = 0 \qquad \text{——(2.3)}$$

$$\frac{\partial u}{\partial x} + \frac{\partial \omega}{\partial p} = \left(\frac{\partial h}{\partial p}\right)\frac{D}{Dt}\left(\frac{\partial p}{\partial z}\right) - \frac{1}{\rho}\frac{D\rho}{Dt} \qquad \text{——(2.4)}$$

$$\Phi = \frac{1}{\gamma}\ln p - \ln \rho \qquad \text{——(2.5)}$$

These equations have been simplified by omitting the y-dimension entirely and putting f, the Coriolis parameter, equal to zero.

The analysis follows through without these assumptions but as special interest is focussed on the motions whose timescales are shorter than f^{-1}, this simplification is convenient. Eqs.(2.1)-(2.5) represent a mixed coordinate set and some manipulation must be performed to obtain consistency in coordinates;

in particular the vertical momentum equation requires modification.

Writing Eq.(2.2) as:

$$-\frac{1}{\rho}\frac{\partial p}{\partial z} = g\left(1+\frac{1}{g}\frac{Dw}{Dt}\right)$$

and making use of the approximation $\left|\frac{1}{g}\frac{Dw}{Dt}\right| << 1$ which is good for all atmospheric motions including cumulonimbus convection, then we have

$$-\rho\frac{\partial h}{\partial p} \doteq \frac{1}{g}\left(1-\frac{1}{g}\frac{Dw}{Dt}\right) + O\left(\frac{1}{g}\frac{Dw}{Dt}\right)^2$$

thus as $w = \frac{Dh}{Dt}$ we can write in pressure coordinates

$$g^2\rho\frac{\partial h}{\partial p} + g = \frac{D}{Dt}\left(\frac{Dh}{Dt}\right) \qquad \text{——(2.6)}$$

It is a characteristic of the atmosphere that practically all motions therein represent small deviations of the physical variables p and ρ from a mean, undisturbed state $p_o(z)$, $\rho_o(z)$. Thus typically deviations are such that $\left|\frac{\delta p}{p_o}\right| \sim \left|\frac{\delta \rho}{\rho_o}\right| << 1$ ($\sim 3\%$).

This mean state can equally well be defined in pressure coordinates by $h_o(p)$ and $\rho_o(p)$, all deviations of variables being defined from these mean values at constant pressure. This use of pressure coordinates requires a careful analysis of the particular set of equations and boundary conditions that are to be used to define the motion. For atmospheric motion, several terms representing what are thought to be irrelevant processes are dropped from our equation set but we will see that in some cases this leads to serious inconsistencies.

The equations are expanded about a state of no motion to facilitate analytic solutions, and tracer parameters n_i, $i = 1,2,3,4$ are attached to terms of particular interest, each n_i has a value unity

when the term in which it appears is retained, or zero when it is neglected.

Define

$$h = h_o(p) + \delta h(x,t\,;\,p)$$

$$\rho = \rho_o(p) + \delta\rho(x,t\,;\,p)$$

$$\Phi = \Phi_o(p) + \delta\Phi(x,t\,;\,p)$$

satisfying $$\frac{\partial h_o}{\partial p} + \frac{1}{g\rho_o} = 0 \qquad \text{——(2.7)}$$

and where the static stability $B = \frac{\partial \Phi_o}{\partial z} = -g\rho_o \frac{\partial \Phi_o}{\partial p}$.

We also define H, the density-scale height by

$$\frac{1}{H} = -\frac{1}{\rho_o}\frac{\partial \rho_o}{\partial z} = g\frac{\partial \rho_o}{\partial p} \quad .$$

B and H are related by: $B + g/c^2 = 1/H$ where c, the speed of sound, B and H are suitably defined mean values independent of pressure.

The simplified equation set then becomes:-

$$\frac{Du}{Dt} + g\frac{\partial}{\partial x}(\delta h) = 0 \qquad \text{——(2.8)}$$

$$\frac{\partial}{\partial t}(\delta\Phi) + u\frac{\partial}{\partial x}(\delta\Phi) - \frac{\omega B}{g\rho_o} = 0 \qquad \text{——(2.9)}$$

$$\delta\Phi = -\frac{\delta\rho}{\rho_o} \qquad \text{——(2.10)}$$

($\delta p \equiv 0$ by definition of δ in p-coordinates)

$$g^2\rho_o \frac{\partial}{\partial p}(\delta h) - g\frac{\delta\rho}{\rho_o} = \frac{D}{Dt}\left(\frac{Dh}{Dt}\right) \qquad \text{——(2.11)}$$

$$\frac{\partial u}{\partial x} + \frac{\partial \omega}{\partial p} = \left(\frac{\partial h}{\partial p}\right)\frac{D}{Dt}\left(\frac{\partial p}{\partial z}\right) - \frac{1}{\rho}\frac{D\rho}{Dt} \qquad \text{——(2.12)}$$

The precise terms retained in the vertical momentum Eq.(2.11) are significant to the simplification of the RHS of Eq.(2.12) for as can readily be seen, the RHS of Eq.(2.12) is identically zero when the pressure is assumed hydrostatic.

The implications of making various approximations in the model are now examined but further simplification of the equations is necessary in order to derive analytical solutions.

Linearising Eqs.(2.8)-(2.11) gives;

$$\frac{\partial u}{\partial t} + g\frac{\partial}{\partial x}(\delta h) = 0 \qquad \text{———(2.13)}$$

$$\frac{\partial}{\partial t}(\delta\Phi) - \frac{\omega B}{g\rho_o} = 0 \qquad \text{———(2.14)}$$

$$\delta\Phi = -\frac{\delta\rho}{\rho_o} \qquad \text{———(2.15)}$$

$$g^2\rho_o\frac{\partial}{\partial p}(\delta h) - g\frac{\delta\rho}{\rho_o} = n_3\frac{\partial^2}{\partial t^2}(\delta h) - n_4\frac{1}{g\rho_o}\frac{\partial\omega}{\partial t} \qquad \text{———(2.16)}$$

Making use of Eq.(2.16) gives us the linearised form of Eq.(2.12).

$$\frac{\partial u}{\partial x} + \frac{\partial\omega}{\partial p} = n_1\frac{1}{g}\frac{\partial^3}{\partial t^3}(\delta h) - n_2\frac{1}{g^2\rho_o}\frac{\partial^2\omega}{\partial t^2} \qquad \text{———(2.17)}$$

where n_i $(i = 1,2,3,4)$ are the tracer parameters.

The tracers n_2 and n_4 follow the gross contribution of the vertical acceleration while n_1 and n_3 follow the arithmetically smaller contribution of local changes in the height field.

Since Eq.(2.17) is derived using Eq.(2.16), for complete algebraic consistency $n_1 = n_3$ and $n_2 = n_4$ although for computational convenience this consistency may be violated if carried out carefully as will be shown.

Eqs.(2.13)-(2.17) have coefficients independent of x and t, and hence solutions can be written in exponential form where each variable is proportional to $\exp i(\lambda x + \sigma t)$.

Thus eliminating u, $\delta\Phi$ and $\delta\rho$ gives:

$$-\frac{i\lambda^2 g}{\sigma}\delta h + \frac{\partial \omega}{\partial p} = n_2 \frac{\sigma^2 \omega}{g^2 \rho_0} - n_1 \frac{i\sigma^3}{g}\delta h \qquad \text{———(2.18)}$$

$$g^2 \rho_0 \frac{\partial}{\partial p}(\delta h) + \frac{\omega B}{i\sigma \rho_0} + n_3 \sigma^2 \delta h + \frac{i\sigma}{g \rho_0} n_4 \omega = 0 \qquad \text{———(2.19)}$$

eliminating either variable gives, with some manipulation —

$$\left\{ p^2 \frac{\partial^2}{\partial p^2} + (n_3 - n_2)\frac{\sigma^2 H}{g} p \frac{\partial}{\partial p} + \left(H^2 (gB - n_4 \sigma^2)\left(\frac{\lambda^2}{\sigma^2} - n_1 \frac{\sigma^2}{g^2}\right) + n_2 \frac{\sigma^2 H^2}{g}\left(\frac{1}{H} - n_3 \frac{\sigma^2}{g}\right) \right) \right\} \psi = 0$$

———(2.20)

where ψ is either ω or $\rho_0 \delta h$.

Eq.(2.20) is a cubic equation in σ^2 and represents two distinct types of motion.

We can solve this equation as an eigenvalue problem by putting $\psi \propto p^{i\nu H + 1/2G}$ where

$$G^{-1} = \left(1 - (n_3 - n_2)\frac{\sigma^2 H}{g}\right)$$

The resulting equation is a cubic equation in σ^2

$$\frac{\sigma^6}{g^2}\left(\frac{(n_3 - n_2)^2}{4} + n_1 n_4 - n_2 n_3\right) - \frac{\sigma^4}{gH}\left(\frac{n_2 + n_3}{2} - n_1 BH\right) + \sigma^2\left(n_4 \lambda^2 + \nu^2 + \frac{1}{4H^2}\right) - gB\lambda^2 = 0$$

———(2.21)

This equation with $n_1 = n_2 = n_3 = n_4 = 1$ represents the full equation with all terms included and takes the form

$$\frac{\sigma^4}{c^2} - \sigma^2(\lambda^2 + \nu^2 + \frac{1}{4H^2}) + gB\lambda^2 = 0 \qquad \text{———(2.22)}$$

The disappearance of the sixth-order term is a significant indication of the spurious nature of this term i.e. a choice of the n_i's such that the term does not disappear represents a physically unrealistic condition.

Eq.(2.22) has two pairs of roots given by:

$$\sigma^2 = \frac{1}{2}c^2(\lambda^2+\nu^2+\frac{1}{4H^2}) \pm \frac{1}{2}c^2(\lambda^2+\nu^2+\frac{1}{4H^2})\left(1-\frac{4gBc^2\lambda^2}{c^4(\lambda^2+\nu^2+\frac{1}{4H^2})^2}\right)^{1/2}$$

The gravity-inertial wave

For $\lambda^2 >> \nu^2+\frac{1}{4H^2}$ we have the solution $\sigma^2 \sim gB$ i.e. motion controlled by buoyancy in tall, narrow columns.

For $\frac{4gB\lambda^2}{c^2(\lambda^2+\nu^2+\frac{1}{4H^2})} << 1$ we obtain the root:

$$\sigma^2 \sim \frac{gB\lambda^2}{\lambda^2+\nu^2+\frac{1}{4H^2}} \qquad \text{——(2.23)}$$

This is the internal gravity-wave oscillation applicable to small-scale convection in which the vertical displacements of particles are controlled by buoyancy (pressure) forces, elastic forces being unimportant.

If hydrostatic equilibrium is assumed throughout the motion (i.e. $n_1 = n_2 = n_3 = n_4 = 0$), the same mode is present (see Eq.(2.21)) but distorted, being without dynamic pressure control i.e.

$$\sigma^2 = gB\frac{\lambda^2}{\nu^2+\frac{1}{4H^2}}$$

The period $\tau = \frac{2\pi}{\sigma}$, is less than the true value given by Eq.(2.23) since:

$$\left\{\frac{\tau_{true}}{\tau_{hydrostatic}}\right\}^2 = 1+\frac{\lambda^2}{\nu^2+\frac{1}{4H^2}} \qquad \text{——(2.24)}$$

For $B<0$ we get a growing solution (σ imaginary) for which the hydrostatic assumption implies an increased growth rate compared to the true value.

Sonic wave

A second solution of Eq.(2.22) for σ^2 sufficiently large

$$\sigma^2 \sim c^2(\lambda^2 + \nu^2 + \frac{1}{4H^2})$$

This is a sound wave propagating with velocity c normal to the lines of constant phase.

Further examination of Eq.(2.21) indicates that this sonic solution only exists for the full equation set (i.e. all $n_i = 1$) since the coefficient of the fourth-order term in σ, $\left(\frac{n_2 + n_3}{2} - n_1 BH\right)$ only takes the form $(1 - BH)$ $[= gH/c^2]$ for $n_1 = n_2 = n_3 = 1$; the choice $n_1 = n_2 = n_3 = 1$ and $n_4 = 0$ retains the sixth-order term in σ and gives spurious solutions.

2.4 THE IMPLICATIONS OF VARIOUS APPROXIMATIONS

The hydrostatic assumption $n_1 = n_2 = n_3 = n_4 = 0$ removes the sonic solution of Eq.(2.21) entirely but thereby distorts the gravity wave solution significantly, for in Eq.(2.24) $\frac{\lambda^2}{\nu^2 + 1/4H^2} \sim 1$ for motions whose horizontal and vertical dimensions are comparable.

It will be observed from Eq.(2.21) that the choice of n_i's such that $n_1 = n_2 = n_3 = 0$, but $n_4 = 1$ introduces no spurious high-order solutions and that the important gravity-wave solution, the only mode remaining, has the undistorted form

$$\sigma^2 = gB \frac{\lambda^2}{\lambda^2 + \nu^2 + \frac{1}{4H^2}}$$

This approximation is therefore very much better than the straightforward hydrostatic approximation, having filtered out sound waves which are meteorologically irrelevant, while at the same time leaving the internal gravity wave undistorted. [This result is incorporated in the form of vertical momentum equation adopted in the derivation of a balance equation (see Chapter III).] This appears to be comparable to omitting $\frac{\partial \rho}{\partial t}$ from the height-coordinate continuity

equation although this term is in part retained in the pressure coordinate form within $\frac{\partial \omega}{\partial p}$.

2.5 BOUNDARIES

The imposition of upper and lower boundary conditions on the linearised system can be examined as an eigenvalue problem; that of solving Eq.(2.20) with upper and lower boundary conditions. The solution of this problem would give frequencies (or phase-speeds) for different values of horizontal and vertical wavelength and static stability. This problem has not been attempted here, but some limiting cases of the solutions are examined.

Free-surface boundary condition: $\omega = 0$

The solution $\omega = 0$ everywhere satisfies this boundary condition provided: $n_1\sigma^4 = \lambda^2 g^2$ (from Eq.(2.18)).

This is a short-wave approximation to the free-surface boundary wave, with a growing component $\sigma^2 = -\lambda g$ if $n_1 = 1$. This indicates that $n_1 = 0$ is a necessary choice if a free-surface boundary condition is used.

Rigid boundary condition: $w = 0$

$w = 0$ everywhere satisfies this boundary condition and implies from linearising that

$$\delta h = \frac{\omega}{i\sigma g \rho_o} \qquad \text{everywhere} \qquad \text{———(2.25)}$$

Thus from Eqs.(2.18) and (2.19) we obtain the frequency solution $\sigma^2 = \lambda^2 c^2$. This is a horizontally propagating 'sound' wave in which there is no vertical motion (a Lamb wave).

If we apply the boundary condition Eq.(2.25) together with Eq.(2.18) we obtain the condition

$$gH \frac{\lambda^2}{\sigma^2} \omega = p \frac{\partial \omega}{\partial p} \qquad \text{at } p = p_s \qquad \text{——(2.26)}$$

This together with the upper boundary condition

$$\omega = 0 \qquad \text{at } p = p_T$$

provides boundary conditions for Eq.(2.20) which we can solve for the simple case of a hydrostatic, neutral atmosphere, a simplification that in retrospect loses no essential features of the solution.

We obtain the solution:

$$\frac{\sigma^2}{\lambda^2} = gH(1 - p_T/p_s)$$

This is the external gravity wave i.e. the wave that has constant specific kinetic energy throughout the fluid depth, having a phase speed c given by $c^2 = gH(1 - p_T/p_s)$.

The analysis given in this section in (x,y,p ; t) coordinates follows that given by Green (1969) for the linearised equations in the (x,y,z ; t) coordinate system.

SUMMARY

As a result of the linearised analysis of our proposed equation set we conclude that with careful modification we can retain only the undistorted modes relevant to the cumulonimbus scale also the use of a free-surface boundary condition allows the external gravity wave in the system.

The extension to an atmosphere in which moist processes occur will not modify the results of the analysis dynamically although terms introduced into the equations will modify the numerical values involved.

CHAPTER III

THE NON-HYDROSTATIC FORMULATION OF THE MODEL

If hydrostatic balance is assumed in the primitive equation set, the integrated hydrostatic equation is used to obtain the heights of the pressure surfaces. This is the normal large-scale modelling procedure justifiable when the motion is quasi-horizontal. Furthermore, the computation of vertical motion by integrating the continuity equation as a diagnostic equation for ω closes the set. In general the significant scale of horizontal divergence is adequately resolved in these models particularly in the fine-mesh models currently being developed [gridlengths ~ 50-100 km].

The use of the hydrostatic assumption in modelling convection might at first sight be considered reasonable for although air in a cumulonimbus updraft undergoes vertical accelerations these are two orders of magnitude smaller than the acceleration due to gravity thus making the vertical acceleration a small term in the momentum equation, the effect of precipitation loading also would appear to make a small contribution; however, this reasoning is not valid as is indicated in the next section.

3.1 ANALYSIS OF MOMENTUM EQUATION

Writing the vertical momentum equation as:

$$\frac{Dw}{Dt} + \frac{1}{\rho}\frac{\partial p}{\partial z} + g = 0 \qquad \text{———(3.1)}$$

where the Coriolis and diffusion terms have been omitted, we observe that this equation is applicable to air containing water provided ρ is the total density of the air and water substance. If any of the water vapour content is condensed to liquid water and this liquid water is in the form of drops falling at their terminal velocity under the influence of gravity

CHAPTER III

THE NON-HYDROSTATIC FORMULATION OF THE MODEL

If hydrostatic balance is assumed in the primitive equation set, the integrated hydrostatic equation is used to obtain the heights of the pressure surfaces. This is the normal large-scale modelling procedure justifiable when the motion is quasi-horizontal. Furthermore, the computation of vertical motion by integrating the continuity equation as a diagnostic equation for ω closes the set. In general the significant scale of horizontal divergence is adequately resolved in these models particularly in the fine-mesh models currently being developed [gridlengths ~ 50-100 km].

The use of the hydrostatic assumption in modelling convection might at first sight be considered reasonable for although air in a cumulonimbus updraft undergoes vertical accelerations these are two orders of magnitude smaller than the acceleration due to gravity thus making the vertical acceleration a small term in the momentum equation, the effect of precipitation loading also would appear to make a small contribution; however, this reasoning is not valid as is indicated in the next section.

3.1 ANALYSIS OF MOMENTUM EQUATION

Writing the vertical momentum equation as:

$$\frac{Dw}{Dt} + \frac{1}{\rho}\frac{\partial p}{\partial z} + g = 0 \qquad \text{———(3.1)}$$

where the Coriolis and diffusion terms have been omitted, we observe that this equation is applicable to air containing water provided ρ is the total density of the air and water substance. If any of the water vapour content is condensed to liquid water and this liquid water is in the form of drops falling at their terminal velocity under the influence of gravity

then a drag term equal to the water's weight must be included i.e.

$$\frac{Dw}{Dt} + \frac{1}{\rho}\frac{\partial p}{\partial z} + g + gl = 0 \qquad \text{———(3.2)}$$

where l = liquid water content in gm/gm; ρ is now the density of air and water vapour only, and the term $l\frac{Dw}{Dt}$ has been neglected.

By expanding the variables as:

$$\left.\begin{aligned} p &= p_s + p' \\ \theta &= \theta_s + \theta' \\ h &= h_s + h' \\ l &= 0 + l \end{aligned}\right\}$$

where the suffix s identifies the undisturbed, droplet-free atmosphere.

Eq.(3.2) can then be written as

$$\frac{Dw}{Dt} - g\left\{\frac{\theta'}{\theta_s} - l - \frac{\partial h'}{\partial p}\Big/\frac{\partial h_s}{\partial p}\right\} = 0 \qquad \text{———(3.3)}$$

This is now an equation in which the gross hydrostatic balance has been removed. It is now apparent that for deep convection the contributions of each term in Eq.(3.3) are of comparable magnitude; thus we have acceleration values of the order of 10^{-1} m.sec^{-2} being equivalent to 3°C of buoyancy, 10 gm.Kg^{-1} of liquid water and 30 m.s^{-1} over 10 km vertical acceleration, all reasonable values for a cumulonimbus updraft. With the hydrostatic assumption Eq.(3.3) reduces to a thickness equation:

$$\frac{\partial h_s'}{\partial p} = \frac{\partial h_s}{\partial p_s}\left(\frac{\theta'}{\theta_s} - 1\right) \qquad \text{———(3.4)}$$

integrable to give h_s'.

3.2 IMPORTANCE OF NON-HYDROSTATIC PRESSURE

We define non-hydrostatic pressure as that part of the change in the total pressure field produced by accelerations as distinct from the changes due to density variations. The significance of the vertical acceleration is demonstrated not only by its

magnitude in the vertical momentum equation but by an essential coupling in the dynamics.

The vertical component of the momentum equation should be interpreted in the same manner as the horizontal components, i.e. as representing the coupling of the wind component to the pressure field through the acceleration terms. This coupling is removed when the hydrostatic assumption is made allowing the vertical velocity to develop without the control of the vertical pressure gradient. This is also demonstrated by forming the energy equation for the hydrostatic system; this equation contains no vertical kinetic energy ($\frac{1}{2}w^2$), obviously an unreal system for cumulonimbus convection.

The inertial term should strictly be included on the synoptic-scale also, but as its contribution is only small, its neglect does not have the catastrophic consequences that arise in the convective model. The neglect on the synoptic or large-scale is however not entirely satisfactory but a practical necessity.

Notation

In the derivation and discussion which follow it will be convenient to partition the height field as $h = h_S + h'$ where h_S is the unperturbed (initial) height field. We also write $h' = h'_S + h''$ where h', the total deviation of the height field, is subdivided into a hydrostatic part h'_S defined by Eq.(3.4) plus a non-hydrostatic part h''. [In applying this partition to other variables it should be noted that by definition ρ'' (and θ'') are identically zero.]

Pressure gradient form

The derivations of the vertical momentum equation and a type of balance equation are given in detail because the use of pressure coordinates makes their form novel.

3.3 VERTICAL MOMENTUM EQUATION

To handle the acceleration term satisfactorily in a pressure-coordinate model requires the transformation of $\frac{Dw}{Dt}$ into pressure coordinate variables.

The complete transformation is complex being of the form:

$$\frac{Dw}{Dt} = \frac{\partial w}{\partial t} + \underset{\sim}{v}.\nabla w + \omega\frac{\partial w}{\partial p} \qquad \text{———(3.5)}$$

with w replaced by $\underset{\text{(i)}}{\frac{\partial h}{\partial t}} + \underset{\text{(ii)}}{\underset{\sim}{v}.\nabla h} + \underset{\text{(iii)}}{\omega\frac{\partial h}{\partial p}}$ ———(3.6)

This is impracticable to use and we look for some suitable simplifications.

The approximation $w \simeq -\omega/g\rho_s$ is a well-known one, readily justified as is seen by examining the orders of magnitude of (i), (ii) and (iii). The use of this approximation however when differentiated, i.e. as

$$\frac{Dw}{Dt} \simeq \frac{D}{Dt}\left(-\frac{\omega}{g\rho_s}\right) \qquad \text{———(3.7)}$$

requires consideration.

Ignoring horizontal advection terms from Eqs.(3.5) and (3.6) and expanding we find that the approximation (3.7) is equivalent to assuming

$$\frac{\partial_p^2 h}{\partial t^2} << \frac{\partial_p w}{\partial t}$$

i.e. the buoyancy generated vertical accelerations are very much larger than the local rates of change in vertical velocity associated with changes in height of the pressure surface.

Considering a column of atmosphere of thickness H, temperature $\bar{\theta}$ subject to a warming throughout of $\overline{\delta\theta}$ then,

$$\frac{\partial_p^2 h}{\partial t^2} \sim \frac{H}{\bar{\theta}}\frac{\overline{\delta\theta}}{\tau_1^2}$$

where τ_1 is the time-scale of the heating $\overline{\delta\theta}$.

The surface pressure change is ignored on this timescale

also: $$\frac{\partial w}{\partial t} \sim g\frac{\overline{\delta\theta}}{\overline{\theta}}$$

Defining τ_2 as the timescale of air following a trajectory through a cumulonimbus then:

$$H \sim \frac{1}{2} g \frac{\overline{\delta\theta}}{\overline{\theta}} \tau_2^2$$

giving $$\frac{\partial_p^2 h}{\partial t^2} \Big/ \frac{\partial w}{\partial t} \sim \frac{\overline{\delta\theta}}{\overline{\theta}} \frac{1}{2}\left(\frac{\tau_2}{\tau_1}\right)^2$$

thus typically $\frac{1}{2}\left(\frac{\tau_2}{\tau_1}\right)^2 \sim 1$ and hence the approximation (3.7) is valid to $\sim \left\{\frac{\overline{\delta\theta}}{\overline{\theta}} \times 100\right\}$ %.

The full dynamical repercussions of these approximations, i.e. of making the hydrostatic assumption and neglecting the density variations both in certain contexts, were examined in Chapter II, §2.3 for the linearised system and the results are a further justification for the form of equation used.

The approximations made could also be justified by the generation of an algorithm whereby the simplified form is used as a first approximation to the complete equation and an iterative (convergent) procedure carried out.

Introducing the simplifications discussed into Eq.(2.6) gives:

$$\frac{D}{Dt}\left(\frac{\omega}{\rho_s}\right) + g^3\rho\frac{\partial h}{\partial p} + g^2 = 0 \qquad \text{———(3.8)}$$

This equation contains the hydrostatically balanced part $\left(\frac{D}{Dt}\left(\frac{\omega}{\rho_s}\right) = 0\right)$

i.e. $$\frac{\partial}{\partial p}\left(h_s + h_s'\right) + \frac{1}{g\rho} = 0 \qquad \text{———(3.9)}$$

Subtracting Eq.(3.9) from Eq.(3.8) gives the non-hydrostatic component of the vertical momentum equation:-

$$\frac{D}{Dt}\left(\frac{\omega}{\rho_s}\right) + g^3\rho\frac{\partial h''}{\partial p} = 0 \qquad \text{———(3.10)}$$

Footnote A scale analysis of the equation set shows that the form of vertical momentum equation Eq.(3.8) is consistent with the approximations made in Eqs.(2.1.1, 2.1.2, and 2.1.3).

3.4 COMPUTATION OF COMPLETE DEVIATION HEIGHT FIELD h'

To compute the complete deviation height field in a non-hydrostatic primitive model, a form of balance equation is used. Ogura and Charney (1962) derived this balance equation for a primitive equation set in an (x,y,z; t) coordinate system. The derivation required here is in pressure coordinates.

It is possible to form an equation for either the complete deviation height field (h') or one for the non-hydrostatic field (h") only.

For computing h', the significant terms arise from the temperature field and the accelerations, while in computing h", only the residual between the hydrostatic field and the accelerations contribute. It would also be possible to obtain h" by computing h and h'_s individually.

The method adopted here was to compute h', chosen as the one having least computational difficulties. [The computing of h" and h'_s separately and adding them was tried and with suitable boundary conditions worked satisfactorily; however h" and h'_s tended to be opposite in sign and grew larger and larger in magnitude leaving h' as a small residual.]

The lower boundary condition on the model is taken as $w = \frac{Dh}{Dt} = 0$ on the 1000 mb surface; this is integrated to give h'_{1000} as a function of time immediately providing a boundary condition for computing h'. [In computing h", the boundary condition was taken as $w = w' = w'_s + w'' = 0$ and $w'_s = w'' = 0$ and hence $h''_{1000} = 0$.]

3.5 BALANCE EQUATION FOR THE HEIGHT FIELD

We take the horizontal momentum equations Eq.(2.1.1) and Eq.(2.1.2) and a slightly modified form of Eq.(3.8) to include liquid water,

i.e. $$\frac{D}{Dt}\left(\frac{\omega}{\rho_s}\right) + g g^{*2} \rho \frac{\partial h}{\partial p} + g g^* = 0 \qquad \text{——(3.11)}$$

where $g^* = g(1 + l)$ (l = liquid water concentration)

The balance equation is formed by taking

$$\frac{\partial}{\partial x} \cdot \text{Eq.(2.1.1)} + \frac{\partial}{\partial y} \cdot \text{Eq.(2.1.2)} + \frac{\partial}{\partial p} \cdot \left\{ \rho_s \times \text{Eq.(3.11)} \right\}$$

Thus
$$\frac{\partial}{\partial t}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial \omega}{\partial p}\right) = -\nabla_2^2 \psi + \frac{\partial}{\partial x}(v\zeta) - \frac{\partial}{\partial y}(u\zeta) - \frac{\partial}{\partial p}\left(\underset{\sim}{v}_2 \cdot \nabla\omega + \rho_s \omega \frac{\partial}{\partial p}\left(\frac{\omega}{\rho_s}\right)\right) - \frac{\partial}{\partial p}\left(g g^{*2} \rho \rho_s \frac{\partial h}{\partial p} + g g^* \rho_s\right) \qquad \text{——(3.12)}$$

The LHS of this equation is zero by continuity hence Eq.(3.12) becomes:

$$g \nabla_2^2 h' + g \frac{\partial}{\partial p}\left(g^{*2} \rho \rho_s \frac{\partial h}{\partial p} + g^* \rho_s\right) + \left\{ \quad \right\} = 0 \qquad \text{——(3.13)}$$

where $\{\ \}$ contains all the non-linear terms in Eq.(3.12).

We can remove the unperturbed height field from Eq.(3.13) in the following way:

$$\rho_s g^* g\left(g^* \rho \frac{\partial h}{\partial p} + 1\right) = g^* g \rho_s \left(g^*(\rho_s + \rho')\left(\frac{\partial h_s}{\partial p} + \frac{\partial h'}{\partial p}\right) + 1\right)$$
$$\approx -g^2\left(l\rho_s + \rho' - g\rho_s^2 \frac{\partial h'}{\partial p}\right)$$

neglecting all terms $O(\ '^2$ and higher

hence Eq.(3.13) becomes

$$g\nabla_2^2 h' + g^3 \frac{\partial}{\partial p}\left(\rho_s^2 \frac{\partial h'}{\partial p}\right) - g^2 \frac{\partial}{\partial p}(l\rho_s + \rho') + \left\{ \quad \right\} = 0 \qquad \text{——(3.14)}$$

ρ is not computed as a dependent variable in our pressure coordinate model, hence in terms of θ (where θ is now virtual potential temperature)

$$\frac{\partial \rho}{\partial p} = \frac{p}{\pi R \theta_s}\left[\frac{(1-\kappa)}{p}\left(1 - \frac{\theta'}{\theta_s}\right) - \frac{1}{\theta_s}\frac{\partial \theta_s}{\partial p} - \frac{1}{\theta_s}\frac{\partial \theta'}{\partial p} + 2\frac{\theta'}{\theta_s^2}\frac{\partial \theta_s}{\partial p}\right]$$

where $\pi = (p/p_o)^{\kappa}$ and $\kappa = R/c_p$

and $\frac{\partial \rho_s}{\partial p} = \frac{p}{\pi R \theta_s}$ [as above with primed variables zero]

then Eq.(3.14) becomes:

$$\nabla^2_2 h' + g^2 \frac{\partial}{\partial p}\left(\rho^2_s \frac{\partial h'}{\partial p}\right) + \frac{gp}{\pi R \theta_s}\left[\frac{(1-\kappa)}{p}\left(\frac{\theta'}{\theta_s} - 1\right) + \frac{1}{\theta_s}\frac{\partial \theta'}{\partial p} - \frac{\partial l}{\partial p} - \frac{1}{\theta_s}\frac{\partial \theta_s}{\partial p}\left(2\frac{\theta'}{\theta_s} - 1\right)\right] + \frac{1}{g}\left\{\quad\right\} = 0 \qquad \text{———(3.15)}$$

The correlation of l and θ' is such that no terms can readily be neglected from the square bracketed term.

This equation, together with suitable boundary conditions, provides a diagnostic equation for the height field in a pressure coordinate model. It is important to notice that this equation is diagnostic because of the disappearance of the local time-derivative $\frac{\partial}{\partial t}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial \omega}{\partial p}\right)$ since $\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial \omega}{\partial p} = 0$.

In the primitive equation height coordinate model of Ogura and Charney, the equivalent time derivative is $\frac{\partial}{\partial t}\left(\frac{\partial}{\partial x}(\bar{\rho}u) + \frac{\partial}{\partial y}(\bar{\rho}v) + \frac{\partial}{\partial z}(\bar{\rho}w)\right)$ [where $\bar{\rho} = \bar{\rho}(z)$ only is related to a basic atmosphere of constant potential temperature $\bar{\theta}$ by: $\bar{\rho} = \frac{p_o}{R\bar{\theta}}\left(\frac{\bar{p}}{p_o}\right)^{1/\gamma}$, an additional approximation].

This derivative disappears if $\nabla_3 \cdot (\bar{\rho}\underset{\sim}{v}) = 0$. Thus we have the convenient elimination of the explicit time-dependence in either coordinate system, the resulting form of balance equation depending on the form of continuity equation used and through particular approximations made in the basic equations.

The height field is computed over a three-dimensional rectangular 'box' with the following boundary conditions:

a) Lower boundary: For a non-hydrostatic model, as for a hydrostatic model, the value of h is required at the lower boundary; this is given by:

$$w = \frac{Dh}{Dt} = 0 \qquad \text{at} \qquad p = p_{surface}$$

which for the model is taken as p = 1000 mb, thus at p = 1000 mb

$$\frac{\partial}{\partial t}(h_s + h') = -(\underset{\sim}{v}.\nabla_2)(h_s + h') - \omega\left(\frac{\partial h_s}{\partial p} + \frac{\partial h'}{\partial p}\right)$$

As $h_s = h_s(p)$ only and $\frac{\partial h_s}{\partial p} >> \frac{\partial h'}{\partial p}$ $\left\{\begin{matrix}\text{equivalent to } \rho'/\rho_s << 1 \\ \text{valid to 1\%}\end{matrix}\right\}$

the time-integration of Eq.(3.16) provides h' as a function of time,

$$\frac{\partial h'}{\partial t} = -(\underset{\sim}{v}.\nabla_2)h' + \frac{\omega}{g\rho_s} \qquad \text{at } p = 1000 \text{ mb} \qquad \text{———(3.16)}$$

b) Upper boundary: We take $\omega = 0$ at p = 100 mb as a (free-surface) boundary condition for integrating the continuity equation but require a boundary condition at 150 mb for h' because of the particular grid configuration used. If we assume $\frac{D}{Dt}(\omega/\rho_s)$ is small at this pressure level then we can define h'_{150} as being that value which satisfies the hydrostatic 'thickness' equation (Eq.(3.4)) and this was the condition used. [The most consistent way of doing this is to include a height level at 100 mb and perform the same computation.]

c) Lateral boundaries: The condition on h' on the lateral boundaries was essentially computed from the horizontal momentum equations and is discussed in Chapter IV, §4.7.

3.6 NUMERICAL SOLUTION OF THE BALANCE-TYPE EQUATION

The Poisson-type elliptic equation derived in the previous section requires to be solved over a 3-D grid of points with equal mesh lengths in the two horizontal directions but unequal mesh lengths in the vertical.

The analytic form of this equation contains the differentiated momentum equations and the continuity equation; the finite-difference form of this balance equation must therefore consist of finite-difference differentiation of the finite-difference

replacements of its component equations formulated previously in the program, an arduous task.

There are two well-known possible solution techniques, the Alternating Direction Implicit method (A.D.I.) or the over-relaxation method (S.O.R.). Miyakoda (1960) carried out comparison computations for various methods and concluded that A.D.I. was most efficient for large arrays but demands large computer storage (at least twice as much as S.O.R.) and requires many more machine instructions. Numerical experiments confirmed Miyakoda's conclusions.

The S.O.R. method provides a good scheme for smaller arrays, having the virtues of simplicity and minimal storage requirements and this was the method used here. The choice of optimum over-relaxation factor is critical and was determined empirically (see Fig.3.1) although a formula for determining an optimum S.O.R. factor is given by Miyakoda for equally-spaced grids.

In practice the solution of this equation by S.O.R. for the (15 x 15) model on the CDC 6600 machine represented 70% of the total computing time, thus any improvement in this part of the computation scheme would yield large benefits.

The rapid advances in spectral methods in the last decade indicate that considerable savings in solution time are achieved by the use of these methods.

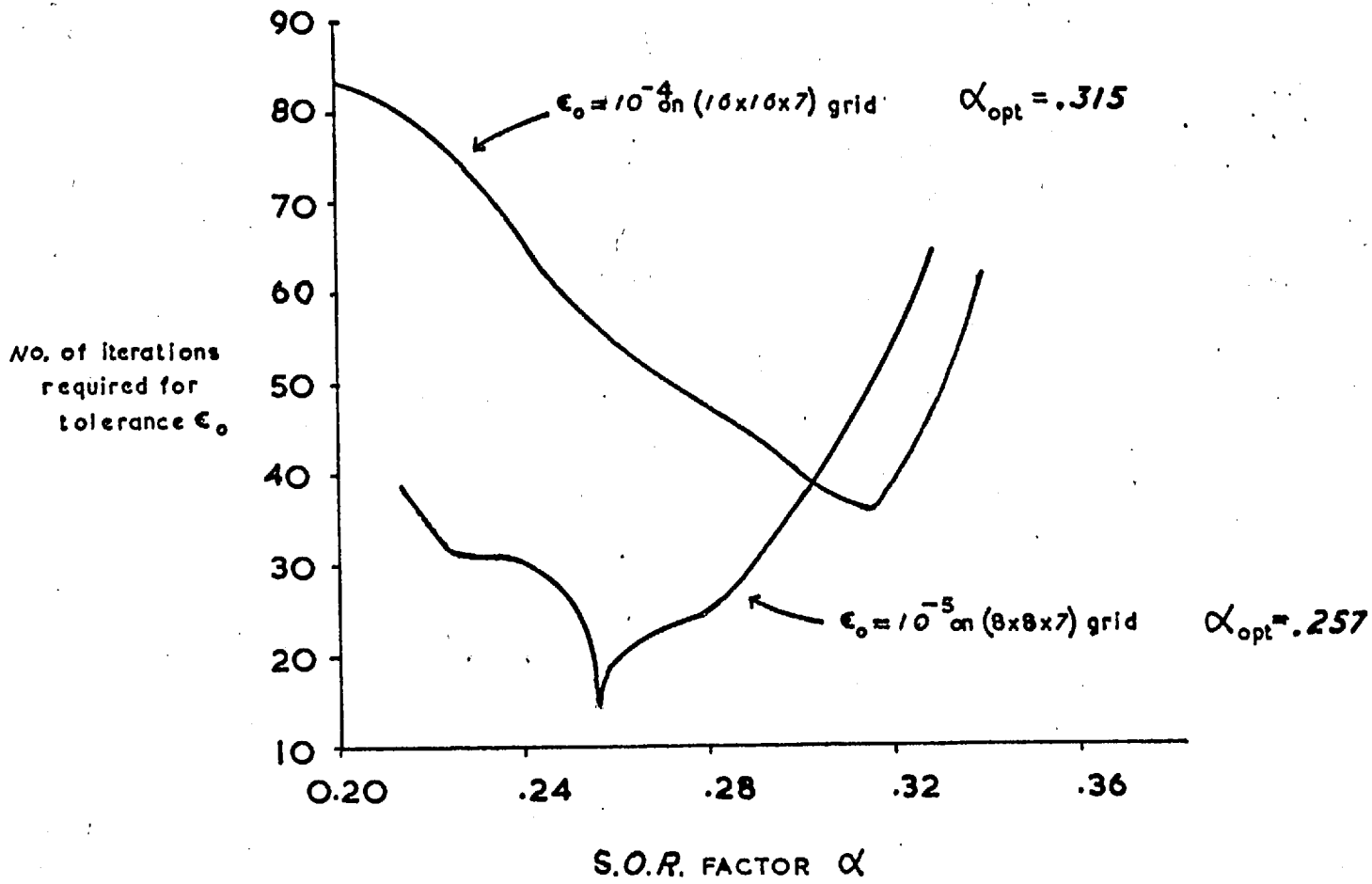


FIG. 3.1 Choice of S.O.R. factor for 3-D balance equation.

CHAPTER IV

BOUNDARY CONDITIONS

4.1 BOUNDARY CONDITIONS

Problems of formulating suitable boundary conditions at upper, lower and lateral boundaries arise in the modelling of convection because the model describes only a limited part of the atmosphere. To approach the problem in a flexible manner requires the use of boundary conditions which allow variety in the initial environmental motions and also permit interactions with the convective system, this contrasts with the modelling convenience of 'closed-box'-type assumptions which impose severe limitations on the model's usefulness.

Attention is focussed here on the lateral boundary conditions, the upper and lower boundaries being considered elsewhere.

There are two possible approaches to the problem:

1) To use a model grid large enough to contain the whole region influenced by the convective system, thus making the choice of boundary conditions non-critical.

2) To choose boundary conditions which correctly relate the model to its 'environment', and thus make the use of a smaller grid feasible.

The first approach requires a grid large enough to contain within the modelled region the fastest-moving waves in the modelling system throughout the total simulation time of the model, this requires dimensions of hundreds of kilometres for models containing free-surface waves, and up to one hundred kilometres for models containing only internal waves. Within present-day limitations on computer storage and economics, this approach is an impractical one for modelling deep convection.

A 2-D model, 50 km in the horizontal and 10 km deep has been used by Takeda (1971) but even these dimensions are difficult to approach in 3-D.

Regarding the second possible approach, the general problem of boundary conditions which relate a primitive model with its environment in an appropriate manner has not been investigated very thoroughly in the past, and the approach used in this model is essentially that of Pearce (1972) although with substantial modification. Pearce carries out an analytic study on the partial differential equations which maintain the external gravity wave. He concludes that it is sufficient for this simplified system to specify either:

(a) The normal velocity component together with either the tangential component or the z-component of vorticity

or (b) Values of the height field.

The internal gravity wave propagation is controlled essentially by the boundary conditions on the temperature, height and vertical motion fields. The moisture q enters very weakly into the gravity wave propagation through its contribution to the density (the virtual temperature effect) and can be regarded as being simply advected across the boundaries.

4.2 BOUNDARY FORMULATION

It was stated above that the specification of v_n (normal component) and v_L (tangential component) were sufficient boundary conditions and these were adopted for the model.

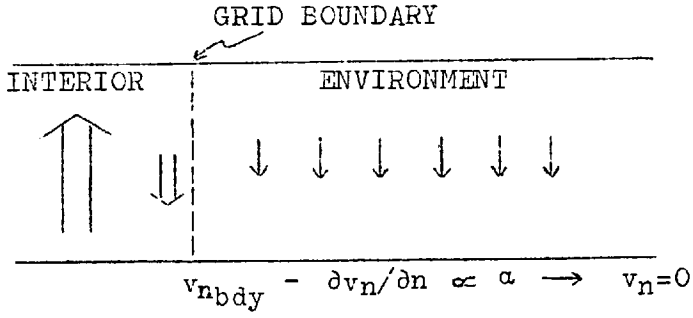
[N.B. for the case of no initial flow v_n and v_L are the boundary wind components of the convective system; for an initial sheared flow v_n and v_L are the <u>perturbation</u> velocities on this initial flow. This applies throughout the chapter.]

This specification of velocity components was incorporated in a manner which allowed detailed modelling of the most active part of the convective system within the computation grid without using a mesh of gridpoints to model the whole of the smoother region surrounding it. In the following sections the methods used are described together with examination of certain difficulties relating to the use of a Cartesian grid, and a less succesful formulation used in the early stages of the project outlined.

4.3 SPECIFICATION OF NORMAL VELOCITY v_n

The normal velocity was computed from the value inside as $v_{n_{bdy}} = \alpha\, v_{n_{inside}}$ where $\alpha(x,y)$ was taken as constant in time and independent of height; the precise use of α and the manner of computation is given in sections 4.5 and 4.6.

The function α can take values $0 \leqslant \alpha \leqslant 1$ and therefore in general represents a drop-off of the velocity to zero at some defined distance. This implied velocity dependence on α can be interpreted as a simple modelling of the vertical motion in the environment surrounding the volume covered by the gridpoints.



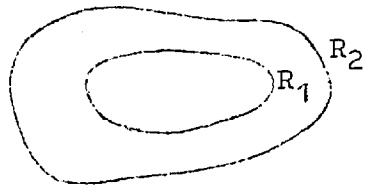
The value $\alpha = 0$ models a closed 'box' in which no flow is allowed across the boundaries implying that all vertical mass fluxes are zero on each isobaric surface within the model grid region.

The device used here of using models with a small number of gridpoints and open boundaries, and lateral boundary conditions based on an assumed environmental behaviour can strictly only be justified by comparison of results with those from models with larger grids. The implied linear normal velocity decrease with distance outwards from the grid bears good comparison with normal velocity (and hence vertical motion) fields from the closed boundary (15 x 15) model as can be seen in Fig. 4.1.

4.4 COMPUTATION OF THE HORIZONTAL DIVERGENCE, D

A problem that arises in computing the horizontal divergence on the boundaries is the following:

Consider two contours R_1 and R_2 in the x-y plane;



then if fluid flows through the annulus R_1R_2, mass flux across R_2 minus mass flux across R_1 (positive outwards) is proportional to the net mass divergence from R_1R_2. This divergence is defined by the normal velocity, the tangential component making no contribution, (property [A]).

Using a grid configuration which is not radially symmetric the specification of the velocity components in the manner described in §4.3 (i.e. directly from interior values) introduces a form of oscillation in the horizontal divergence field whereby although property [A] is satisfied, the divergence field around

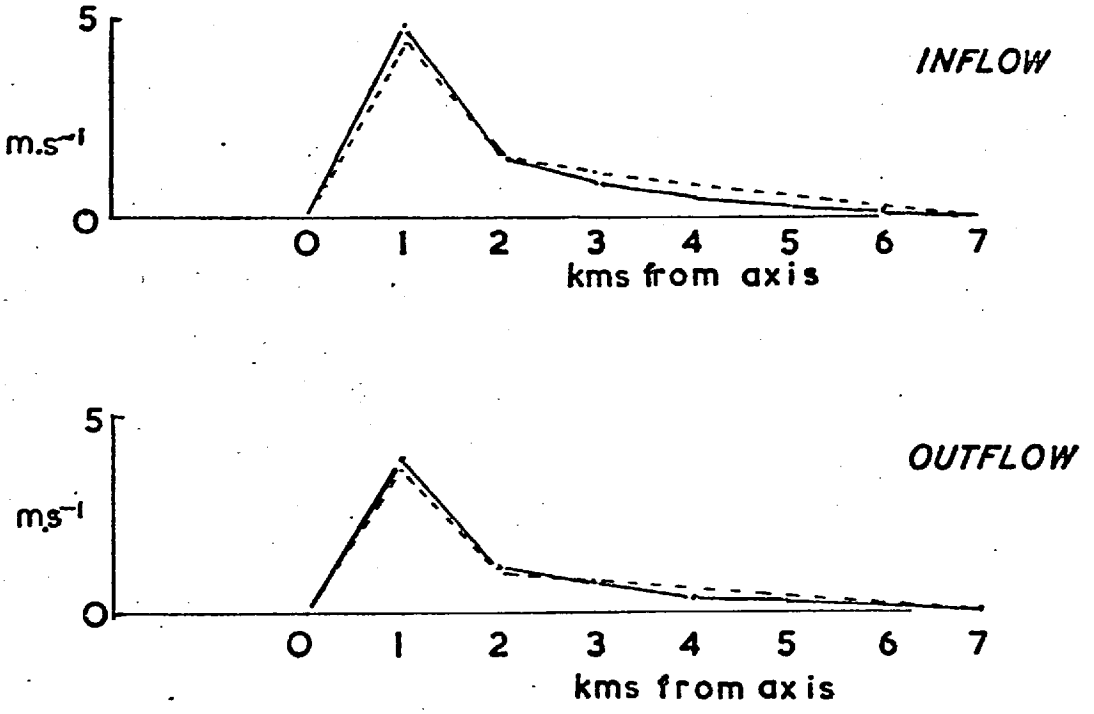
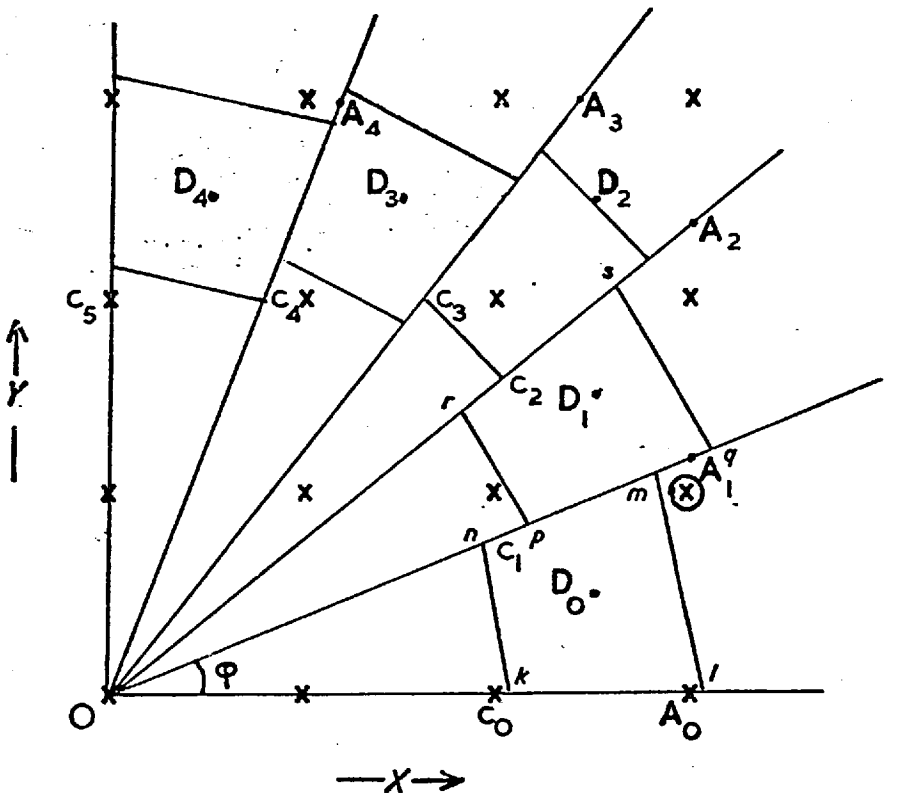


FIG 4.1 Normal velocity comparison of models showing variation outwards from convective cell



Points (x) hold the velocity components (u,v)

FIG 4.2 Horizontal grid section for boundary computation

the boundary annulus contained a symmetric oscillation. This was apparent in the early stages of the model development and arises essentially from the geometry of the grid configuration, being due to the inability of the Cartesian grid with the simple boundary velocity specification to accurately describe a substantially radially symmetric flow without considerable modification. The errors in the divergence field result from non-zero contributions to the gridsquare divergence values by the finite-difference representations of the tangential velocities.

Attempts to eliminate this problem resulted in the following formulation of the boundary divergence computation whereby the Cartesian grid near the boundaries is treated in a quasi-polar coordinate manner such as to minimise this geometric problem introduced by having corners on the grid boundary.

4.5 COMPUTATION OF BOUNDARY DIVERGENCE

The following finite-difference procedure was adopted which did not result in any significant oscillation in the divergence field round the boundaries. The boundary velocity components resulting from simple specification in terms of interior values are improved upon whilst retaining the principle discussed in §4.3.

Referring to Fig. 4.2 , the mean values of the divergence for the trapeziums klmn, pqrs, etc. (where mn = kl = 1 gridlength) are each computed as the sum of a radial component DRAD, and a tangential component DTAN, such that $\sum$ DTAN round a 'ring' of trapeziums are zero.

The algebraic forms used for this computation are as follows:

Defining VRAD and VTAN as velocity components along and at right angles to the radial lines in Fig. 4.2 and with VRAD and VTAN computed from the (u,v) values by simple trigonometry then:

$$\text{DRAD(at } D_o) = \beta_1\big(\text{VRAD}(C_o) + \text{VRAD}(C_1)\big)(\alpha - 1)$$

$$\text{DTAN}(\ D_o\ ) = \beta_2\big(\text{VTAN}(C_1) - \text{VTAN}(C_o)\big)(\alpha' + 1)$$

and $$D\ (\ D_o\ ) = \text{DRAD}(D_o) + \text{DTAN}(D_o)$$

β_1 and β_2 are constants related to the geometry of the particular trapezium and α is the function (taken as a constant) discussed in section §4.3. α' has the same interpretation as α and is given the value zero. For a symmetric model centred at the origin O, VTAN should be zero everywhere, in practice this is not so because of the grid configuration, the choice of $\alpha' = 0$ however ensures that VTAN near to the boundary is taken as zero.

4.6 COMPUTATION OF BOUNDARY VELOCITY COMPONENTS

Both the normal and tangential velocities are computed from the implied VRAD values at the points A_i by linear interpolation. For example (see Fig. 4.2), the computation of u and v at the point ⊗ is given by:

$$u = \gamma\,\text{VRAD}(A_o) + \delta\,\text{VRAD}(A_1).\cos\phi$$

$$v = \delta\,\text{VRAD}(A_1).\sin\phi$$

where γ and δ are the interpolation factors: $[\gamma = 1 - \delta]$.

It should be appreciated that the imposition of radial symmetry is only applied weakly through the boundary values and any asymmetry in the interior is still allowed to develop.

4.7 THE COMPUTATION OF THE ISOBARIC HEIGHTS, h' ON THE BOUNDARY

As previously mentioned in Chapter III §3.5, the deviation height field h' is computed from the horizontal momentum equation:

$$g\frac{\partial h'}{\partial n} = -\frac{\partial v_n}{\partial t} + (v_L + v_L^o)\zeta - \omega\frac{\partial}{\partial p}(v_n + v_n^o) - \frac{1}{2}\frac{\partial}{\partial n}\left([v_n + v_n^o]^2 + [v_L + v_L^o]^2\right)$$

(where the zero superscript identifies the unperturbed wind field).

The RHS of this equation can be evaluated and hence also h' on the boundary using the interior value of h'.

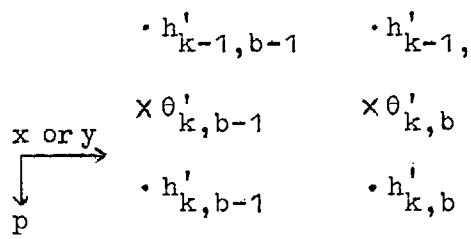
4.8 COMPUTATIONAL INSTABILITY ASSOCIATED WITH THE NON-LINEAR TERMS ON THE BOUNDARY

The non-linear term $\frac{\partial}{\partial n}\left\{(v_n + v_n^o)^2 + (v_L + v_L^o)^2\right\}$ was found to be the source of computational instability and it was therefore omitted. This instability was particularly marked when the ambient wind field was non-zero. The reason for this can be seen by writing $\frac{1}{2}\frac{\partial}{\partial n}(v_n + v_n^o)^2 = v_n\frac{\partial v_n}{\partial n} + v_n^o\frac{\partial v_n}{\partial n}$; the multiplier v_n^o thus enlarges the effect of the instability. Since $\frac{\partial h'}{\partial n}$ was found to be small on the boundary it was not felt worthwhile to look for a more sophisticated treatment of the non-linear terms.

4.9 THE COMPUTATION OF THE TEMPERATURE: θ' ON THE LATERAL BOUNDARIES

The boundary values of θ' were computed from the boundary values of h' using the thickness equation (Eq.(3.4)). This assumes the atmosphere to be hydrostatic at the boundaries, a good approximation in a region where the vertical accelerations are very small. In practice solving h' from a balance equation produced values of h' near the boundaries that contained substantially non-zero values of h'' and the resulting temperature fields on the boundary did not give good continuity with inside values computed

from the advection equation. Discontinuities in a field at adjacent gridpoints computed by different means are to be expected and a solution found to this problem was to formulate a new thickness equation in terms of the computed values at interior points as follows:-



Thus if hydrostatic:— $(h'_{k-1,b-1} - h'_{k,b-1}) \propto \theta'_{k,b-1}$, the constant of proportionality is then used to derive $\theta'_{k,b}$ from the thickness values:

i.e. $$\theta'_{k,b} = \theta'_{k,b-1} \left(\frac{h'_{k-1,b} - h'_{k,b}}{h'_{k-1,b-1} - h'_{k,b-1}} \right)$$

This procedure although subject to errors implicit in the h'-computation produced very much smoother θ' fields.

The importance of the boundary conditions on θ' and h' is reflected in the propagation behaviour of the internal gravity wave energy across the boundaries and this is discussed in §4.11.

4.10 AN ALTERNATIVE BOUNDARY FORMULATION

As referred to in §4.2, an alternative but less successful boundary condition on the motion field was tried earlier in the project. The theory used by Pearce (1972) suggests that the specification of horizontal divergence D, and the vertical component of absolute vorticity ζ at all boundary points are sufficient boundary conditions. This result was applied in the model by specifying the boundary divergence using an interpolation between the immediate interior value and an assumed zero divergence

in the extra-grid environment (c.f. treatment of v_n discussed earlier) while ζ was taken as equal to the Coriolis parameter at inflow points and computed assuming continuity with inside values at outflow points. The normal and tangential velocities were computed from the specified D_{bdy} and ζ_{bdy}.

When tried on the (7 x 7) model, these boundary conditions proved to exercise insufficient control on the normal velocity.

The choice of $\alpha < 1$ implies that the divergence and hence the vertical motion goes to zero at some specified distance in the environment. The normal velocity is not however constrained similarly.

4.11 INITIAL TESTS OF THE MODEL

During the development of any numerical model there is a continuing requirement for rigorous testing of the computer program and modelling procedures and techniques. This is particularly true of three-dimensional models where there is increased programming associated with the extra dimension.

Two useful tests were performed on the model; one as a straightforward consistency check on the programming, and the other as a test of the model itself under reasonably well-defined conditions of no ambient wind and a water-vapour free atmosphere.

The programming check consisted of running the models with no ambient wind and zero Coriolis parameter. This allowed it to evolve under conditions of symmetry in a Cartesian coordinate system, any errors leading to asymmetry immediately becoming apparent. It is felt that some equivalent type of testing is a very necessary part of program development, particularly in models which use real data, as any non-symmetries which are

induced by errors can easily remain hidden using highly unsymmetric real data fields.

The second test of the model discussed in §4.12 proved to be an interesting study in its own right.

Tests of the microphysics and thermodynamic sections of the program were carried out separately prior to their inclusion in the full model.

4.12 THE DRY MODEL

The model was run with no water vapour the convection being initiated by a heating function applied during the early stages of the integration. This heating perturbation (0.01°C/sec for 3 minutes at a group of gridpoints —see Chapter VI, §6.1) forced an internal gravity oscillation and the behaviour of this oscillation in time was examined. The oscillation frequency can be derived theoretically for the linearised system (see Chapter II, §2.3); the amplitude behaviour - its damping in time with the corresponding energy propagation away from the initial source is considered in the following section, and comparisons of oscillation characteristics in the model are then made.

4.13 ENERGY TRANSMISSION ASSOCIATED WITH GRAVITY WAVE PROPAGATION

From the linearised equation system studied in Chapter II internal and external (free-surface) gravity waves are supported in the model, the external wave being a limiting case of the internal mode. The rate of transmission of energy by waves in a fluid is equal to the waves' group speed, c_g where $c_g = \partial\sigma/\partial\lambda$, σ being the frequency and λ the wavenumber; hence for the external wave — $\sigma^2 \sim \lambda^2 gH$ — the horizontal phase and group speeds are coincident.

For internal wave propagation $\sigma^2 \sim gB\lambda^2/(\lambda^2 + \nu^2 + \frac{1}{4H_\rho^2})$

giving $c_g \sim \left(\frac{gB}{\lambda^2 + \nu^2 + \frac{1}{4H_\rho^2}}\right)^{1/2} \left(1 - \frac{\lambda^2}{\lambda^2 + \nu^2 + \frac{1}{4H_\rho^2}}\right) = c_{ph}\left(1 - \frac{\lambda^2}{\lambda^2 + \nu^2 + \frac{1}{4H_\rho^2}}\right)$

It can readily be seen that for waves with $\lambda/\nu \gg 1$, $c_g \ll c_{ph}$ ($c_g \longrightarrow 0$ as $\lambda/\nu \longrightarrow \infty$). Oscillations whose vertical wavelength is much larger than their horizontal wavelength (tall, thin columns) propagate energy away only very slowly and thus exhibit slow amplitude damping in time. For very much broader oscillations $c_g \longrightarrow c_{ph}$ and rapid damping results.

It is interesting to note that internal waves in a hydrostatic system do not exhibit this horizontal wavelength dependence but have a group speed equal to their horizontal phase speed.

For gravity waves excited by convection in the atmosphere, λ/ν will in general be greater than or comparable to unity and hence the waves have a group speed several times smaller than their phase speed. The main region of influence of the convective circulation will be governed by the group energy propagation and will therefore be quite small during the clouds lifetime. This suggests that compensating downdrafts associated with a cumulus are contained within a diameter or so of the cloud edge; there are difficulties however in observing such an effect.

4.14 CLOSED BOUNDARIES

For the model run with closed lateral boundaries (i.e. no flow normal to the boundaries), the only energy lost from the system is from internal dissipation associated with the numerical scheme used. Fig. 4.3 (curves A) shows the gravity oscillation in the 500 mb ω' and θ' fields with closed boundaries. The temperature field oscillation is virtually undamped with a period of $\sim$10 mins although with a tendency for the period to shorten a little in time,

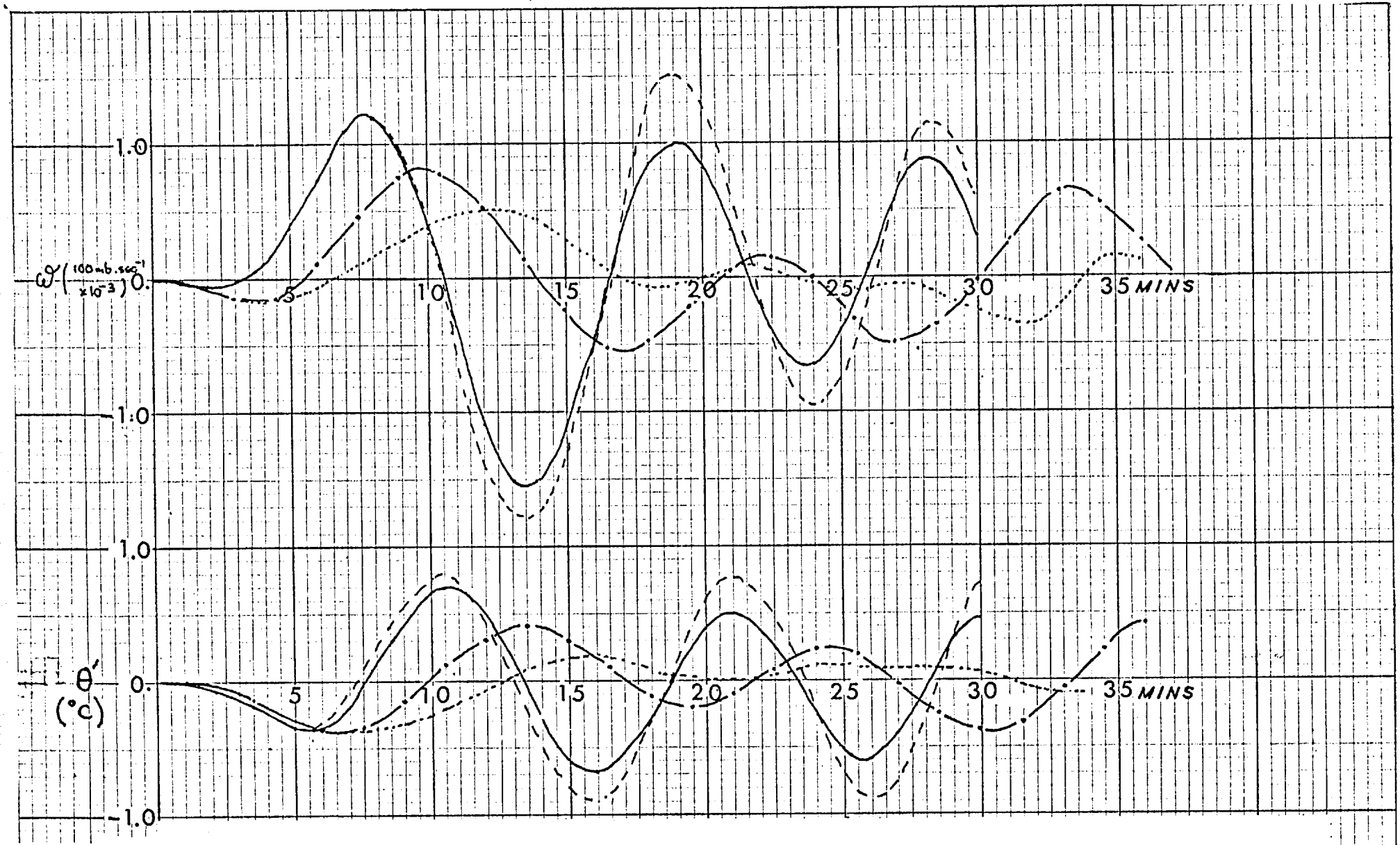


Fig. 4.3 Oscillations in 500 mb ω and θ' fields on the axis of symmetry $[\theta'_{500} = \frac{1}{2}(\theta'_{400} + \theta'_{600})]$

(A) Closed boundary ---------
(B) Open boundary ———— } L ~ 4 km (L = horizontal wavelength)
(C) Open boundary —·—·—·— L ~ 12 km
(D) Open boundary ····· L ~ 20 km

and demonstrates the expected difference in phase from that of the vertical velocity oscillation ($\frac{1}{4}$ period). The ω-oscillation is more irregular although its period follows closely the characteristics of the θ' one. The varying amplitude is due to the interaction of several vertical modes together with propagating and reflecting components. Most of these irregularities are removed from the θ' oscillation in forming the average of θ'_{400} and θ'_{600} to obtain θ'_{500}.

The extent of non-linearities associated with the finite-amplitude oscillation was found to be small when very small amplitude oscillations were compared with the ones shown here.

The model oscillation period is in good agreement with the linearised analysis solution (for an unbounded fluid) derived in Chapter II, §2.3 evaluated using model values for wavenumbers and static stability.

The application of vertical smoothing (filtering) referred to in Chapter II, §2.3 was observed to lengthen the oscillation period by $\approx$20%. For a linearised system containing only the internal gravity mode, the period for the vertically smoothed oscillation is approximated by:

$$\tau \approx \tau_o \left(1 + \frac{5K^2}{H_\nu^4} \tau_o^2\right)$$

where τ_o is the unsmoothed model oscillation, H_ν = vertical wavelength and K = smoothing coefficient. (This result is derived by including in the equations containing internal waves a diffusion term of the form $K' \frac{\partial^2}{\partial p^2}(\ )$ and carrying out a linearised analysis.)

Evaluating this for the smoothed model oscillation ($K \sim 10^5 m^2 s^{-1}$) gives a period increase comparable to the modelled increase.

4.15 OPEN BOUNDARIES

With open boundaries, the λ-dependence of the amplitude damping is illustrated by curves B, C, and D in Fig. 4.3 . For identical initial forcing, curves A and B demonstrate the small energy propagation of the open boundary case (B) compared to the closed boundary case (A) for large λ/ν . The curves C and D show the more rapid damping for longer wavelengths.

There is evidence of interference from reflected waves at the later stages (30+ minutes) in curve C indicating that the boundary formulation does not transmit the propagating gravity waves completely; however since the group speed of radiating waves is small, the reflected energy will not interfere significantly with the convection during its lifetime.

As a consequence of the gravity wave scales in the convective model ($\lambda/\nu \sim 1$ or larger), the oscillations are substantially maintained over the cloud lifetime and the effect of these oscillations on the development of motions is significant; this is discussed in Chapter VI.

CHAPTER V

THERMODYNAMICS AND CLOUD PHYSICS

This chapter outlines the thermodynamics and cloud physics simulated in the model. It is shown that the use of pressure as a vertical coordinate yields considerable advantages in the latent heating and cooling computation.

Having adopted a cloud physics parameterisation it was found necessary to experiment with the numerical values of the parameters involved in order to obtain reasonable cloud and rainwater contents. The details are discussed near the end of the chapter.

5.1 CONDENSATION AND EVAPORATION

As the model is for deep convection (implying 'large' deviations of variables) a method was developed to handle the latent heat processes with a reasonably high accuracy.

The potential temperature and specific humidity of the air are calculated at each gridpoint using the simplified forms of Eqs.(2.1.4 and 2.1.5)

$$\frac{D\theta}{Dt} = 0 \qquad \text{——(5.1)}$$

$$\frac{Dq}{Dt} = 0 \qquad \text{——(5.2)}$$

which assume no mixing by small-scale processes.

Thus
$$\Delta\theta = -\Delta t\left\{ \underset{\sim}{v}_2 \cdot \nabla\theta' + \omega \frac{\partial\theta}{\partial p} \right\}$$

$$\Delta q = -\Delta t\left\{ \underset{\sim}{v}_2 \cdot \nabla q + \omega \frac{\partial q}{\partial p} \right\}$$

and hence
$$\theta(t+\Delta t) = \theta(t) + \Delta\theta$$
$$q(t+\Delta t) = q(t) + \Delta q$$

(Actual computation of $\Delta\theta$ and Δq in the model is however by a different time-integration scheme used.)

At each timestep q is compared with $q_s(\theta)$ —the saturated value at potential temperature θ, if $[q - q_s(\theta)] \leq 0$ no adjustment is made, however if $[q - q_s(\theta)] > 0$, the specific humidity and temperature are adjusted to saturation with production of liquid water. The evaporation process is handled in an exactly analogous manner.

5.2 THE ADJUSTMENT PROCEDURE

From the 1st Law of Thermodynamics we have, for the process outlined above:

$$L\,dr = c_p\,dT + r\,c_{p_v}\,dT - V\,dp \qquad \text{———(5.3)}$$

where c_p, c_{p_v} are the specific heats of air and water vapour respectively and r is the mixing ratio, other symbols follow the usual notation. Eq.(5.3) can then be written to a high degree of approximation as:

$$L\,dq \doteqdot c_p\,dT - V\,dp$$

where the approximation is somewhat better than writing

$$L\,dr \doteqdot c_p\,dT - V\,dp$$

In pressure coordinates with gridpoints on pressure surfaces the assumption that the latent heat release occurs at constant pressure is almost exact, the only error introduced being due to the pressure changes of the air during the finite timestep of the model. Errors always occur in applying such an equation to mean values over a grid volume, but these, a function of the grid resolution are unavoidable. Hence we use the relationship

$$L\,dq \doteqdot c_p\,dT \qquad \text{———(5.4)}$$

Thus for one timestep

$$L(q - q_s(T_F)) = c_p(T_F - T) \qquad \text{——(5.5)}$$

(suffix F denotes the value after adjustment)

We now make use of the Taylor expansion:

$$q_s(T_F) = q_s(T) + \Delta T q_s'(T) + \frac{(\Delta T)^2}{2!} q_s''(T) + \frac{(\Delta T)^3}{3!} q_s'''(T) + O(\Delta T)^4$$

where the prime denotes differentiation w.r.t. T, and $\Delta T = T_F - T$.

Substituting in Eq.(5.5) gives:

$$q - \left\{ q_s(T) + \Delta T q_s'(T) + \frac{(\Delta T)^2}{2!} q_s''(T) + ... \right\} = \frac{c_p}{L} \Delta T \qquad \text{——(5.6)}$$

We can obtain the derivatives of $q_s(T)$ from the Clausius-Clapeyron (C-C) equation and hence solve Eq.(5.6) for ΔT. $q_s(T)$ was computed from Tetens (1936) formula approximating the integrated C-C equation:

i.e. $$e_s(T) = 6.11 \times 10^{\left(\frac{aT}{T+b}\right)} \qquad [T \text{ in } °C] \qquad \text{——(5.6a)}$$

together with —

$$q_s(T) = \frac{\epsilon . e_s(T)}{p - e_s(1-\epsilon)} \qquad \epsilon = 0.622$$

The constants a and b are taken as:

$a = 7.5 \quad b = 237.5 \quad T \geq 0$

$a = 9.5 \quad b = 265.5 \quad T < 0$

5.3 ACCURACY

The accuracy of solution of equations concerned with the latent heat release has to be examined carefully as any systematic errors can have substantial effect on the buoyancy and hence on the dynamics.

Eq.(5.6) can be written as

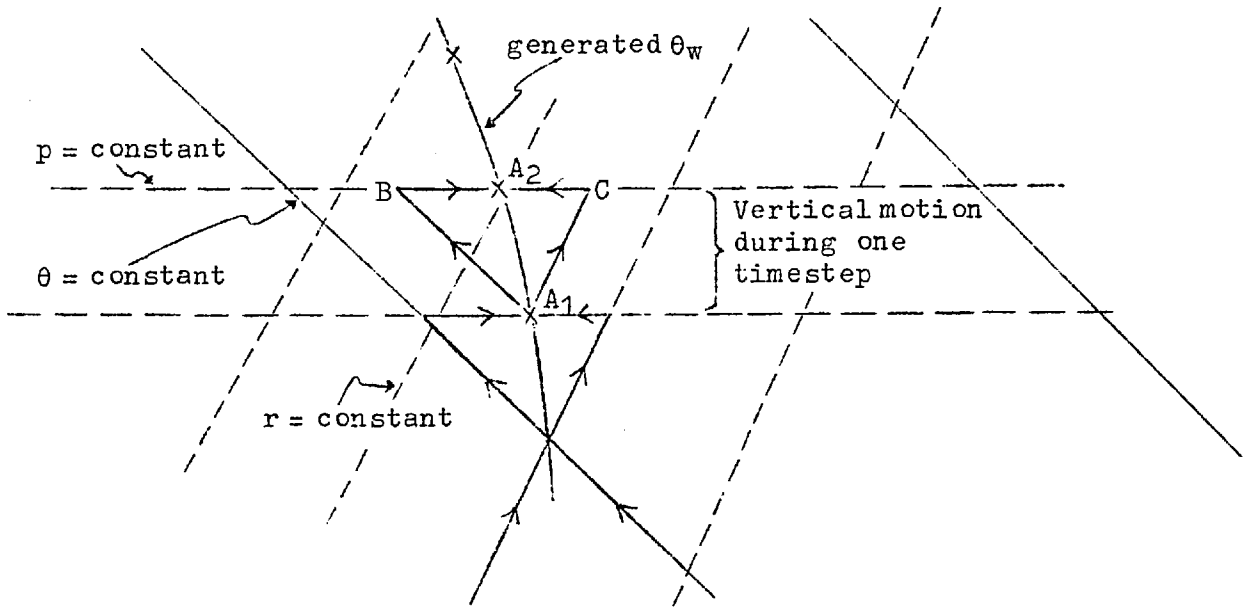
$$a_o + a_1 \Delta T + a_2 (\Delta T)^2 + a_3 (\Delta T)^3 + ... = 0 \qquad \text{——(5.7)}$$

where $a_o = q_s(T) - q$; $a_1 = c_p/L + q_s'(T)$; $a_2 = \frac{1}{2} q_s''(T)$ etc.

and in general $a_i >> a_{i+1}$

$i = 0, 1, 2, \ldots$

Eq.(5.7) is solved in some truncated form at every timestep. The associated ΔT is thus, in general, small ($|\Delta T| << 1$), especially for short timesteps and the relative sizes of the a_i are such that the equation can be truncated after only a few terms. For much of the saturated region the linear form of Eq.(5.7) would be accurate to one per cent, but in the extreme case of updraft speeds ~ 50 m.s^{-1}, $\Delta T \sim 1 - 2°C$, so that in practice the quadratic form of Eq.(5.7) was used.



DIAGRAMMATIC REPRESENTATION OF CONDENSATION COMPUTATION

With reference to the sketch above:

i) Solving Eq.(5.1) corresponds to path A_1B

ii) Solving Eq.(5.2) corresponds to path A_1C

iii) The adjustment procedure then locates point A_2.

The thermodynamic process assumed in this calculation is essentially the pseudo-wet adiabatic one i.e. that in which the

heat capacity of the liquid water is neglected together with the effect of freezing. These two effects are subtractive and in general contribute only a small error. In tropical deep convection, in which the errors are largest, it is possible for the representation of the thermodynamic process to be in error by $\approx 2°C$ through the full depth of troposphere.

5.4 COMPUTATION OF $e_s(T)$

The computation of the saturated vapour pressure e_s (w.r.t. water) is a necessary part of modelling a moist convective system and poses an interesting modelling problem first discussed by Ogura and Phillips (1962).

The dependence of e_s on temperature is given by the Clausius-Clapeyron equation:

$$\frac{de_s}{dT} = \frac{L_v}{R_v} \cdot \frac{e_s}{T^2} \qquad \text{——(5.8)}$$

where L_v = Latent heat of condensation

R_v = Gas constant for water vapour

T = Absolute temperature

This equation can be integrated to give:

$$\ln(e_s) = -\frac{L_v}{R_v}\frac{1}{T} + \text{constant} \qquad \text{——(5.9)}$$

Consider an atmosphere horizontally stratified, then we define temperature, pressure and potential temperature by p_o, $T_o(p_o)$, $\theta_o(p_o)$. Then for the actual modelled system we can consider these variables in the form:

$$p = p_o + \Delta p$$

$$T = T_o + \Delta T \qquad \text{where } \left|\frac{\Delta p}{p}\right|, \left|\frac{\Delta T}{T}\right|, \left|\frac{\Delta \theta}{\theta}\right| << 1 \ .$$

$$\theta = \theta_o + \Delta\theta$$

Expanding Eq.(5.9) and neglecting terms $O(\Delta T^2)$ and higher, we obtain after suitable manipulation:

$$e_s(T) \approx \frac{e_s(T_o)}{\exp\left(-\frac{L_v}{R_v T_o}\right)} \left\{ \exp\left(-\frac{L}{RT_o} + \frac{L}{RT_o}\frac{\Delta T}{T_o} \right) \right\}$$

[Eq.(5.6a) is essentially analogous to this equation and the following comments apply to it.]

Thus we have:

$$e_s(T) \doteq e_s(T_o) \exp \alpha \frac{\Delta T}{T_o} \qquad \alpha = \alpha(T_o)$$

where $\frac{\alpha \Delta T}{T_o} << 1$ in general

Expanding the exponential and neglecting terms $O(\Delta T^2)$ and higher gives

$$e_s(T) - e_s(T_o) = e_s(T_o)\alpha \frac{\Delta T}{T}$$

From the definition of potential temperature

$$\frac{\Delta T}{T_o} = \frac{\Delta \theta}{\theta_o} + \kappa \frac{\Delta p}{p_o}$$

Thus the computed departure of saturated vapour pressure from the undisturbed temperature value (T_o) is approximated by:

$$e_s(T) - e_s(T_o) \doteq e_s(T_o)\alpha(T_o) \left\{ \frac{\Delta \theta}{\theta_o} + \kappa \frac{\Delta p}{p_o} \right\} \qquad \text{——(5.10)}$$

It is common in computer models to use θ as the temperature variable so that when computing $e_s(T)$ it is necessary to convert the θ gridpoint values to temperatures. If the gridpoints are at fixed heights then T must be obtained from θ using the pressure at that gridpoint. The simplest value to take for the pressure is p_o —the undisturbed value. This by examination of Eq.(5.10) gives a large inaccuracy as $\kappa \frac{\Delta p}{p_o}$ in deep convection can be comparable in size to $\frac{\Delta \theta}{\theta_o}$.

The true T should therefore be calculated using the total pressure field at a gridpoint. This total pressure field is computed from a balance equation which in itself is temperature dependent; thus the computation involves a highly complex, implicit set of relationships.

Researchers have not found suitable methods of handling this problem satisfactorily and it is a major deficiency of existing deep convection models; Takeda (1971) refers to this problem especially, as he effectively uses the unperturbed pressure field p_o in computing the condensation rate.

A possible solution investigated by Murray (1970) is that of using Lagrangian techniques for the thermodynamics. This would pose some problems in three-dimensions however.

When the coordinate system is such that the gridpoints are in surfaces of constant pressure, Δp is by definition, zero. The problems of computing Eq.(5.10) discussed above do not therefore arise.

5.5 CLOUD PHYSICS MODELLING

The earlier part of this chapter has examined the process of condensation (and evaporation) of water vapour to cloudwater. The computer model could be run with only this process simulated so that all liquid water condensed took no further part in the dynamics (other than re-evaporation at the place of condensation).

This is obviously not a justifiable assumption on any grounds other than simplicity when modelling deep convection as the conversions of cloudwater to rainwater and the subsequent evaporation and drag of the raindrops all play an essential part in the life-history of a shower-cloud as will be seen in Chapter VI.

5.6 PARAMETERISATIONS

The parameterising of these micro-physical processes in a form suitable for inclusion in numerical models has been studied by several researchers and was reviewed by Kessler (1969).

Kessler, Das (1964), Srivastava (1967), Weinstein (1968) and Simpson (1969) all tested their parameterisations on one-dimensional cloud models.

Two dimensional models have also been developed with microphysics included, notably by Arnason et al (1968), Takeda (1966, 1971), Orville and Liu (1969) and Orville and Sloan (1971).

It was not the immediate aim of the work described in this thesis to study and experiment with parameterisations (although the models developed will provide the working basis for such research), and so a parameterisation already developed and tested, that of Liu and Orville (1969) was adopted. This parameterisation was originally developed for two-dimensional, shallow cumulus clouds (< 3.5 km) but has recently been extended to models of 10 km depth. In applying this parameterisation to the model described in this thesis, alterations were made to adapt to deeper clouds as discussed later.

The parameterisation was formulated in height coordinates originally and this was retained for speed of adaptation in the model [using unperturbed height field values]. This will be remedied as soon as possible although no significant errors are introduced.

Consider the partition of water substance into three parts:-

Water vapour — r (mixing ratio)

Cloud water — l_c

Rain water — l_r , all in units of $gm.kg^{-1}$

then making the assumption that water vapour and cloudwater are advected together gives:-

$$\frac{D}{Dt}(r + l_c) = - PROD \qquad \text{——(5.11)}$$

where PROD is a production term discussed in more detail in §5.9.

The equation for continuity of rainwater leads to

$$\frac{Dl_r}{Dt} = PROD + \frac{\partial}{\partial z}(l_r V_t) + l_r V_t \frac{1}{\rho}\frac{\partial \rho}{\partial z} \qquad \text{——(5.12)}$$

where V_t is the mean terminal velocity of the rainwater drops relative to the air.

V_t is parameterised in the form

$$V_t \approx 5.32\, l_r^{1/5} \ m.s^{-1} \qquad \text{(with } l_r \text{ in } gm.kg^{-1}\text{).}$$

For derivation and comparison with alternatives see Liu and Orville (1969). This concept of a mean terminal velocity is not very satisfactory as the variation in fall-speeds due to the range of drop sizes is a not insignificant property affecting a cloud's life-cycle.

Takeda (1971) has improved on the above parameterisation by computing the space (2-D) and time variation of the size distribution of drops in seven discrete radii ranges, the smallest being cloud droplets.

5.7 THE PRODUCTION TERM — PROD

This was first introduced by Srivastava (1967) and can be written:-

$$PROD = P_1 + P_2 + P_3$$

where P_1 parameterises the evaporation of raindrops

P_2 parameterises the accretion of l_c by l_r

P_3 parameterises the conversion of l_c to l_r.

All three parameterisations are derived from the physics of a single drop together with assumed raindrop spectra.

P_1 :

Written as:- $P_1 = \beta (r - r_{sat})$, where $(r - r_{sat})$ is the saturation deficit. β is a parameter with a wide range of values and should, for deep convection especially, be expressed as a function of its dependent variables i.e. l_r, r_{sat}, spectrum properties, Reynolds No. etc. A typical value of β is $\sim 10^{-3} sec^{-1}$.

P_2 :

Written as:- $P_2 = \gamma l_c l_r^{0.95}$. Again the form of γ is complex and in particular contains air density dependence and a collection efficiency coefficient. Values of γ used were in the range 10^{-2}—10^{-3}. sec^{-1}. This parameter proved to be very important and difficulty was found in maintaining reasonable cloud and rainwater water concentrations at the same gridpoint i.e. the presence of rainwater working through process P_2 tended to dry the cloud out rapidly.

This situation would be improved with a better terminal velocity parameterisation.

P_3 :

Written as:- $P_3 = \alpha(l_c - l_{crit})$ where l_{crit} is a critical cloudwater value below which there is no auto-conversion. A typical value of α is 10^{-3} sec^{-1}. The choice of α was obviously important as too small a value delayed the formation of significant rainwater values ($\alpha = 0 \longrightarrow$ no rainwater); the precise value chosen for l_{crit} however had little effect on the large clouds being modelled.

The assumptions made in this precipitation parameterisation are:

(a) Waterdrops in an unsaturated region evaporate until either they disappear or the region becomes saturated.

(b) No freezing, sublimation or melting occur.

For deep convection in general (b) is a poor assumption, however if the situation modelled is a tropical one, as in this model, only small errors will be introduced.

5.8 VALUES OF α, β, γ and l_{crit}

In practice these parameters were adjusted to give reasonable cloud and rainwater concentrations i.e. the model was developed purely as a simulation of the atmosphere. The deficiencies of the parameterisations become more marked in deep convection when the cloud is bigger and wetter, and when the microphysical processes occur very much quicker than in the cumulus cloud models for which the parameterisation was originally developed.

There is a need for improvement in this part of the modelling technique. The adaptation of the parameterisations to include the air density dependence explicitly is a good case in point.

5.9 RAINFALL

In view of the limitations of the vertical resolution and of the parameterisation, the rainfall rates and totals are only a guide to the model's ability to cope with this part of the cloud simulation and are formulated to check that the rainwater is removed from the model at a realistic rate.

5.10 SUMMARY OF THE CLOUD PHYSICS

The development and sophistication of the microphysics has not been a major concern of this study and while the basic processes have been included there is scope for considerable improvement. With better spatial resolution the model would however provide a stepping-off point for a study of microphysical modelling and its effect on the convection in ways which one or two dimensional models cannot.

As will be seen in Chapter VI, the cloud physics plays a large part in determining the dynamics of the cloud and this stresses the need for sophistication of the parameterisations.

CHAPTER VI

INITIALISATION AND DISCUSSION OF SIMULATION RESULTS

In previous chapters the mathematical foundation of the model has been given in detail. Before examining the model's ability to simulate the life-cycle of a cumulonimbus system in sheared and unsheared environments, the initiation process is discussed.

6.1 INITIALISATION

For a model which endeavours to examine the life-cycle of a cumulonimbus cloud, the evolution of the early, small, thermal-like cumulus clouds cannot be modelled in detail because the resolution of the grid adopted for the former is too coarse for the latter. Consequently the initial atmosphere, horizontally stratified, was perturbed in the lowest layers on the scale of a cumulonimbus cloud. This has the added advantage of compressing the early organising stages of the convective motion into a short computational period. The importance of the dimensions of initial perturbations has been examined by several authors, Takeda (1971) in particular shows convincingly that the dimensions have very little effect on the properties of the fully developed cloud although, as one would expect, the early development exhibits more substantial dependence. This dependence on the dimensions of the initial disturbance is an example of interaction between the convective motion and internal gravity oscillations and this interaction will feature in later discussions.

A: POSSIBLE FORMS OF INITIAL PERTURBATION

It has been customary for cumulus convection models to be initiated by a localised warm 'bubble', i.e. a region of excess temperature, usually ~ 0.5-$1.0^{\circ}C$, situated in the lower part of the model atmosphere. This buoyancy induces rising motion and, if of

sufficient magnitude, subsequently leads to latent heat release due to condensation. The warm 'bubble' however has smaller relative humidity than the environment at the same level and thus delays condensation. Murray (1970) suggests that an initial moisture 'bubble' is more realistic; this produces buoyancy through the density deficit. If the bubble relative humidity is high, condensation heating will dominate very quickly. In view of the proposed initiation on the scale of a cumulus both types of impulse would appear to be relvant as they can bring the model to the point where latent heating dominates beyond which stage a strong convective circulation develops.

B: THE INITIALISATION ADOPTED IN THE MODEL

The initial horizontally stratified atmosphere can be chosen arbitrarily with input to the program of potential temperature, specific humidity (or mixing ratio) and wind components at the pressure levels specified.

For the simulations described in this thesis the sounding shown in Fig.6.1 was used (apart from one exception which will be discussed later). Originally the sounding was that of a mean tropical (Caribbean) sounding (Jordan, 1958). Two modifications were however made: firstly the available potential energy for low-level parcel ascent was reduced above 400 mb to allow the 'overshoot' region of air ascending from low-levels to be better resolved. The second modification was necessary in view of the finite-difference interpretation of the sounding, the temperature and moisture values at 925 mb represent a mean for the 850-1000 mb layer; as far as the energetics of this low-level air is concerned, the temperature and moisture values at 925 mb should represent the mean θ_w (pseudo-wet-bulb potential temperature) value for

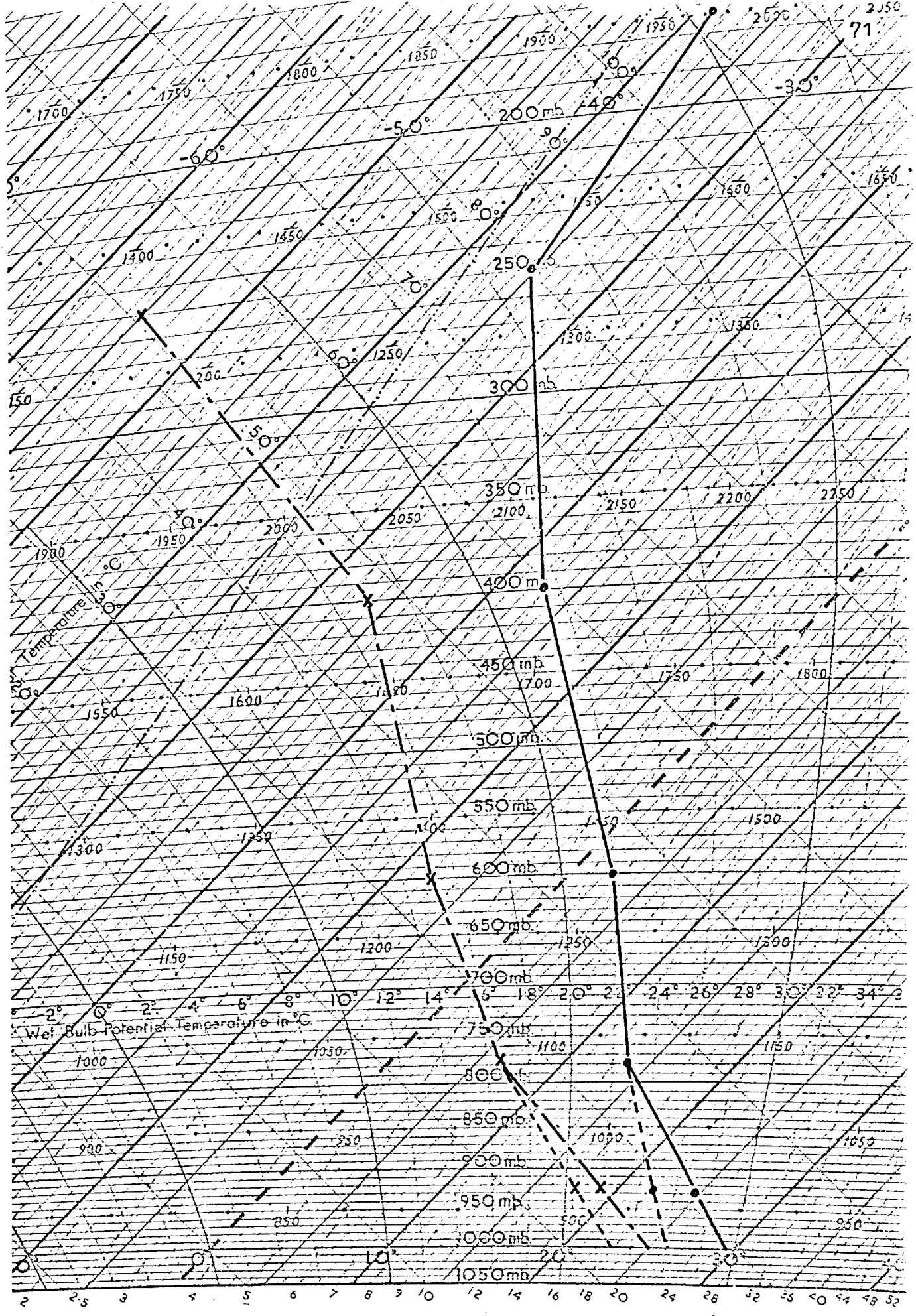


FIG. 6.1
MODEL SOUNDING (DOTTED DEPARTURE SEE TEXT)

this layer on the original sounding. The 925 mb values were therefore adjusted accordingly. This same adjustment could have been applied to all other levels but since the misrepresentation has much less important consequences at these levels this was not done.

The initial perturbation was formed by raising the specific humidity 1 $gm.kg^{-1}$ above the undisturbed value in a 'box' two kilometres square occupying a depth corresponding to a 150 mb pressure difference. An artificial heat source was also assumed within this box to force the convective motion.

The energy input of the applied temperature and humidity perturbations should be small relative to the available energy, since otherwise the resultant motion would become too dependent on the details of the initial perturbation.

An example of this effect appeared when using the sounding for which the low-level adjustment had not been carried out (dotted lines in Figure).

This sounding represents an atmosphere with little available potential energy for air ascending from low-levels, in which deep moist convection is unlikely to occur naturally. The effect of low-level perturbations in temperature and moisture is to excite gravity oscillations. In view of the discussion on energy propagation in Chapter IV, §4.13, these oscillations tend to persist, their amplitude being proportional to the perturbation heating (Fig.6.2). Under certain conditions the life-cycle of a cloud will be seen to be linked with the period of the gravity oscillation, although the interaction will only be significant when the energy of the oscillation is comparable with the available potential energy as measured by the 'positive area' on a tephigram i.e. when the latter is small.

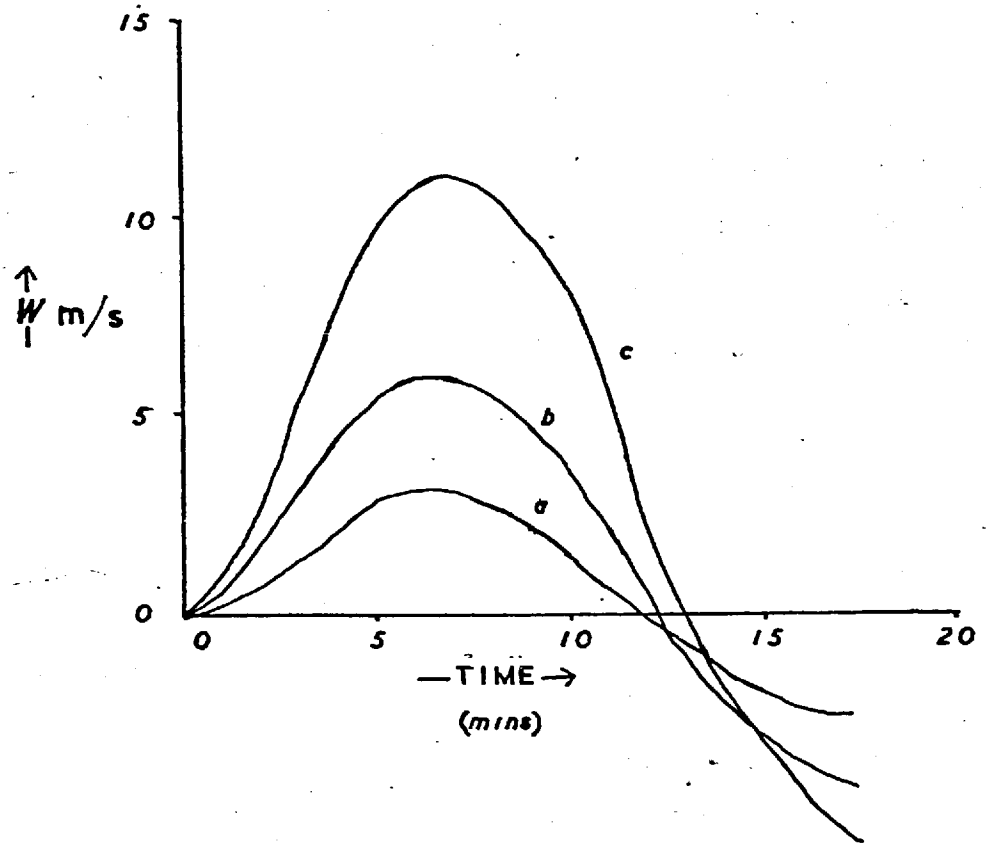


Fig. 6.2 Axis vertical velocity at 700 mb for different magnitudes of initial heating function

(a) 0.01, (b) 0.02, (c) 0.04 $^{\circ}C.sec^{-1}$

6.2 INTERPRETATION OF RESULTS

The numerical solutions obtained by simulations such as those presented here must be physically interpreted with care, for the results contain not only the physics and dynamics of the phenomenon under investigation but also effects due to the model's limitations associated mainly with numerical truncation and the modelling assumptions; although the modeller naturally tries to minimise these effects. An example of this was given by Murray (1970) who compared cloud simulations from 2-D rectangular Cartesian and axially-symmetric models. The differences in model cloud properties were large (up to 100%) and since the same physical system was modelled in the two cases the differences must be interpreted as being associated with the properties of the models.

The three-dimensionality of the model presented here minimises purely geometrical effects but direct physical comparisons of results must still be carried out with caution.

6.3 DIAGRAMS

For convenience of presentation the diagrams use a linear scale of pressure in the vertical, thus distorting the vertical dimension (as shown in Fig.(i)); this distortion is not however very large.

The use both in diagrams and text of vertical velocity - w rather than the pressure tendency - ω is for ease of assimilation. w is computed from the relationship:

$$w = \frac{Dh}{Dt} = \frac{\partial h'}{\partial t} + \underset{\sim}{v}_2 \cdot \nabla h' + \omega \frac{\partial h}{\partial p}$$

w is computed at the points of the (15 x 15) grid i.e. the (u,v) ones,

allowing vector representation of the winds in a given plane. This computation of w does however represent a smoothing of the vertical motion field since the ω used in the above equation is a four-point average. This is somewhat undesirable but a problem inherent in staggered grid systems. The wind vector diagrams of flow in the plane of the paper are given for selected stages in the clouds history, arrow lengths being proportional to the wind speed.

The machine-plotted contours [red > 0; blue < 0] are based on linearly interpolated values at every 100 mb level. The precise grid of points shown is identified by the bracket: $[x,y=y_n,p]$ or $[x=x_n,y,p]$ or $[x,y,p=p_n]$ where the suffix n is the m^{th} gridpoint of that dimension (e.g. $[x,y=y_n,p]$ is the (x,p) plane at a value $y=y_n$).

6.4 SIMULATION RUNS

Four simulation runs were completed, designated RUNS 1 - 4. Each run had the same environmental temperature and moisture profiles (as shown in Fig.6.1) but with the ambient wind profiles as follows:-

RUN 1 No ambient wind

RUN 2 No ambient wind (but cloud physics parameterisation different from (1))

RUN 3 Constant vertical shear $(\partial u/\partial p) = 20\,m.sec^{-1}\,(850\,mb)^{-1}$ the ambient wind assumed as in Fig.(ii) below [u-component] and v-component zero

RUN 4 As (3) but with the wind at 1000 mb turned through 45° i.e. a v-component at 1000 mb equal to the u-component at that level and with the v-component zero at all higher levels.

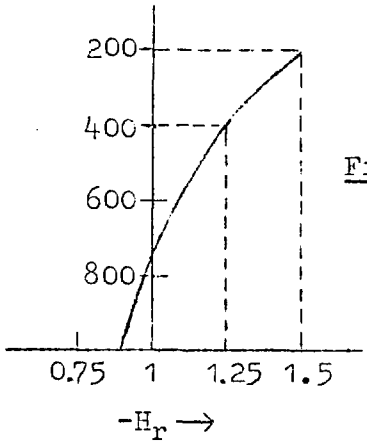


Fig.(i)

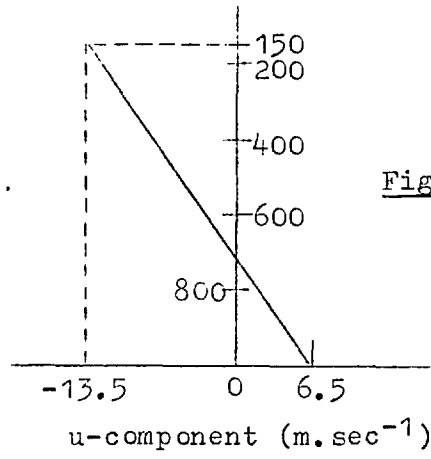


Fig.(ii)

where $H_r = \frac{\text{True vertical scale}}{\text{Implied vertical scale in diagrams}}$

6.5 DESCRIPTION OF RUN 1

(No ambient wind; Coriolis parameter zero; cloud physics parameters $\alpha = 10^{-3}s^{-1}$; $\beta = 10^{-3}s^{-1}$; $\gamma = 10^{-3}s^{-1}$)

The evolution of the convection is presented in Figs.6.3 - 6.19. Cloud first forms at about 7½ mins when the maximum updraft speed is ≈ 5 m.s^{-1}. These early development stages are characterised by low pressure (negative deviation height field) under the updraft (inflow) and high pressure (positive deviation height field) at outflow (e.g. Fig.6.3). A gravity oscillation in the vertical is also apparent. Rainwater first forms at 775 mb at $\approx$15 mins with cloudwater values of ~ 4 gm.kg^{-1}; at about this time the largest vertical velocities and temperature excesses are found — ≈ 10 m.s^{-1} at 700 mb and +1.9°C at 600 mb. From 15 mins very rapid changes occur under the dominating influence of the liquid water load and rainwater evaporation; the fields at 17½ mins show this effect clearly with evaporative cooling of the air coming into the updraft and the reversal of the vertical deviation pressure gradient. Inflow previously at both the 1000 and 850 mb levels soon ceases at 1000 mb with inflow largest at $\sim$ 700 mb thereafter.

By 20 mins only the top of the cloud contains rising air - being free of significant water loading, the cloud is at its maximum size at this time. The period 20-25 mins sees the development of a deep downdraft with heavy rainfall down to the surface. This downdraft extends from ~ 500 mb to the surface with as much as a 2°C temperature deficit through the lowest 200 mb. A maximum height deviation at 1000 mb of 8 m ($\cong$ 0.8 mb) is produced with the low-level outflow attaining 10 $m.s^{-1}$. Most of the downdraft air originates from the 500-700 mb layer where there is a pressure minimum and the minimum θ_w of the sounding, a result consistent with the concept of downdraft air having middle-level origins (Ludlam, 1963).

By 30 mins, the downdraft has reduced to 4 $m.s^{-1}$ with cloud water at upper levels only (~ 0.5 $gm.kg^{-1}$); below this cloud the warming of air due to descent now exceeds any evaporative cooling as the cloud has rained out and positive potential temperature deviations are apparent except at the lowest level. About 5 mm of rainfall occurs in the period 18-30 mins.

Fig.6.20 shows in more detail the evolution of the cloud properties on its axis of symmetry.

6.6 IMPORTANCE OF CLOUD PHYSICS PARAMETERISATION

The results of RUN 1 clearly show the dominating control of the cloud physics particularly in the later stages; in view of this, a closer analysis of the relationship in time and space of the cloud and rainwater concentrations was made. Although both liquid water variables contribute to the dynamics through a drag term in the equations, the cloudwater is assumed to advect with the flow only, while the rainwater has a terminal velocity component as well as flow advection. The changing proportions

of both liquid variables are therefore important to the overall liquid distribution in the cloud. RUN 1 showed the onset of rain to be delayed with the transformation of cloud to rain allowing values of cloudwater in excess of 2 gm.kg^{-1} to exist until after 23 mins with a maximum value $\sim$5 gm.kg^{-1} in the earlier stages. This has the effect of spreading the water load over a deep layer eventually resulting in the deep downdraft observed.

The dominating coalescence process was therefore accelerated (by an exaggerated amount) in RUN 2. The equivalent parameterisation used by Kessler (1969) suggested a larger value (for deep convection) of the accretion parameter γ.

As far as the evaporation parameter (β) was concerned the modelled surface temperature deficits would appear to be reasonable, although less evaporative cooling would reduce the large downdraft speeds.

6.7 RUN 2 [Figs.6.21 - 6.34]

(As RUN 1 but with the accretion parameter $\gamma = 10^{-2}s^{-1}$)

Until rain forms there are of course no differences from the results of RUN 1, results are therefore presented from T = 15 mins. The differences from this point are quite substantial; as expected much larger rainfall concentrations are produced at low levels with cloudwater values < 1 gm.kg^{-1}; buoyancy with ascent is maintained above 700 mb up to 30 mins. The development of a downdraft is again marked but limited to the lowest 200 mb; although containing a slightly larger temperature deficit ($\sim$ -2.5°C) the downdraft is somewhat smaller in magnitude principally through its smaller vertical extent. In both RUNS 1 and 2, a ring of 'cold' air spreads out at the lowest levels. The rainfall duration

is considerably longer in RUN 2 resulting in three times as much total rainfall as in RUN 1.

For RUNS 3 and 4, a value for γ of $2 \times 10^{-3} s^{-1}$ was chosen with the two other parameters α and β unchanged ($= 10^{-3} s^{-1}$).

6.8 RUN 3

(Ambient wind (u-component) as in Fig.(ii) p. 76. Coriolis parameter zero; cloud physics parameters:- $\alpha = 10^{-3} s^{-1}$; $\beta = 10^{-3} s^{-1}$; $\gamma = 2 \times 10^{-3} s^{-1}$)

Some difficulties were encountered at the first attempts to include environmental shear, mainly in the early growth stages of the convection. The initial 'forced' cumulus tended to grow only very slowly or to dissipate under the influence of the shear and smoothing terms in the equations. A further problem was the propagation and advection of the cloud across the model grid resulting (after some time) in convection near the lateral boundaries and consequent computational problems. The shear configuration was chosen with a view to minimising this problem - with some success - although this run still had to be terminated before the cloud had completed its life-cycle.

In view of these difficulties in the early stages, the first 20 mins are treated as an initialisation period and the main interest focussed on later times. It would be possible to compress this first twenty minute period into a shorter time making a total simulation times given in the diagrams an overestimate.

As previously mentioned, the developing convection was found to be very sensitive to the magnitudes of the smoothing terms; the following smoothing coefficients were used in the run reproduced here: $\underset{\sim}{K_v} = 500\ m^2.sec^{-1}$ for $T > 5$ mins;

$K_\theta = K_q = 0$ for $T \leq 25$ mins and for $T > 25$ mins, a maximum effective K_θ and $K_q \sim 500\ m^2.s^{-1}$ where K_θ and K_q are defined from a $K'|\nabla^2|\nabla^2$ formulation (see Chapter II, §2.2).

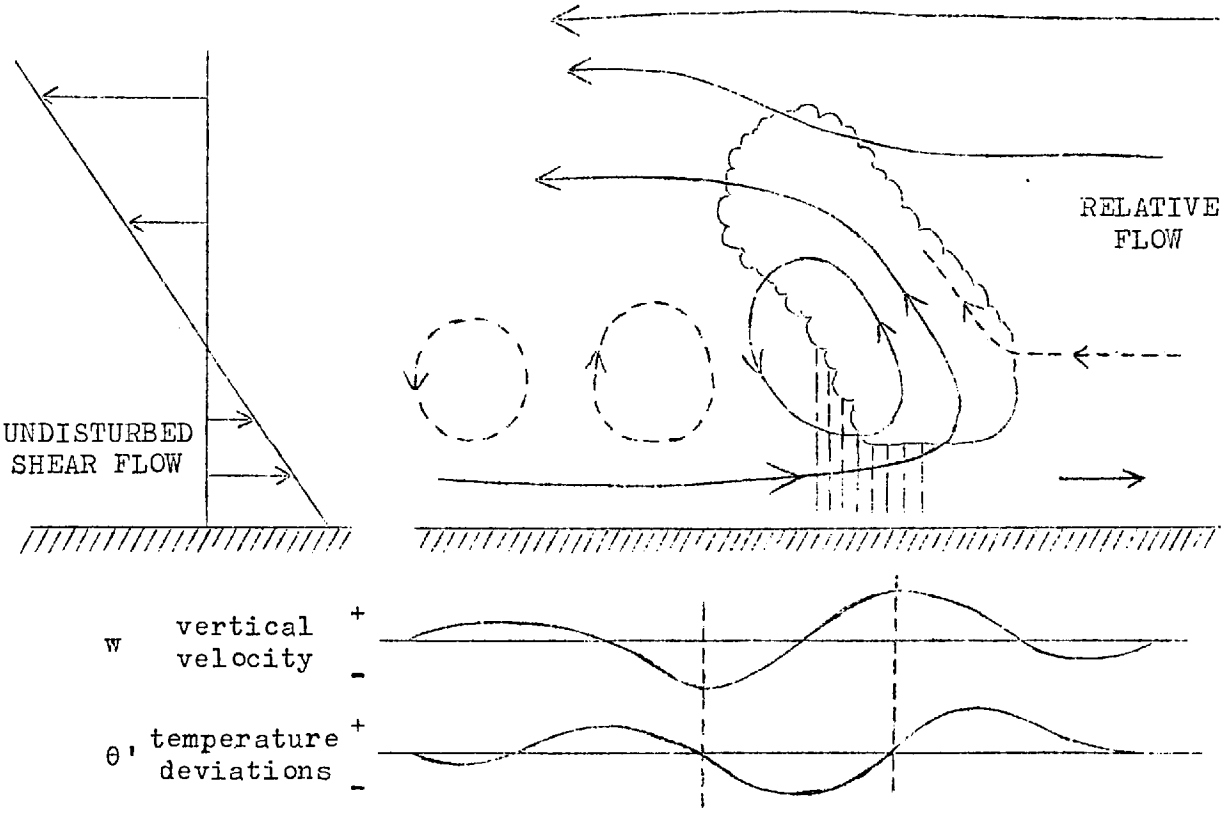
Figures 6.35 - 6.50 show the results of RUN 3, all figures (except Figs.6.46 and 6.47) representing fields through the line of symmetry along the mean wind.

The initialisation stages $T < 20$ mins produce a small cloud with updraft speeds $\approx 3\ m.s^{-1}$ and cloudwater values $\approx 2\ gm.kg^{-1}$. This cloud extends to ~500 mb by 30 mins and a circulation is apparent with the ascending air forming cloud and warming through latent heat release while on the upstream side of the cloud, 'cold' air descends. This descending branch is part of the wave-like form of the convective system and is strengthened by continuous evaporation of cloudwater on the upstream side with temperature deviations $\approx \pm 2°C$ in the two branches.

This convective system exhibits only slow changes between 27½ mins and 40 mins while moving in the direction of the low level wind with a well-defined speed of $\approx 3\ m.s^{-1}$ corresponding to the 850 mb level environment wind speed. The fields at 37½ mins best describe this system given in schematic form below. The wave-structure is clearly shown with the phase-change between w and θ' well-defined.

The slow changes that occur during the period 27½ - 40 mins are mostly concerned with the formation of rainwater. Rain first forms at ≈ 32 mins and reaches the ground at ≈ 37½ mins extending the chilled region down to the surface. By 42½ mins the downdraft branch of the circulation is strengthening and 'cold' air spreads out at low levels. This divergence of air has important implications which are discussed later. The cloud

leans down the shear with a growing anvil in the later stages. Beyond 45 mins the proximity of the lateral boundary spoiled the computation.



The structure of the convection at right angles to the mean flow has a symmetric form very similar to the unsheared cases discussed previously and an example is shown in Fig.6.46.

The nature of this convective system and its observed quasi-steady behaviour poses several questions, particularly about the large cloudwater evaporation. If the model retains too much water in cloudwater form the evaporative cooling is exaggerated, for cloudwater is assumed to evaporate during a timestep (3 secs) until either saturation is reached or the cloudwater has disappeared. The simulated values of 1 - 3 $gm.kg^{-1}$ do not however appear unduly higher than those observed (Mason, 1962, C.U.P.). Excessive explicit mixing of cloud and clear air might also exaggerate the evaporation of cloudwater but this is not the case here where smoothing terms have been

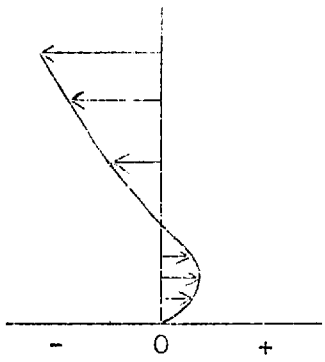
kept as small as possible.

The convective system modelled in RUN 3 represents one 'mode' in which a cumulonimbus can exist for a given initial environment. The development of a downdraft of cold air spreading out at low levels produces regions of low-level convergence, these regions do not however reinforce existing updrafts. The system therefore does not have a 'long' lifetime although attaining a quasi-steady state before the rainwater evaporation becomes dominant. For a long-lasting convective system with the 'steady-overturning' character observed in severe storm studies there exists a requirement for the downdraft to stimulate and strengthen convective development on the downshear side (Ludlam, 1963).

The character of the convective motion in RUN 3 suggests that modifications could be made to the initial conditions or to the cloud physics possibly such as to encourage downshear movement of the low-level downdraft air thus promoting continual growth on the downshear side. One can speculate as to what modification would achieve this aim. An intensification of the downdraft would be one possibility, either by increasing the intensity of the convection through alterations in the initial sounding or by increasing the rainwater evaporation through the parameterisation. The intensifying of the convection would however modify the whole system including its steering-level and it is not clear whether the desired effect would be achieved.

Reference to observation suggests possibly a better modification. The large-scale environment conducive to the development of intense storms is characterised by a pronounced veer in the winds with height; this turning of the wind with

height can give a wind component profile such that the lower-level wind has a maximum and reduces to a small value near the surface as sketched below. The effect of this configuration in the model would be to allow the downdraft to spread out downshear more readily.



u-component (say)

This form of wind profile has been used by Takeda (1971) in a 2-D model. He shows that this allows the modelling of a "long-lasting" cloud with organisation of the updraft and downdraft with the importance of this wind profile on precipitation particle trajectories emphasised.

6.9 RUN 4

Results from RUN 4 in which a low-level turning of the wind with height was modelled but without a low-level maximum as sketched above showed only small differences from RUN 3, mainly in the direction of movement of the cloud, and results are not shown. The results do however represent a convective system with no symmetry in space whatsoever - the first known model results of their kind.

In retrospect the low-level turning of the wind should be included so as to test the ideas discussed in the previous section and this will be done as soon as possible.

RUN 1

Figs. 6.3 - 6.20

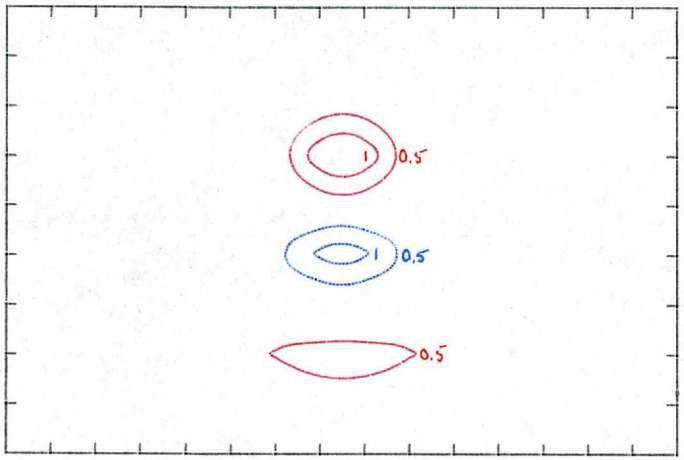
No ambient wind

Coriolis parameter zero

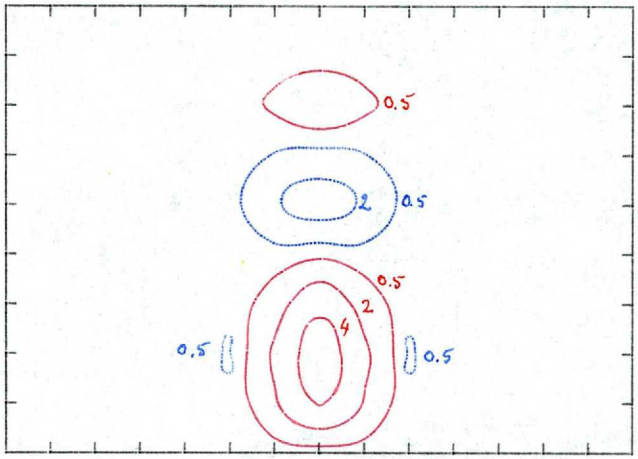
$\alpha = 10^{-3}\ s^{-1}$

$\beta = 10^{-3}\ s^{-1}$

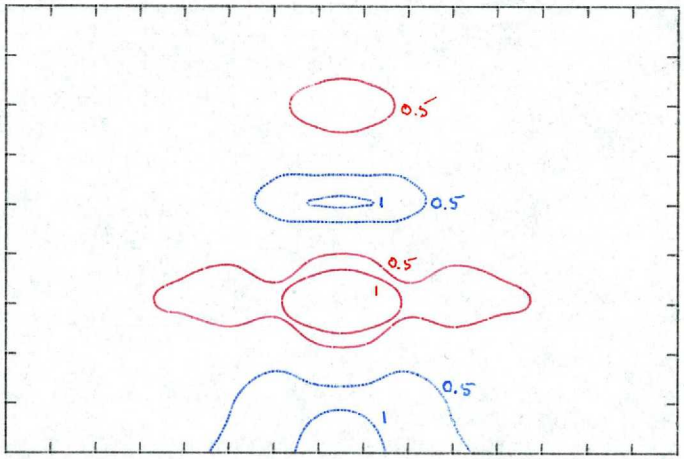
$\gamma = 10^{-3}\ s^{-1}$



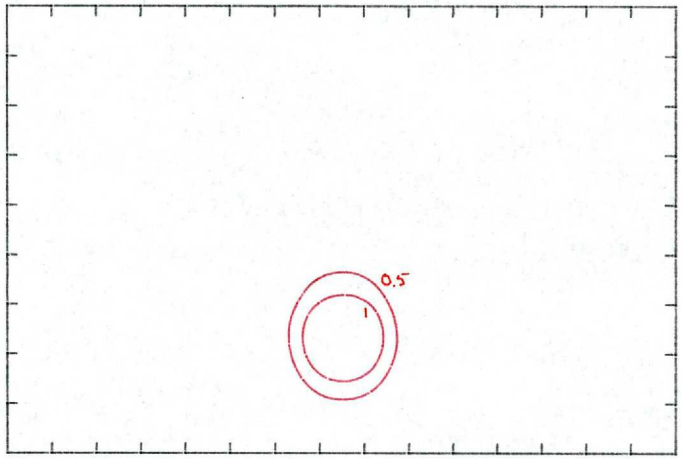
POTENTIAL TEMPERATURE DEVIATION IN DEG. C. AT 10.0 MINS.



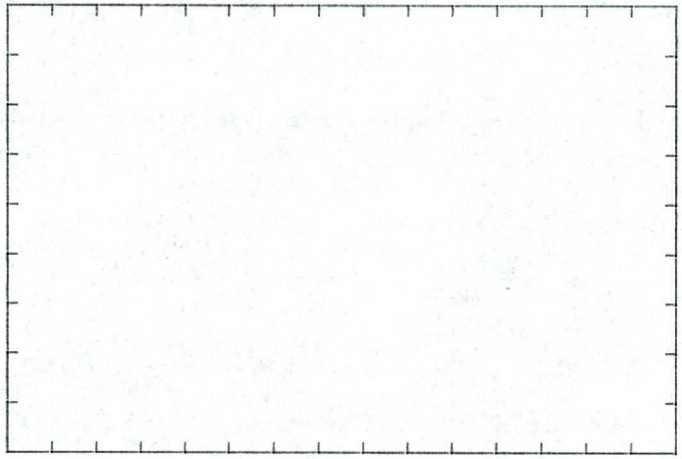
VERTICAL VELOCITY IN METRES/SEC. AT 10.0 MINS



DEVIATION HEIGHT FIELD IN METRES AT 10.0 MINS.

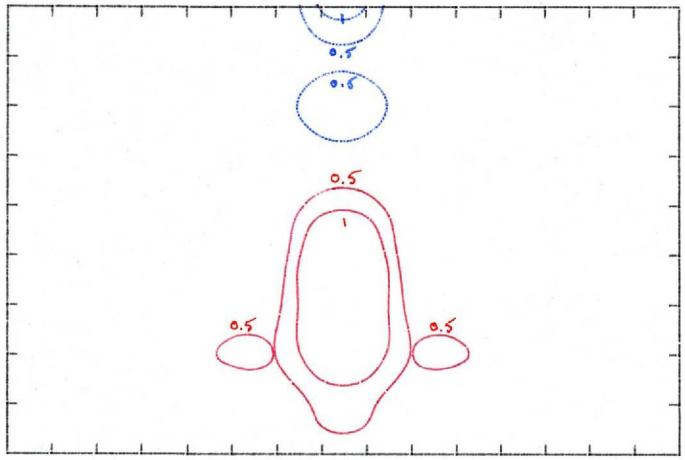


CLOUDWATER CONTENT IN GM/KG. AT 10.0 MINS

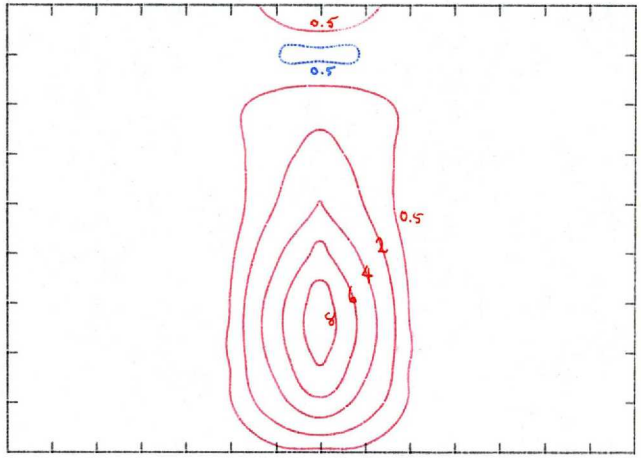


RAINWATER CONTENT IN GM/KG. AT 10.0 MINS

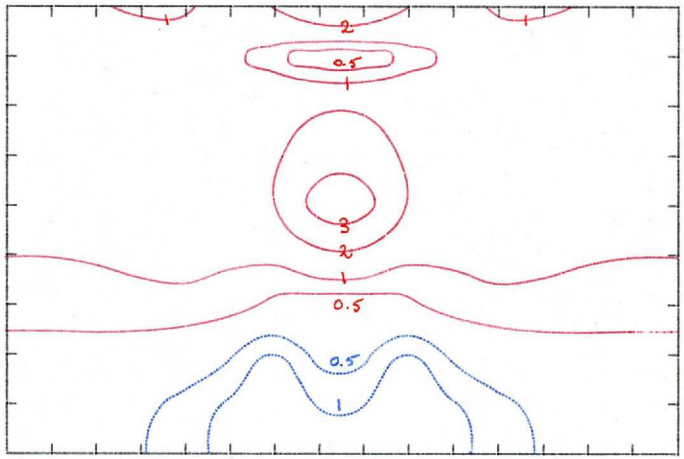
Fig. 6.3



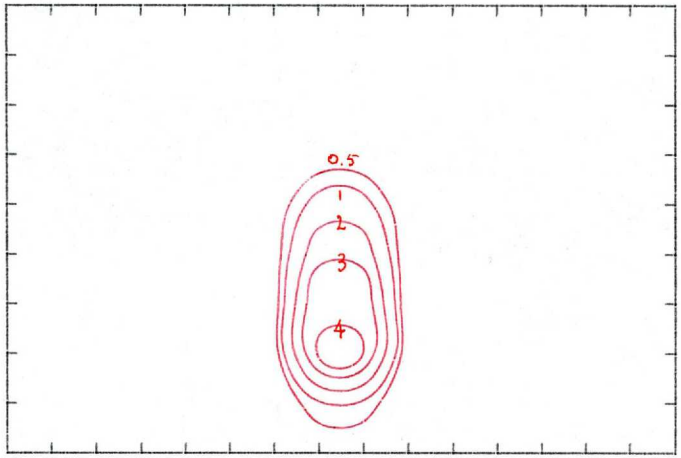
POTENTIAL TEMPERATURE DEVIATION IN DEG. C. AT 15.0 MINS.



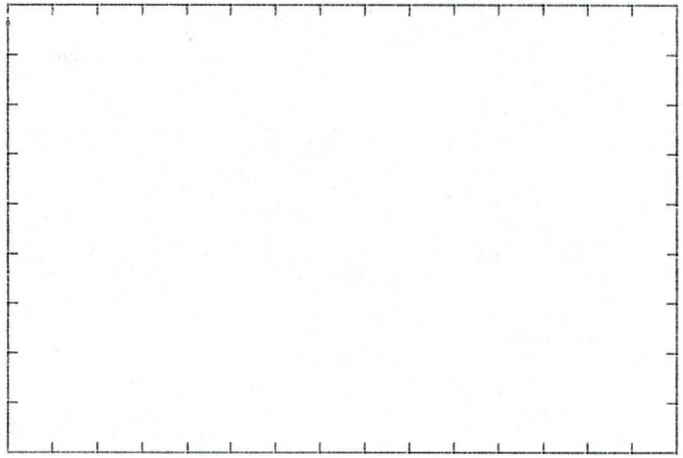
VERTICAL VELOCITY IN METRES/SEC. AT 15.0 MINS



DEVIATION HEIGHT FIELD IN METRES AT 15.0 MINS.



CLOUDWATER CONTENT IN GM/KG. AT 15.0 MINS



RAINWATER CONTENT IN GM/KG. AT 15.0 MINS

Fig. 6.4

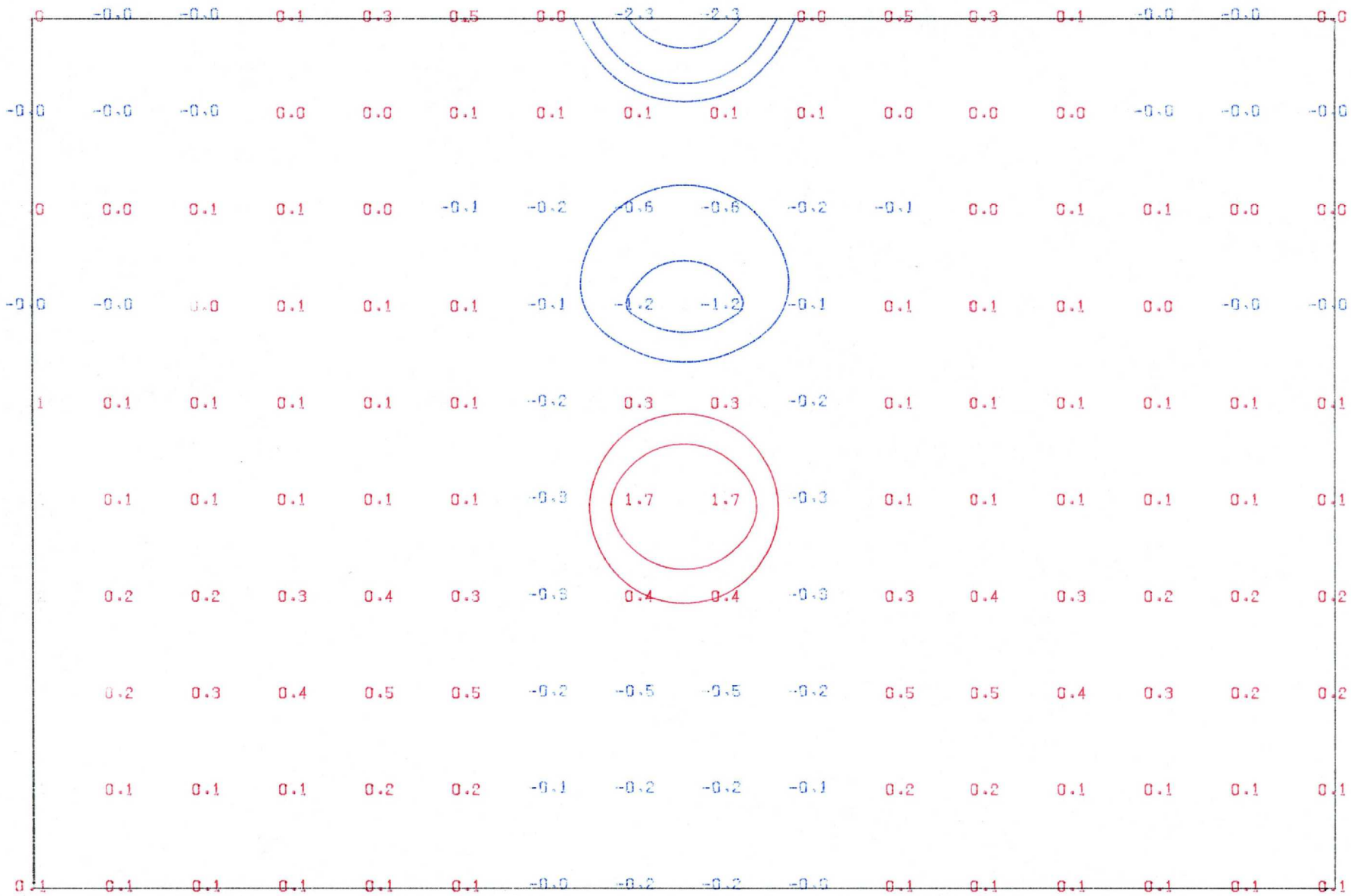


Fig. 6.5 POTENTIAL TEMPERATURE DEVIATION IN DEG. C. AT 17.5MINS.

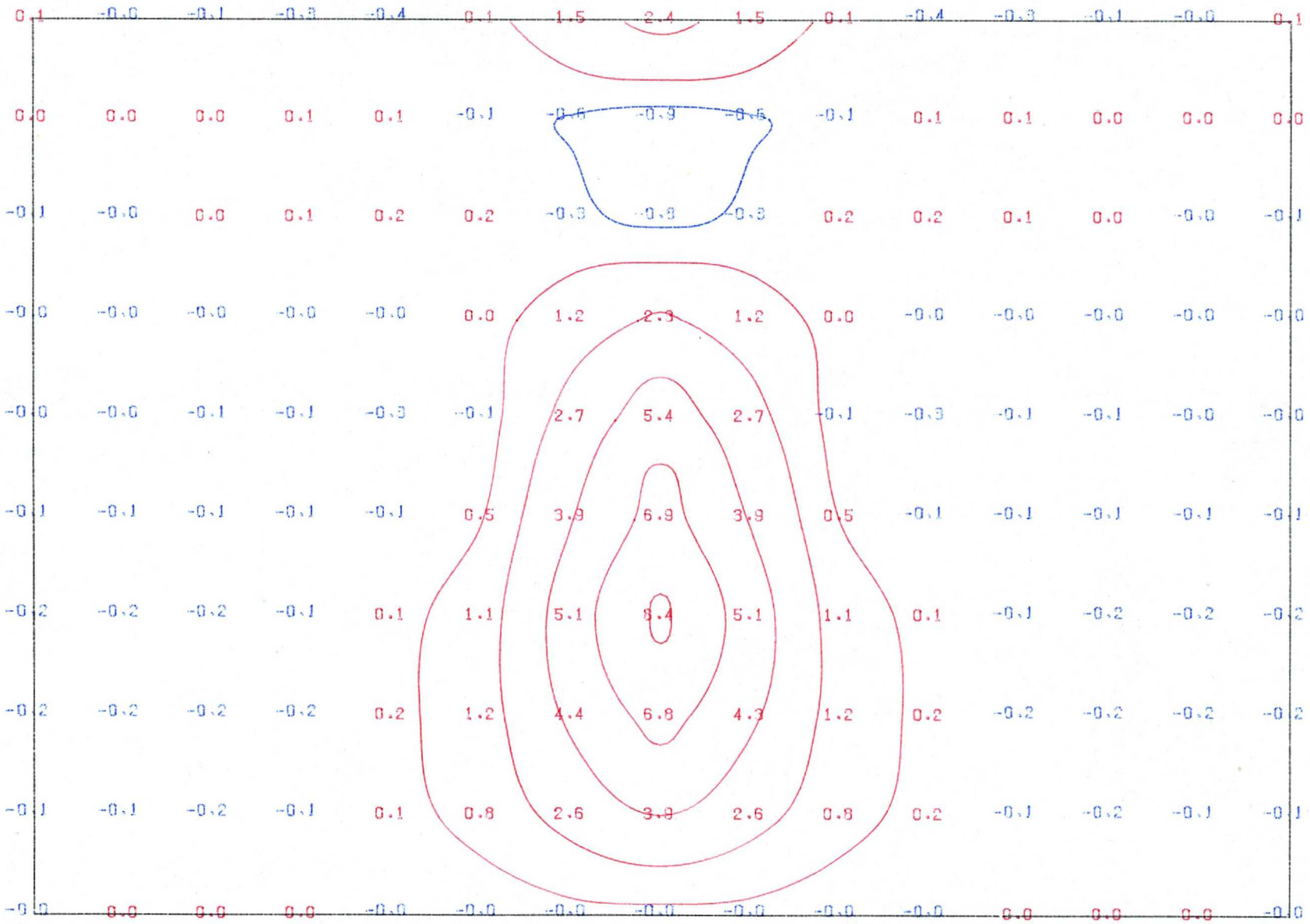


Fig. 6.6 VERTICAL VELOCITY IN METRES/SEC. AT 17.5MINS

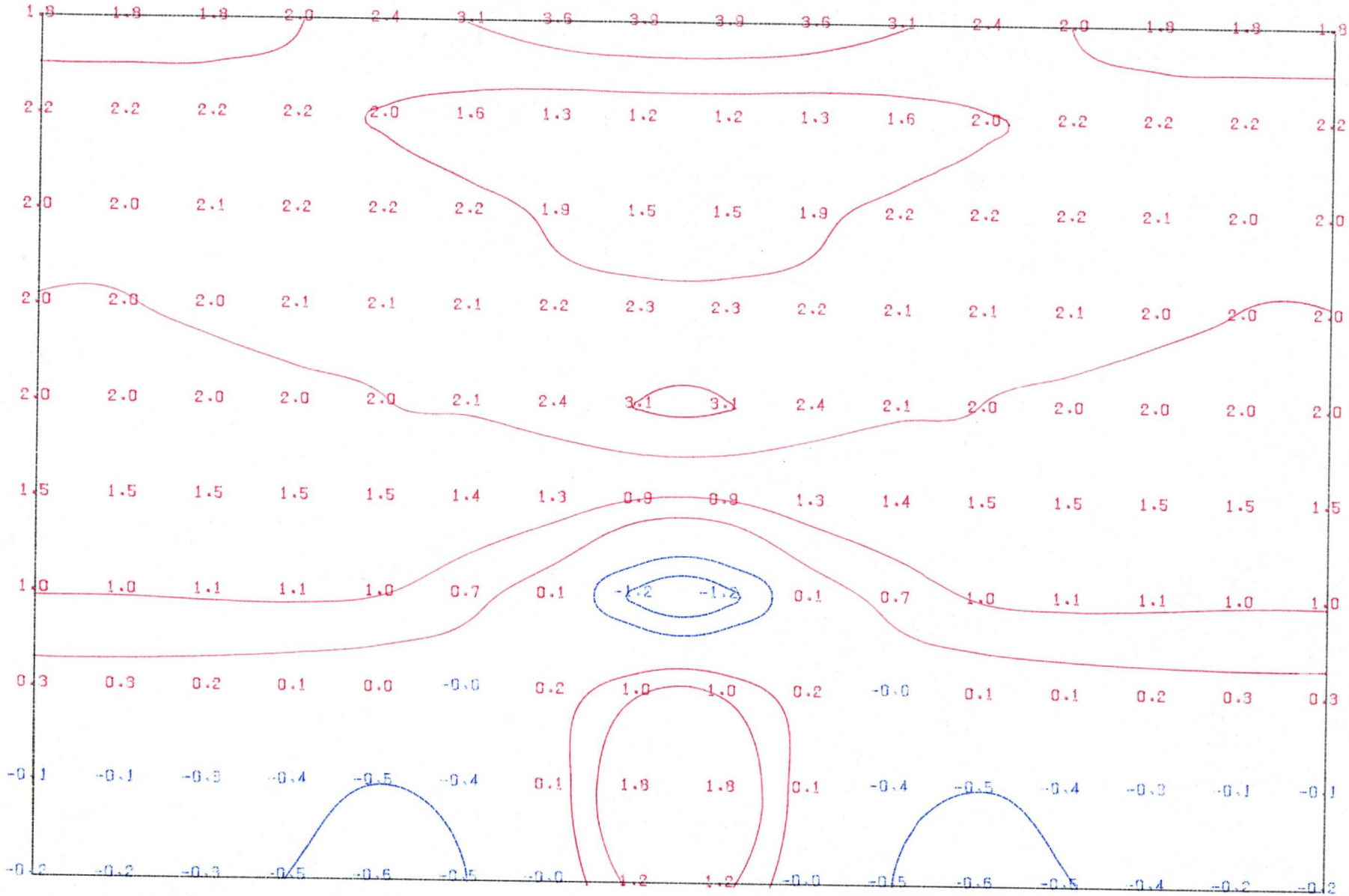


Fig. 6.7 DEVIATION HEIGHT FIELD IN METRES AT 17.5MINS.

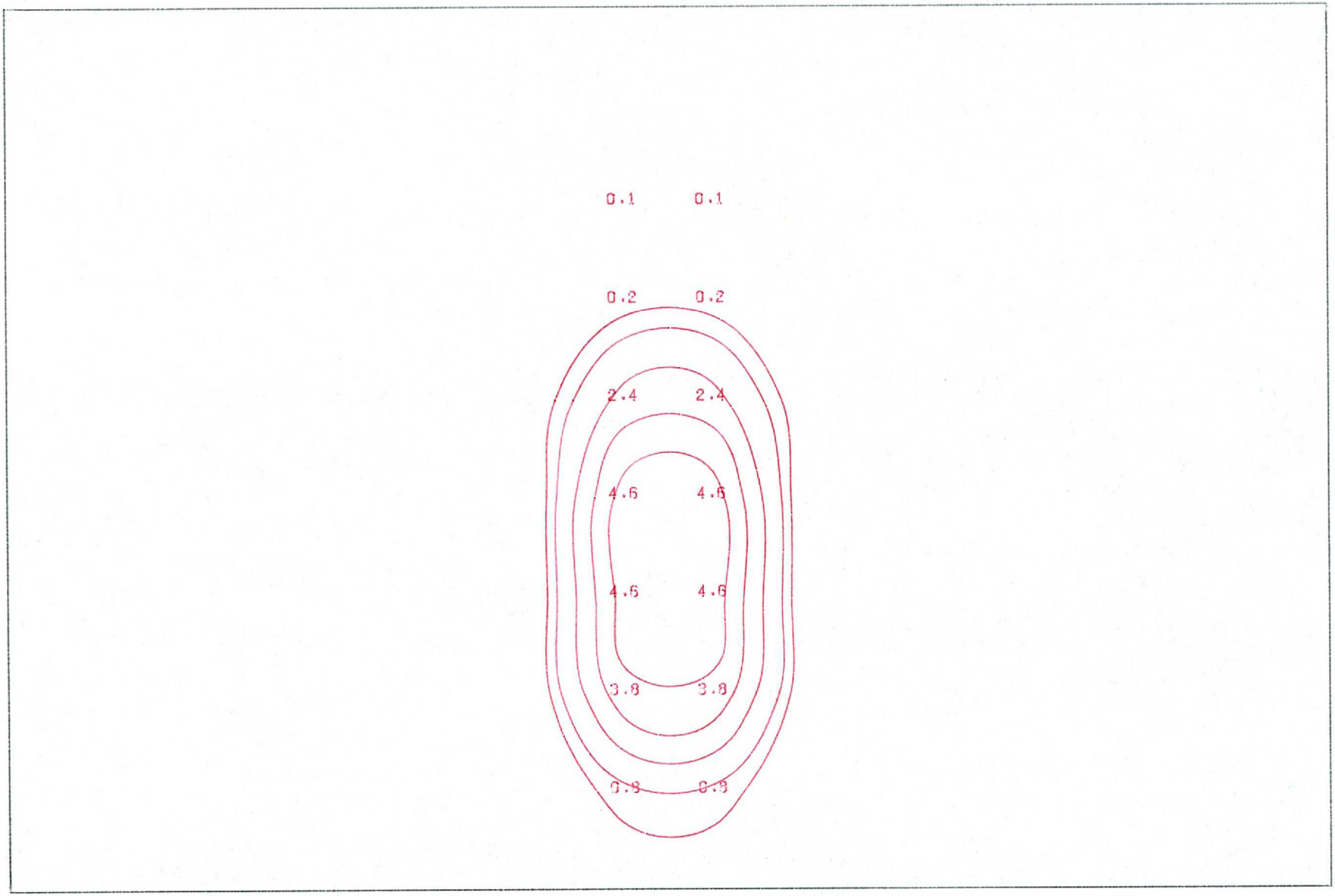


Fig. 6.8 CLOUDWATER CONTENT IN GM/KG- AT 17.5MINS

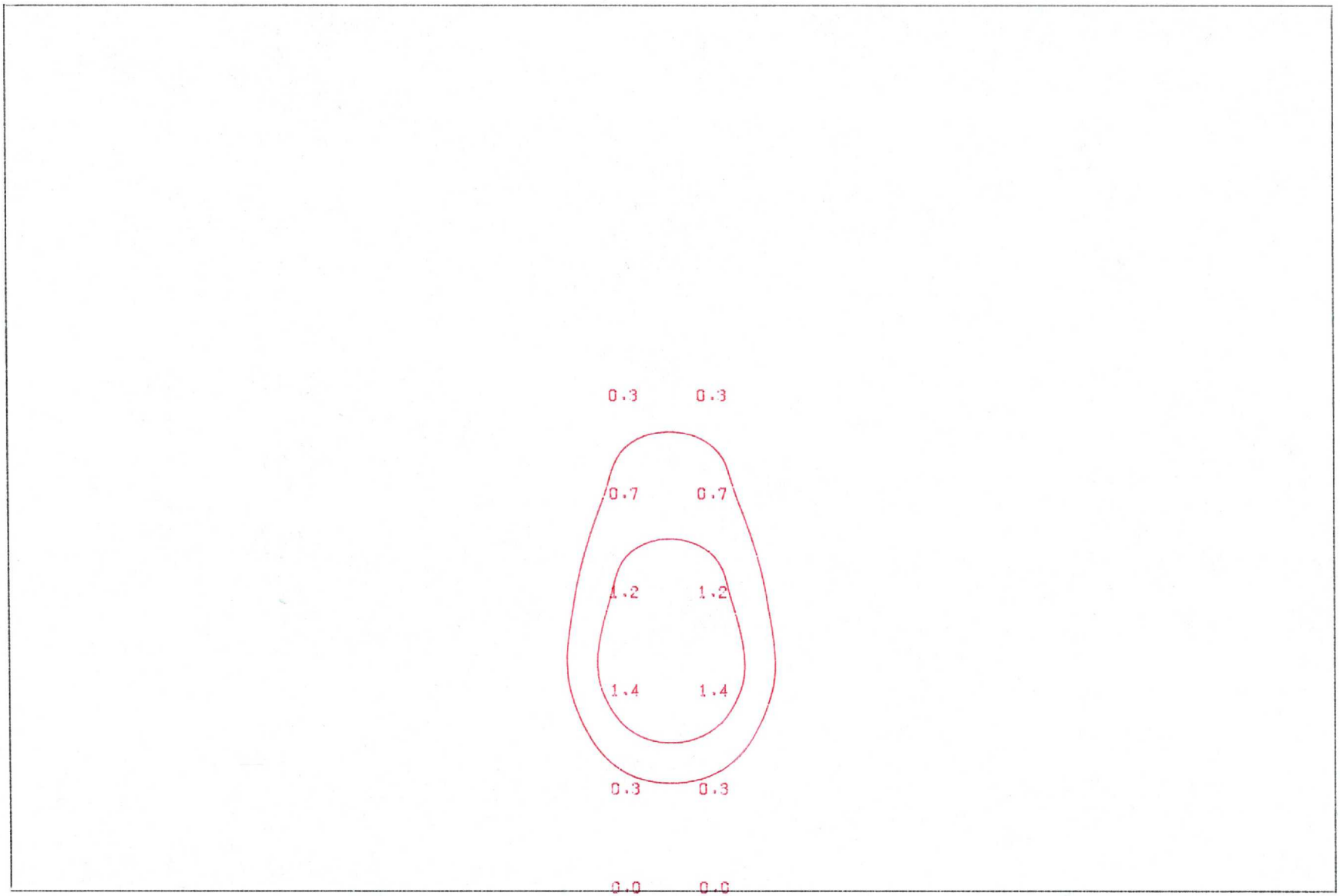


Fig. 6.9 RAINWATER CONTENT IN GM/KG. AT 17.5MINS

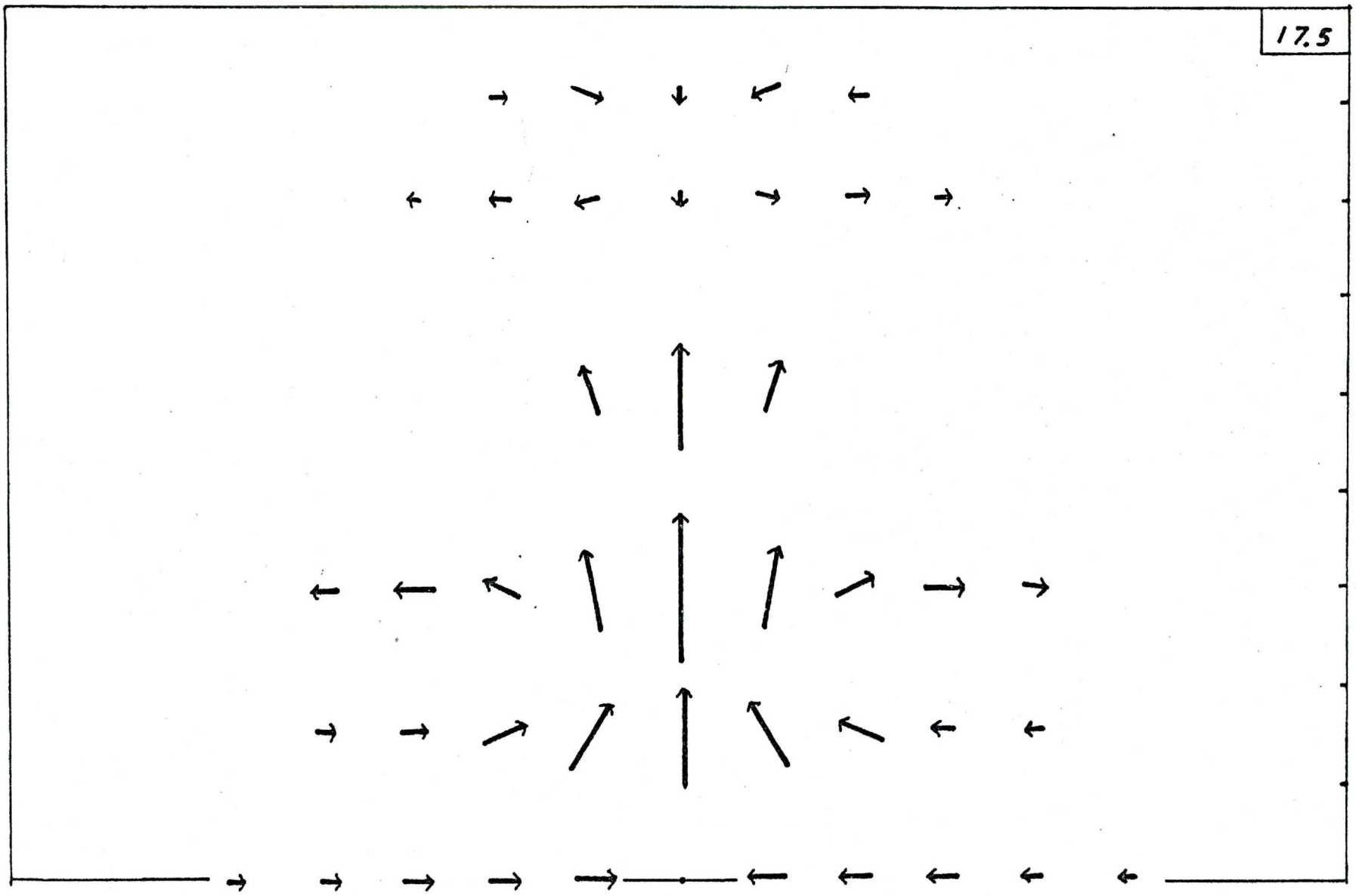
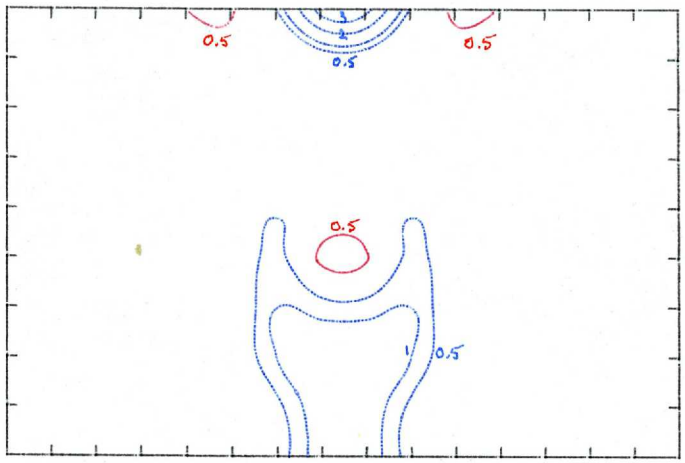
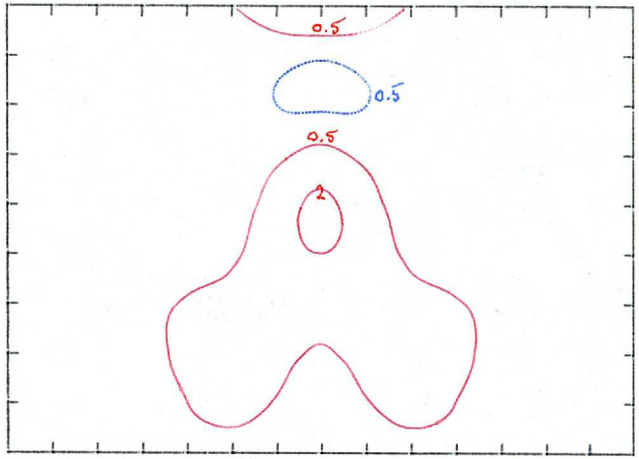


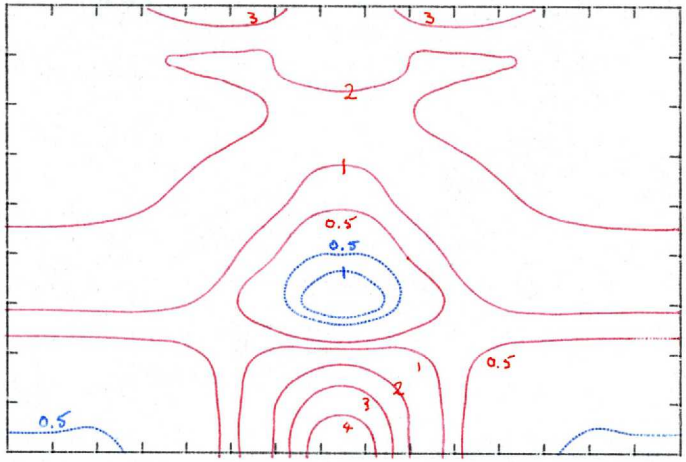
FIG 6.10 *RUN I.* [X, Y=Y$_s$, P] *WIND VECTORS* 10 m/s



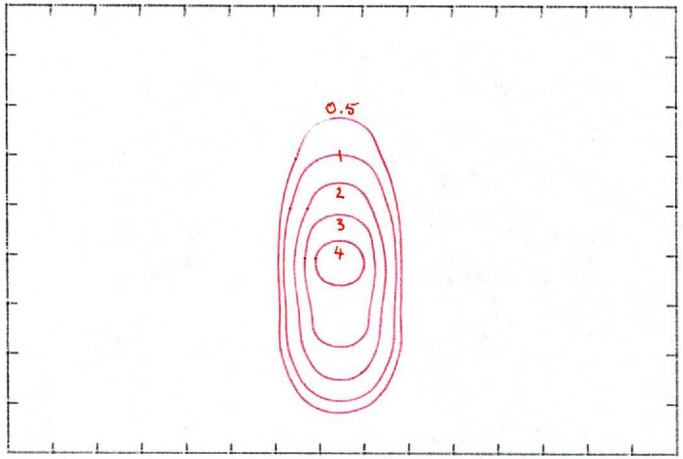
POTENTIAL TEMPERATURE DEVIATION IN DEG. C. AT 20.0 MINS.



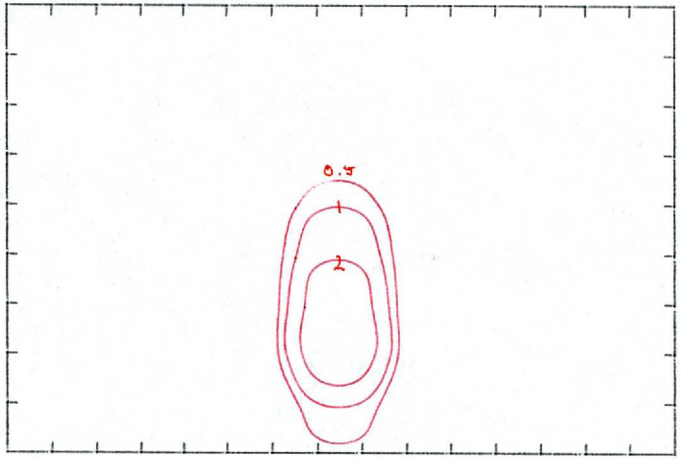
VERTICAL VELOCITY IN METRES/SEC. AT 20.0 MINS



DEVIATION HEIGHT FIELD IN METRES AT 20.0 MINS.



CLOUDWATER CONTENT IN GM/KG. AT 20.0 MINS



RAINWATER CONTENT IN GM/KG. AT 20.0 MINS

Fig. 6.11

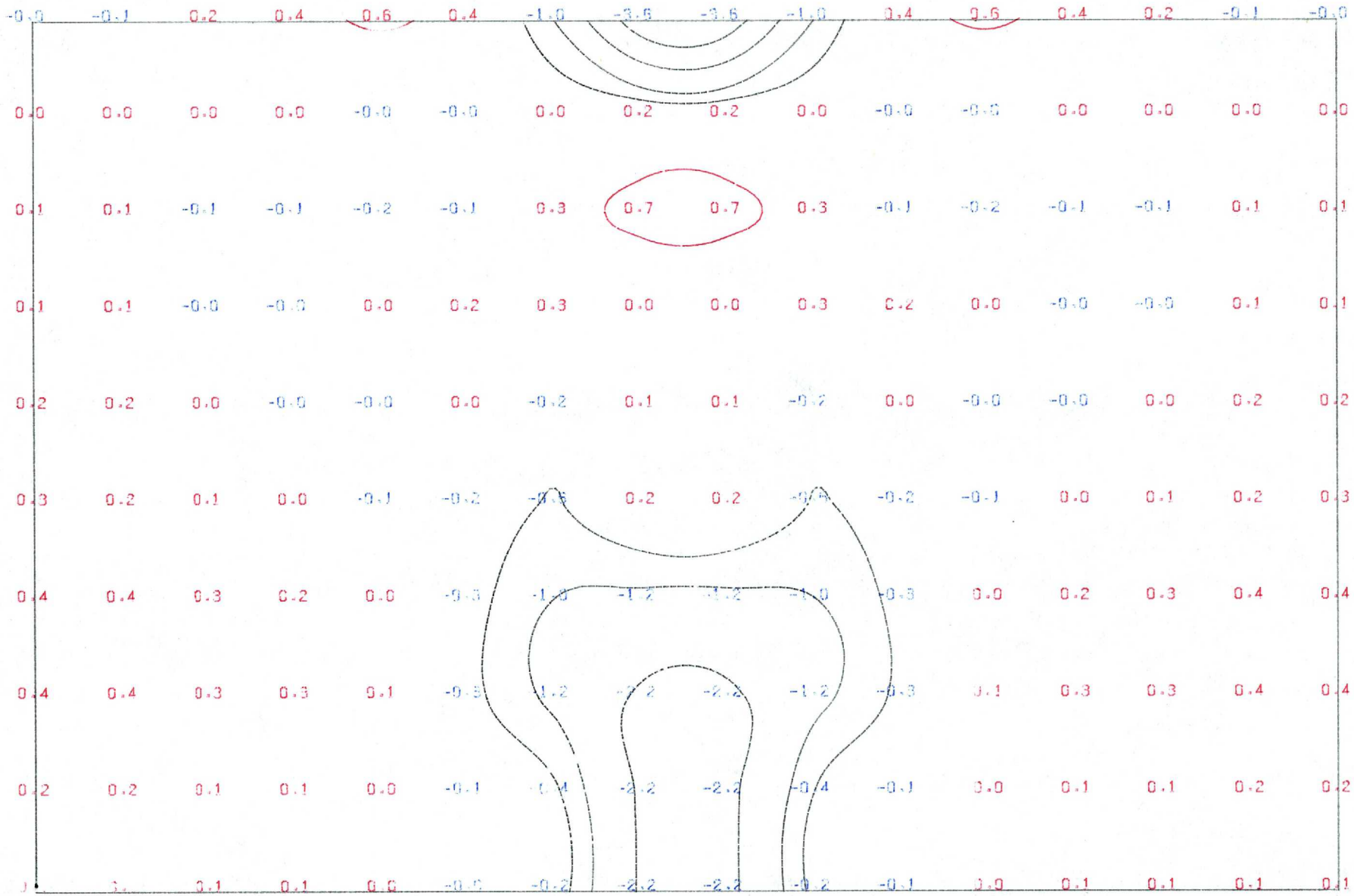


Fig. 6.12 [illegible] TEMPERATURE DEVIATION IN DEG. C. AT 22.5MINS.

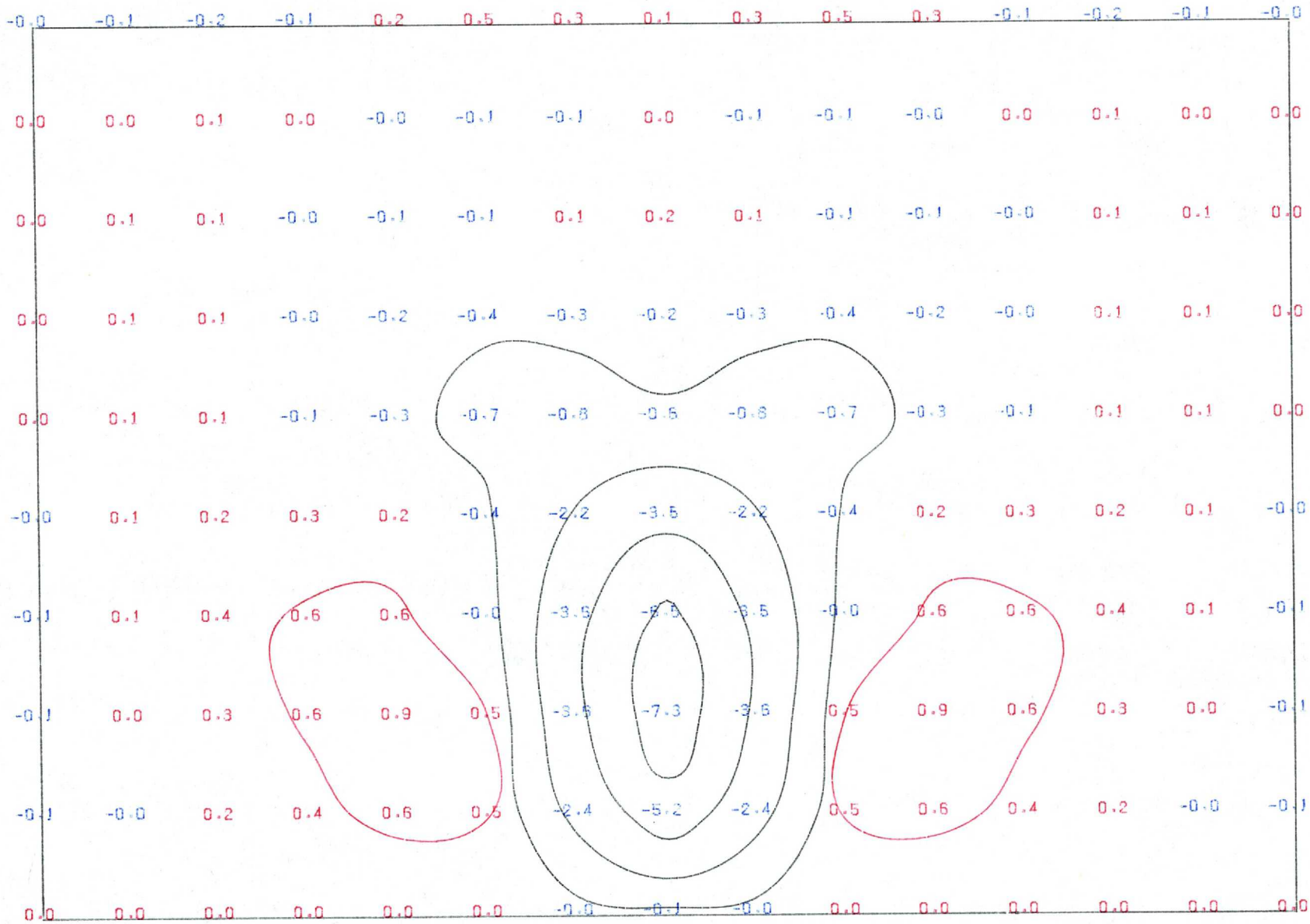


Fig. 6.13 VERTICAL VELOCITY IN METRES/SEC. AT 22.5MINS

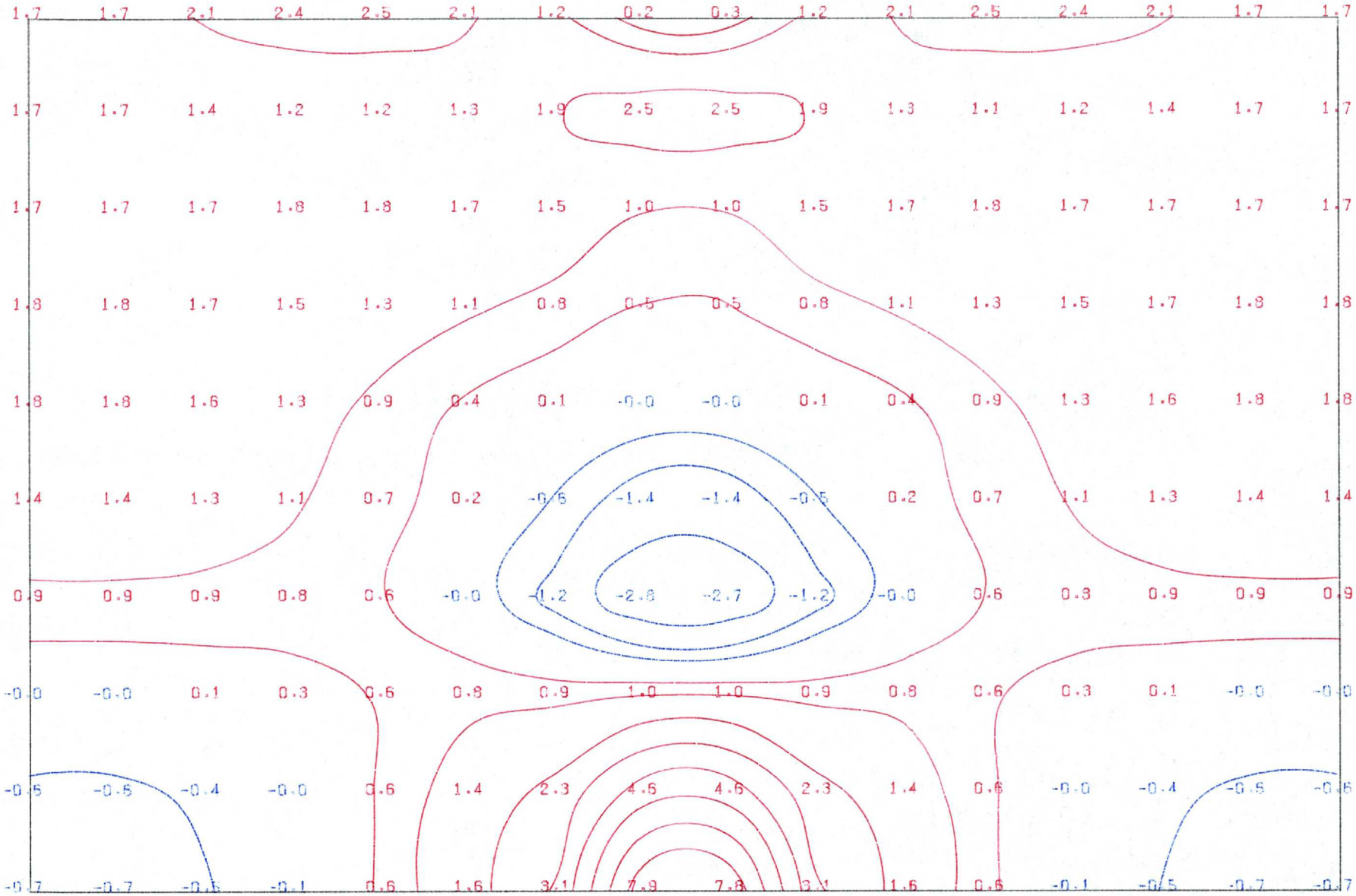


Fig. 6.14 DEVIATION HEIGHT FIELD IN METRES AT 22.5MINS.

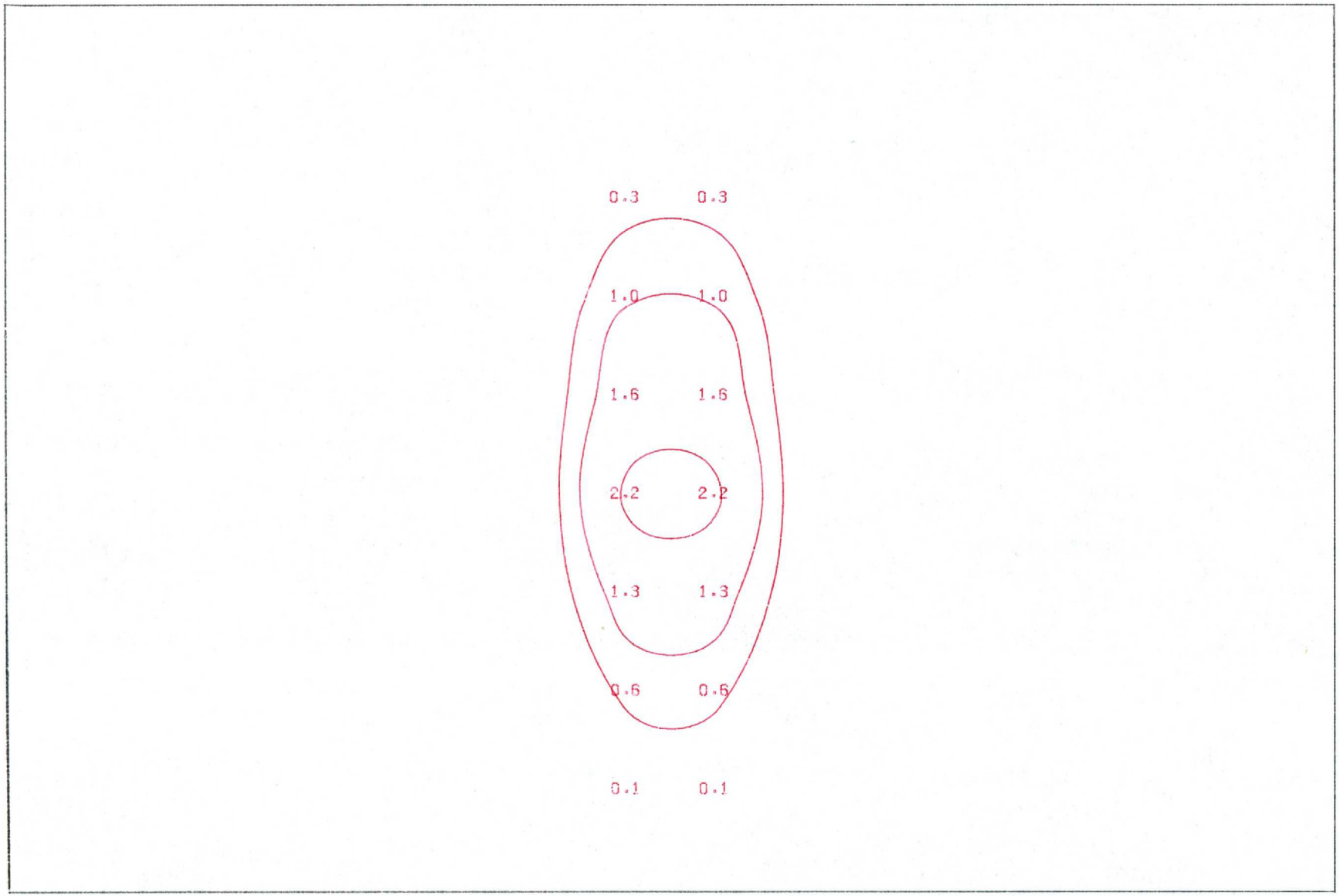


Fig. 6.15 CLOUDWATER CONTENT IN GM/KG. AT 22.5MINS

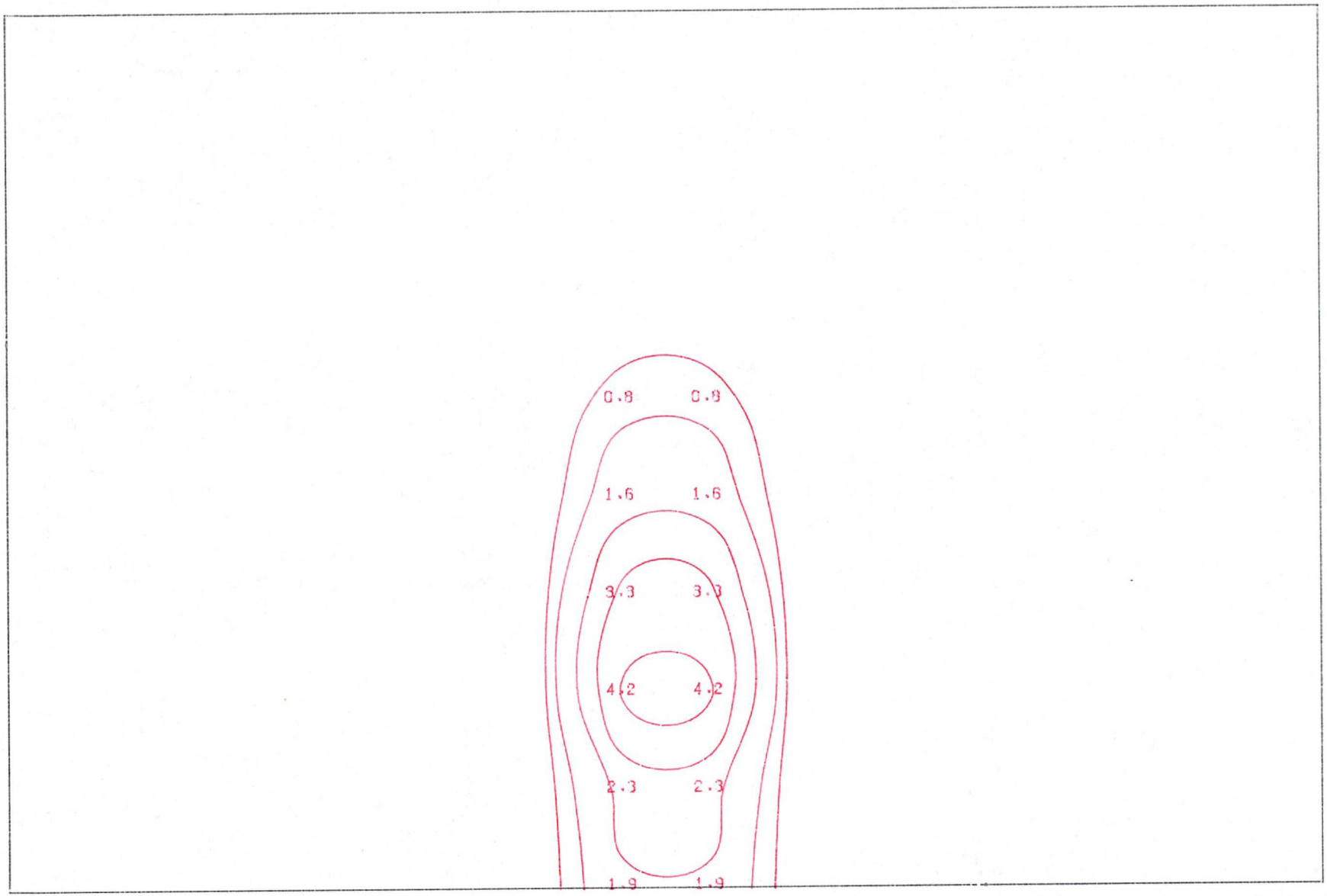


Fig. 6.16 RAINWATER CONTENT IN GM/KG AT 22.5MINS

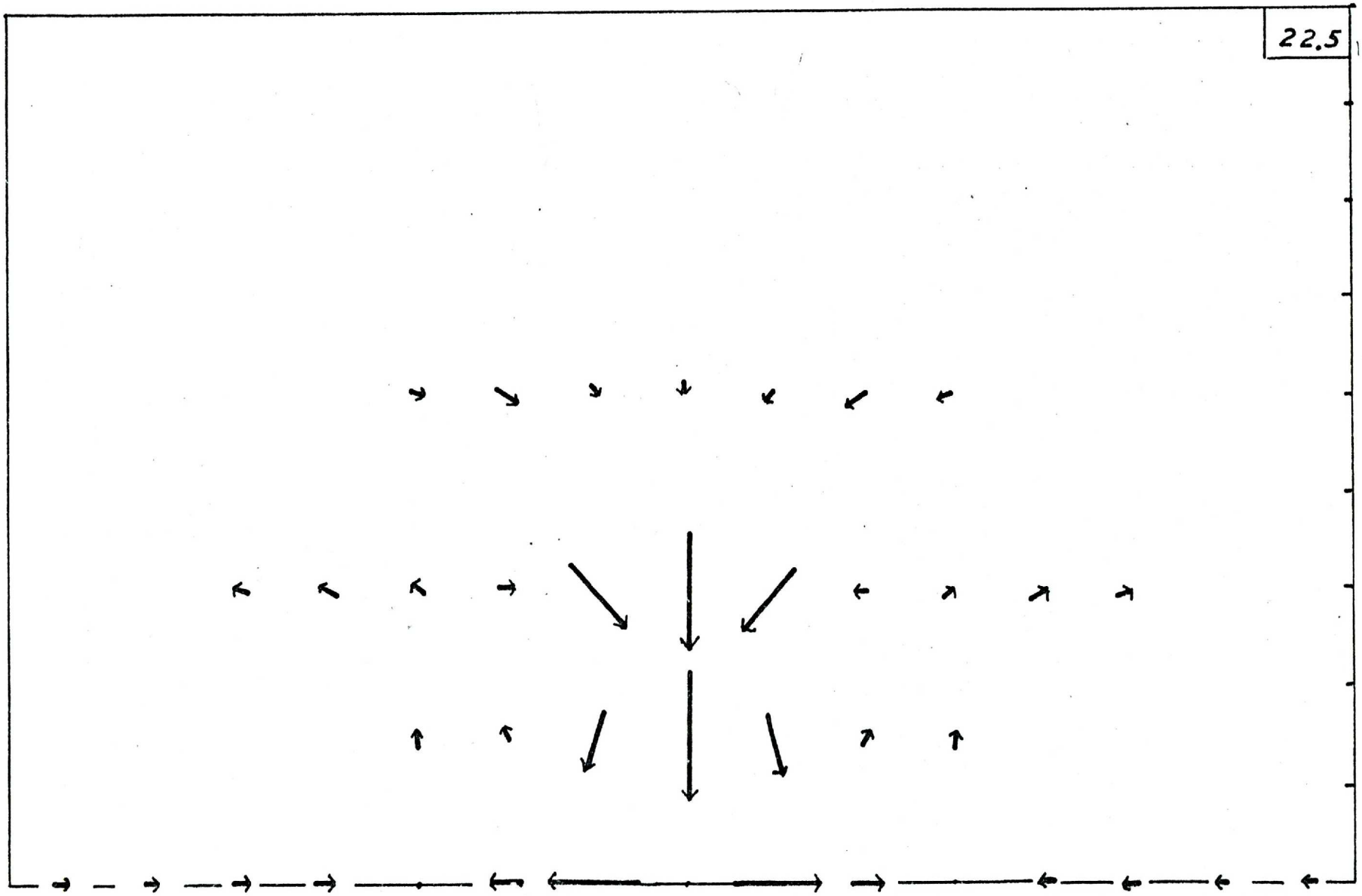
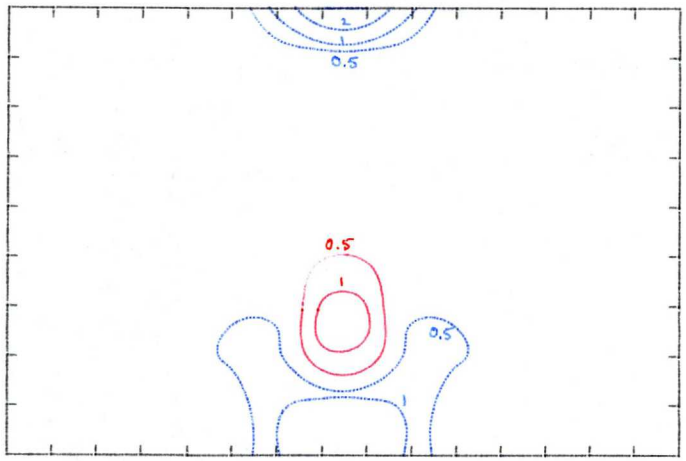
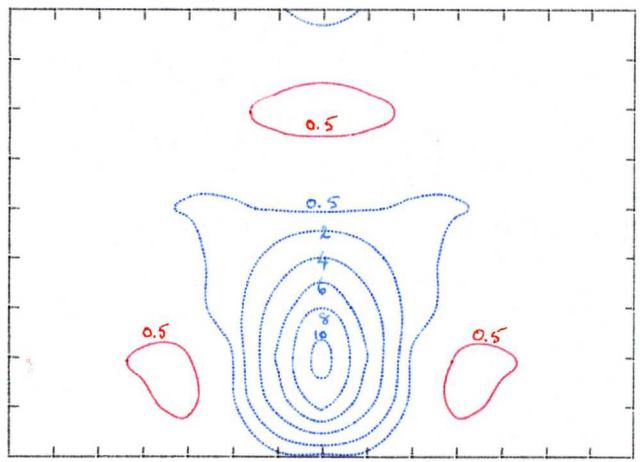


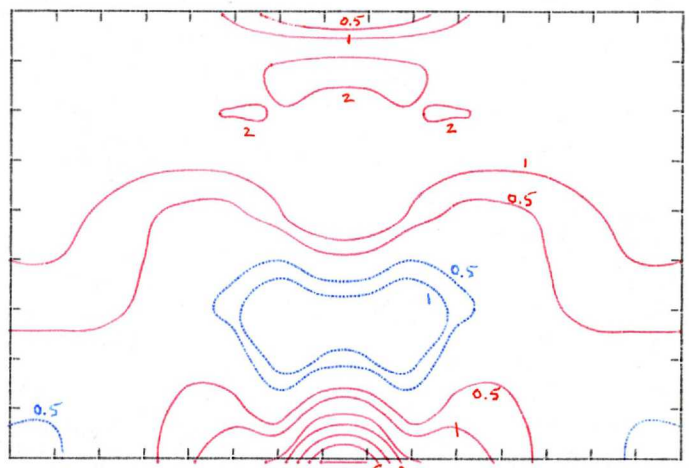
FIG 6.17 *RUN I* [X , Y=Y$_s$, P] *WIND VECTORS* —— 10 m/s



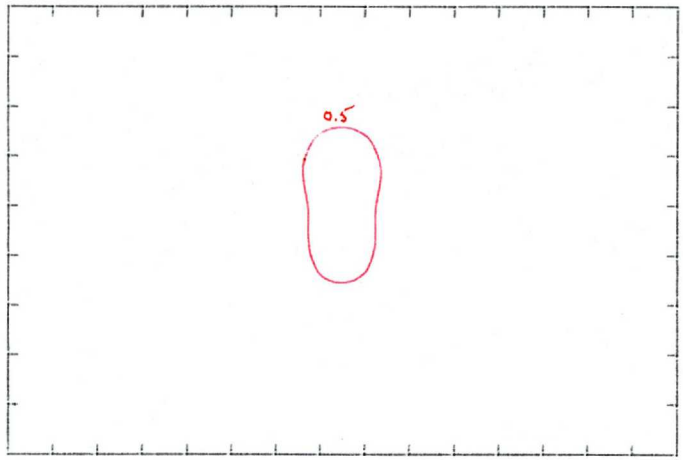
POTENTIAL TEMPERATURE DEVIATION IN DEG. C. AT 25.0 MINS.



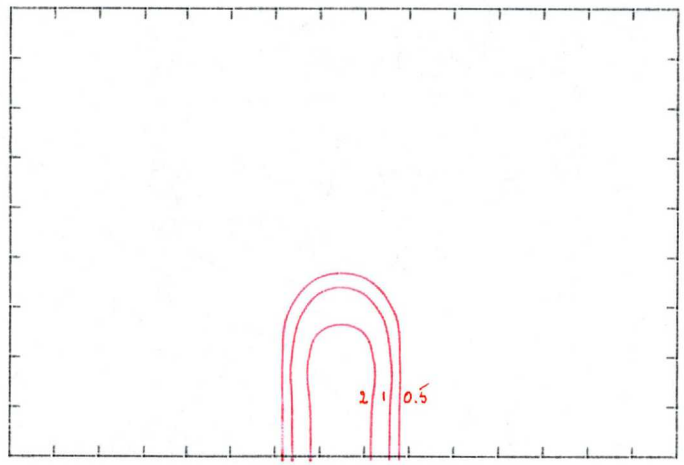
VERTICAL VELOCITY IN METRES/SEC. AT 25.0 MINS



DEVIATION HEIGHT FIELD IN METRES AT 25.0 MINS.

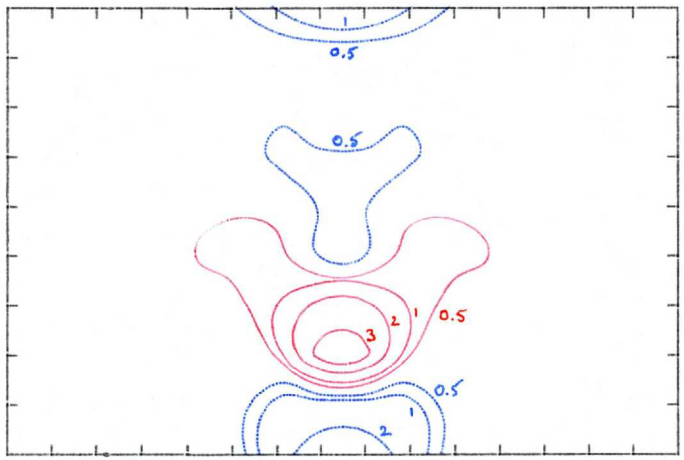


CLOUDWATER CONTENT IN GM/KG. AT 25.0 MINS

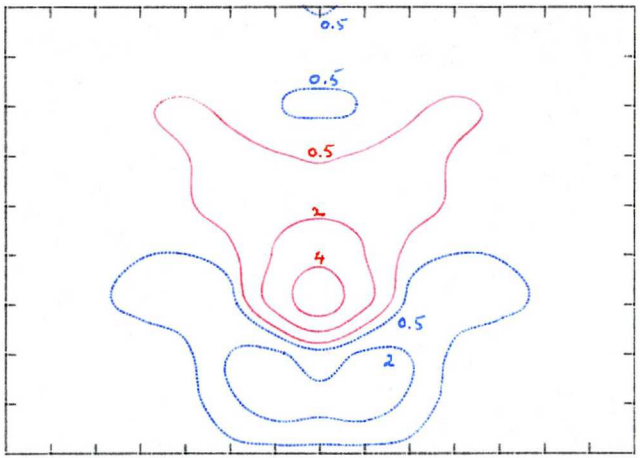


RAINWATER CONTENT IN GM/KG. AT 25.0 MINS

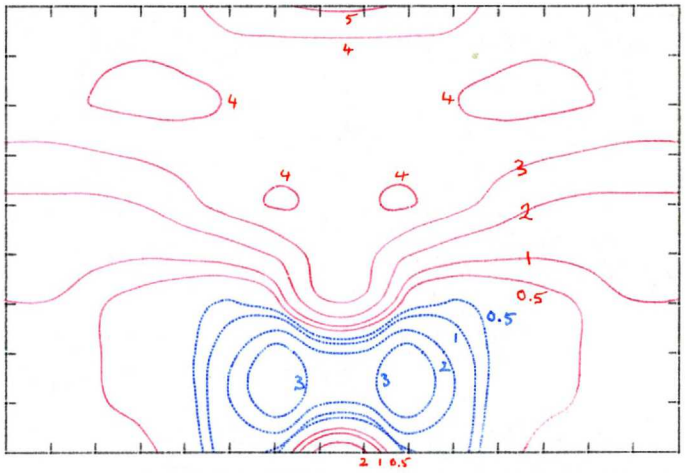
Fig. 6.18



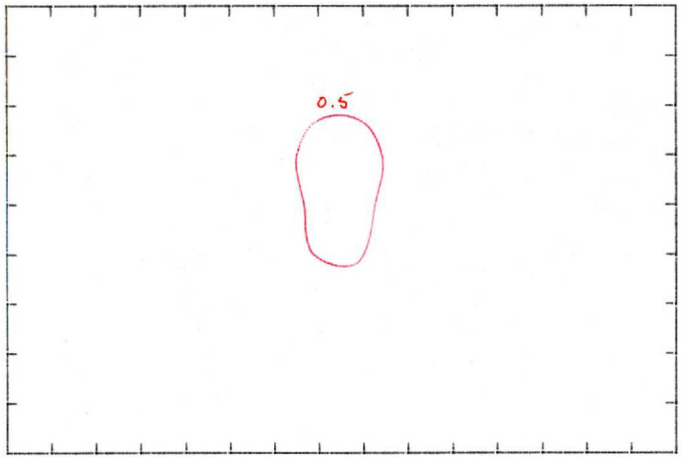
POTENTIAL TEMPERATURE DEVIATION IN DEG. C. AT 30.0 MINS.



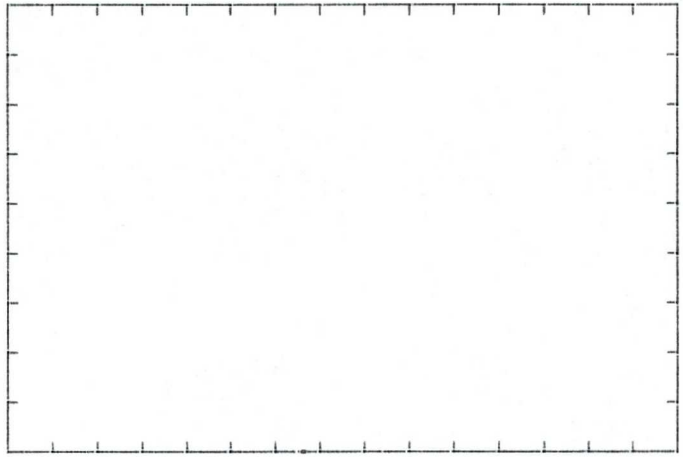
VERTICAL VELOCITY IN METRES/SEC. AT 30.0 MINS



DEVIATION HEIGHT FIELD IN METRES AT 30.0 MINS.



CLOUDWATER CONTENT IN GM/KG. AT 30.0 MINS



RAINWATER CONTENT IN GM/KG. AT 30.0 MINS

Fig. 6.19

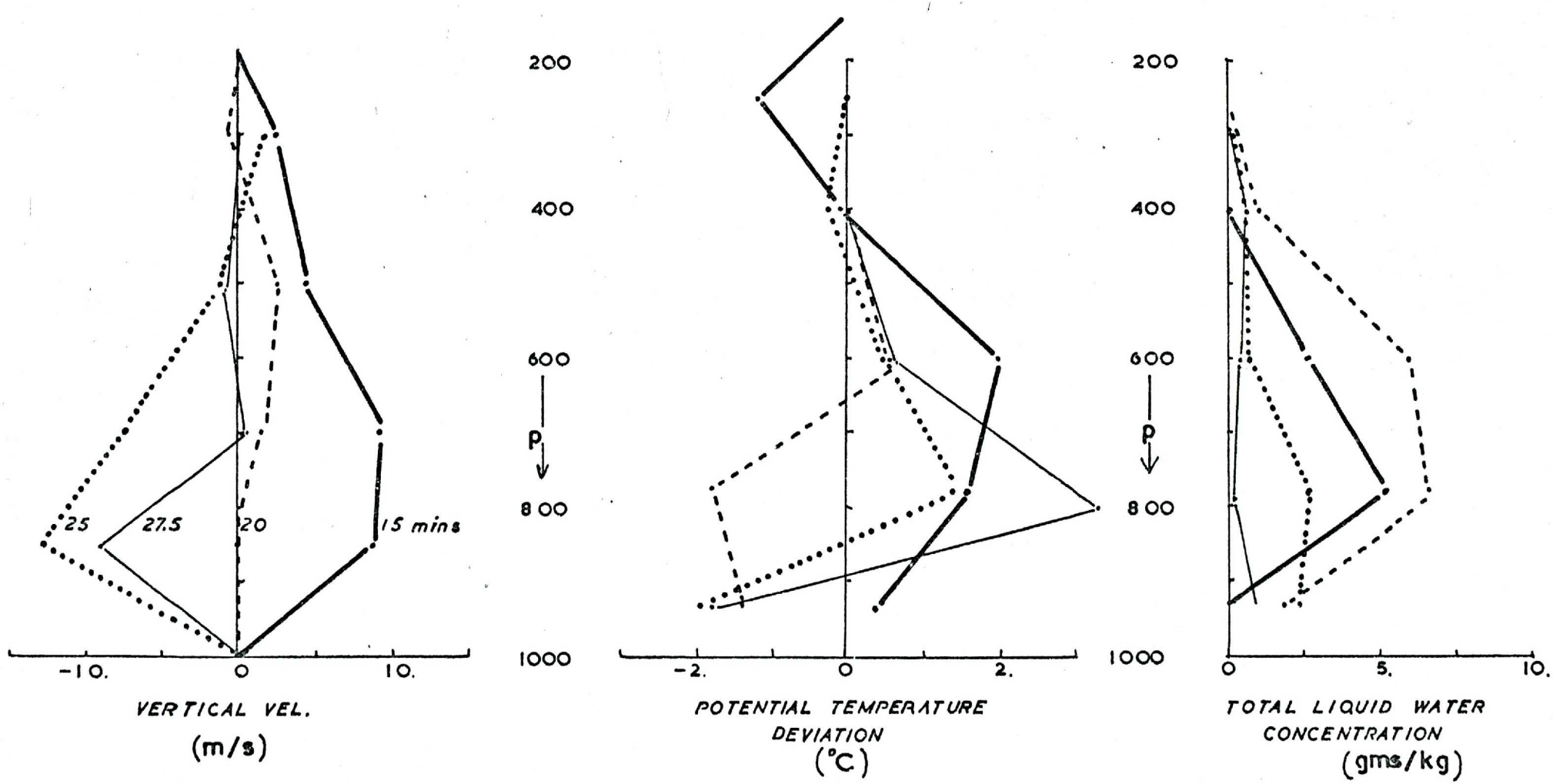


Fig. 6.20 Axis values of variables during lifecycle of cloud

RUN 2

Figs. 6.21 - 6.34

(as RUN 1 but with $\gamma = 10^{-2}\ s^{-1}$)

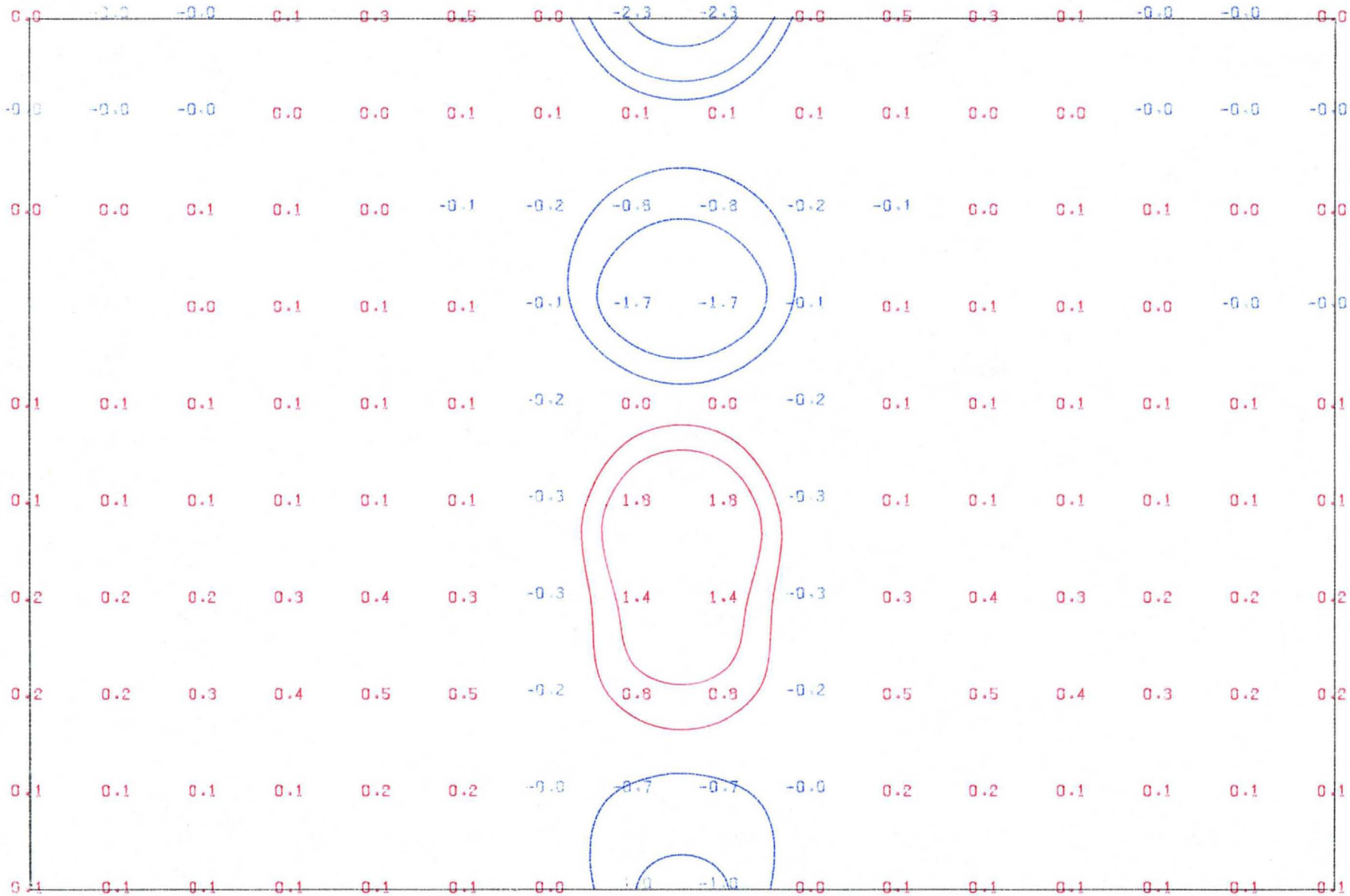


Fig. 6.21 POTENTIAL TEMPERATURE DEVIATION IN DEG. C AT 17.5MINS.

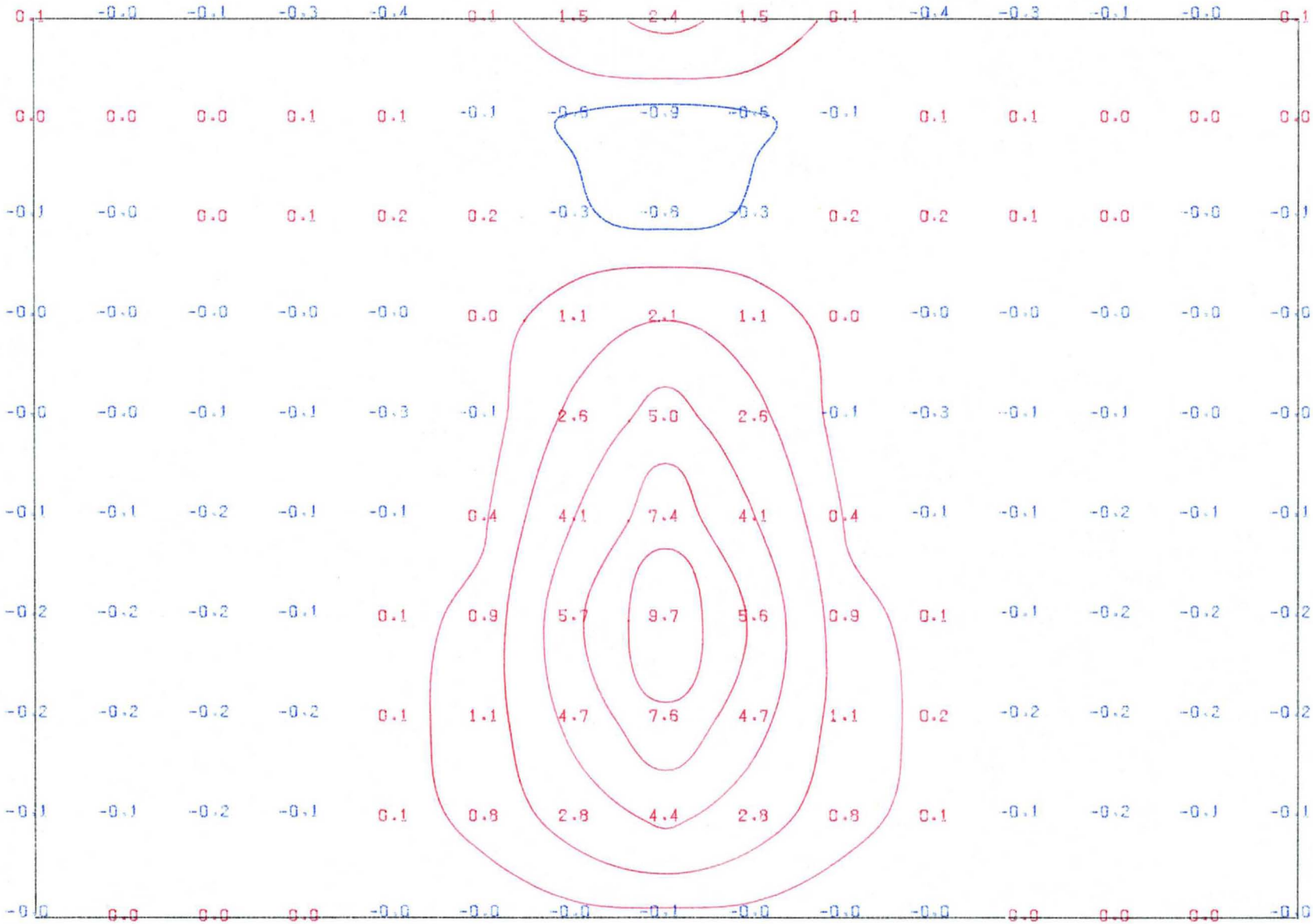


Fig. 6.22 VERTICAL VELOCITY IN METRES/SEC. AT 17.5MINS

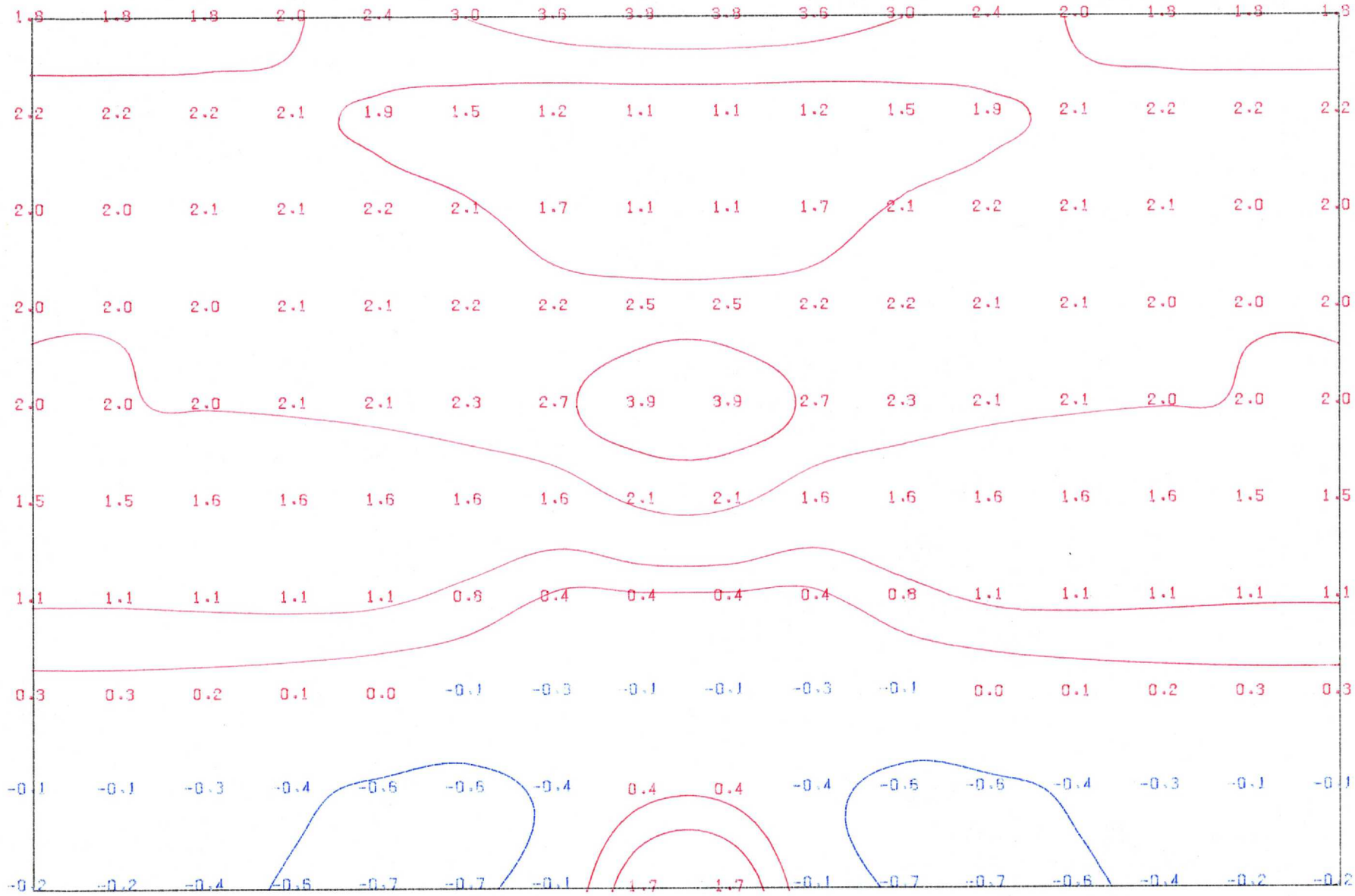


Fig. 6.23 DEVIATION HEIGHT FIELD IN METRES AT 17.5MINS.

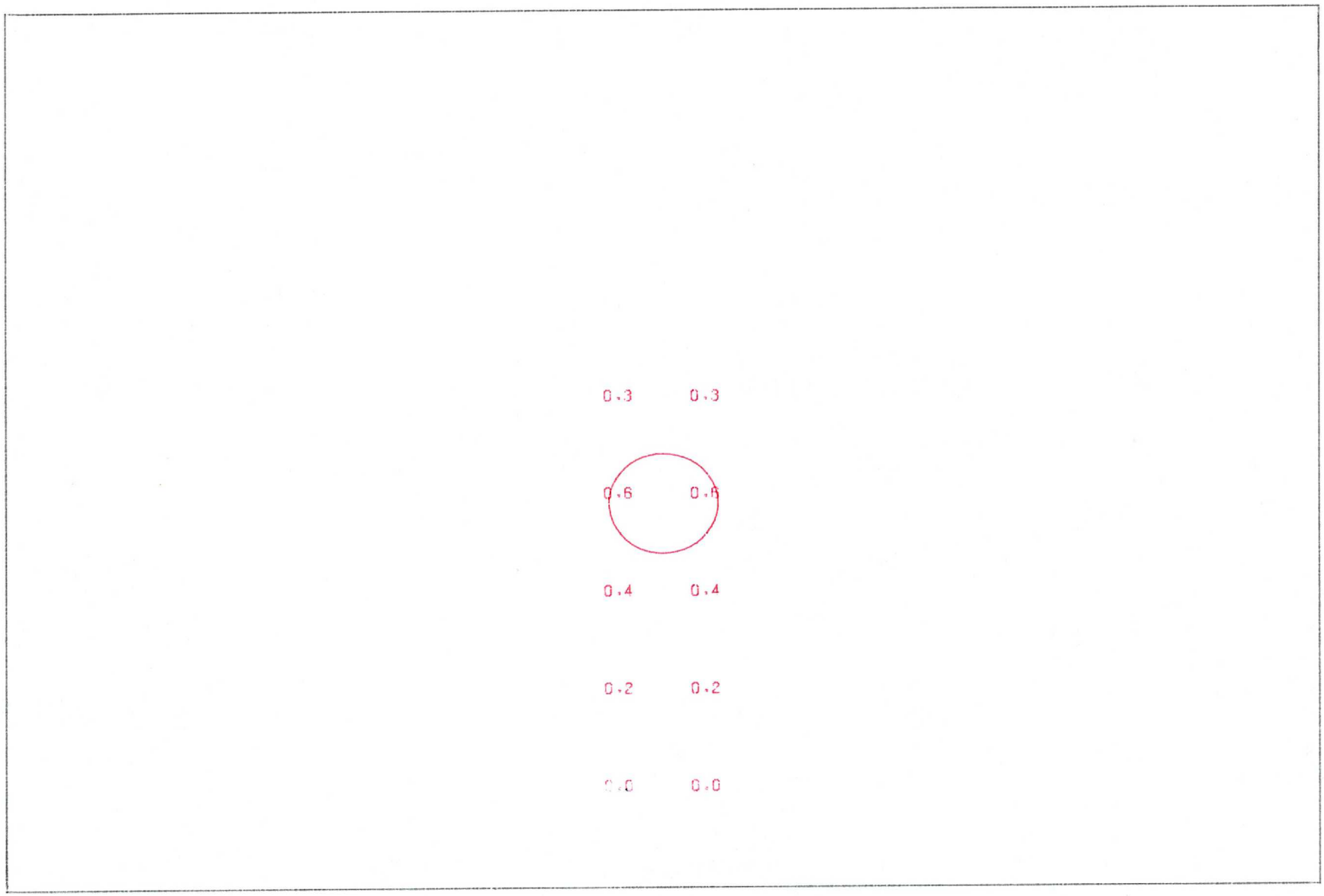


Fig. 6.24 CLOUDWATER CONTENT IN GM/KG. AT 17.5MINS

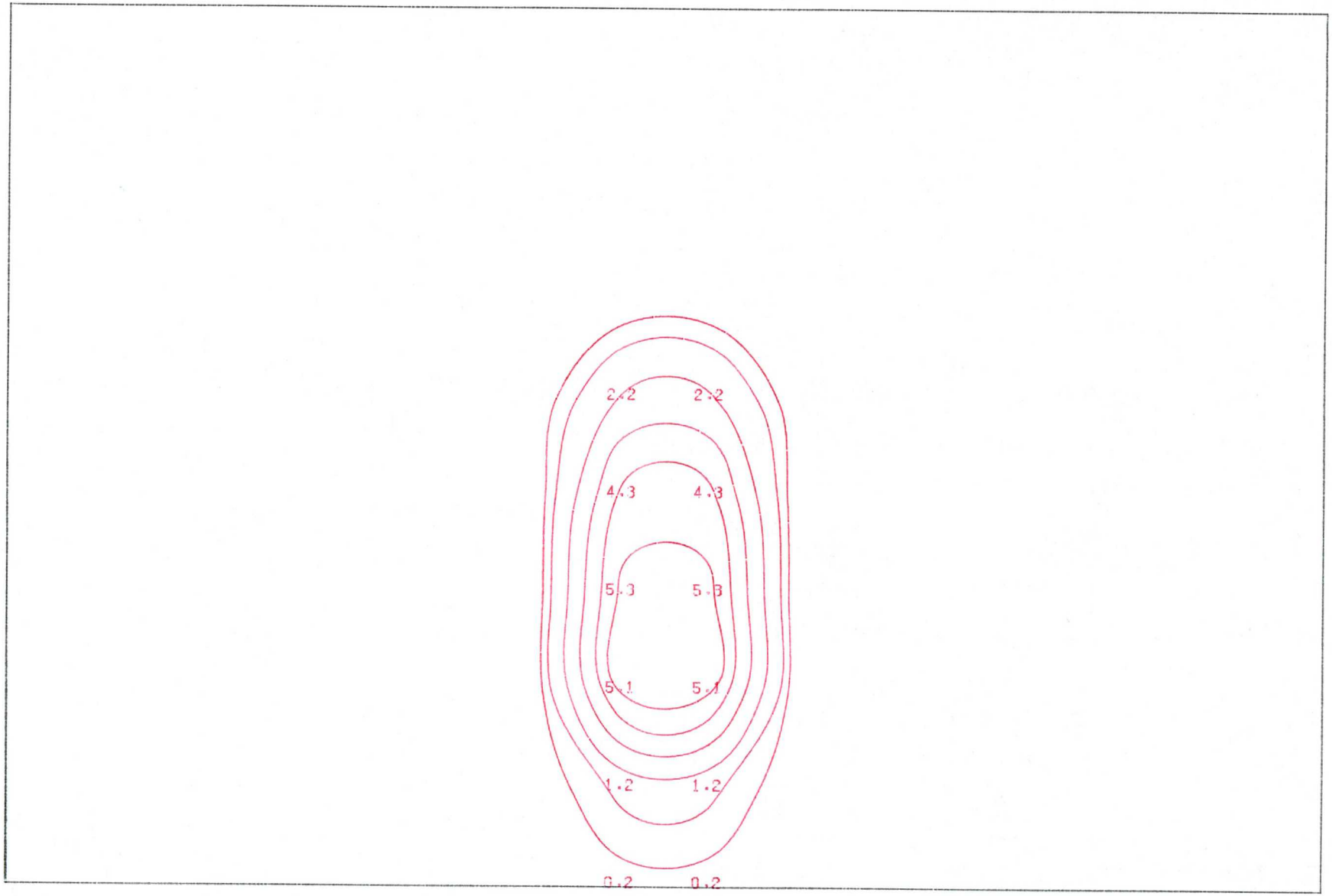


Fig. 6.25 RAINWATER CONTENT IN GM/KG. AT 17.5MINS

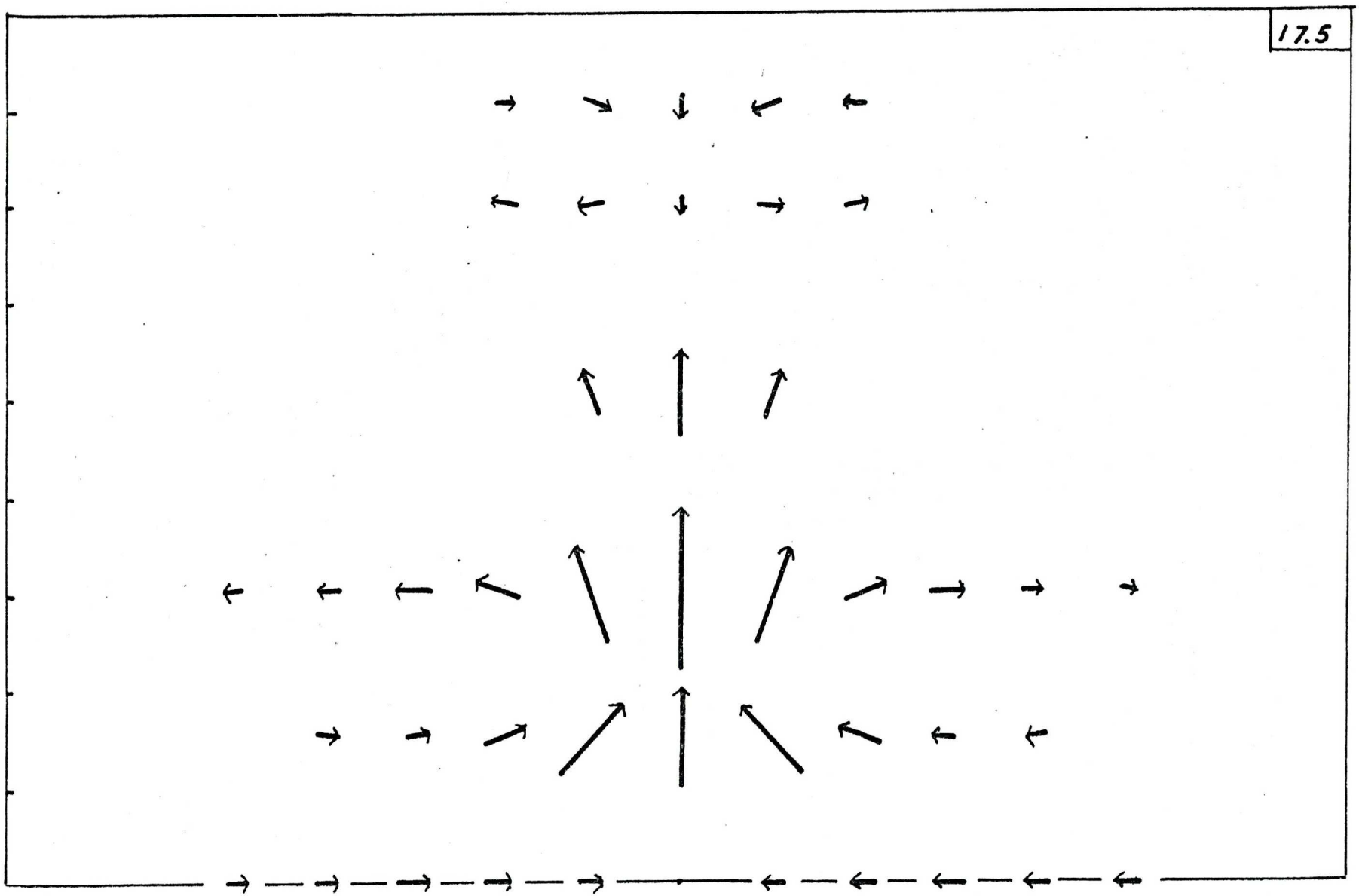
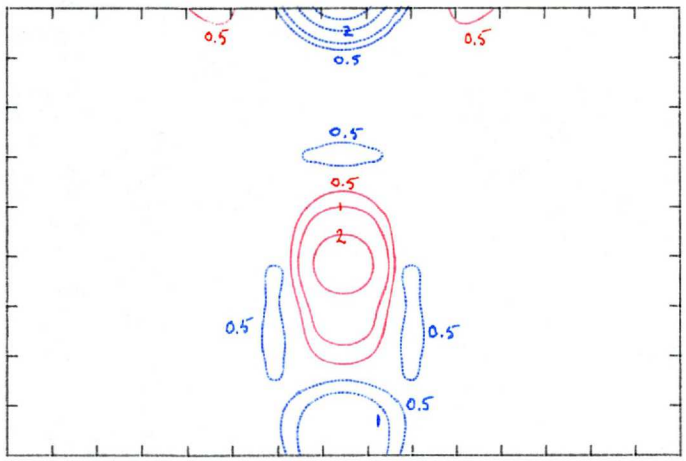
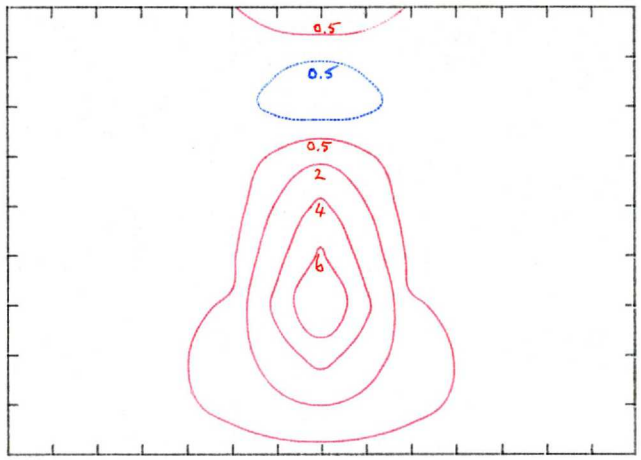


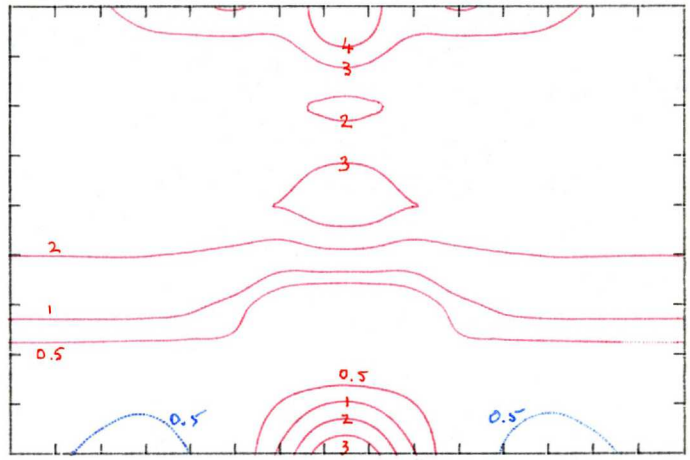
FIG 6.26 *RUN 2* [X, Y=Y_β, P] *WIND VECTORS* 10 m/s



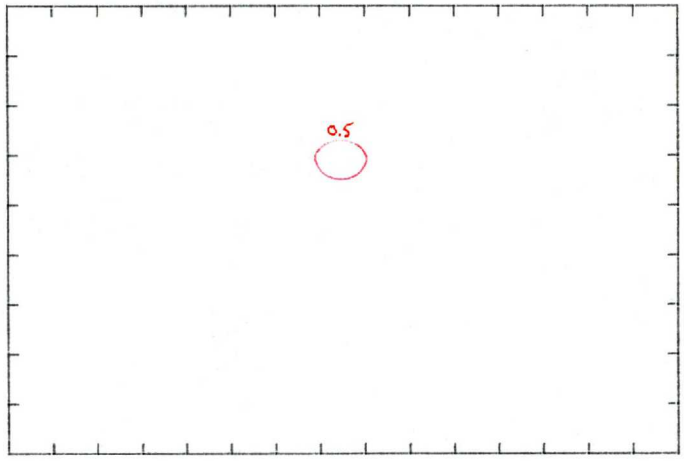
POTENTIAL TEMPERATURE DEVIATION IN DEG. C. AT 20.0 MINS.



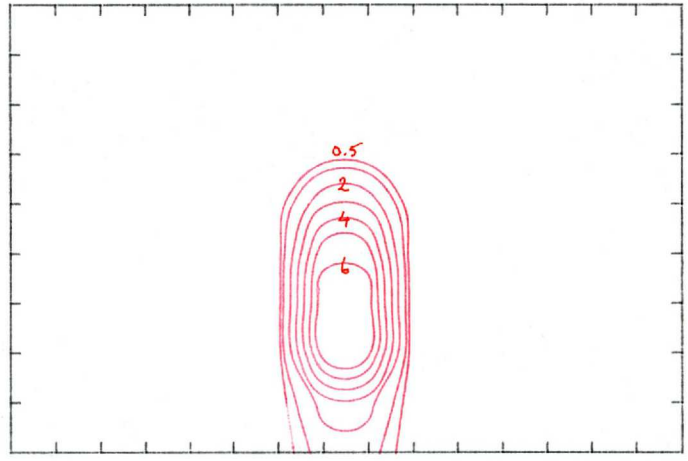
VERTICAL VELOCITY IN METRES/SEC. AT 20.0 MINS



DEVIATION HEIGHT FIELD IN METRES AT 20.0 MINS.



CLOUDWATER CONTENT IN GM/KG. AT 20.0 MINS



RAINWATER CONTENT IN GM/KG. AT 20.0 MINS

Fig. 6.27

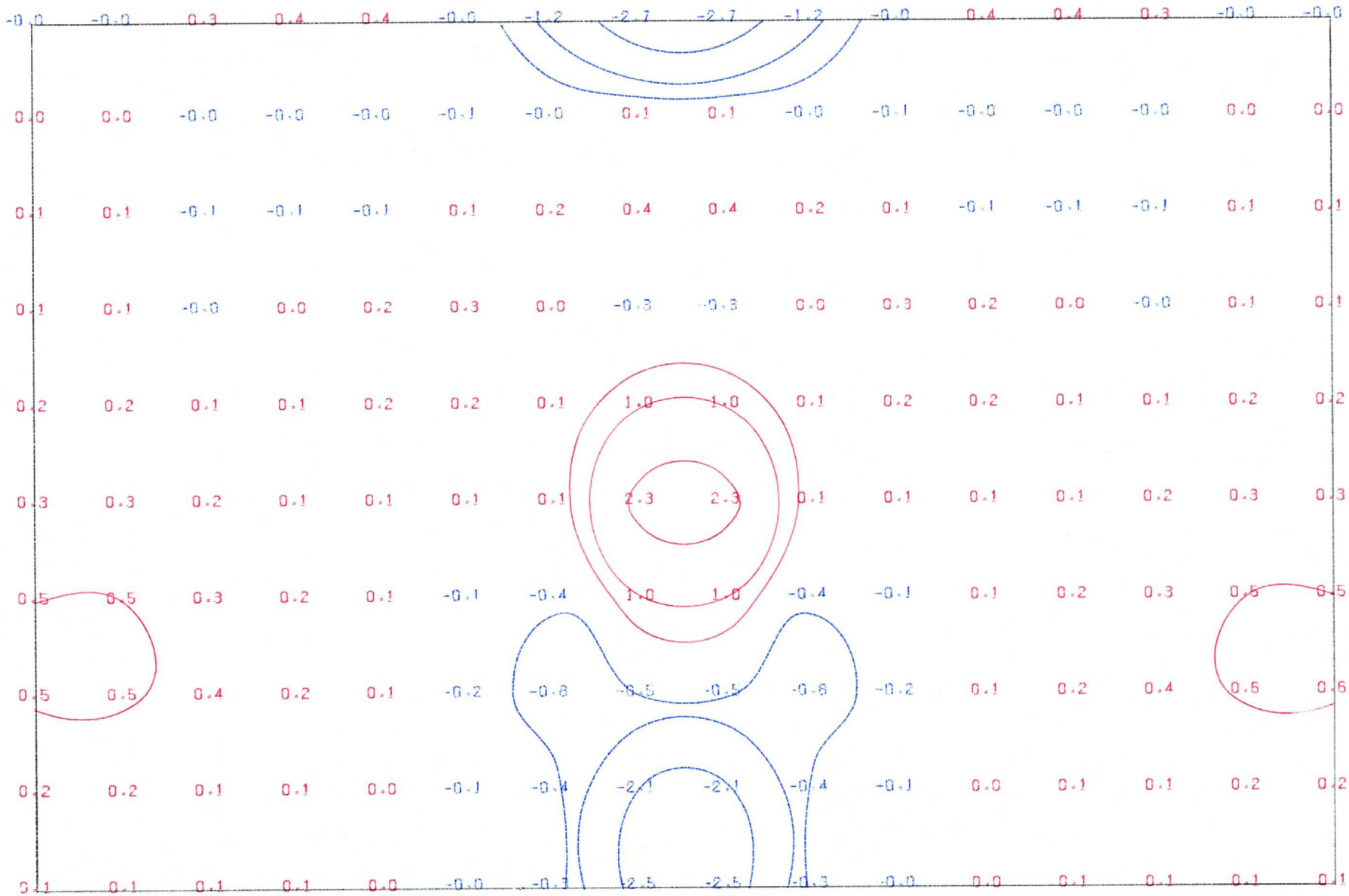


Fig. 6.28 POTENTIAL TEMPERATURE DEVIATION IN DEG. C. AT 25.0MINS.

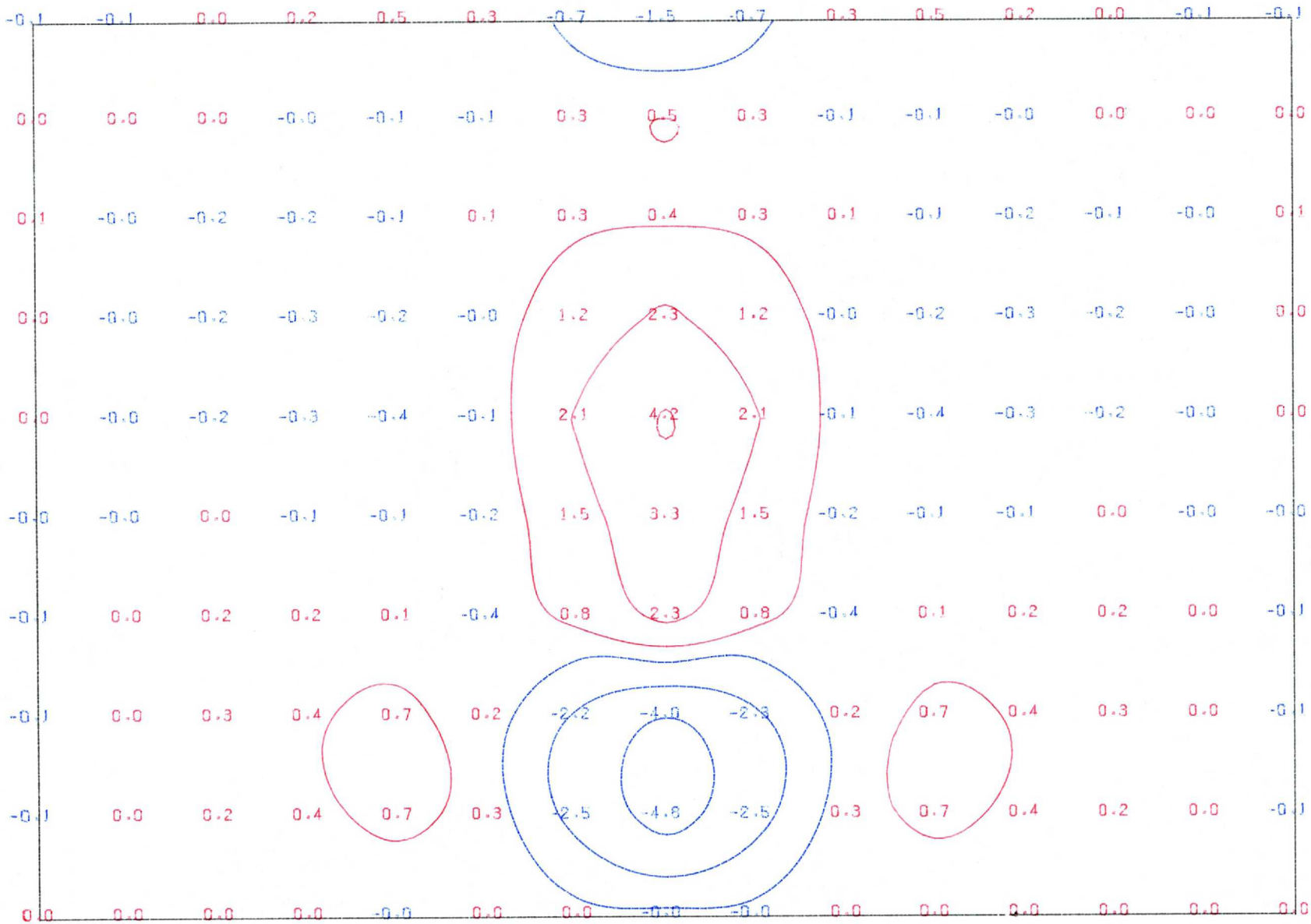


Fig. 6.29 VERTICAL VELOCITY IN METRES/SEC. AT 25.0MINS

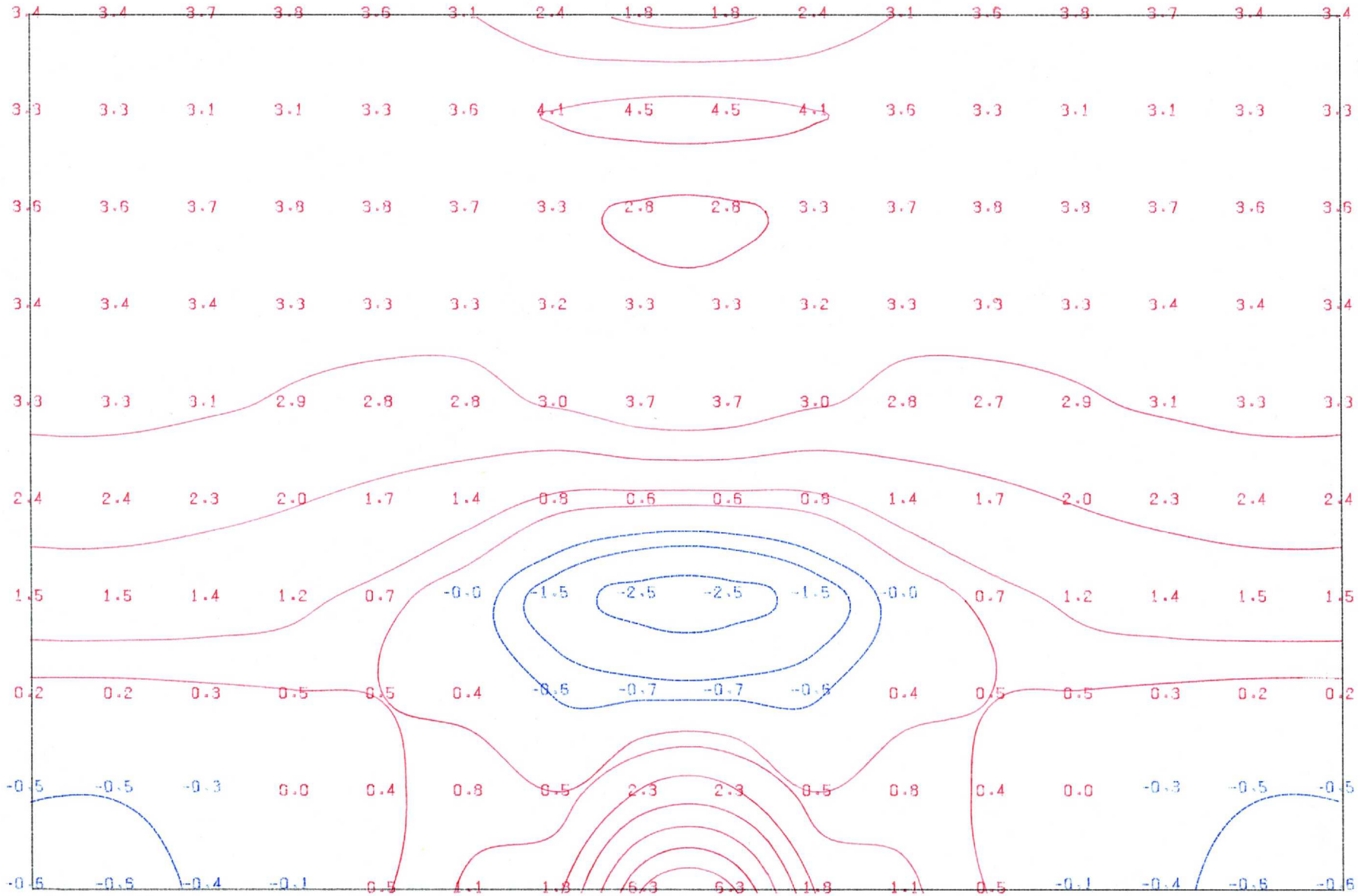


Fig. 6.30 DEVIATION HEIGHT FIELD IN METRES AT 25.0MINS.

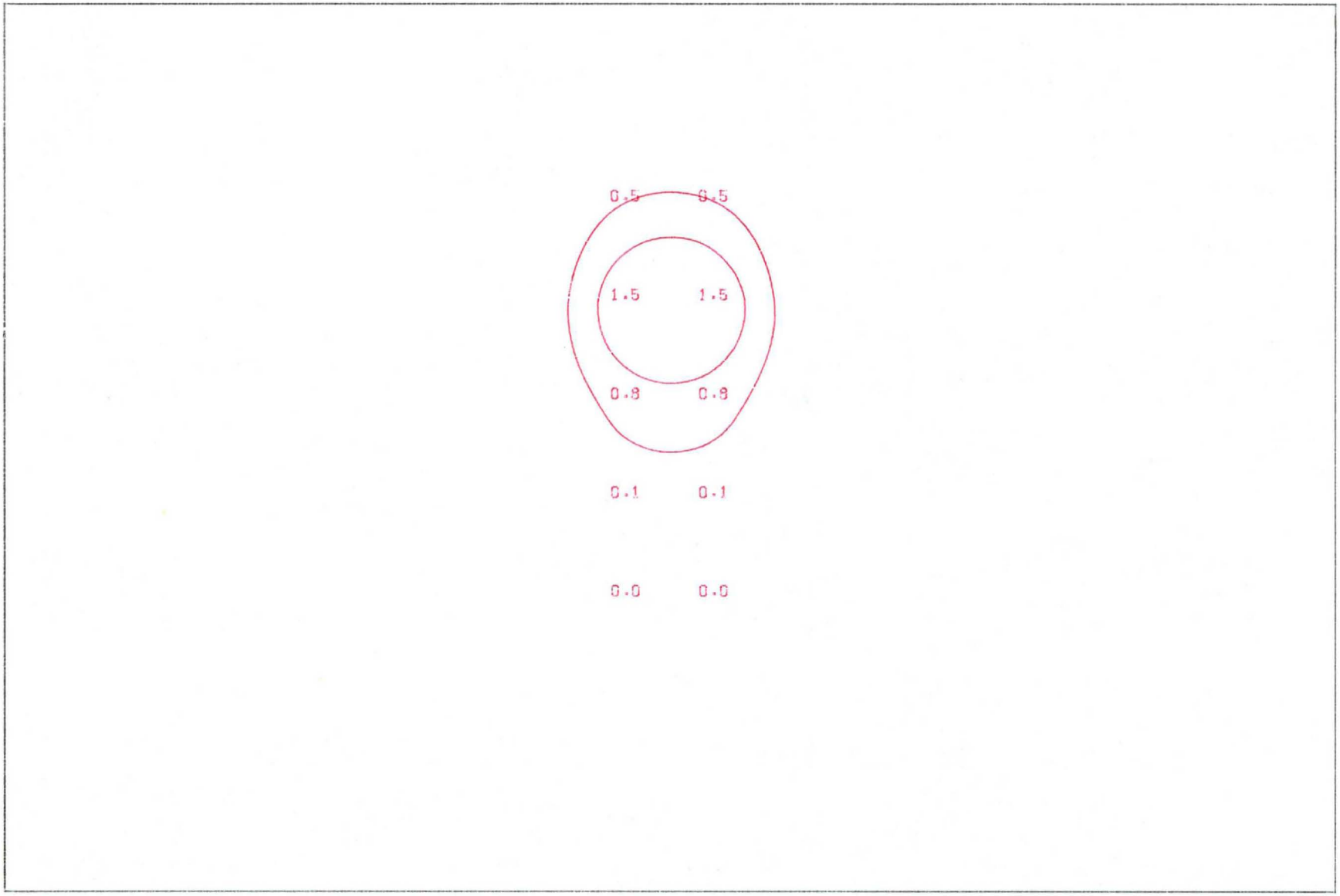


Fig. 6.31 CLOUDWATER CONTENT IN GM/KG. AT 25.0MINS

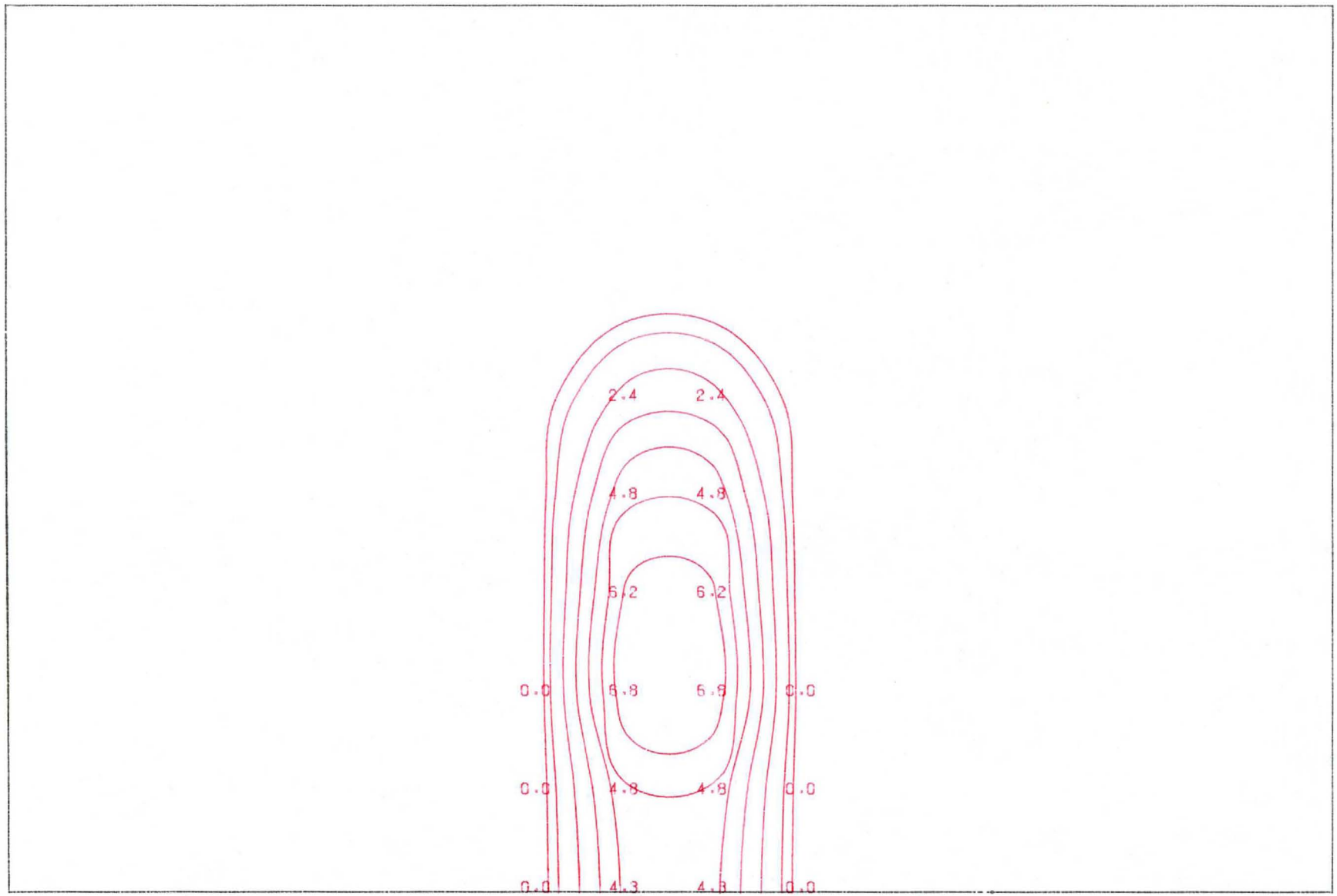


Fig. 6.32 RAINWATER CONTENT IN GM/KG. AT 25.0MINS

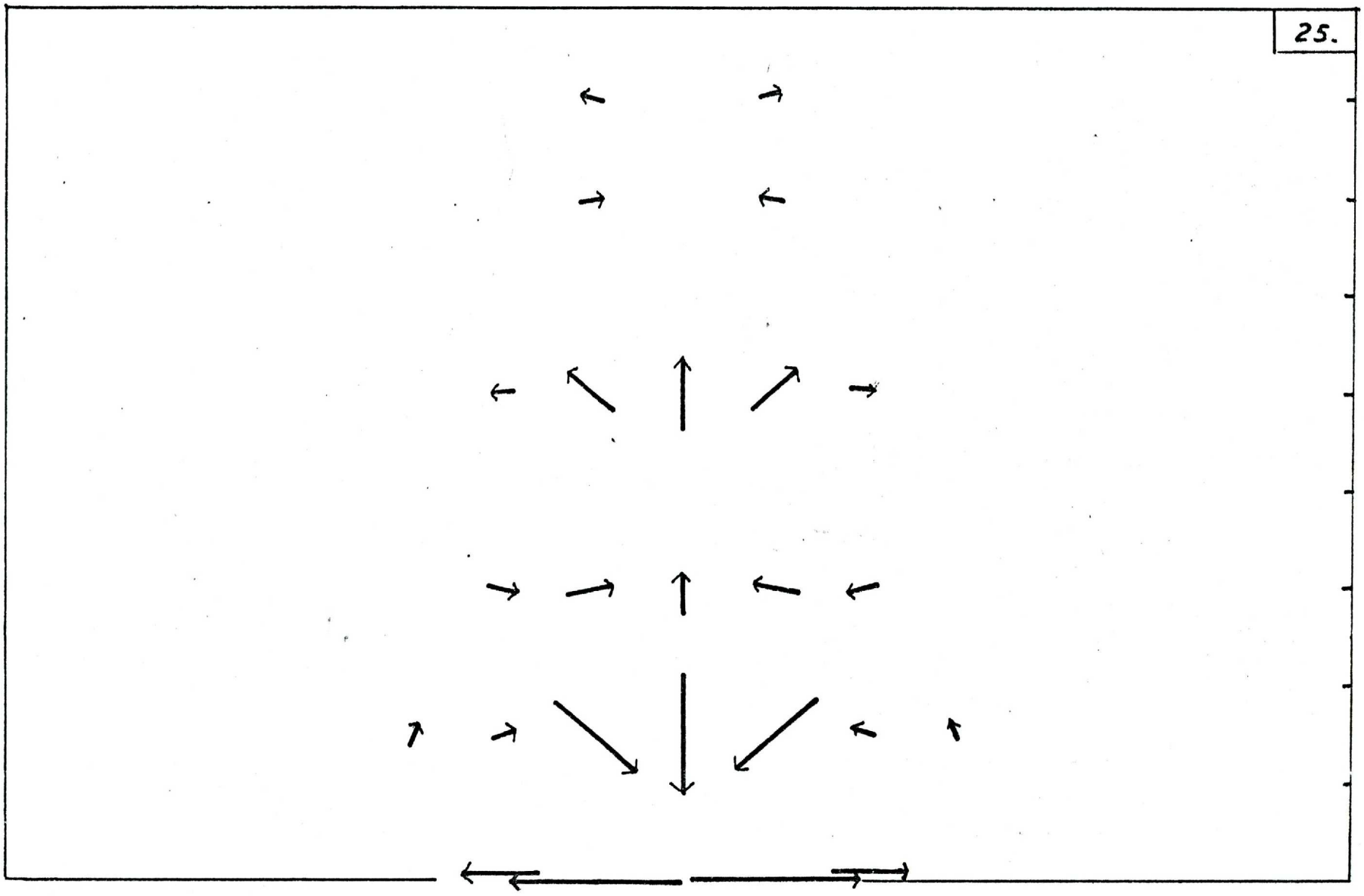
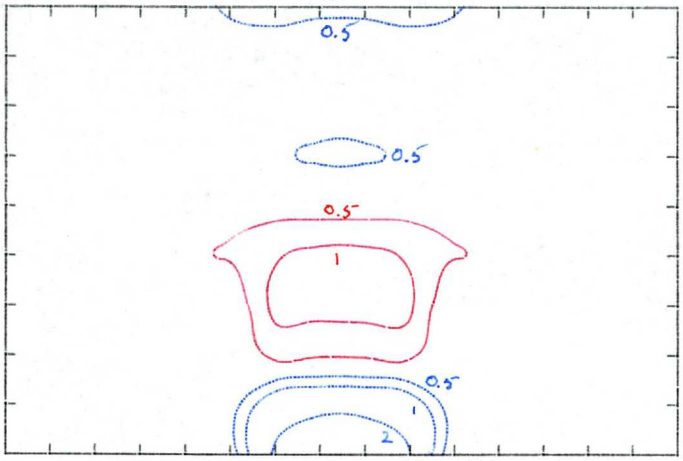
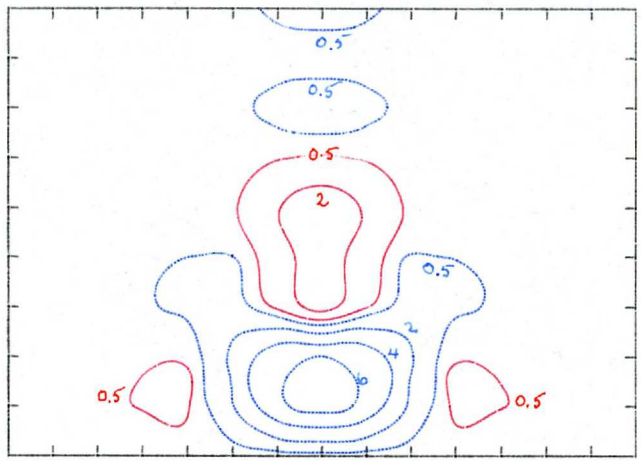


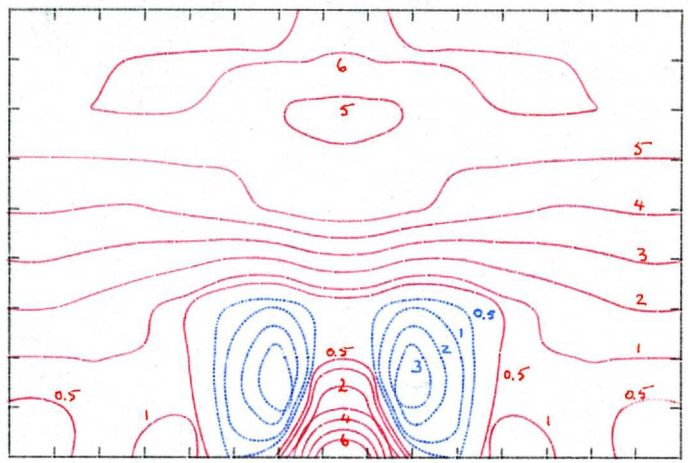
FIG 6.33 *RUN 2* [x, y-y_s, p] *WIND VECTORS*



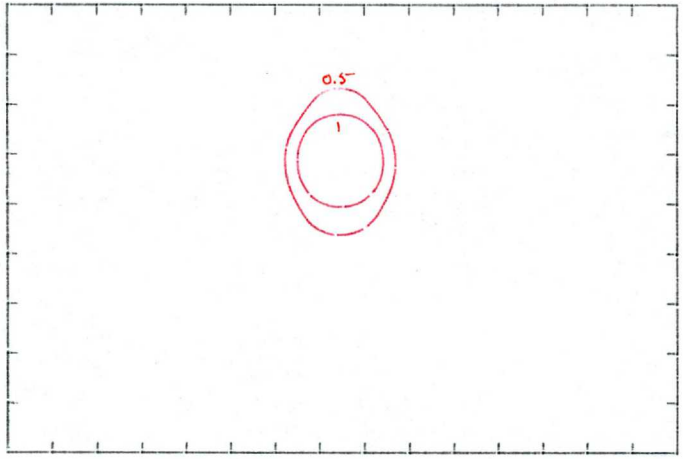
POTENTIAL TEMPERATURE DEVIATION IN DEG. C. AT 30.0 MINS.



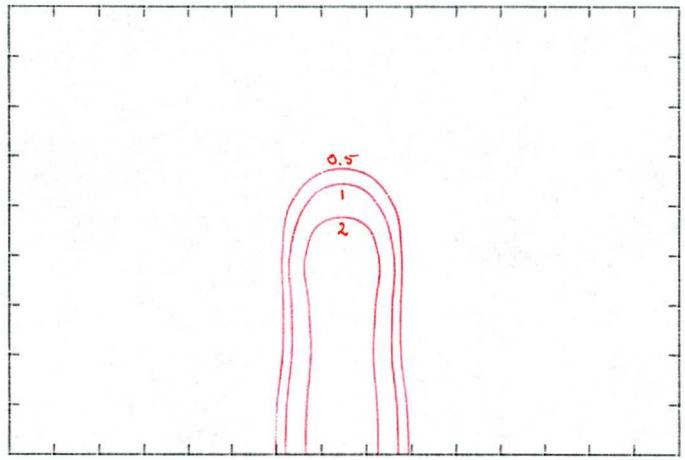
VERTICAL VELOCITY IN METRES/SEC. AT 30.0 MINS



DEVIATION HEIGHT FIELD IN METRES AT 30.0 MINS.



CLOUDWATER CONTENT IN GM/KG. AT 30.0 MINS



RAINWATER CONTENT IN GM/KG. AT 30.0 MINS

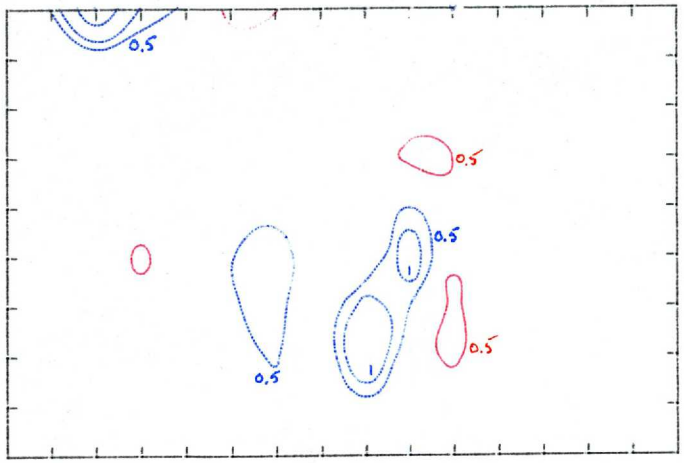
Fig. 6.34

RUN 3

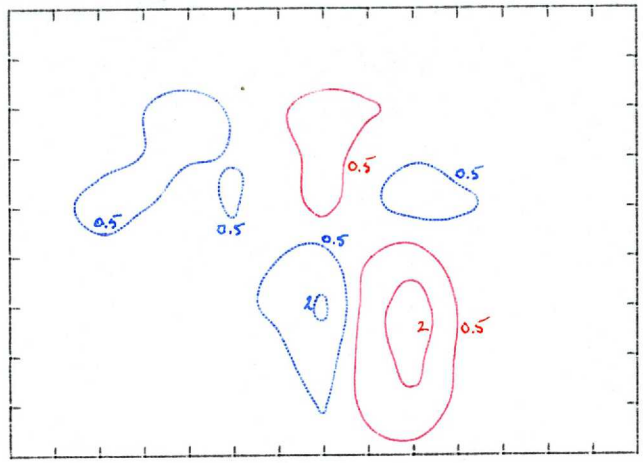
Figs. 6.35 - 6.50

Initial sheared flow (see text)
Coriolis parameter zero

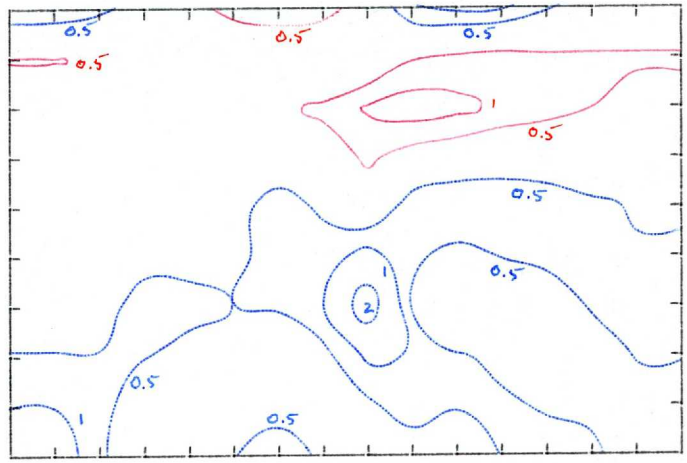
$\alpha = 10^{-3}\ s^{-1}$
$\beta = 10^{-3}\ s^{-1}$
$\gamma = 2 \times 10^{-3}\ s^{-1}$



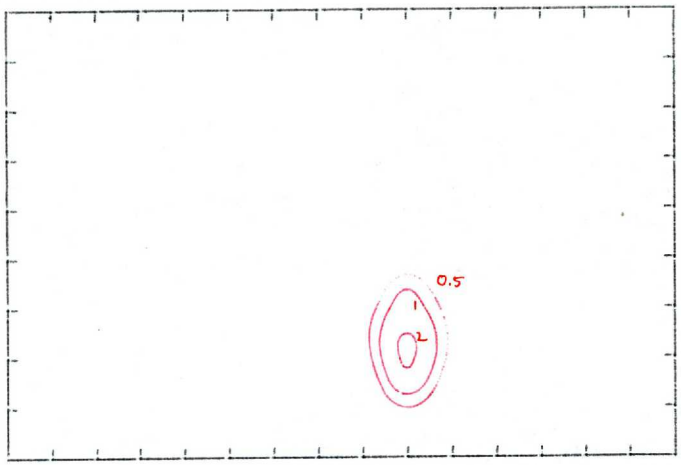
POTENTIAL TEMPERATURE DEVIATION IN DEG. C. AT 25.0 MINS.



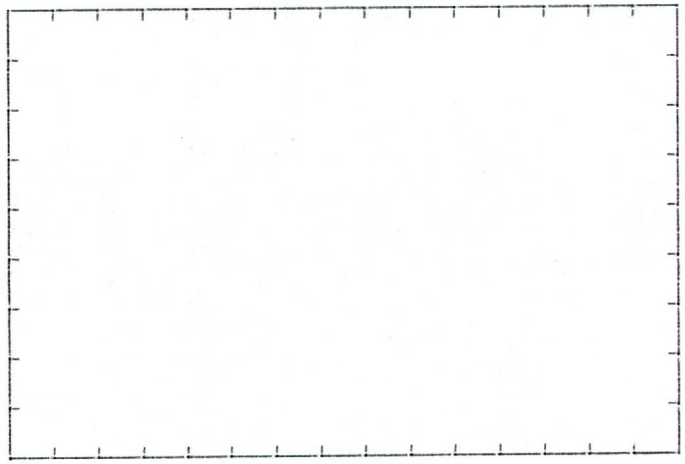
VERTICAL VELOCITY IN METRES/SEC. AT 25.0 MINS



DEVIATION HEIGHT FIELD IN METRES AT 25.0 MINS.



CLOUDWATER CONTENT IN GM/KG. AT 25.0 MINS



RAINWATER CONTENT IN GM/KG. AT 25.0 MINS

Fig. 6.35

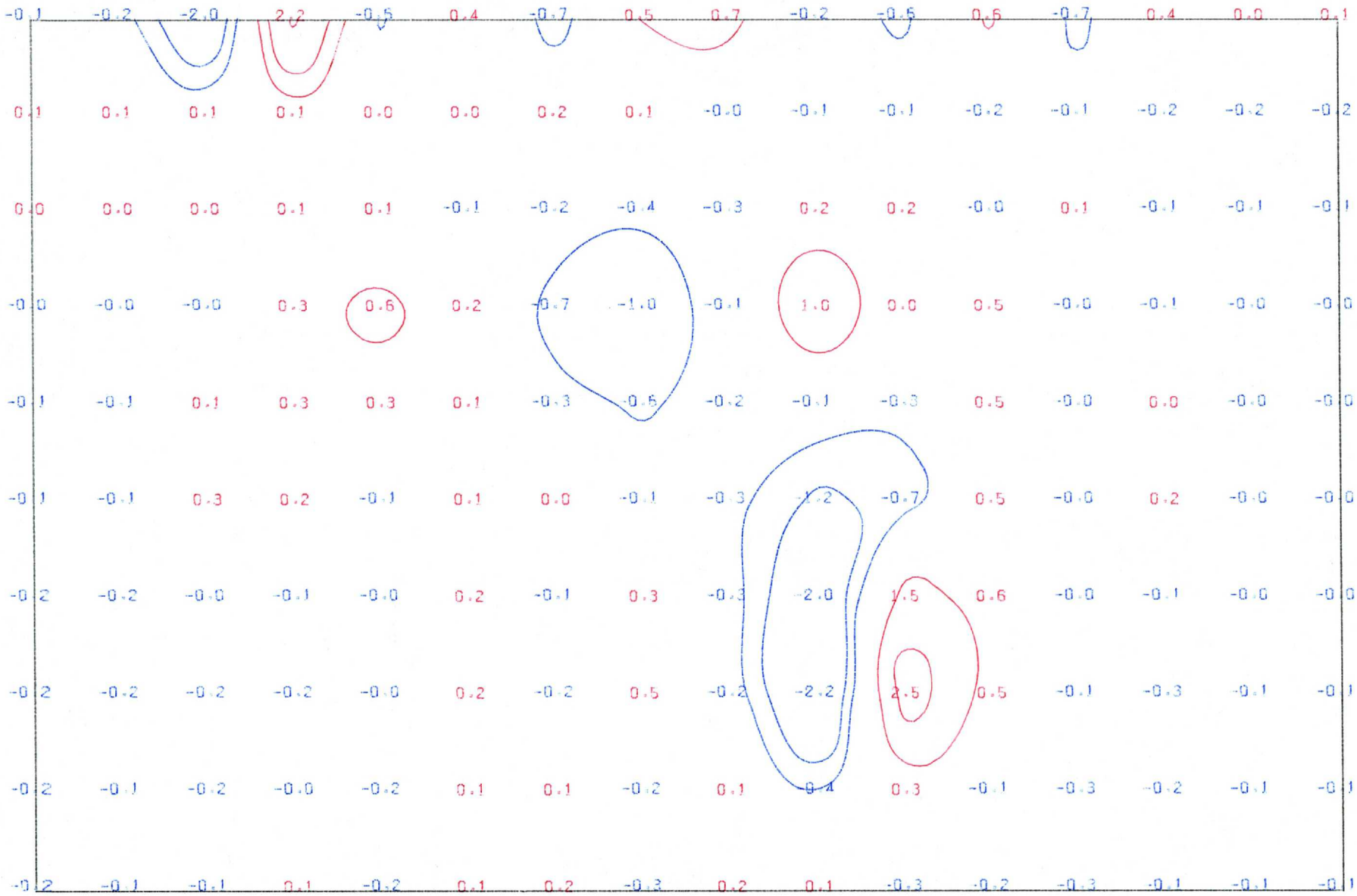


Fig. 6.36 POTENTIAL TEMPERATURE DEVIATION IN DEG. C. AT 30.0MINS.

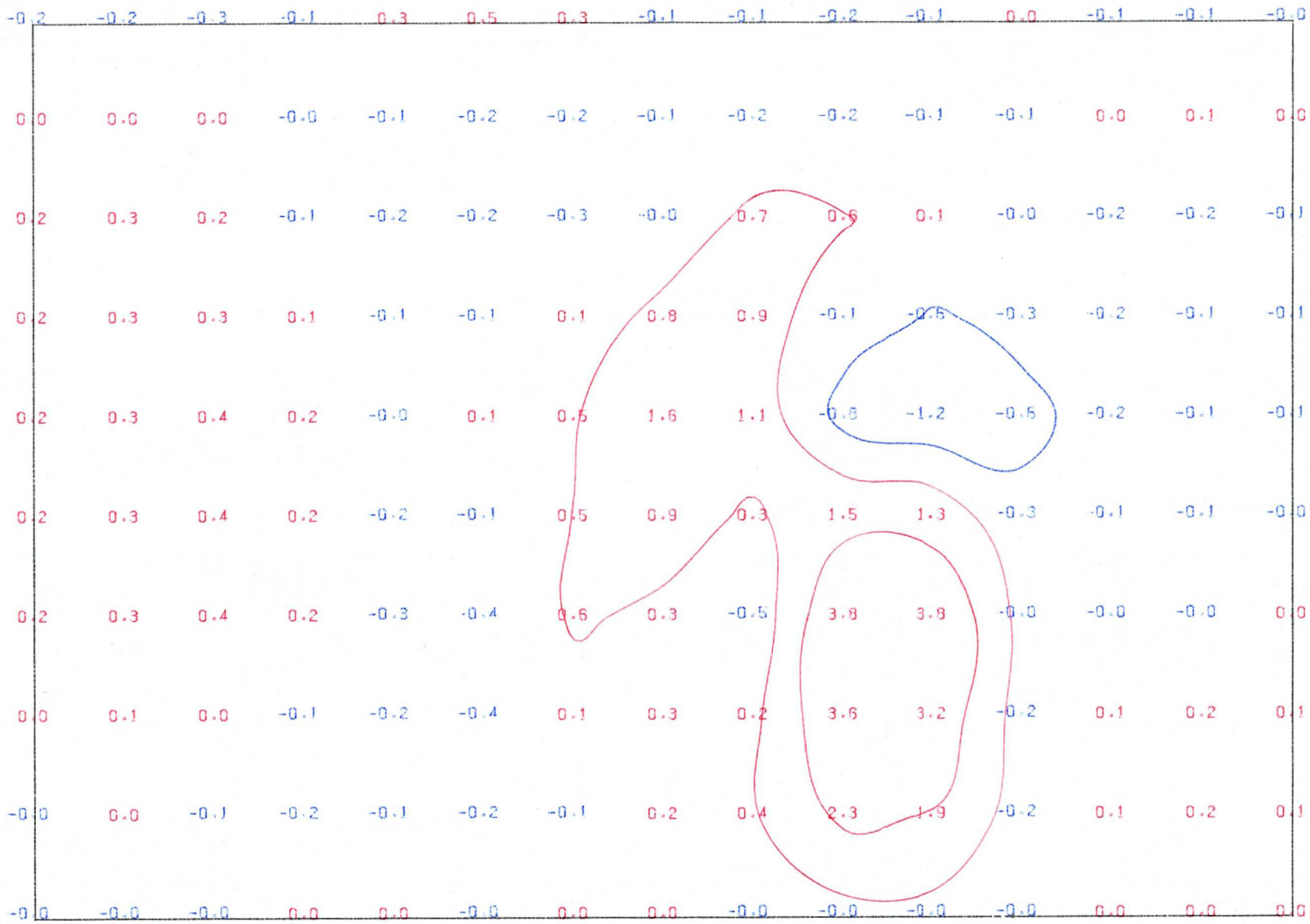


Fig. 6.37 VERTICAL VELOCITY IN METRES/SEC. AT 30.0MINS

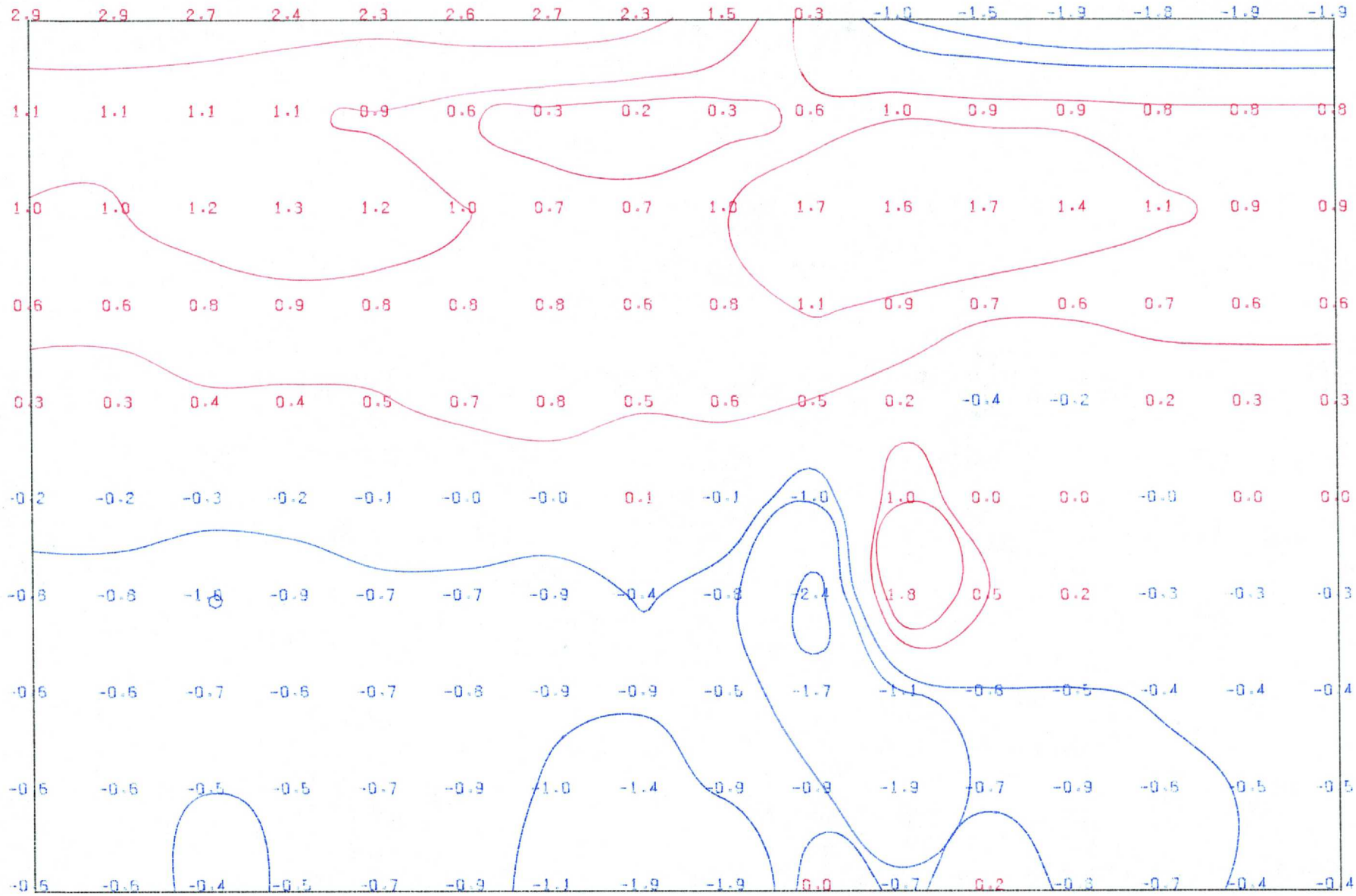


Fig. 6.38 DEVIATION HEIGHT FIELD IN METRES AT 30.0MINS.

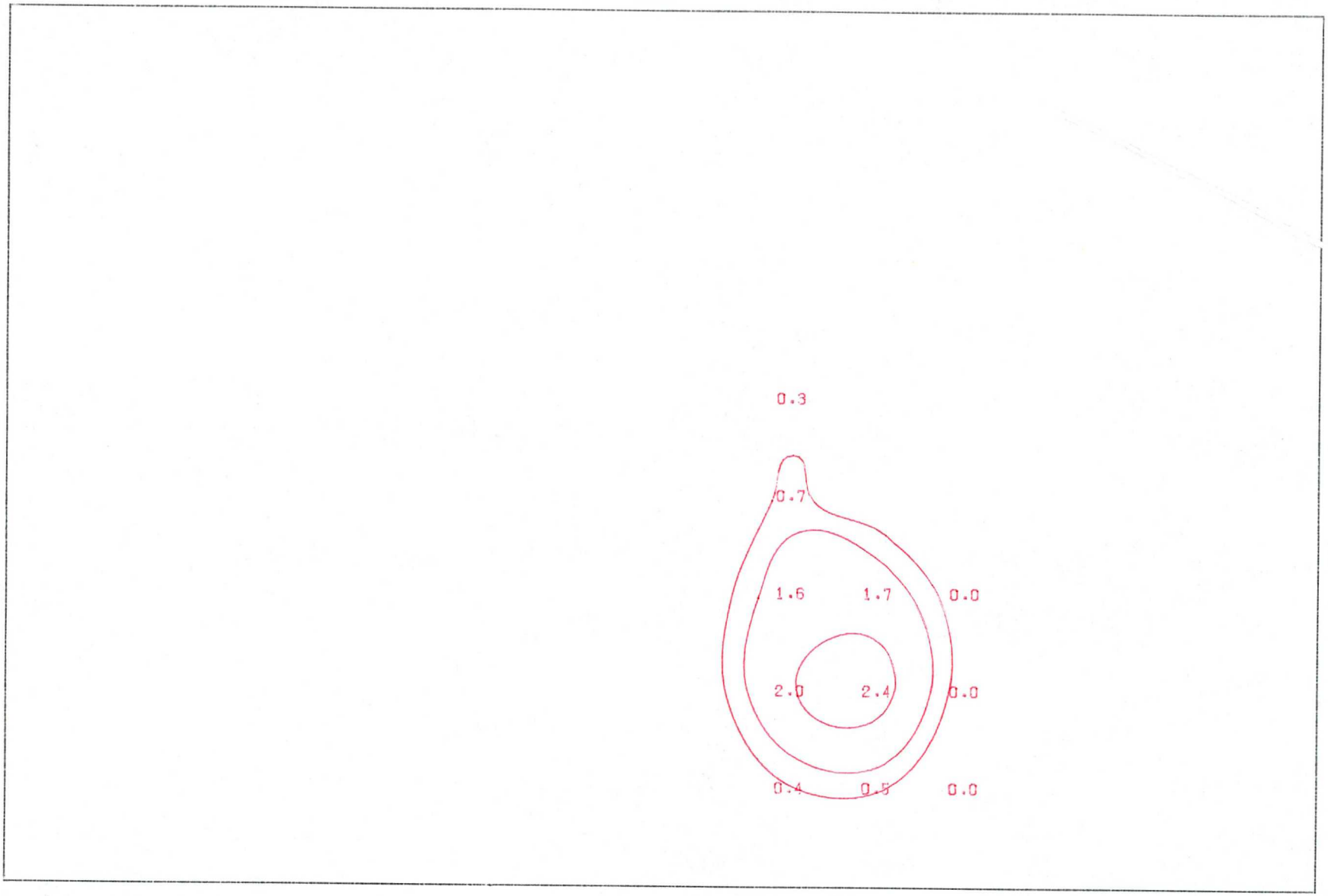


Fig. 6.39 CLOUDWATER CONTENT IN GM/KG. AT 30.0MINS

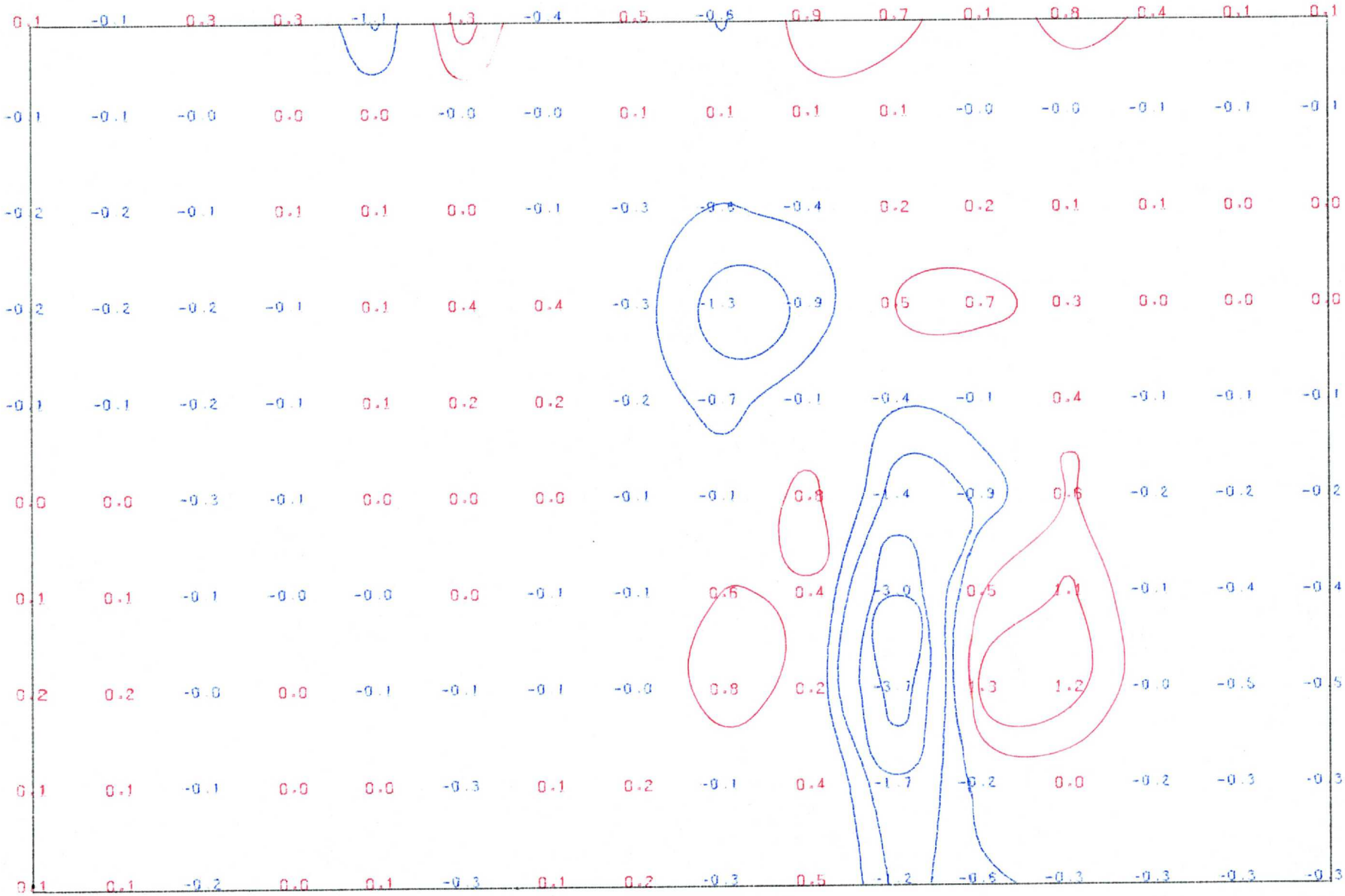


Fig. 6.40 POTENTIAL TEMPERATURE DEVIATION IN DEG. C. AT 37.5MINS.

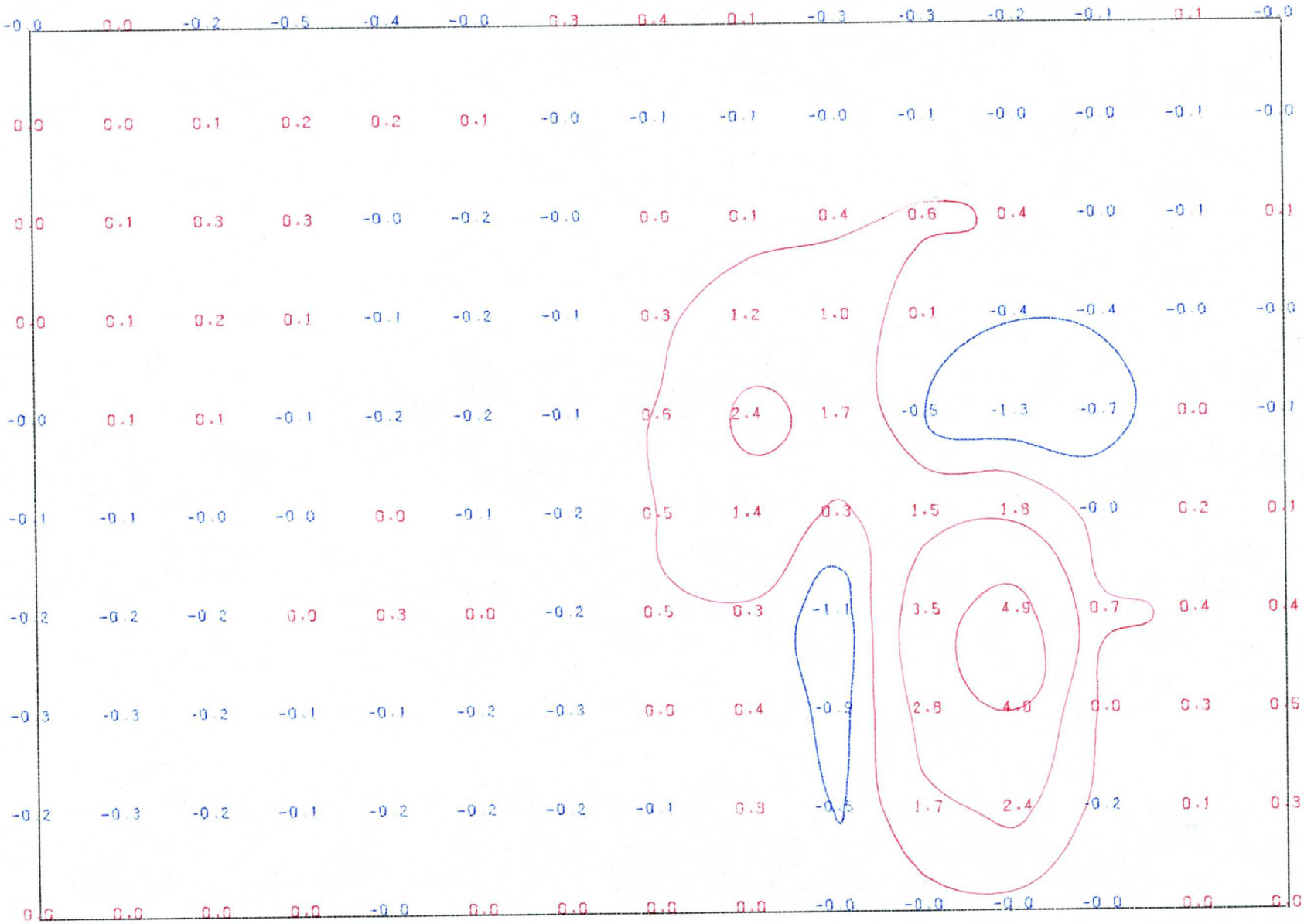


Fig. 6.41 VERTICAL VELOCITY IN METRES/SEC. AT 37.5MINS

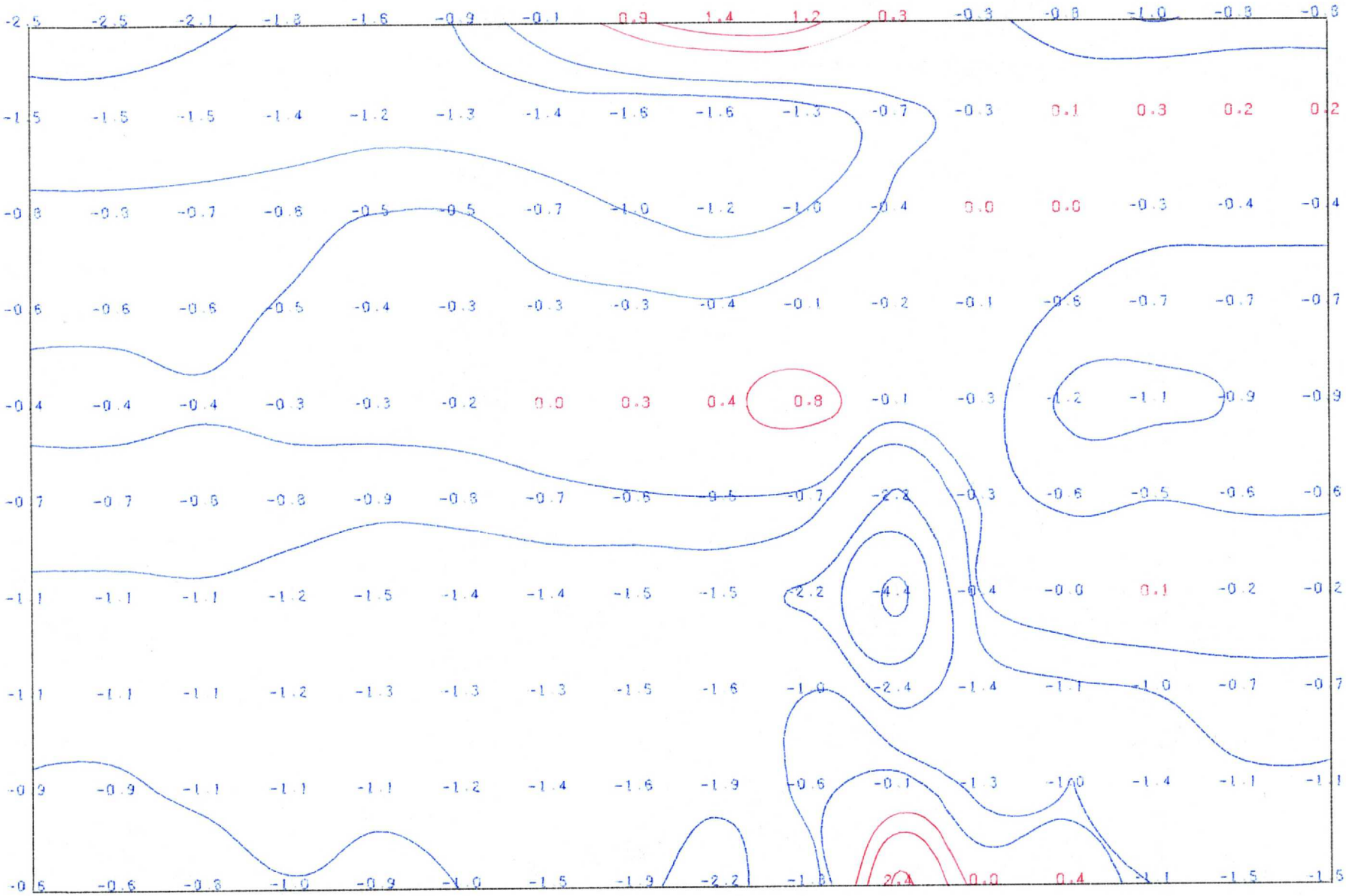


Fig. 6.42 DEVIATION HEIGHT FIELD IN METRES AT 37.5MINS.

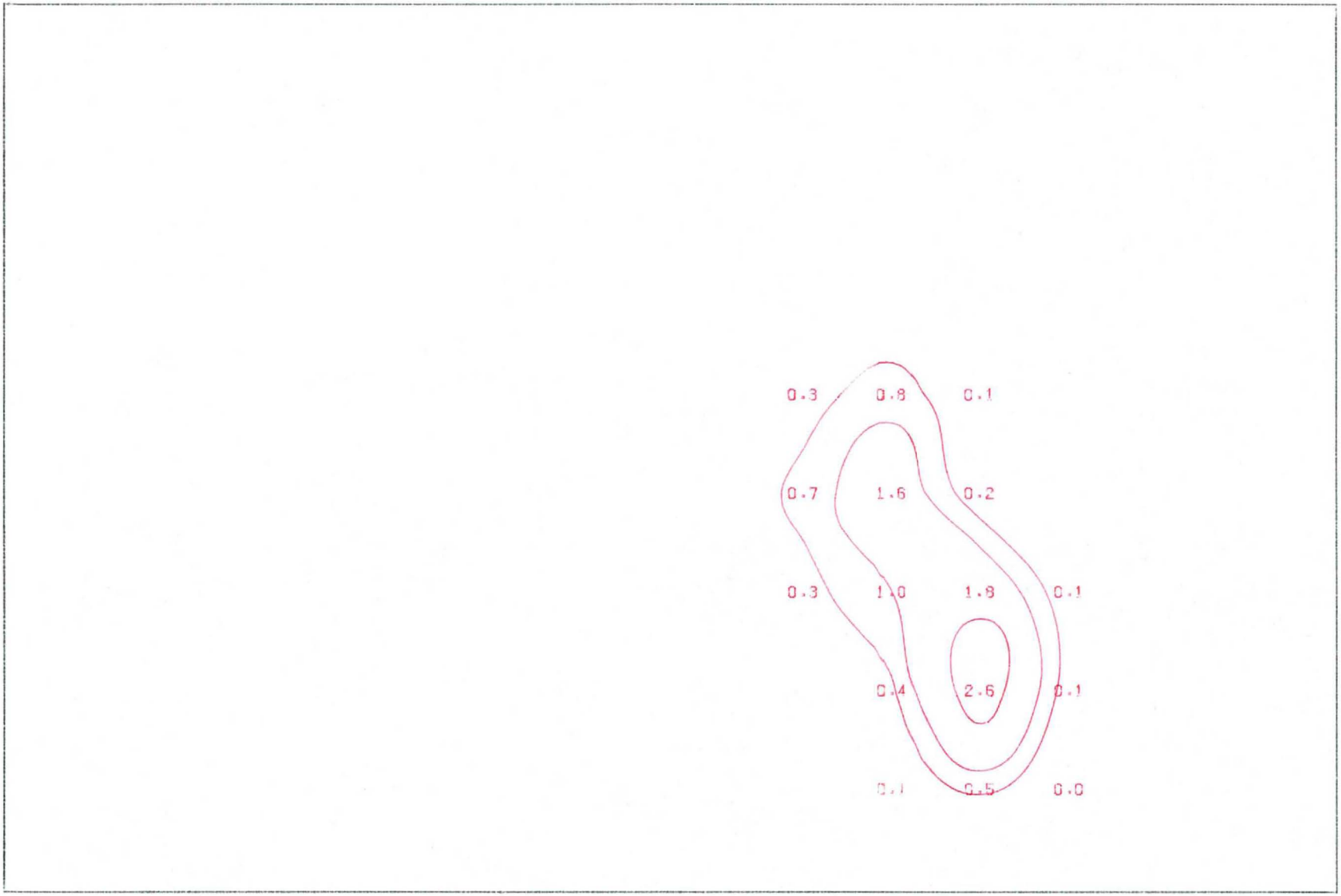


Fig. 6.43 CLOUDWATER CONTENT IN GM/KG. AT 37.5MINS

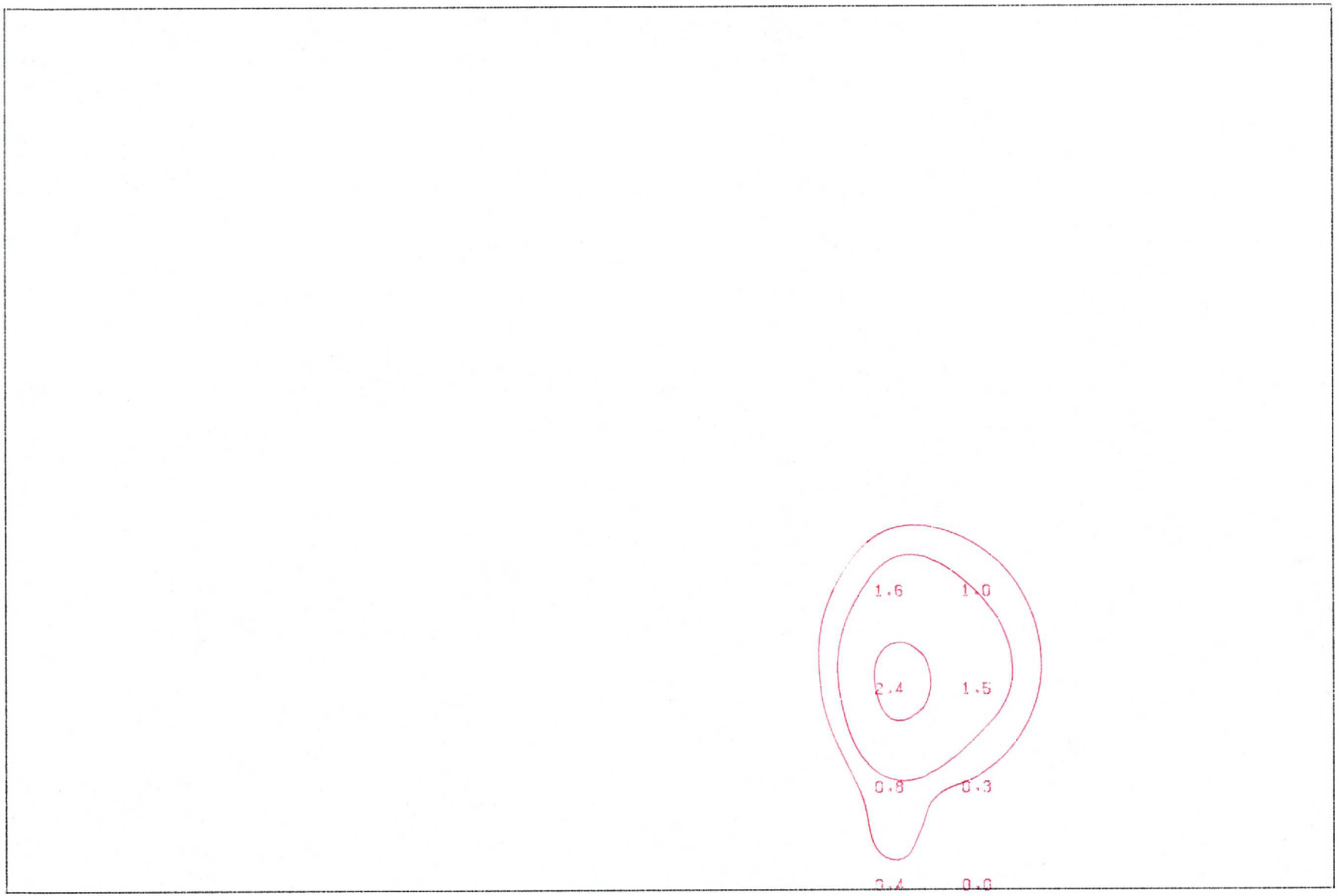


Fig. 6.44 RAINWATER CONTENT IN GM/KG. AT 37.5MINS

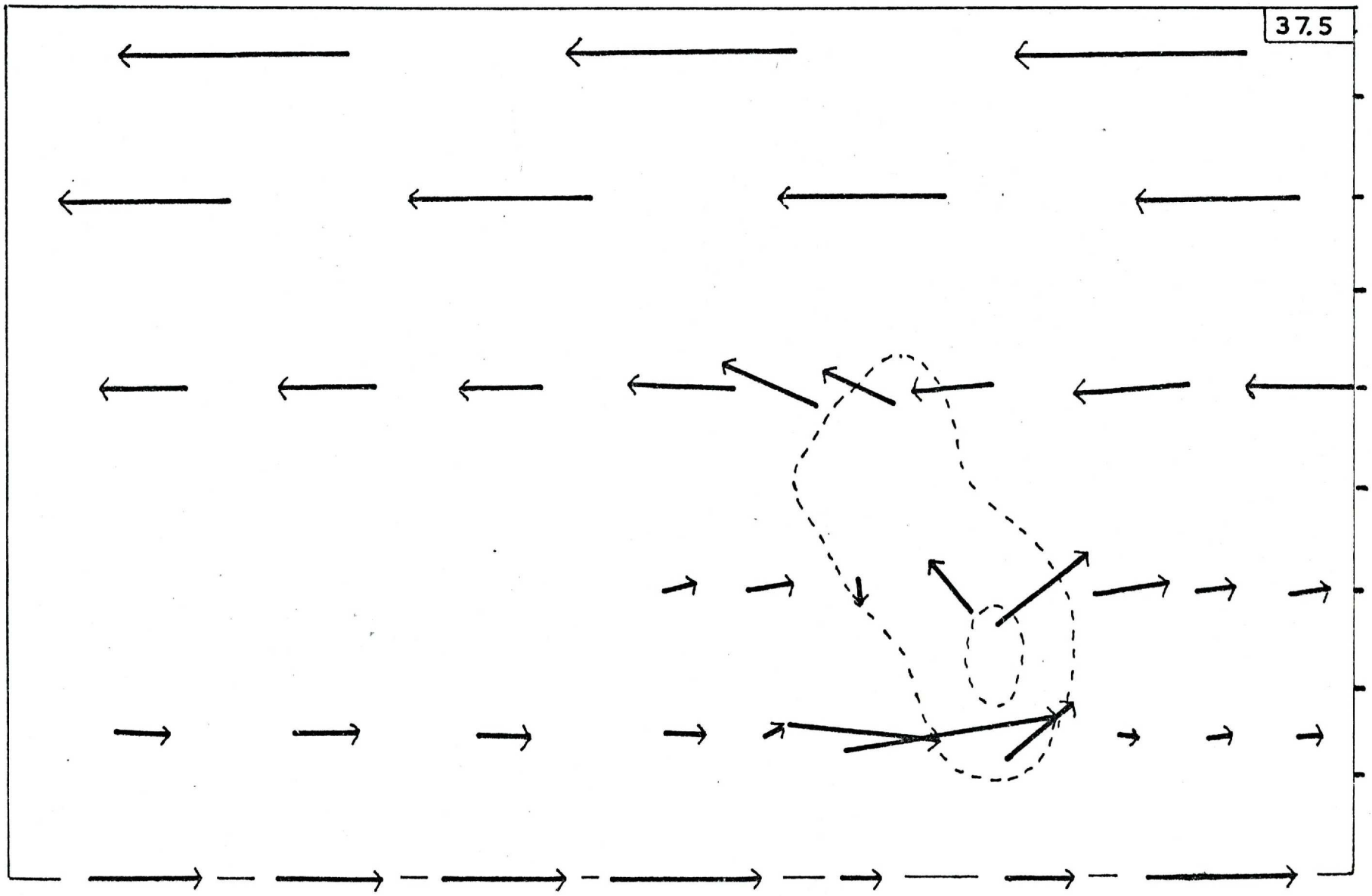
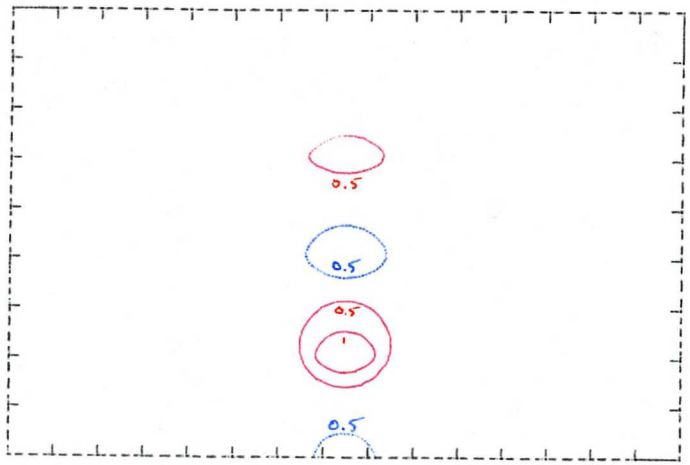
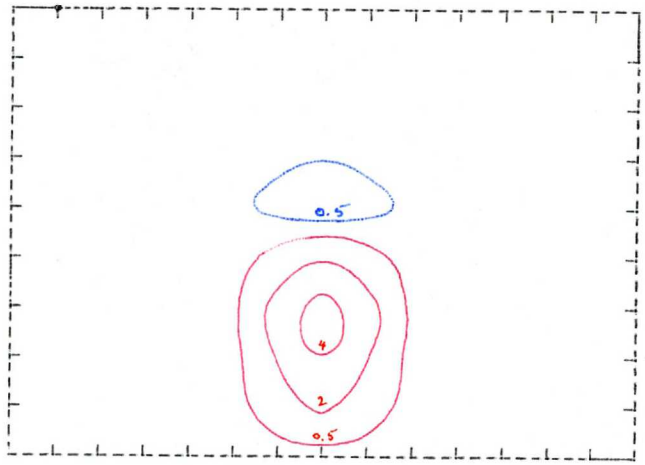


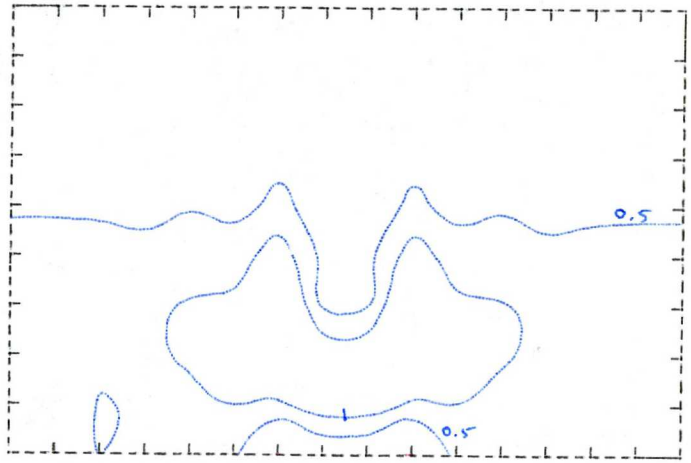
Fig. 6.45 *RUN* 3 [X , Y=Y_s,P] WIND VECTORS



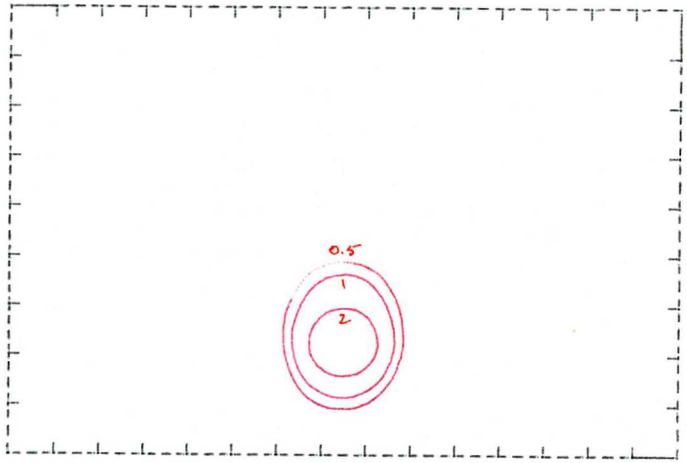
POTENTIAL TEMPERATURE DEVIATION IN DEG. C. AT 37.5 MINS.



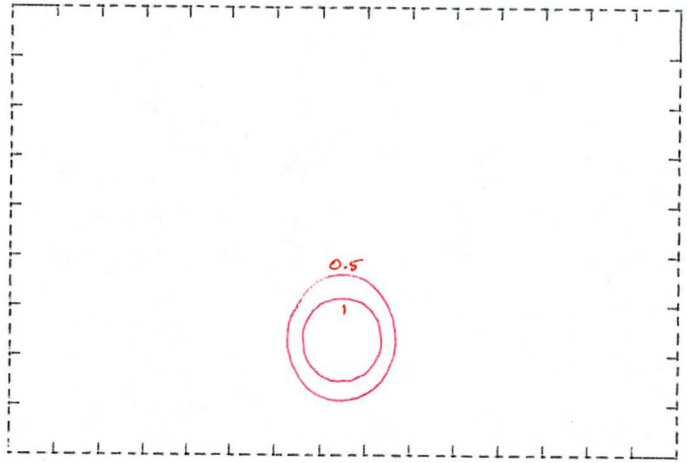
VERTICAL VELOCITY IN METRES/SEC. AT 37.5 MINS



DEVIATION HEIGHT FIELD IN METRES AT 37.5 MINS.



CLOUDWATER CONTENT IN GM./KG. AT 37.5 MINS



RAINWATER CONTENT IN GM/KG. AT 37.5 MINS

Fig. 6.46

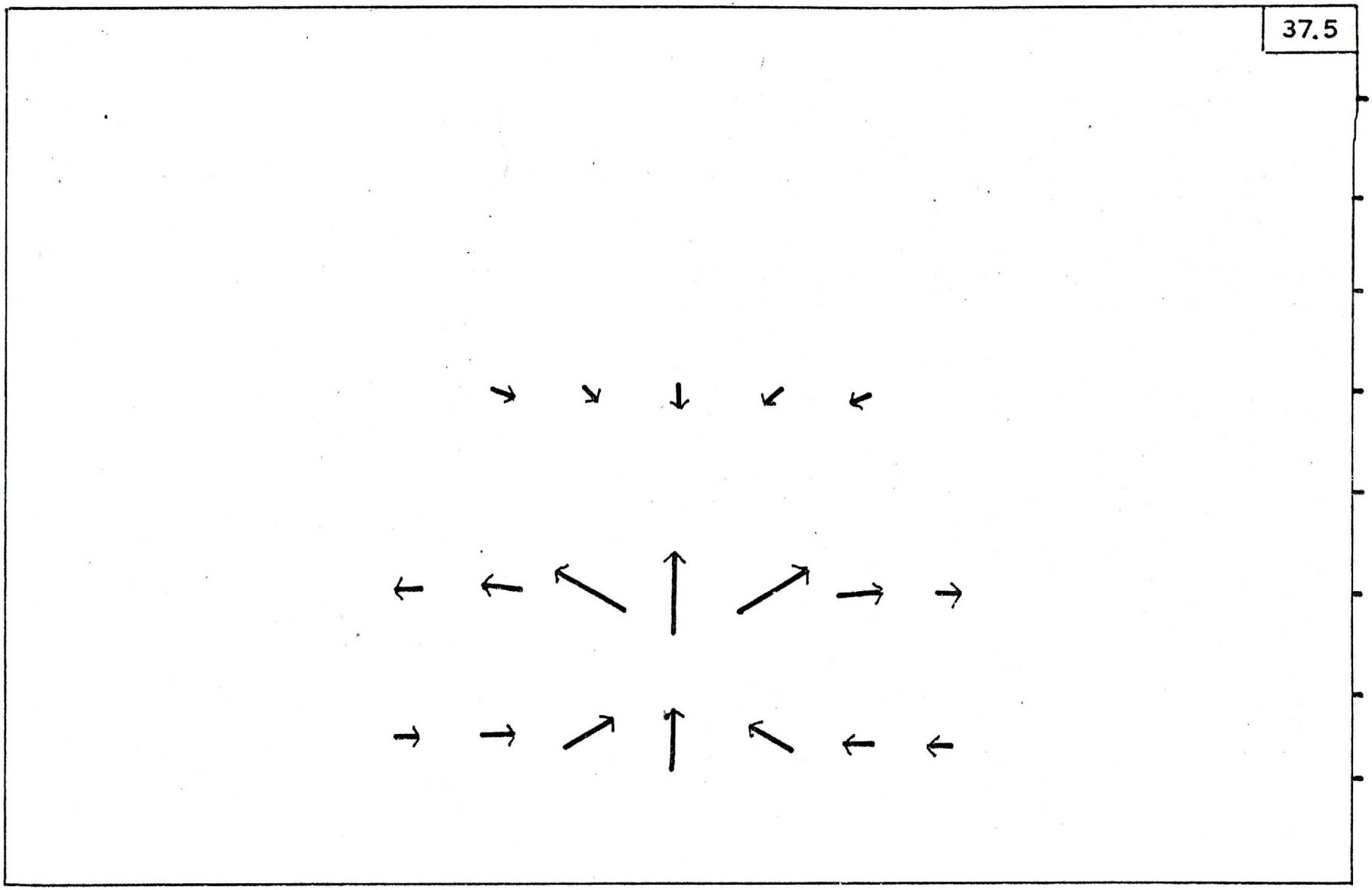
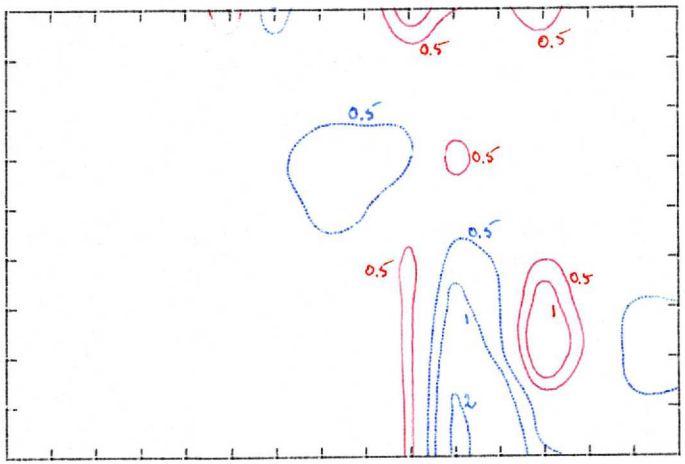
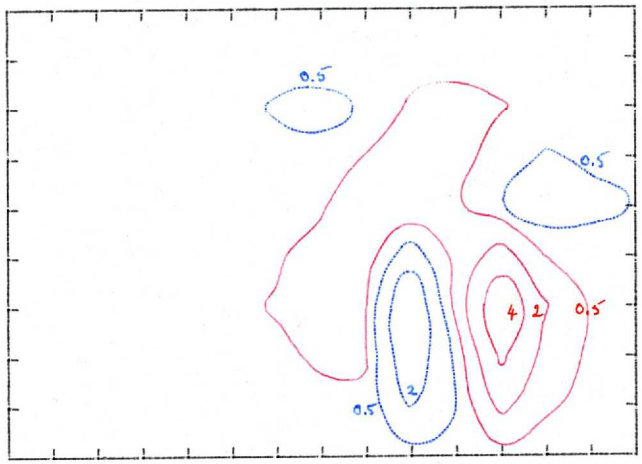


Fig. 6.47 *RUN 3* [$X=X_{12}, Y, P$] WIND VECTORS

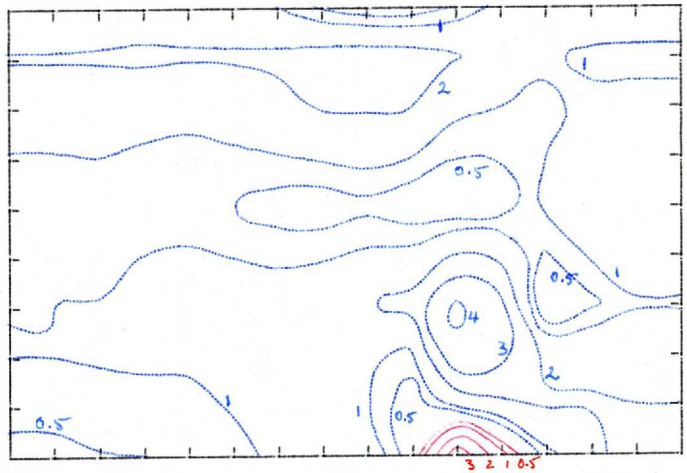
10 m/s



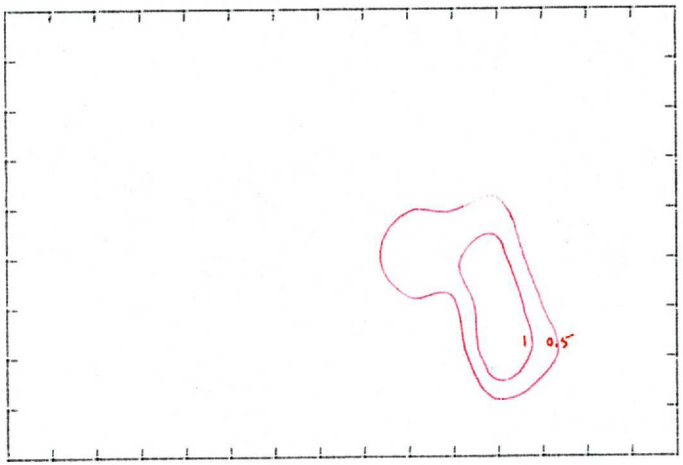
POTENTIAL TEMPERATURE DEVIATION IN DEG. C. AT 40.0 MINS.



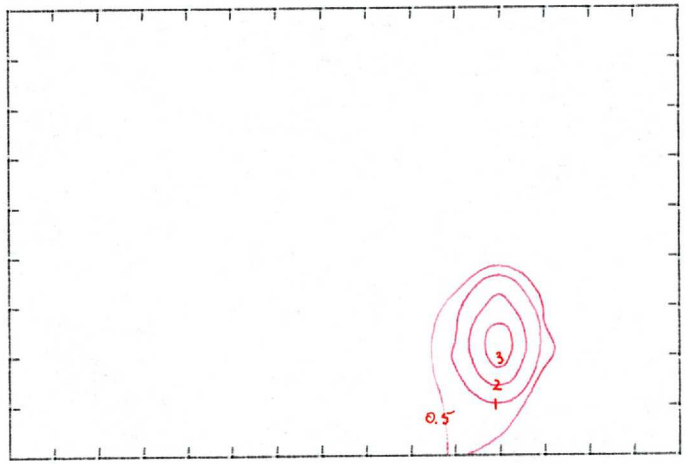
VERTICAL VELOCITY IN METRES/SEC. AT 40.0 MINS



DEVIATION HEIGHT FIELD IN METRES AT 40.0 MINS.

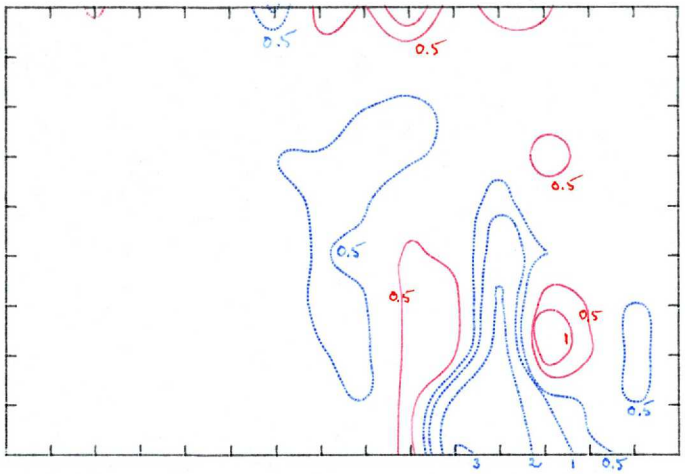


CLOUDWATER CONTENT IN GM/KG. AT 40.0 MINS

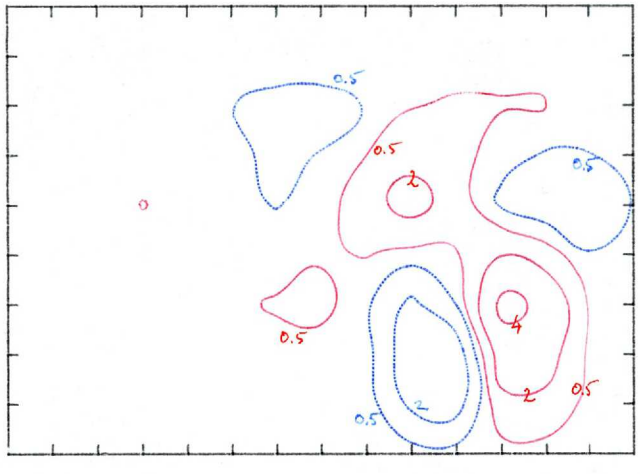


RAINWATER CONTENT IN GM/KG. AT 40.0 MINS

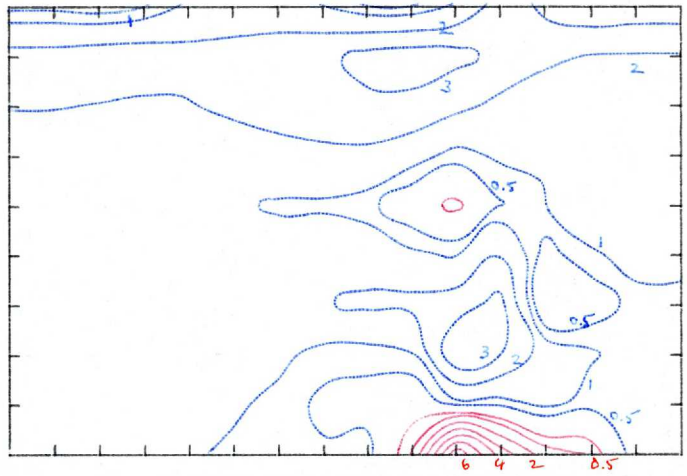
Fig. 6.48



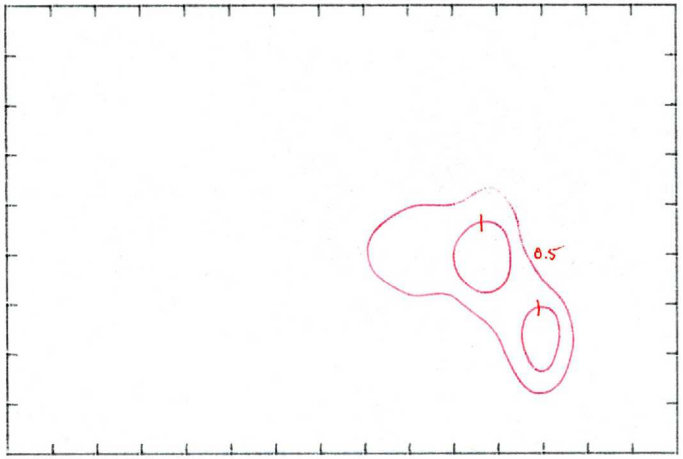
POTENTIAL TEMPERATURE DEVIATION IN DEG. C. AT 42.5 MINS.



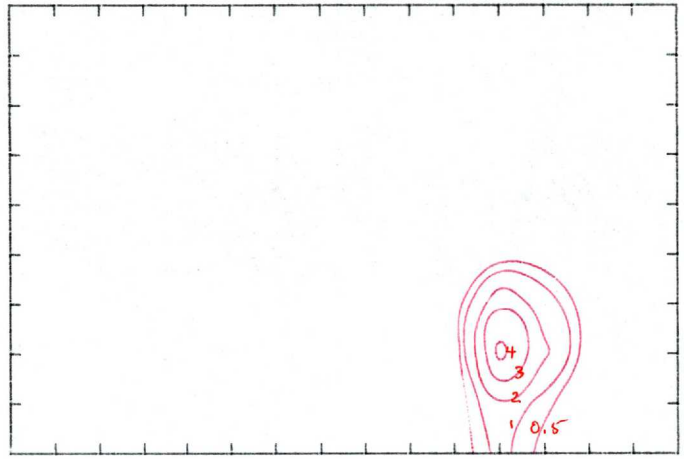
VERTICAL VELOCITY IN METRES/SEC. AT 42.5 MINS



DEVIATION HEIGHT FIELD IN METRES AT 42.5 MINS.

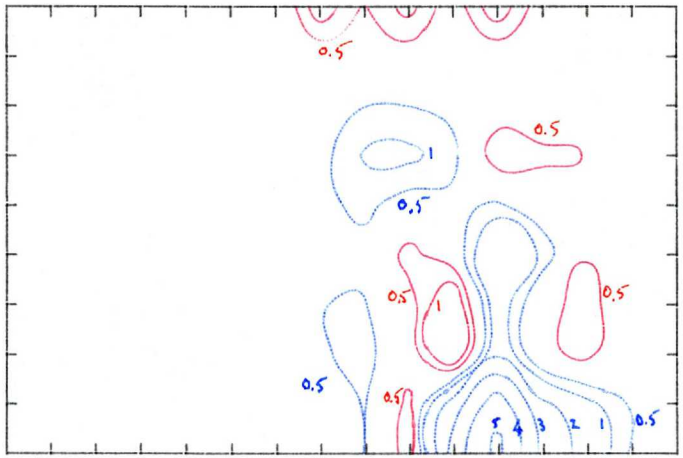


CLOUDWATER CONTENT IN GM/KG. AT 42.5 MINS

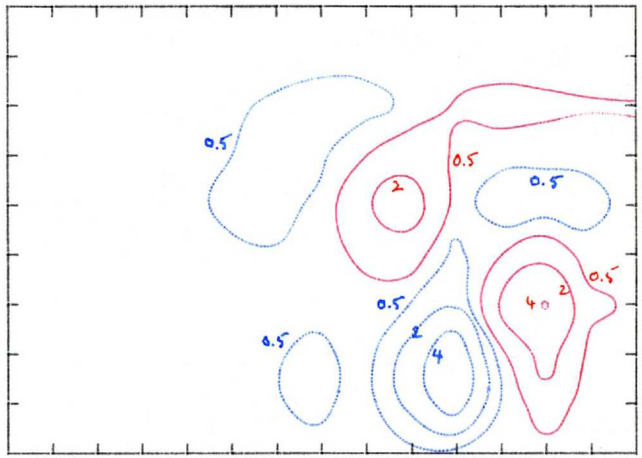


RAINWATER CONTENT IN GM/KG. AT 42.5 MINS

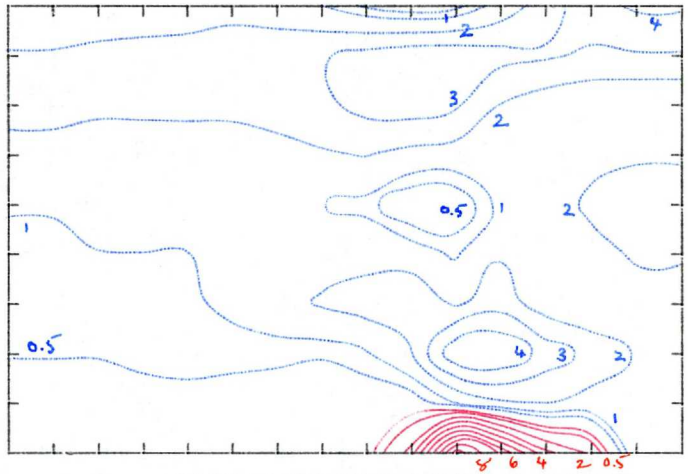
Fig. 6.49



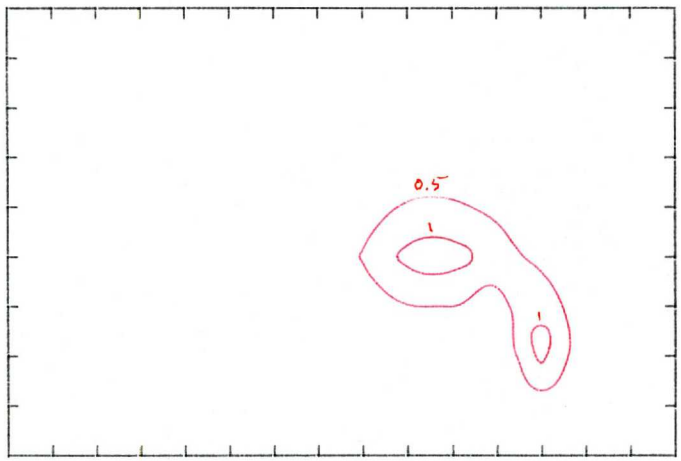
POTENTIAL TEMPERATURE DEVIATION IN DEG. C. AT 45.0 MINS.



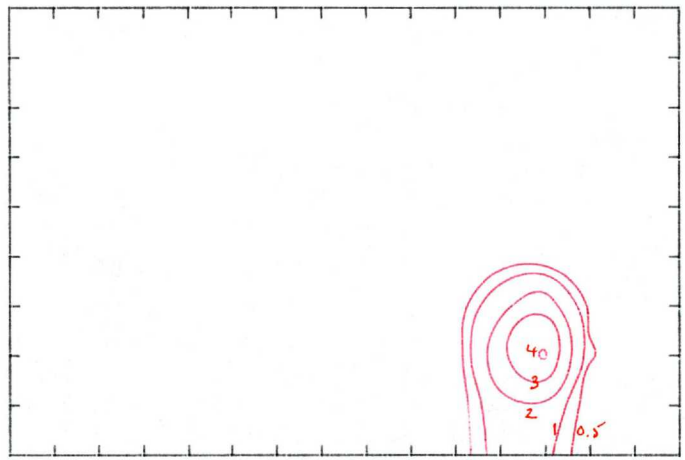
VERTICAL VELOCITY IN METRES/SEC. AT 45.0 MINS



DEVIATION HEIGHT FIELD IN METRES AT 45.0 MINS.



CLOUDWATER CONTENT IN GM/KG. AT 45.0 MINS



RAINWATER CONTENT IN GM/KG. AT 45.0 MINS

Fig. 6.50

CHAPTER VII

SUMMARY AND CONCLUDING REMARKS

In the preceding chapters, a detailed description of a numerical model of the atmosphere has been given, with examples of simulations of deep convection under differing environmental conditions. Several of the major limitations of existing models have been removed thus realising a substantially more flexible and adaptable research tool for future work.

The main advantages gained are as follows:-

i) Full three-dimensionality in space

ii) Accurate computations of the thermodynamics of the vapour-liquid transitions

iii) Open lateral boundaries

iv) The inclusion of the spatial variations of density.

The model also extends to the 100 mb level thus virtually eliminating upper boundary effects inherent in existing models.

Inevitably, these improvements and extensions have been gained at the expense of other factors although in general with no serious disadvantages introduced.

The extension to three-dimensions in space made heavy demands on computing facilities and programming, and when combined with open lateral boundaries, introduced boundary condition formulation problems. This aspect of the model proved to be a difficult problem and one which has not been satisfactorily or completely solved; further work is needed on a consistent set of dynamical boundary conditions for fluid flowing across a gridpoint model boundary.

The improved thermodynamic computations and the inclusion

of the density variations are directly the result of using pressure as the vertical coordinate, these advantages were gained by reformulating the modelling equations in pressure coordinates. This use of pressure coordinates with a free-surface boundary condition permits only a short computational timestep - a fact detrimental to the overall economy of computing time but beneficial to the computation of the thermodynamics and cloud physics.

The practicability of modelling deep convection in 3-D is demonstrated and several points noted, both as physical features and as modelling effects and deficiencies.

Despite the resolution and relative crudity of the parameterisation of liquid water, the sensitivity of the simulated convection to this parameterisation is an encouraging feature - for the substantial rcle of the liquid water phase in the cloud's lifecycle, while showing the importance of the parameterisation, emphasises the essential response of the model to physical influence. The rapid changes that occur at certain times in the cloud's history are governed primarily by the cloud physical processes.

The parameterisation deficiencies have been referred to previously but it is clear that an improved handling of the spectrum of raindrop fall-speeds is desirable. The modelling of the freezing and sublimation processes has been attempted only in one-dimensional models and the inclusion of these processes is not envisaged in the immediate future.

The sensitivity of the models to explicit smoothing (diffusion) terms was marked. Although these terms were included for computational reasons, the physical effects were significant. The simulations proved to be particularly sensitive to smoothing on the moisture field and time-dependent smoothing was introduced

to minimise this in the early growth stages. It has been customary for previous models to incorporate a constant coefficient of eddy diffusion on variables both in space and time; in view of experiences here, this must be considered a poor modelling assumption and improved ways of modelling turbulent and subgrid-scale transfer processes should be investigated. The low resolution of the model would necessitate parameterisation of a large part of the spectrum of small-scale motions. The resolution problem also extends to the cloud physics, for the use of gridpoint values representing mean quantities over such large volumes is a crude representation of cloud properties.

No energy-conservation integrals and transformations have been given here although preliminary computations have been conducted on the smaller (7 x 7 x 7) model. The demands on computer time and storage of the present model were such as to make it impracticable. A diagnostic program will be run in the future, and much of the groundwork has already been done. The present model requires approximately two hours computing time for a forty-minute simulation run but reprogramming and the use of improved solution techniques (see Chapter III, §3.6) are expected to reduce this by a factor of two at least.

DISCUSSION OF THE MODEL WITH REGARD TO FUTURE WORK

In this thesis a model has been developed with which future research into convection in the atmosphere will be pursued. In particular, examination of the basic dynamics of convection (and in its present form — deep convection) is proposed with the parameterisation problem as one important objective. A series of 'experiments' can be carried out in which the characteristics of the convection are studied in relation to different initial atmospheric conditions.

The convective systems discussed in Chapter VI, §6.8 provide examples of the types of flow studied as a dynamical problem by Moncrieff and Green (1972), and support the ideas developed there, these in turn provide a basis for further modelling investigations particularly with regard to the transfers of momentum, heat and moisture.

The modelling of storms using initial wind fields and soundings similar to those occasions for which descriptive models have been developed from detailed observation (e.g. Browning and Ludlam, 1962; Browning, 1965), is a practical possibility with a larger version of the model, although certain aspects can however be tried on the present model as suggested in Chapter VI, §6.8. It should be possible to construct 3-D flow patterns and compare with those of descriptive models.

The experiments referred to, together with the sophistication and improvement of the model indicate interesting, varied and profitable lines of research in the future.

APPENDIX 1. THE NUMERICAL COMPUTATION SCHEME

a) THE FINITE-DIFFERENCE FORMULAE

Using the notation ϕ (i.e. without a suffix) to denote $\phi(l\Delta x, m\Delta y, p_k)$ and ϕ_* to denote $\phi\{(l+\tfrac{1}{2})\Delta x, (m+\tfrac{1}{2})\Delta y, p_{k+\frac{1}{2}}\}$, the pressure levels p_k(1000, 850, 700, 500, 300, 200 and 150 mb) corresponding respectively to k = 7, 6 ... 1, the space derivatives in Eqs.(2.1.1), (2.1.2), (2.1.4), (2.1.5) are replaced to give

$$\frac{\partial u}{\partial t} = -\delta_x\overline{\psi}^y + v\overline{\zeta}^{xy} - \overline{\omega}^{xy}\delta_p u \qquad \text{——(A1)}$$

$$\frac{\partial v}{\partial t} = -\delta_y\overline{\psi}^x - u\overline{\zeta}^{xy} - \overline{\omega}^{xy}\delta_p v \qquad \text{——(A2)}$$

$$\frac{\partial \theta_*}{\partial t} = -\delta_x(\overline{u_*}^{py}\,\overline{\theta_*}^x) - \delta_y(\overline{v_*}^{px}\,\overline{\theta_*}^y) - \delta_p(\omega_*\,\overline{\theta_*}^p) + Q'_* \qquad \text{——(A3)}$$

$$\frac{\partial q_*}{\partial t} = -\delta_x(\overline{u_*}^{py}\,\overline{q_*}^x) - \delta_y(\overline{v_*}^{px}\,\overline{q_*}^y) - \delta_p(\omega_*\,\overline{q_*}^p) + Q''_* \qquad \text{——(A4)}$$

where Q'_* and Q''_* are computed as described in Chapter V

and $$\psi_* = gh_* + \frac{1}{2}\left\{(\overline{u_*}^{xy})^2 + (\overline{v_*}^{xy})^2\right\} \qquad \text{——(A5)}$$

where
$$\overline{\phi}^x = \frac{1}{2}\left[\phi\{(l+\tfrac{1}{2})\Delta x, m\Delta y, p_k\} + \phi\{(l-\tfrac{1}{2})\Delta x, m\Delta y, p_k\}\right]$$

$$\overline{\phi}^y = \frac{1}{2}\left[\phi\{l\Delta x, (m+\tfrac{1}{2})\Delta y, p_k\} + \phi\{l\Delta x, (m-\tfrac{1}{2})\Delta y, p_k\}\right]$$

$$\delta_x\phi = \frac{1}{\Delta x}\left[\phi\{(l+\tfrac{1}{2})\Delta x, m\Delta y, p_k\} - \phi\{(l-\tfrac{1}{2})\Delta x, m\Delta y, p_k\}\right]$$

$$\delta_y\phi = \frac{1}{\Delta x}\left[\phi\{l\Delta x, (m+\tfrac{1}{2})\Delta y, p_k\} - \phi\{l\Delta x, (m-\tfrac{1}{2})\Delta y, p_k\}\right]$$

and $\delta_p\phi$ in Eqs.(A1) and (A2) is a quadratic interpolant through values at three adjacent levels.

The expression $\delta_p(\phi_*)$ in Eqs.(A3) and (A4) is:

$$\left[\phi\left\{(l+\tfrac{1}{2})\Delta x,\ (m+\tfrac{1}{2})\Delta y,\ p_{k+\frac{1}{2}}\right\} - \phi\left\{(l+\tfrac{1}{2})\Delta x,\ (m+\tfrac{1}{2})\Delta y,\ p_{k-\frac{1}{2}}\right\}\right] \Big/ 2(p_{k+1} - p_k)$$

and the value of $\zeta\left\{(l+\tfrac{1}{2})\Delta x,\ (m+\tfrac{1}{2})\Delta y,\ p_k\right\}$ from which $\overline{\zeta}^{xy}$ values are formed is:

$$\delta_x\left(\overline{v}^{y}\left\{(l+\tfrac{1}{2})\Delta x,\ (m+\tfrac{1}{2})\Delta y,\ p_k\right\}\right) - \delta_y\left(\overline{u}^{x}\left\{(l+\tfrac{1}{2})\Delta x,\ (m+\tfrac{1}{2})\Delta y,\ p_k\right\}\right)$$

The finite-difference analogue for Eq.(2.1.3) integrated w.r.t. pressure is given by the trapezium rule:

$$\omega_{l+\frac{1}{2},m+\frac{1}{2},k} = -\sum_{i=1}^{k} \frac{1}{2}\left(D_{l+\frac{1}{2},m+\frac{1}{2},i-1} + D_{l+\frac{1}{2},m+\frac{1}{2},i}\right)\left(p_i - p_{i-1}\right) \qquad \text{——(A6)}$$

where $D_{l+\frac{1}{2},m+\frac{1}{2},i} = \delta_x \overline{u}^{y}_{l+\frac{1}{2},m+\frac{1}{2},i} + \delta_y \overline{v}^{x}_{l+\frac{1}{2},m+\frac{1}{2},i}$

The space derivatives in Eq.(3.16) are replaced to give

$$\frac{\partial h}{\partial t}_{l+\frac{1}{2},m+\frac{1}{2},7} = -\overline{\overline{u}^{y}_{l+\frac{1}{2},m+\frac{1}{2},7}\ \delta_x h_{l,m+\frac{1}{2},7}}^{x} - \overline{\overline{v}^{x}_{l+\frac{1}{2},m+\frac{1}{2},7}\ \delta_y h_{l+\frac{1}{2},m,7}}^{y} + \frac{1}{g\rho_7}\omega_{l+\frac{1}{2},m+\frac{1}{2},7}$$

——(A7)

Reference to the finite-difference form of Eq.(3.15) has already been made. The advection equations for rainwater and cloudwater are treated in an exactly analogous form to Eq.(A4).

b) TIME-INTEGRATION AND COMPUTATIONAL STABILITY

The replacement of the time derivatives and the times to which the above finite-difference replacements refer are essentially as follows.

Starting at time t, the $u(t + \Delta t)$ and $v(t + \Delta t)$ fields are first computed from Eqs.(A1) and (A2) to yield $\omega(t + \Delta t)$

from Eq.(A6); these are subsequently used in Eqs.(A3), (A4) and (A7) to compute fields of θ, q and h_7 at time $(t+\Delta t)$.

The analysis of the time-integration for both a linearised system and the non-linear equation system has been carried out by Pearce (1972). The results of his analysis relevant here can be summarised as follows.

For the linearised system containing only the external gravity wave and quasi-horizontal advective waves, the use of consecutive forward timesteps in the form $\phi(t+\Delta t) = \phi(t) + \left[\frac{\partial \phi}{\partial t}\right]_t \Delta t$ (with $[\partial\phi/\partial t]$ computed from RHS of prognostic equation) - requires the stability criterion $\Delta t < \Delta x/c$ (c = gravity wave speed) to be satisfied. This scheme is unstable however when the non-linear terms are included.

The scheme adopted in the model consists basically of alternate timesteps, the first with the calculation of temporary values and the second without. Temporary values of variables are formed using the relationship:

$$\phi(\text{temp}) = \phi(t) + \left[\frac{\partial \phi}{\partial t}\right]_t \Delta t$$

and these used to compute $\phi(t+\Delta t)$ using

$$\phi(t+\Delta t) = \phi(t) + \left[\frac{\partial \phi}{\partial t}\right]_{\text{temp}} \Delta t$$

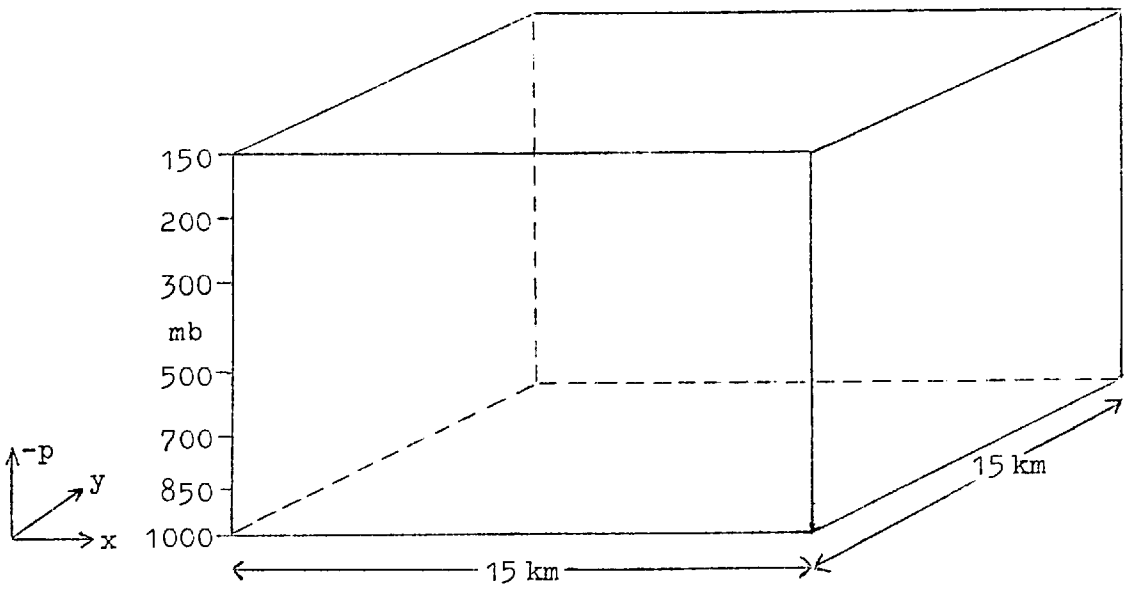
where $[\partial\phi/\partial t]_{\text{temp}}$ is computed using temporary values in the RHS of the prognostic equation.

The computation of temporary values every timestep gives a scheme stable for the 'meteorological wave' but only of 1^{st} order accuracy; the use of alternate timesteps introduces slight instability but gives a 2^{nd} order accurate scheme. This slight

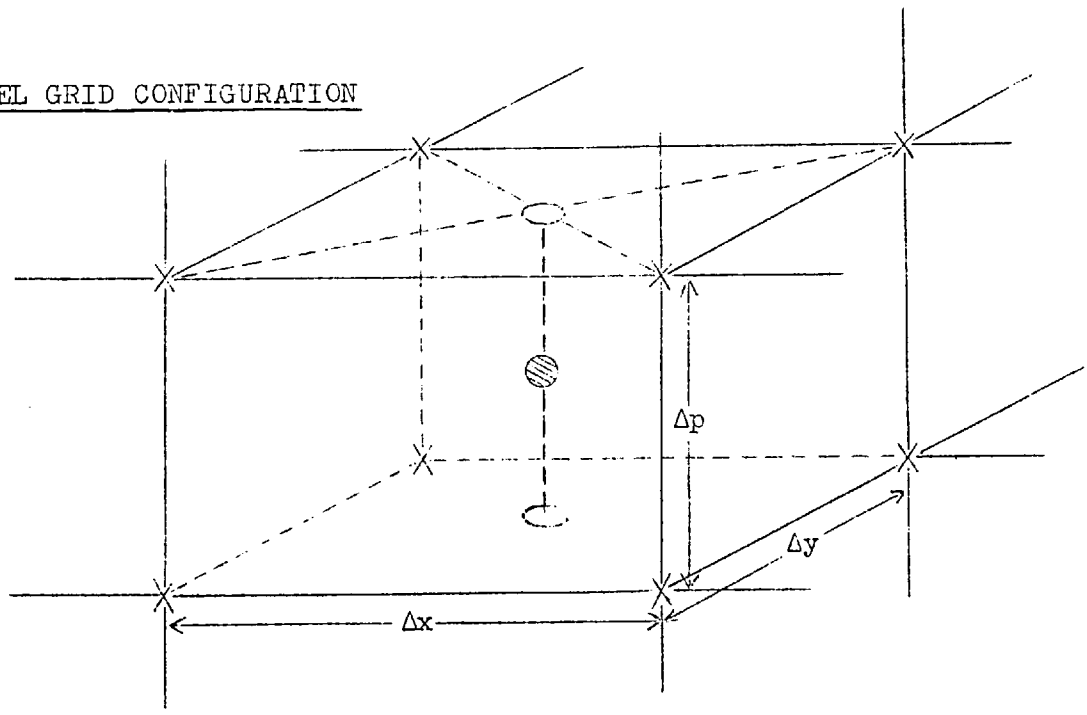
instability is so small that the scheme is effectively stable for the meteorological wave for use in the model here; the use of occasional repeated calculation of temporary values may be used to ensure stability in models incorporating very large numbers of timesteps.

This scheme is not theoretically stable as far as the external gravity waves are concerned but the use of a small amount of smoothing on the height field through Eq.(A7) is sufficient to ensure stability.

MODEL DIMENSIONS



MODEL GRID CONFIGURATION



X	hold	u, v	Δx = Δy = 1000 m
⬭	"	h, ω	Δp — as defined by pressure levels
◍	"	θ, q, l_c, l_r	

APPENDIX 2 - PROGRAM LISTING

A PRIMITIVE EQUATION MODEL FOR THE NUMERICAL SIMULATION
OF A CUMULONIMBUS CLOUD IN THREE SPACE DIMENSIONS.

THE MODEL INCORPORATES THE FOLLOWING FEATURES

1 THE VERTICAL CO-ORDINATE IS PRESSURE
2 THE AMBIENT WIND, TEMPERATURE AND MOISTURE FIELDS ARE SPECIFIED AS INITIAL DATA
3 A CLOUD MICROPHYSICAL PARAMETERISATION IS USED TO SIMULATE RAINFALL
4 THE FLOW NORMAL TO THE BOUNDARIES IS TIME-DEPENDENT AT THE BOUNDARIES

NUMERICAL METHODS USED INCLUDE

1 TIME-INTEGRATION BY A MODIFIED EULER-BACKWARD SCHEME
2 CENTRED SPACE-DIFFERENCES
3 S.O.R. SOLUTIONS TO THE BALANCE EQUATION FOR THE HEIGHT FIELD

COMPUTER REQUIREMENTS

THE PROGRAM AS WRITTEN REQUIRES 60K CORE STORAGE PLUS A MUCH SMALLER DISK AREA. 5 MINUTES SIMULATED TIME REQUIRES 14 MINUTES OF THE CDC 6600 COMPUTER'S TIME.
THE SIMULATION RUNS ARE BUILT UP AS A SEQUENCE OF FILES ON TAPE, EACH FILE REPRESENTING DATA FIELDS EVERY 2.5 MINUTES. THIS TAPE IS THEN AVAILABLE AS INPUT FOR CONTOURING PROGRAMS ETC.
A COMPLETE SIMULATION RUN CONSISTS OF INDIVIDUAL RUNS OF THE PROGRAM LISTED HERE, PROVIDING TWO FILES AT A TIME, WITH A SLIGHTLY MODIFIED PROGRAM TO GENERATE THE FIRST FILE.

CONSTANTS

NAME	MEANING
DELT	TIMESTEP - 3 SECONDS.
DELP	VERTICAL GRADIENT IN UNITS OF 100MBS.
DM	HORIZONTAL GRIDLENGTH - 1000 METRES

VARIABLE NAMES

UST, VST	AMBIENT HORIZONTAL VELOCITY COMPONENTS (IF NON-ZERO) IN METRES/SEC.
U15, V15	VARIABLE PART OF HORIZONTAL VELOCITY COMPONENTS IN METRES/SEC.
HDIVP	HORIZONTAL DIVERGENCE
WP14	OMEGA - DP/DT IN UNITS OF 100MB/SEC.
ZETA14	VERTICAL COMPONENTS OF VORTICITY
HITE16	HEIGHT DEVIATION FROM A PRESSURE SURFACE IN METRES.
PSI15	PSI FUNCTION IN METRES SQUARED/SEC.SQUARED
Q16	SPECIFIC HUMIDITY IN GMS/KG.
QL	CLOUDWATER CONCENTRATION IN GMS/KG.
RAIN	RAINWATER CONCENTRATION IN GMS/KG.
THS, QST	UNPERTURBED POTENTIAL TEMPERATURE AND HUMIDITY FIELDS IN DEG.C AND GMS/KG.
OUTFLO	INFLOW/OUTFLOW INDICATOR

VARIABLES ENDING WITH -INCR REPRESENT TIMESTEP INCREMENTS OF BASIC FIELDS.

TAPE 3 IS THE MAIN RECORD TAPE.
TAPES 1 AND 7 ARE TEMPORARY SCRATCH AREAS ON DISC.

```
      PROGRAM NIMSHE(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,                  MAINP   1
     1               TAPE1,TAPE2,TAPE3,TAPE7)                                MAINP   2
      DIMENSION QL(16,16,7),DQL(16,16,7),HINCR(16,16,7),HITE16(16,16,7),MAINP   3
     1          TH16(16,16,7),PSI(16,16,7),PSI16(16,16,7),THINCR(16,16,7),MAINP   4
     2          RAIN(16,16,7),Q16(16,16,7),QINCR(16,16,7),THWINC(16,16,7),MAINP   5
     3            UR(15,15,7),U15(15,15,7),UINCR(15,15,7),     WW(15,15,7),MAINP   6
     4            VR(15,15,7),V15(15,15,7),VINCR(15,15,7),  WP14(14,14,7),MAINP   7
     5                  F(15),OUTFLO(56,7),HDIVP(14,14,7),ZETA14(14,14,7),MAINP   8
     6                        DELP(7),THICK(7),ACOUT(7),BCOUT(7),QCOUT(7)  MAINP   9
      LOGICAL    OUTFLO,MARK                                                MAINP  10
      EXTERNAL   OUT                                                        MAINP  11
      COMMON     CCX,DM,KOUNT2,MARK,KOUNT1,DELT,IND                         MAINP  12
      COMMON     /BLK1/ U15,V15,WP14                                        MAINP  13
      COMMON     /BLK2/ Q16,TH16                                            MAINP  14
      COMMON     /BLK3/ ZETA14                                              MAINP  15
      COMMON     /BLK4/ UINCR,VINCR                                         MAINP  16
      COMMON     /BLK6/ HITE16                                              MAINP  17
      COMMON     /BLK13/ OUTFLO                                             MAINP  18
      COMMON     /BLK20/ BCOUT,QCOUT,ACOUT,IOUT,JOUT                        MAINP  19
      COMMON     /BLK21/ HDIVP,QL                                           MAINP  20
      DATA       THICK/.5,.5,1.,2.,2.,1.5,1.5/                              MAINP  21
      DATA       DELP /0.5,0.5,1.,2.,2.,1.5,1.5/                            MAINP  22
      EQUIVALENCE (UR(1),U15(1)),(VR(1),V15(1)),(PSI(1) ,PSI 16(1))         MAINP  23
          CALL ONTRAP(OUT)                                                  MAINP  24

      DO 3003 II=1,25                                                       MAINP  25
    1 READ(3) MARK,KOUNT2,MP,LASTF,ISTOP                                    MAINP  26
      IF(ENDFILE 3) 1,6                                                     MAINP  27
    6 IF(LASTF.EQ.1) BACKSPACE 3                                            MAINP  28
      IF(LASTF.EQ.1) GO TO 3005                                             MAINP  29
      DO 3003 K=1,7                                                         MAINP  30
      READ   (3) ((U15    (I,J,K),I=1,15),J=1,15)                         MAINP  31
      READ   (3) ((V15    (I,J,K),I=1,15),J=1,15)                         MAINP  32
      READ   (3) ((WP14   (I,J,K),I=1,14),J=1,14)                         MAINP  33
      READ   (3) ((ZETA14 (I,J,K),I=1,14),J=1,14)                         MAINP  34
      READ   (3) ((PSI16  (I,J,K),I=1,16),J=1,16)                         MAINP  35
      READ   (3) ((HITE16 (I,J,K),I=1,16),J=1,16)                         MAINP  36
```

```
      READ  (3) ((Q16    (I,J,K),I=1,16),J=1,16)                                  MAINP 37
      READ  (3) ((QL     (I,J,K),I=1,16),J=1,16)                                  MAINP 38
      READ  (3) ((TH16   (I,J,K),I=1,16),J=1,16)                                  MAINP 39
      READ  (3) ((RAIN   (I,J,K),I=1,16),J=1,16)                                  MAINP 40
      READ  (3) (OUTFLO(I,K),I=1,65)                                              MAINP 41
 3003 CONTINUE                                                                    MAINP 42
 3005 LASTF=0                                                                     MAINP 43
          CALL OUT15 (MP14,14,14,7,0)                                             MAINP 44
      DELT=3.                                                                     MAINP 45
      KOUNT1=0                                                                    MAINP 46
      DM=1000.                                                                    MAINP 47
      D=1.                        $CCX=0.                                         MAINP 48
      ISTART=KOUNT2                                                               MAINP 49
      NUMST=ISTART+2*MP-1                                                         MAINP 50
      DO 5 I=1,15                                                                 MAINP 51
    5 F(I)=0.                                                                     MAINP 52

      DO 10 IA=ISTART,NUMST                                                       MAINP 53
      MARK=.NOT.MARK                                                              MAINP 54
      KOUNT1=KOUNT1+1    $    KOUNT2=KOUNT2+1                                     MAINP 55
          CALL INCRUV(DELT,Q15,V15,MP14,ZETA14,UINCR,VINCR,PSI16,HITE16,          MAINP 56
     1                F,QINCR,THINCR)                                            MAINP 57

      DO 100 K=1,7          $DO 100 I=2,14          $DO 100 J=2,14                MAINP 58
      UR(I,J,K)=UR(I,J,K)+UINCR(I,J,K)                                            MAINP 59
  100 VR(I,J,K)=VR(I,J,K)+VINCR(I,J,K)                                            MAINP 60
      IF(KOUNT2.GT.150) GO TO 34                                                  MAINP 61

      DO 33 J=1,15                                                                MAINP 62
      UR(1,J,7)=UR(2,J,7)=10.                                                     MAINP 63
   33 UR(1,J,6)=UR(2,J,6)=6.5                                                     MAINP 64
   34     CALL NEW (PSI16,ZETA14,J15,V15,HITE16,MP14,F,DELT,HINCR,Q16            MAINP 65
     1              ,TH16,RAIN,QINCR,THINCR,THWINC,0,MP)                         MAINP 66

      DO 200 I=2,15         $DO 200 J=2,15          $DO 200 K=1,7                 MAINP 67
      HINCR(I,J,K)=Q16(I,J,K)/(Q16(I,J,K)+QL(I,J,K))                              MAINP 68
  200 Q16(I,J,K)=Q16(I,J,K)+QL(I,J,K)                                             MAINP 69
```

```
      REWIND 7                                                          MAINP 70
      WRITE(7) (((RAIN(I,J,K),I=1,16),J=1,16),K=1,7)                    MAINP 71
      REWIND 7                                                          MAINP 72
         CALL INCTHQ(U15,V15,Q16,TH16,WP14,RAIN,THINCR,QINCR,THWINC,    MAINP 73
     1               AN,DELT,FNN)                                       MAINP 74
      READ(7) (((RAIN(I,J,K),I=1,16),J=1,16),K=1,7)                     MAINP 75

      DO 201 I=2,15          $DO 201 J=2,15          $DO 201 K=1,7      MAINP 76
      Q16(I,J,K)=Q16(I,J,K)-QL(I,J,K)                                   MAINP 77
      QINCR(I,J,K)=QINCR(I,J,K)*HINCR(I,J,K)                            MAINP 78
      DQL(I,J,K)=QINCR(I,J,K)*(1.-HINCR(I,J,K))/HINCR(I,J,K)            MAINP 79
  201 QL(I,J,K)=QL(I,J,K)+DQL(I,J,K)                                    MAINP 80
   11    CALL BDY(U15,V15,OUTFLO)                                       MAINP 81
                                                                        MAINP 82

      DO 102 I=2,15          $DO 102 J=2,15          $DO 102 K=1,7      MAINP 83
      TH16(I,J,K)=TH16(I,J,K)+THINCR(I,J,K)                             MAINP 84
  102 Q16(I,J,K)=Q16(I,J,K)+QINCR(I,J,K)                                MAINP 85
      REWIND 1                                                          MAINP 86
      WRITE(1)(((VINCR(I,J,K),I=1,15),J=1,15),K=1,7)                    MAINP 87
         CALL ADVRAIN(UR,VR,WP14,RAIN,DELT,OUTFLO,VINCR,HINCR)          MAINP 88

      DO 2 I=2,15            $DO 2 J=2,15            $DO 2 K=1,7        MAINP 89
    2 RAIN(I,J,K)=RAIN(I,J,K)-HINCR(I,J,K)                              MAINP 90
      REWIND 1                                                          MAINP 91
      READ(1)(((VINCR(I,J,K),I=1,15),J=1,15),K=1,7)                     MAINP 92
         CALL SLVTQF(Q16,TH16,QL,THINCR,QINCR,DQL)                      MAINP 93
         CALL PHYSIC(TH16,DELT,DQL,Q16,QL,HINCR)                        MAINP 94

      DO 4 I=2,15            $DO 4 J=2,15            $DO 4 K=1,7        MAINP 95
      RAIN(I,J,K)=RAIN(I,J,K)+HINCR(I,J,K)                              MAINP 96
    4 IF(ABS(RAIN(I,J,K)).LT.0.01) RAIN(I,J,K)=0.                       MAINP 97
      IF(KOUNT1.EQ.MP) WRITE(6,72)                                      MAINP 98
      IF(KOUNT1.EQ.MP) CALL OUT15(RAIN,16,16,7,0)                       MAINP 99
```

```
      CALL NEW (PSI16,ZETA14,U15,V15,HITE16,VP14,F,DELT,HINCR,Q16      MAINP100
     1          ,TH16,RAIN,CINCR,THINCR,THINC,1,MP)                    MAINP101
      IF(KOUNT1.EQ.MP) CALL OUT                                        MAINP102
      IF(KOUNT1.EQ.MP) CALL ZERO(MA,U15,V15,HINCR,HITE16,VP14,DM,DELT) MAINP103
      IF(KOUNT1.EQ.MP) CALL OUT15(VR,15,15,7,0)                        MAINP104
      IF(KOUNT1.NE.MP) GO TO 543                                       MAINP105
      WRITE (3) MARK,KOUNT2,MP,LASTF,ISTOP                             MAINP106
      DO 3004 K=1,7                                                    MAINP107
      WRITE (3) ((U15    (I,J,K),I=1,15),J=1,15)                       MAINP108
      WRITE (3) ((V15    (I,J,K),I=1,15),J=1,15)                       MAINP109
      WRITE (3) ((VP14   (I,J,K),I=1,14),J=1,14)                       MAINP110
      WRITE (3) ((ZETA14 (I,J,K),I=1,14),J=1,14)                       MAINP111
      WRITE (3) ((PSI16  (I,J,K),I=1,16),J=1,16)                       MAINP112
      WRITE (3) ((HITE16 (I,J,K),I=1,16),J=1,16)                       MAINP113
      WRITE (3) ((Q16    (I,J,K),I=1,16),J=1,16)                       MAINP114
      WRITE (3) ((CL     (I,J,K),I=1,16),J=1,16)                       MAINP115
      WRITE (3) ((TH16   (I,J,K),I=1,16),J=1,16)                       MAINP116
      WRITE (3) ((RAIN   (I,J,K),I=1,16),J=1,16)                       MAINP117
      WRITE (3) (OUTFLO(I,K),I=1,56)                                   MAINP118
 3004 CONTINUE                                                         MAINP119
      ENDFILE 3                                                        MAINP120
  543 IF (KOUNT1.EQ.MP) KOUNT1=0                                       MAINP121
   10 CONTINUE                                                         MAINP122
      CALL OUT15(VR,15,15,7,0)                                         MAINP123
      LASTF=1                                                          MAINP124
      WRITE(3) MARK,KOUNT2,MP,LASTF,ISTOP                              MAINP125
      ENDFILE 3                                                        MAINP126
   72 FORMAT(1HO,*RAINWATER FIELD IN GM/KG*,/)                         MAINP127
      STOP                                                             MAINP128
      END                                                              MAINP129
```

```
      SUBROUTINE INCRUV(DELT,UR,VR,WP,ZETA,UINCR,VINCR,PSI,HEIGHT,      NCRUV   1
     1                            F,EN2,EN3)                            NCRUV   2
C* * *    COMPUTES HORIZONTAL VELOCITY INCREMENTS                  * * *NCRUV   3
      DIMENSION HEIGHT(16,16,7),UINCR(15,15,7),ZETA(14,14,7),DELP(7),   NCRUV   4
     1               PSI(16,16,7),VINCR(15,15,7),  WP(14,14,7),  Y1(7),  NCRUV   5
     2               EN2(16,16,7),   UR(15,15,7),  Z2(14,14),    Y2(7),  NCRUV   6
     3               EN3(16,16,7),   VR(15,15,7),   F(15),       X1(7),  NCRUV   7
     4              PSI2(16,16),     U2(15,15),    V2(15,15),    X2(7)   NCRUV   8
      LOGICAL    MARK                                                   NCRUV   9
      COMMON     CCX,DM,KOUNT2,MARK                                     NCRUV  10
      DATA       DELP/0.5,0.5,1.,2.,2.,1.5,1.5/                         NCRUV  11
      E1=1.                    $E2=1.                  $CAPA=0.0005      NCRUV  12

      DO 505 I=1,15            $DO 505 J=1,15          $DO 505 K=1,7     NCRUV  13
      UINCR(I,J,K)=0.                                                  NCRUV  14
  505 VINCR(I,J,K)=0.                                                  NCRUV  15

      DO 520 I=2,14            $DO 520 J=2,14          $DO 525 K=1,7     NCRUV  16
      X1(K)=UR(I,J,K)                                                  NCRUV  17
  525 X2(K)=VR(I,J,K)                                                  NCRUV  18
          CALL DDP (X1,Y1)                                             NCRUV  19
          CALL DDP (X2,Y2)                                             NCRUV  20
      DO 520 K=1,7                                                     NCRUV  21
      UINCR(I,J,K)=Y1(K)                                               NCRUV  22
  520 VINCR(I,J,K)=Y2(K)                                               NCRUV  23
```

```
      DO 530 I=2,14          $DO 530 J=2,14          $DO 530 K=1,7                NCRUV 24
      AVZETA=AV4(ZETA,I-1,J-1,K,14,14,7)                                          NCRUV 25
      AVGP  =AV4(GP  ,I-1,J-1,K,14,14,7)                                          NCRUV 26
      UINCR(I,J,K)=((PSI(I,J,K)+PSI(I,J+1,K)-PSI(I+1,J,K)-PSI(I+1,J+1,K)NCRUV 27
     1           )/(2.*DY)+F2*VR(I,J,K)*AVZETA-AVGP*UINCR(I,J,K)*E1)*DELT NCRUV 28
      VINCR(I,J,K)=((PSI(I,J+1,K)+PSI(I+1,J+1,K)-PSI(I,J,K)-PSI(I+1,J,K)NCRUV 29
     1           )/(2.*DX)-F2*UR(I,J,K)*AVZETA-AVGP*VINCR(I,J,K)*E1)*DELT NCRUV 30
                                                                               NCRUV 31
  530 CONTINUE                                                                 NCRUV 32
      IF (.NOT.MARK) GO TO 561

      DO 521 I=2,14          $DO 521 J=2,14          $DO 526 K=1,7                NCRUV 33
      X1(K)=UR(I,J,K)+UINCR(I,J,K)                                             NCRUV 34
  526 X2(K)=VR(I,J,K)+VINCR(I,J,K)                                             NCRUV 35
          CALL DDP(X1,Y1)                                                      NCRUV 36
          CALL DDP(X2,Y2)                                                      NCRUV 37
      DO 521 K=1,7                                                             NCRUV 38
      EN2(I,J,K)=Y1(K)                                                         NCRUV 39
  521 EN3(I,J,K)=Y2(K)                                                         NCRUV 40

      DO 560 K=1,7           $DO 550 I=1,15          $DO 550 J=1,15               NCRUV 41
      U2(I,J)=UR(I,J,K)+UINCR(I,J,K)                                           NCRUV 42
  550 V2(I,J)=VR(I,J,K)+VINCR(I,J,K)                                           NCRUV 43

      DO 555 I=1,14          $DO 555 J=1,14                                    NCRUV 44
      IP1=I+1                $JP1=J+1                                          NCRUV 45
      Z2(I,J)=(V2(IP1,J)-V2(I,J)+V2(IP1,JP1)-V2(I,JP1)+U2(I,JP1)-U2(I,J)NCRUV 46
     1        +U2(IP1,JP1)-U2(IP1,J))/(2.*DX)+(F(J)+F(J+1))*0.5             NCRUV 47
  555 PSI2(IP1,JP1)=0.01*HEIGHT(IP1,JP1,K)+((U2(I,J)+U2(I,JP1)+U2(IP1,J)NCRUV 48
     1        +U2(IP1,JP1))**2+(V2(I,J)+V2(I,JP1)+V2(IP1,J)                 NCRUV 49
     2        +V2(IP1,JP1))**2)/32.                                         NCRUV 50
```

```
      DO 560 I=2,14          $DO 560 J=2,14                                    NCRUV 51
      AVZETA=(Z2(I-1,J-1)+Z2(I-1,J)+Z2(I,J-1)+Z2(I,J))*0.25                    NCRUV 52
      UINCR(I,J,K)=((PSI2(I,J)+PSI2(I,J+1)-PSI2(I+1,J)-PSI2(I+1,J+1))          NCRUV 53
     1           /(2.*DM)+E2*V2(I,J)*AVZETA+CCX*(U2(I+1,J)-U2(I-1,J))         NCRUV 54
     2           /(2.*DM))*DELT                                               NCRUV 55
      VINCR(I,J,K)=((PSI2(I,J+1)+PSI2(I+1,J+1)-PSI2(I,J)-PSI2(I+1,J))          NCRUV 56
     1           /(2.*DM)-E2*U2(I,J)*AVZETA+CCX*(V2(I+1,J)-V2(I-1,J))         NCRUV 57
     2           /(2.*DM))*DELT                                               NCRUV 58
  560 CONTINUE                                                                 NCRUV 59

      DO  11 I=2,14          $DO  11 J=2,14          $DO  11 K=1,7             NCRUV 60
      UINCR(I,J,K)=UINCR(I,J,K)-AV4(WP,I-1,J-1,K,14,14,7)*EN2(I,J,K)           NCRUV 61
     1           *DELT                                                        NCRUV 62
   11 VINCR(I,J,K)=VINCR(I,J,K)-AV4(WP,I-1,J-1,K,14,14,7)*EN2(I,J,K)           NCRUV 63
     1           *DELT                                                        NCRUV 64
  561 CONTINUE                                                                 NCRUV 65

      DO 119 I=2,14          $DO 119 J=2,14          $DO 119 K=1,7             NCRUV 66
      UINCR(I,J,K)=UINCR(I,J,K)+CAPA*DELT*ALAPL(UR,I,J,K,15,15,7)              NCRUV 67
  119 VINCR(I,J,K)=VINCR(I,J,K)+CAPA*DELT*ALAPL(VR,I,J,K,15,15,7)              NCRUV 68
      RETURN                                                                   NCRUV 69
         END                                                                   NCRUV 70
```

```
      SUBROUTINE INCTHQ(UR,VR,Q1,TH8,WP,THS,THINCR,QINCR,THWINC,          NCTHQ   1
     1                                AN,DELT,FNN)                        NCTHQ   2
C* * *     COMPUTES TEMPERATURE AND SPECIFIC HUMIDITY INCREMENTS     * * *NCTHQ   3
      DIMENSION THINCR(16,16,7),THS(16,16,7),UR(15,15,7),DELP(7),         NCTHQ   4
     1          THWINC(16,16,7),TH8(16,16,7),VR(15,15,7),THS(7),         NCTHQ   5
     2           QINCR(16,16,7), Q1(16,16,7),WP(14,14,7)                 NCTHQ   6
      LOGICAL    MARK,MARK1                                              NCTHQ   7
      COMMON     CCX,DM,KOUNT2,MARK                                      NCTHQ   8
      DATA       DELP/0.5 ,0.5 , 1. , 2. , 2. ,1.5 ,1.5 /                NCTHQ   9
      DATA       THS /424.,385.,353.,332.,318.,308.,304./                NCTHQ  10

      MARK1=.TRUE.              $SK1=0.0001          $SK0=0.00005         NCTHQ  11
      E1=1.                     $E3=1.               $SUM=0.              NCTHQ  12

  200 DO 2090 I=2,13            $DO 2090 J=2,13      $DO 2090 KK=1,7      NCTHQ  13
      IP1=I+1                   $JP1=J+1             $IP2=I+2    $JP2=J+2 NCTHQ  14
      K=8-KK                    $KP1=K+1             $KM1=K-1             NCTHQ  15
      IF (K.EQ.1) GO TO 202                                               NCTHQ  16
      W1=AVPY4(UR,IP1,J,KM1,15,15,7)-CCX                                  NCTHQ  17
      W5=AVPY4(UR,I,J,KM1,15,15,7)-CCX                                    NCTHQ  18
      W2=AVPX4(VR,I,J,KM1,15,15,7)                                        NCTHQ  19
      W6=AVPX4(VR,I,JP1,KM1,15,15,7)                                      NCTHQ  20
      GO TO 203                                                           NCTHQ  21
  202 W1=(UR(IP1,J,1)+UR(IP1,JP1,1))*0.25 -CCX                            NCTHQ  22
      W5=(UR(I,J,1)+UR(I,JP1,1))*0.25-CCX                                 NCTHQ  23
      W2=(VR(I,J,1)+VR(IP1,J,1))*0.25                                     NCTHQ  24
      W6=(VR(I,JP1,1)+VR(IP1,JP1,1))*0.25                                 NCTHQ  25
  203 CONTINUE                                                            NCTHQ  26
      W4=TH8(IP2,JP1,K)+TH8(I,JP1,K)+TH8(IP1,JP2,K)+TH8(IP1,J,K)          NCTHQ  27
     1          -4.*TH8(IP1,JP1,K)                                        NCTHQ  28
      ADVTH=((W1+W5)*(TH8(IP2,JP1,K)-TH8(I,JP1,K))+(W2+W6)*(TH8(IP1,J,    NCTHQ  29
     1          K)-TH8(IP1,JP2,K)))*0.25/DM                               NCTHQ  30
```

```
                        THINCR(IP1,JP1,K)=(-ADVTH+BK1*W4*ABS(W4))*   NCTHQ 31
     1                        DELT                                   NCTHQ 32
      IF(K.EQ.7)              THINCR(IP1,JP1,7)=THINCR(IP1,JP1,7)-(WP(I,J,NCTHQ 33
     1                        7)+WP(I,J,6))*DELT*(303.5-TH8(IP1,JP1,NCTHQ 34
     2                        7)-THS(7))/1.5                         NCTHQ 35
      IF(K.NE.1.AND.K.NE.7)  THINCR(IP1,JP1,K)=THINCR(IP1,JP1,K)-WP(I,J,KNCTHQ 36
     1                        )*DELT*(TH8(IP1,JP1,KP1)+THS(KP1)-TH8(NCTHQ 37
     2                        IP1,JP1,K)-THS(K))/(DELP(K)+DELP(KP1))NCTHQ 38
     3                        -WP(I,J,KM1)*DELT*(TH8(IP1,JP1,K)+THS(NCTHQ 39
     4                        K)-TH8(IP1,JP1,KM1)-THS(KM1))/(DELP(K)NCTHQ 40
     5                        +DELP(KM1))                            NCTHQ 41
      IF(K.EQ.1)              THINCR(IP1,JP1,K)=THINCR(IP1,JP1,K)-DELT*WP(NCTHQ 42
     1                        I,J,K)*(TH8(IP1,JP1,KP1)+THS(KP1)-TH8(NCTHQ 43
     2                        IP1,JP1,K)-THS(K))/(DELP(K)*2.)        NCTHQ 44

                         QINCR(IP1,JP1,K)=-((W1+W5)*(Q1(IP2,JP1,K)-QNCTHQ 45
     1                        1(I,JP1,K))+(W2+W6)*(Q1(IP1,J,K)-Q1(IPNCTHQ 46
     2                        1,JP1,K)))*0.25*DELT/DM                NCTHQ 47
      IF(K.EQ.7)              QINCR(IP1,JP1,7)=QINCR(IP1,JP1,7)-(WP(I,J,7NCTHQ 48
     1                        )+WP(I,J,6))*DELT*(17.-Q1(IP1,JP1,7))/NCTHQ 49
     2                        1.5                                    NCTHQ 50
      IF(K.NE.1.AND.K.NE.7)  QINCR(IP1,JP1,K)=QINCR(IP1,JP1,K)-WP(I,J,K)NCTHQ 51
     1                        *DELT*(Q1(IP1,JP1,KP1)-Q1(IP1,JP1,K))/NCTHQ 52
     2                        (DELP(K)+DELP(KP1))-WP(I,J,KM1)*DELT*(NCTHQ 53
     3                        Q1(IP1,JP1,K)-Q1(IP1,JP1,KM1))/(DELP(KNCTHQ 54
     4                        )+DELP(KM1))                           NCTHQ 55
      IF(K.EQ.1)              QINCR(IP1,JP1,K)=QINCR(IP1,JP1,K)-DELT*WP(INCTHQ 56
     1                        ,J,K)*(Q1(IP1,JP1,KP1)-Q1(IP1,JP1,K))/NCTHQ 57
     2                        (DELP(K)*2.)                           NCTHQ 58
                             QINCR(IP1,JP1,K)=QINCR(IP1,JP1,K)+BKQ*DELT*NCTHQ 59
     1                        ALAPL(Q1,IP1,JP1,K,16,16,7)*ABS(ALAPL(NCTHQ 60
     2                        Q1,IP1,JP1,K,16,16,7))                 NCTHQ 61
2090 CONTINUE                                                       NCTHQ 62
```

```
      DO 100 K=1,7          $DO 100 I=2,15          $DO 100 J=2,15          NCTHQ 63
     ,IF(I.NE.2.AND.I.NE.15.AND.J.NE.2.AND.J.NE.15) GO TO 100              NCTHQ 64
      KM1=K-1                                                              NCTHQ 65
      IF(K.EQ.1) GO TO 15                                                  NCTHQ 66
      V1=AVPX4(VR,I-1,J-1,KM1,15,15,7)                                     NCTHQ 67
      U1=AVPY4(UR,I,J-1,KM1,15,15,7)                                       NCTHQ 68
      U2=AVPY4(UR,I-1,J-1,KM1,15,15,7)                                     NCTHQ 69
      V2=AVPX4(VR,I-1,J,KM1,15,15,7)                                       NCTHQ 70
      GO TO 19                                                             NCTHQ 71
   15 U1=(UR(I,J-1,1)+UR(I,J,1))*0.25                                      NCTHQ 72
      U2=(UR(I-1,J-1,1)+UR(I-1,J,1))*0.25                                  NCTHQ 73
      V1=(VR(I-1,J-1,1)+VR(I,J-1,1))*0.25                                  NCTHQ 74
      V2=(VR(I-1,J,1)+VR(I,J,1))*0.25                                      NCTHQ 75
   19 CONTINUE                                                             NCTHQ 76
      ADX=(U1*(TH8(I+1,J,K)-TH8(I,J,K))-U2*(TH8(I-1,J,K)-TH8(I,J,K)))      NCTHQ 77
     1           /(2.*DX)                                                  NCTHQ 78
      ADY=(V1*(TH8(I,J-1,K)-TH8(I,J,K))-V2*(TH8(I,J+1,K)-TH8(I,J,K)))      NCTHQ 79
     1           /(2.*DY)                                                  NCTHQ 80
      IF(K.EQ.7) GO TO 14                                                  NCTHQ 81
      AT=TH8(I,J,K+1)+THS(K+1)-TH8(I,J,K)-THS(K)                           NCTHQ 82
      IF(K.EQ.1) GO TO 13                                                  NCTHQ 83
      BT=TH8(I,J,K-1)+THS(K-1)-TH8(I,J,K)-THS(K)                           NCTHQ 84
      GO TO 99                                                             NCTHQ 85
   14 AT=TH8(I,J,K)+THS(K)-TH8(I,J,K-1)-THS(K-1)                           NCTHQ 86
      BT=TH8(I,J,K-1)+THS(K-1)-TH8(I,J,K)-THS(K)                           NCTHQ 87
      GO TO 99                                                             NCTHQ 88
   13 BT=0.                                                                NCTHQ 89
      KM1=1                                                                NCTHQ 90
   99 ADP=(WP(I-1,J-1,K)*AT-WP(I-1,J-1,KM1)*BT)/(2.*DELP(K))               NCTHQ 91
```

```
      ADX=ADY=0.                                                          NCTHQ 92
      THINCR(I,J,K)=-(ADX+ADY+ADP)*DELT                                   NCTHQ 93
      ADX=(U1*( Q1(I+1,J,K)- Q1(I,J,K))-U2*( Q1(I-1,J,K)- Q1(I,J,K)))    NCTHQ 94
     1           /(2.*DM)                                                 NCTHQ 95
      ADY=(V1*( Q1(I,J-1,K)- Q1(I,J,K))-V2*( Q1(I,J+1,K)- Q1(I,J,K)))    NCTHQ 96
     1           /(2.*DM)                                                 NCTHQ 97
      IF(K.EQ.7) GO TO 81                                                 NCTHQ 98
      AQ=Q1(I,J,K+1)-Q1(I,J,K)                                            NCTHQ 99
      IF(K.EQ.1) GO TO 82                                                 NCTHQ100
      BQ=Q1(I,J,K-1)-Q1(I,J,K)                                            NCTHQ101
      GO TO 88                                                            NCTHQ102
   81 AQ=Q1(I,J,K)-Q1(I,J,K-1)                                            NCTHQ103
      BQ=Q1(I,J,K-1)-Q1(I,J,K)                                            NCTHQ104
      GO TO 88                                                            NCTHQ105
   82 BQ=0.                                                               NCTHQ106
      KM1=1                                                               NCTHQ107
   88 ADP=(WP(I-1,J-1,K)*AQ-WP(I-1,J-1,KM1)*BQ)/(2.*DELP(K))              NCTHQ108
      QINCR(I,J,K)=-(ADX+ADY+ADP)*DELT                                    NCTHQ109
  100 CONTINUE                                                            NCTHQ110
      MARK1=.NOT.MARK1                                                    NCTHQ111
      IF (MARK1) GO TO 208                                                NCTHQ112
      IF (.NOT.MARK) GO TO 210                                            NCTHQ113

      DO 205 I=2,15          $DO 205 J=2,15          $DO 205 K=1,7        NCTHQ114
      THW(I,J,K)=TH8(I,J,K)                                               NCTHQ115
      THWINC(I,J,K)=Q1(I,J,K)                                             NCTHQ116
      TH8(I,J,K)=TH8(I,J,K)+THINCR(I,J,K)                                 NCTHQ117
  205 Q1(I,J,K)=Q1(I,J,K)+QINCR(I,J,K)                                    NCTHQ118
      GO TO 206                                                           NCTHQ119

  208 DO 207 I=2,15          $DO 207 J=2,15          $DO 207 K=1,7        NCTHQ120
      TH8(I,J,K)=THW(I,J,K)                                               NCTHQ121
  207 Q1(I,J,K)=THWINC(I,J,K)                                             NCTHQ122
  210 RETURN                                                              NCTHQ123
          END                                                             NCTHQ124
```

```
      SUBROUTINE BDYDIV(U,V,HDIVP,DM,ALPHA,MP,KL,VRAD)                          BDYDV   1
C* * *     COMPUTES BOUNDARY HORIZONTAL DIVERGENCES                        * * *BDYDV   2
      DIMENSION U(15,15,7),DRAD(52),VRAD(53),HDIVP(14,14,7),                    BDYDV   3
     1          V(15,15,7),DTAN(52),VTAN(52),A(12)                            BDYDV   4
      DATA      A/0.052175,0.046901,0.040315,0.032735,0.022622,0.011697,      BDYDV   5
     1            0.056823,0.060971,0.065512,0.060543,0.073521,0.076029/      BDYDV   6

      ALDM=(ALPHA-1.)/(2.17*DM)                                                 BDYDV   7
      ALDM2=(ALPHA+1.)/(2.*DM)                          SSD=0.                  BDYDV   8
      DO 10 I=1,12                                                              BDYDV   9
      J=14                        SL=I+13                                       BDYDV  10
      NM=6                        SNM=13-I              SM=15-I                 BDYDV  11
      IF(I.LE.6) NN=-6                                                          BDYDV  12
      NM6=NM/6                    SNN=NM+NN                                     BDYDV  13
      N=13-NN                     SIJ=I                                         BDYDV  14
      IF(I.GT.6) IJ=13                                                          BDYDV  15
      VRAD(I)=NM6 *(N*U(I+1,J,KL)+NN*U(I+2,J,KL))*A(IJ)-(N*V(I+1,J,KL)+         BDYDV  16
     1            NN*V(I+2,J,KL))*A(IJ+6)                                       BDYDV  17
      VTAN(I)=NM6 *(N*V(I+1,J,KL)+NN*V(I+2,J,KL))*A(IJ)+(N*U(I+1,J,KL)+         BDYDV  18
     1            NN*U(I+2,J,KL))*A(IJ+6)                                       BDYDV  19
      VRAD(L)=(N*U(J,M,KL)+NN*U(J,M-1,KL))*A(IJ+6)+NM6 *(N*V(J,M,KL)+NN*        BDYDV  20
     1            V(J,M-1,KL))*A(IJ)                                            BDYDV  21
      VTAN(L)=(N*V(J,M,KL)+NN*V(J,M-1,KL))*A(IJ+6)-NM6 *(N*U(J,M,KL)+NN*        BDYDV  22
     1            U(J,M-1,KL))*A(IJ)                                            BDYDV  23
      L=I+26                      SJ=2                                          BDYDV  24
      VRAD(L)=-NM6 *(N*U(M,J,KL)+NN*U(M-1,J,KL))*A(IJ)+(N*V(M,J,KL)+NN*V        BDYDV  25
     1            (M-1,J,KL))*A(IJ+6)                                           BDYDV  26
      VTAN(L)=-NM6 *(N*V(M,J,KL)+NN*V(M-1,J,KL))*A(IJ)-(N*U(M,J,KL)+NN*U        BDYDV  27
     1            (M-1,J,KL))*A(IJ+6)                                           BDYDV  28
      L=I+39                                                                    BDYDV  29
      VRAD(L)=-(N*U(J,I+1,KL)+NN*U(J,I+2,KL))*A(IJ+6)-NM6 *(N*V(J,I+1,KL        BDYDV  30
     1            )+NN*V(J,I+2,KL))*A(IJ)                                       BDYDV  31
      VTAN(L)=-(N*V(J,I+1,KL)+NN*V(J,I+2,KL))*A(IJ+6)+NM6 *(N*U(J,I+1,KL        BDYDV  32
     1            )+NN*U(J,I+2,KL))*A(IJ)                                       BDYDV  33
   10 CONTINUE                                                                  BDYDV  34
```

```
      DO 20 I=10,52,13                                                  BDYDV 35
      VRAD(I+3)=VRAD(I+2)                                               BDYDV 36
      VTAN(I+3)=VTAN(I+2)                                               BDYDV 37
      VRAD(I+2)=VRAD(I+1)                                               BDYDV 38
      VTAN(I+2)=VTAN(I+1)                                               BDYDV 39
      VRAD(I+1)=VRAD(I)                                                 BDYDV 40
      VTAN(I+1)=VTAN(I)                                                 BDYDV 41
      VRAD(I)=VRAD(I-1)                                                 BDYDV 42
      VTAN(I)=VTAN(I-1)                                                 BDYDV 43
      VRAD(I-1)=VRAD(I-2)                                               BDYDV 44
      VTAN(I-1)=VTAN(I-2)                                               BDYDV 45
      VRAD(I-2)=VRAD(I-3)                                               BDYDV 46
      VTAN(I-2)=VTAN(I-3)                                               BDYDV 47
   20 CONTINUE                                                          BDYDV 48
      VRAD(7)=-V(8,14,KL)                                               BDYDV 49
      VTAN(7)= U(8,14,KL)                                               BDYDV 50
      VRAD(20)= U(14,8,KL)                                              BDYDV 51
      VTAN(20)= V(14,8,KL)                                              BDYDV 52
      VRAD(33)= V(8,2,KL)                                               BDYDV 53
      VTAN(33)=-U(8,2,KL)                                               BDYDV 54
      VRAD(46)=-U(2,8,KL)                                               BDYDV 55
      VTAN(46)=-V(2,8,KL)                                               BDYDV 56
```

```
      DO 30 I=2,52                                                      BDYDV 57
      DRAD(I)=ALDR*(VRAD(I)+VRAD(I-1))                                  BDYDV 58
      DTAN(I)=ALDR2*(VTAN(I)-VTAN(I-1))                                 BDYDV 59
   30 CONTINUE                                                          BDYDV 60
      DRAD(1)=ALDR*(VRAD(1)+VRAD(52))                                   BDYDV 61
      DTAN(1)=ALDR2*(VTAN(1)-VTAN(52))                                  BDYDV 62

      DO 40 I=2,13                                                      BDYDV 63
      K=41-I                          $J=1                              BDYDV 64
      HDIVP(I,J,KL)=DRAD(K)+DTAN(K)                                     BDYDV 65
      HDIVP(J,I,KL)=DRAD(I+39)+DTAN(I+39)                               BDYDV 66
   40 CONTINUE                                                          BDYDV 67

      DO 50 I=1,13                                                      BDYDV 68
      K=28-I                          $J=14                             BDYDV 69
      HDIVP(I,J,KL)=DRAD(I)+DTAN(I)                                     BDYDV 70
      HDIVP(J,I,KL)=DRAD(K)+DTAN(K)                                     BDYDV 71
   50 CONTINUE                                                          BDYDV 72
      HDIVP(1,1,KL)=DRAD(40)+DTAN(40)                                   BDYDV 73
      HDIVP(14,14,KL)=DRAD(14)+DTAN(14)                                 BDYDV 74

      DO 60 I=1,52                                                      BDYDV 75
      SD=SD+DTAN(I)                                                     BDYDV 76
   60 CONTINUE                                                          BDYDV 77
      RETURN                                                            BDYDV 78
          END                                                           BDYDV 79
```

```
      SUBROUTINE BDYUV(UR,VR,VRAD,UINCR,VINCR,ALPHA,K)                    BDYUV   1
C* * *     COMPUTES BOUNDARY HORIZONTAL VELOCITY COMPONENTS          * * *BDYUV   2
      DIMENSION UR(15,15,7),UINCR(15,15,7),VRAD(53),                      BDYUV   3
     1          VR(15,15,7),VINCR(15,15,7),A(14)                         BDYUV   4
      DATA      A/0.,0.152057,0.294086,0.419058,0.524097,                 BDYUV   5
     1            0.609711,0.678280,0.734203,0.792624,                    BDYUV   6
     2            0.851658,0.907959,0.955779,0.988372,1.0/                BDYUV   7

      ALFA=ALPHA/14.          $MN=9                                       BDYUV   8
      VRAD(53)=VRAD(1)                                                    BDYUV   9

      DO 10 J=33,72,13        $DO 10 I=1,14                               BDYUV  10
      IJ=I+J                  $II=14-I                                    BDYUV  11
      IF(IJ.GT.53) IJ=IJ-52                                               BDYUV  12
      IF(IJ.LT.41.AND.IJ.GT.26) N=1                                       BDYUV  13
      IF(IJ.LT.15.OR.IJ.EQ.53)  N=15                                      BDYUV  14
      IF(IJ.LT.48.AND.IJ.GT.40.AND.I.NE.1) N=I-6                          BDYUV  15
      IF(IJ.GT.46.AND.I .NE.14) N=I+8                                     BDYUV  16
      IF(IJ.GT.14.AND.IJ.LT.22.AND.I.NE.I) N=II+8                         BDYUV  17
      IF(IJ.GT.20.AND.IJ.LT.28.AND.I.NE.14) N=II-6                        BDYUV  18
      IF(IJ.GT.14.AND.IJ.LT.41) MN=MN-1                                   BDYUV  19
      IF(IJ.LE.14.OR.IJ.GE.41) MN=MN+1                                    BDYUV  20
      M=MN-N                  $IA=I                                       BDYUV  21
      IF(J.EQ.46.OR.J.EQ.72) IA=15-I                                      BDYUV  22
      IB=IA+1                                                             BDYUV  23
      IF(J.EQ.46.OR.J.EQ.72) IB=IA-1                                      BDYUV  24
      IF(I.EQ.14)IB=IA                                                    BDYUV  25
```

```
      ISIGN=-1                    ;JSIGN=-1                                          BDYUV 26
      IF(J.EQ.33.OR.J.EQ.72) JSIGN=1                                                 BDYUV 27
      IF(J.EQ.59.OR.J.EQ.72)  ISIGN=1                                                BDYUV 28

      UU=UR(M,N,K)                ;VU=VR(M,N,K)                                      BDYUV 29
      UR(M,N,K)=ISIGN*ALFA*(II*VRAD(IJ)*A(IB )+I*VRAD(IJ-1)*A(IA ))                  BDYUV 30
      VR(M,N,K)=JSIGN*ALFA*(II*VRAD(IJ)*A(15-IB))+I*VRAD(IJ-1)*A(15 -IA))BDYUV 31

      IF(I.EQ.14)GO TO 20                                                            BDYUV 32
      GO TO 40                                                                       BDYUV 33
   20 IF(J.EQ.46.OR.J.EQ.72) GO TO 30                                                BDYUV 34

      VR(M,N,K)=0.                                                                   BDYUV 35
      GO TO 40                                                                       BDYUV 36
   30 UR(M,N,K)=0.                                                                   BDYUV 37

   40 IF(I.EQ.1) GO TO 50                                                            BDYUV 38
      GO TO 70                                                                       BDYUV 39
   50 IF(J.EQ.46.OR.J.EQ.72) GO TO 60                                                BDYUV 40

      UR(M,N,K)=ISIGN*ALFA*II*VRAD(IJ)*A(IB)                                         BDYUV 41
      GO TO 70                                                                       BDYUV 42
   60 VR(M,N,K)=JSIGN*ALFA*II*VRAD(IJ)*A(15-IB)                                      BDYUV 43

   70 CONTINUE                                                                       BDYUV 44
      UINCR(M,N,K)=UR(M,N,K)-UU                                                      BDYUV 45
      VINCR(M,N,K)=VR(M,N,K)-VU                                                      BDYUV 46
   10 CONTINUE                                                                       BDYUV 47
      RETURN                                                                         BDYUV 48
         END                                                                         BDYUV 49
```

```
      SUBROUTINE BDY(UR,VR,OUTFLO)                                          BDY     1
C* * *    FORMS  INFLOW/OUTFLOW  ARRAY                              * * *BDY     2
      DIMENSION UR(15,15,7),VR(15,15,7),OUTFLO(56,7),THS(7)                 BDY     3
      LOGICAL   OUTFLO                                                      BDY     4
      COMMON    CCX,DM                                                      BDY     5
      DATA      THS/424.,385.,353.,332.,318.,308.,304./                     BDY     6

      DO 9000 KK=1,7                                                        BDY     7
      K=8-KK                                                                BDY     8

      DO 110 I=1,56                                                         BDY     9
  110 OUTFLO(I,K)=.FALSE.                                                   BDY    10
      IF((UR(1,1,K)-CCX).LT.0..AND.VR(1,1,K).GT.0.) OUTFLO(1,K)=.TRUE.      BDY    11

      DO 4 I=2,14                                                           BDY    12
    4 IF (VR(I,1,K).GT.0.)    OUTFLO(I,K)=.TRUE.                          BDY    13
      IF((UR(15,1,K)-CCX).GT.0..AND.VR(15,1,K).GT.0.)OUTFLO(15,K)=.TRUE.  BDY    14

      DO 8 J=2,14                                                           BDY    15
      J1=J+14                                                               BDY    16
    8 IF((UR(15,J,K)-CCX).GT.0.)    OUTFLO(J1,K)=.TRUE.                   BDY    17
      IF((UR(15,15,K)-CCX).GT.0..AND.VR(15,15,K).LT.0.)                     BDY    18
     1           OUTFLO(29,K) =  .TRUE.                                     BDY    19

      DO 12 I=2,14                                                          BDY    20
      I1=44-I                                                               BDY    21
   12 IF(VR(I,15,K).LT.0.)    OUTFLO(I1,K)=.TRUE.                         BDY    22
      IF((UR(1,15,K)-CCX).LT.0..AND.VR(1,15,K).LT.0.)OUTFLO(43,K)=.TRUE.  BDY    23

      DO 16 J=2,14                                                          BDY    24
      J1=58-J                                                               BDY    25
   16 IF((UR(1,J,K)-CCX).LT.0.)    OUTFLO(J1,K)=.TRUE.                    BDY    26
 9000 CONTINUE                                                              BDY    27
      RETURN                                                                BDY    28
         END                                                                BDY    29
```

```
      SUBROUTINE NEW(PSI,ZETA,UR,VR,HEIGHT,MP,F,DELT,HINCR,Q1,TH,THW,    NEW     1
     1                 QINCR,THINCR,THWINC,ISWICH,MP)                     NEW     2
C* * *     CALLS BOUNDARY SUBROUTINES AND HEIGHT FIELD COMPUTATION  * * *NEW     3
      DIMENSION  HEIGHT(16,16,7),UINCR(15,15,7),HDIVP(14,14,7),THICK(7),NEW     4
     1           THWINC(16,16,7),VINCR(15,15,7), ZETA(14,14,7), HFAC(7),NEW     5
     2           THINCR(16,16,7),    UR(15,15,7),    MP(14,14,7),  THS(7),NEW     6
     3            HINCR(16,16,7),    VR(15,15,7),CORRN(14,14),     QST(7),NEW     7
     4            QINCR(16,16,7),    Q1(16,16,7),    VR(16,16),    UST(7),NEW     8
     5              THW(16,16,7),    QL(16,16,7),     F(15),       H(60,7),NEW     9
     6             EDGI(60),       PERI(56),       VRAD(92)               NEW    10
      EQUIVALENCE(H(1),CORRN(1)),(EDGI(1),PERI(1))                        NEW    11
      LOGICAL     MARK                                                    NEW    12
      COMMON      /BLK 4/ UINCR,VINCR                                     NEW    13
      COMMON      /BLK21/ HDIVP,QL                                        NEW    14
      COMMON      CCX,DT,KOUNT2,MARK                                      NEW    15
      DATA        KOUNTA/3/                                               NEW    16
      DATA        THICK/.5,.5,1.,2.,2.,1.5,1.5/                           NEW    17
      DATA        HFAC /6.5248,5.101,7.9582,11.4107,8.4512,5.2751,4.6606/NEW    18
      DATA        THS  /424.,385.,353.,332.,318.,308.,304./               NEW    19
      DATA        QST  /0.,0.,0.,1.1,3.6,7.8,13.,5/                       NEW    20
      DATA        UST  /-15.5,-12.5,-10.,-5.25,-0.5,3.,6.5/               NEW    21

      D=DT/1000.              SC5=10000.                                  NEW    22
      IF(ISWICH.EQ.1)  GO TO 99                                           NEW    23
      NUMINC=1                SALPHA=1.                                   NEW    24
```

```
   DO 25 I=1,14          $DO 25 J=1,14          $DO 25 K=1,7                NEW    25
25 HDIVP(I,J,K)=(UR(I+1,J,K)+UR(I+1,J+1,K)+VR(I,J,K)+VR(I+1,J,K)-UR(INEW    26
  1           ,J,K)-UR(1,J+1,K)-VR(I,J+1,K)-VR(I+1,J+1,K))/(2.*DM)      NEW    27

   DO 77 KL=1,7          $DO 88 I=1,15          $DO 88 J=1,15               NEW    28
   VR(I,J,7)=VR(I,J,7)-6.5                                                  NEW    29
88 UR(I,J,KL)=UR(I,J,KL)-UST(KL)                                           NEW    30
      CALL BDYDIV(UR,VR,HDIVP,DM,ALPHA,MP,KL,VRAD)                         NEW    31
      CALL BDYUV(UR,VR,VRAD,UINCR,VINCR,ALPHA,KL)                          NEW    32

   DO 89 I=1,15          $DO 89 J=1,15                                      NEW    33
   VR(I,J,7)=VR(I,J,7)+6.5                                                  NEW    34
89 UR(I,J,KL)=UR(I,J,KL)+UST(KL)                                           NEW    35
77 CONTINUE                                                                NEW    36

   DO 61 I=1,14          $DO 61 J=1,14                                      NEW    37
   WP(I,J,1)=-THICK(1)*0.5*HDIVP(I,J,1)                                     NEW    38
   DO 61 K=2,7                                                              NEW    39
61 WP(I,J,K)=WP(I,J,K-1)-THICK(K)*0.5*(HDIVP(I,J,K-1)+HDIVP(I,J,K))        NEW    40
   DO 62 I=1,14          $DO 62 J=1,14          $DO 62 K=1,7                NEW    41
62 ZETA(I,J,K)=(VR(I+1,J,K)+VR(I+1,J+1,K)-VR(I,J,K)-VR(I,J+1,K)            NEW    42
  1          -UR(I,J,K)-UR(I+1,J,K)+UR(I,J+1,K)+UR(I+1,J+1,K))/(2.*DM)NEW    43
  2          +F(J)*0.5+F(J+1)*0.5                                          NEW    44
   RETURN                                                                   NEW    45
99 K=7                                                                      NEW    46
```

```
      DO  3  I=2,15              $DO  3  J=2,15                                        NEW     47
    3 HINCR(I,J,7)=DELT*(GP(I-1,J-1,7)*900.-0.25*((UR(I-1,J,K)+UR(I-1,J-NEW     48
     1           1,K))*(HEIGHT(I,J,K)-HEIGHT(I-1,J,K))+(UR(I,J,K)+UR(I,J-1NEW     49
     2           ,K))*(HEIGHT(I+1,J,K)-HEIGHT(I,J,K)))/DX-0.25*((VR(I+1,J-NEW     50
     3           1,K)+VR(I,J-1,K))*(HEIGHT(I,J-1,K)-HEIGHT(I,J,K))+(VR(I-1NEW     51
     4           ,J,K)+VR(I,J,K))*(HEIGHT(I,J,K)-HEIGHT(I,J+1,K)))/DX+CCX*NEW     52
     5           (HEIGHT(I+1,J,K)-HEIGHT(I-1,J,K))/(2.*DX))+CS*DELT*ALAPL(NEW     53
     6           HEIGHT,I,J,K,16,16,7)/(DX*DX)                                      NEW     54
      IF (.NOT.MAPK) GO TO 6                                                        NEW     55

      DO 131 I=2,15              $DO 131 J=2,15                                     NEW     56
      CORRN(I-1,J-1)=HEIGHT(I,J,7)                                                  NEW     57
  131 HEIGHT(I,J,7)=HEIGHT(I,J,7)+HINCR(I,J,7)                                      NEW     58
      K=7                                                                           NEW     59

      DO  53  I=2,15             $DO  53  J=2,15                                    NEW     60
   53 HINCR(I,J,7)=DELT*(GP(I-1,J-1,7)*900.-0.25*((UR(I-1,J,K)+UR(I-1,J-NEW     61
     1           1,K))*(HEIGHT(I,J,K)-HEIGHT(I-1,J,K))+(UR(I,J,K)+UR(I,J-1NEW     62
     2           ,K))*(HEIGHT(I+1,J,K)-HEIGHT(I,J,K)))/DX-0.25*((VR(I+1,J-NEW     63
     3           1,K)+VR(I,J-1,K))*(HEIGHT(I,J-1,K)-HEIGHT(I,J,K))+(VR(I-1NEW     64
     4           ,J,K)+VR(I,J,K))*(HEIGHT(I,J,K)-HEIGHT(I,J+1,K)))/DX+CCX*NEW     65
     5           (HEIGHT(I+1,J,K)-HEIGHT(I-1,J,K))/(2.*DX))+CS*DELT*ALAPL(NEW     66
     6           HEIGHT,I,J,K,16,16,7)/(DX*DX)                                      NEW     67

      DO  54  I=2,15             $DO  54  J=2,15                                    NEW     68
   54 HEIGHT(I,J,7)=CORRN(I-1,J-1)                                                  NEW     69
    6 CONTINUE                                                                      NEW     70

      DO  300  I=2,15            $DO  300  J=2,15                                   NEW     71
  300 HEIGHT(I,J,7)=HEIGHT(I,J,7)+HINCR(I,J,7)                                      NEW     72
          CALL  HFIELD(UR,VR,GINCR,GP,HEIGHT,THGINC,DELT,HDIVP,TH,THS,HFACNEW     73
     1           ,G1,GL,THW,THINCR)                                                 NEW     74
```

```
      DO 20 K=1,7              $DO 10 I=2,15          $DO 10 J=2,15          NEW    75
      II=I-1                   $JJ=J-1                                       NEW    76
      VM(I,J)=(AV4(UR,II,JJ,K,15,15,7))**2+(AV4(VR,II,JJ,K,15,15,7))**2     NEW    77
   10 PSI(I,J,K)=9.81*HEIGHT(I,J,K)+VM(I,J)*0.5                             NEW    78
      EDG1(1)=-UR(1,1,K)*ZETA(1,1,K)*DM                                     NEW    79

      DO 210 I=2,14                                                         NEW    80
  210 EDG1(I)=-UR(I,1,K)*(ZETA(I-1,1,K)+ZETA(I,1,K))*0.5*DM                 NEW    81
      EDG1(15)=-UR(15,1,K)*ZETA(14,1,K)*DM                                  NEW    82
      EDG1(16)=VR(15,1,K)*ZETA(14,1,K)*DM                                   NEW    83

      DO 220 J=2,14                                                         NEW    84
  220 EDG1(J+15)=VR(15,J,K)*(ZETA(14,J-1,K)+ZETA(14,J,K))*0.5*DM            NEW    85
      EDG1(30)=VR(15,15,K)*ZETA(14,14,K)*DM                                 NEW    86
      EDG1(31)=UR(15,15,K)*ZETA(14,14,K)*DM                                 NEW    87

      DO 230 I=2,14                                                         NEW    88
      L=46-I                                                                NEW    89
  230 EDG1(L)=UR(I,15,K)*(ZETA(I-1,14,K)+ZETA(I,14,K))*0.5*DM               NEW    90
      EDG1(45)= UR(1,15,K)*ZETA(1,14,K)*DM                                  NEW    91
      EDG1(46)=-VR(1,15,K)*ZETA(1,14,K)*DM                                  NEW    92

      DO 240 J=2,14                                                         NEW    93
      L=61-J                                                                NEW    94
  240 EDG1(L)=-VR(1,J,K)*(ZETA(1,J-1,K)+ZETA(1,J,K))*0.5*DM                 NEW    95
      EDG1(60)=-VR(1,1,K)*ZETA(1,1,K)*DM                                    NEW    96
      DMDELT=DM/DELT                                                        NEW    97

      DO 129 I=1,15                                                         NEW    98
      J=16-I                                                                NEW    99
      EDG1(I)=EDG1(I)-VINCR(I,1,K)*DMDELT                                   NEW   100
      EDG1(I+15)=EDG1(I+15)-UINCR(15,I,K)*DMDELT                            NEW   101
      EDG1(I+30)=EDG1(I+30)+VINCR(J,15,K)*DMDELT                            NEW   102
  129 EDG1(I+45)=EDG1(I+45)+UINCR(1,J,K)*DMDELT                             NEW   103
```

```
      DO 140 I=1,14                                                    NEW  104
      EDG1(I)=(EDG1(I)+EDG1(I+1))*0.5                                  NEW  105
      EDG1(I+15)=(EDG1(I+15)+EDG1(I+16))*0.5                           NEW  106
      EDG1(I+30)=(EDG1(I+30)+EDG1(I+31))*0.5                           NEW  107
  140 EDG1(I+45)=(EDG1(I+45)+EDG1(I+46))*0.5                           NEW  108

      DO 133 I=2,15                                                    NEW  109
      L=46-I                                                           NEW  110
      PSI(I,1,K)=PSI(I,2,K)+EDG1(I-1)                                  NEW  111
  133 PSI(I,16,K)=PSI(I,15,K)+EDG1(L)                                  NEW  112

      DO 132 J=2,15                                                    NEW  113
      L=61-J                                                           NEW  114
      PSI(16,J,K)=PSI(15,J,K)+EDG1(J+14)                               NEW  115
  132 PSI(1,J,K)=PSI( 2,J,K)+EDG1(L)                                   NEW  116
      PSI(16,1,K)=-PSI(15,2,K)+PSI(15,1,K)+PSI(16, 2,K)                NEW  117
      PSI(16,16,K)=-PSI(15,15,K)+PSI(16,15,K)+PSI(15,16,K)             NEW  118
      PSI(1,16,K)=-PSI(2,15,K)+PSI(2,16,K)+PSI(1,15,K)                 NEW  119
      PSI(1,1,K)=-PSI(2,2,K)+PSI(1,2,K)+PSI(2,1,K)                     NEW  120

      DO 161 I=1,15                                                    NEW  121
  161 PERI(I)=UR(I,1,K)*UR(I,1,K)+VR(I,1,K)*VR(I,1,K)                  NEW  122
      DO 162 J=2,15                                                    NEW  123
  162 PERI(J+14)=UR(15,J,K)*UR(15,J,K)+VR(15,J,K)*VR(15,J,K)           NEW  124
      DO 63 I=30,43                                                    NEW  125
      I1=44-I                                                          NEW  126
   63 PERI(I)=UR(I1,15,K)*UR(I1,15,K)+VR(I1,15,K)*VR(I1,15,K)          NEW  127
      DO 64 I=44,56                                                    NEW  128
      J1=58-I                                                          NEW  129
   64 PERI(I)=UR(1,J1,K)*UR(1,J1,K)+VR(1,J1,K)*VR(1,J1,K)              NEW  130
         CALL EXTONE (VM,PERI)                                         NEW  131
```

```
      DO 135 I=1,15                                                          NEW  132
      J=17-I                                                                 NEW  133
      H(I,K)=(PSI(I,1,K)-0.5* VM(I,1))/9.81                                  NEW  134
      H(I+15,K)=(PSI(16,I,K)-0.5* VM(16,I))/9.81                             NEW  135
      H(I+30,K)=(PSI(J,16,K)-0.5*VM(J,16))/9.81                              NEW  136
  135 H(I+45,K)=(PSI(1,J ,K)-0.5*VM(1,J ))/9.81                              NEW  137
   20 CONTINUE                                                               NEW  138
      IF(KOUNT2.EQ.1) GO TO 55                                               NEW  139

      DO 22 K=2,7           $DO 136 I=2,15                                   NEW  140
      M=17-I                                                                 NEW  141
      TH(16,I,K)=(HEIGHT(16,I,K-1)-HEIGHT(16,I,K))*TH(15,I,K)/(HEIGHT(       NEW  142
     1           15,I,K-1)-HEIGHT(15,I,K))                                   NEW  143
      TH (I,1,K)=(HEIGHT(I,1,K-1)-HEIGHT(I,1,K))*TH (I,2,K)/(HEIGHT(I,2      NEW  144
     1           ,K-1)-HEIGHT(I,2,K))                                        NEW  145
      TH(M,16,K)=(HEIGHT(M,16,K-1)-HEIGHT(M,16,K))*TH(M,15,K)/(HEIGHT(       NEW  146
     1           M,15,K-1)-HEIGHT(M,15,K))                                   NEW  147
  136 TH (1,M,K)=(HEIGHT(1,M,K-1)-HEIGHT(1,M,K))*TH (2,M,K)/(HEIGHT(2,M      NEW  148
     1           ,K-1)-HEIGHT(2,M,K))                                        NEW  149
      TH(16,1,K)=(HEIGHT(16,1,K-1)-HEIGHT(16,1,K))*TH(15,2,K)/(HEIGHT(       NEW  150
     1           15,2,K-1)-HEIGHT(15,2,K))                                   NEW  151
      TH(1,16,K)=(HEIGHT(1,16,K-1)-HEIGHT(1,16,K))*TH(2,15,K)/(HEIGHT(       NEW  152
     1           2,15,K-1)-HEIGHT(2,15,K))                                   NEW  153
      TH (1,1,K)=(HEIGHT(1,1,K-1)-HEIGHT(1,1,K))*TH (2,2,K)/(HEIGHT(2,2      NEW  154
     1           ,K-1)-HEIGHT(2,2,K))                                        NEW  155
      TH(16,16,K)=(HEIGHT(16,16,K-1)-HEIGHT(16,16,K))*TH(15,15,K)/(HEIG      NEW  156
     1           HT(15,15,K-1)-HEIGHT(15,15,K))                              NEW  157
   22 CONTINUE                                                               NEW  158

   55 DO 138 I=1,15                                                          NEW  159
      J=17-I                                                                 NEW  160
      THINCR(I,1,1)=0.                                                       NEW  161
      THINCR(16,I,1)=0.                                                      NEW  162
      THINCR(J,16,1)=0.                                                      NEW  163
  138 THINCR(1,J,1)=0.                                                       NEW  164
      RETURN                                                                 NEW  165
         END                                                                 NEW  166
```

```
      SUBROUTINE HFIELD(VR,VR,AVVP,VP,HP,STAT,DELT,RMS,TH,THS,HFAC,Q1    HFELD   1
     1                 ,QL,RAIN,HDIVP)                                   HFELD   2
C* * *     COMPUTES DEVIATION HEIGHT FIELD BY ITERATION METHOD (SOR)* * *HFELD   3
      DIMENSION AVVP(16,16,7),Q1(16,16,7),HP(16,16,7),HDIVP(16,16,7),    HFELD   4
     1          RAIN(16,16,7),QL(16,16,7),UR(15,15,7),   RMS(14,14,7),   HFELD   5
     2          STAT(16,16,7),TH(16,16,7),VR(15,15,7),    VP(14,14,7),   HFELD   6
     3          HFAC(7),      THS(7),       RO(7),        DELP(7)        HFELD   7
      COMMON    CCX,DX,KOUNT2                                            HFELD   8
      DATA      ALFA/0.315/                                              HFELD   9
      DATA      DELP/0.5,0.5,1.,2.,2.,1.5,1.5/                           HFELD  10
      DATA      RO/254.,320.,434.,655.,862.,1015.,1152./                 HFELD  11

      CAPA=500.                                                          HFELD  12

      DO 22 I=1,14          DO 22 J=1,14          DO 22 K=1,7            HFELD  13
   22 HDIVP(I+1,J+1,K)=RMS(I,J,K)                                        HFELD  14

      DO 44 K=1,7           DO 33 I=2,15                                 HFELD  15
      HDIVP(1 ,I ,K)=HDIVP(2 ,I ,K)                                      HFELD  16
      HDIVP(16,I ,K)=HDIVP(15,I ,K)                                      HFELD  17
      HDIVP(I ,1 ,K)=HDIVP(I ,2 ,K)                                      HFELD  18
   33 HDIVP(I ,16,K)=HDIVP(I ,15,K)                                      HFELD  19
      HDIVP(1 ,1 ,K)=HDIVP(2 ,2 ,K)                                      HFELD  20
      HDIVP(16,16,K)=HDIVP(15,15,K)                                      HFELD  21
      HDIVP(1 ,16,K)=HDIVP(2 ,15,K)                                      HFELD  22
   44 HDIVP(16,1 ,K)=HDIVP(15,2 ,K)                                      HFELD  23
      DDX=DX*DX*2.          ENREL=0                                      HFELD  24
```

```
      CALL SUMRHS(UR,VR,WP,AVWP,DM,RHS,RO)                              HFELD 25
      CALL STATIC(TH,THS,DELP,STAT,Q1,QL,RAIN)                          HFELD 26

    DO 23 I=2,15          $DO 23 J=2,15          $DO 23 K=2,6           HFELD 27
 23 AVWP(I,J,K)=CAPA*ALAPL(HDIVP,I,J,K,16,16,7)/4.905                   HFELD 28
    IC=0                                                                HFELD 29
999 CONTINUE                                                            HFELD 30

    IC=IC+1               $D=0.                                         HFELD 31
    IF(IC.EQ.5) IC=1                                                    HFELD 32
    DO 34 K=2,6                                                         HFELD 33
    RR=96.24*RO(K)*RO(K)*1.E-14                                         HFELD 34
    DPK=DELP(K)+DELP(K+1)                                               HFELD 35
    DRDK=DDM*192.47*RO(K)* (RO(K+1)-RO(K-1))/DPK                        HFELD 36
    DK=DELP(K)/DELP(K+1)                                                HFELD 37
    RRP=RR/(0.5*DPK)                                                    HFELD 38
    DRDK1=DRDK/1.E14                                                    HFELD 39

    DO 34 L1=2,15         $DO 34 L2=2,15                                HFELD 40
    GO TO (101,102,103,104), IC                                         HFELD 41
101 I=L2                  $J=L1                                         HFELD 42
    GO TO 105                                                           HFELD 43
102 J=L2                  $I=17-L1                                      HFELD 44
    GO TO 105                                                           HFELD 45
103 I=17-L2               $J=17-L1                                      HFELD 46
    GO TO 105                                                           HFELD 47
104 J=17-L2               $I=L1                                         HFELD 48
105 CONTINUE                                                            HFELD 49
    TM=1.                 $T=1.                                         HFELD 50
```

```
      HXX=HP(I+1,J+1,K)+HP(I+1,J-1,K)-2.*HP(I,J,K)                              HFELD 51
      HYY=HP(I-1,J+1,K)+HP(I-1,J-1,K)-2.*HP(I,J,K)                              HFELD 52
      HPP=((HP(I,J,K+1)-HP(I,J,K))/DELP(K+1)-(HP(I,J,K)-HP(I,J,K-1              HFELD 53
     1            ))/DELP(K))                                                   HFELD 54
      HPP=HPP*DRP                                                               HFELD 55
      DHDP=(DK*(HP(I,J,K+1)-HP(I,J,K))+(HP(I,J,K)-HP(I,J,K-1))/DK)/DPK          HFELD 56
      RES=HXX+HYY+DDM*HPP+(STAT(I,J,K)-RHS(I-1,J-1,K))*DDM/7.81                 HFELD 57
      RES=RES+DHDP*DRDK1                                                        HFELD 58
      RES=RES+AVWP(I,J,K)                                                       HFELD 59
      HP(I,J,K)=HP(I,J,K)+ALFA*RES                                              HFELD 60
      IF(ABS(RES).GT.D) D=RES                                                   HFELD 61
   34 CONTINUE                                                                  HFELD 62
      IF(ABS(D).GT.0.000001)NREL=NREL+1                                         HFELD 63
      IF(NREL.GT.100) RETURN                                                    HFELD 64
      IF(ABS(D).GT.0.000001)GO TO 999                                           HFELD 65
      WRITE(6,47) NREL                                                          HFELD 66

      DO 35 I=2,15          $DO 35 J=2,15                                       HFELD 67
   35 HP(I,J,1)=HP(I,J,2)+HFAC(2)*TH(I,J,2)                                     HFELD 68
      RETURN                                                                    HFELD 69
   47 FORMAT(1X,17HNO OF ITERATIONS=,I4)                                        HFELD 70
         END                                                                    HFELD 71
```

```
      SUBROUTINE STATIC(TH8,THS,DELP,STAT,Q1,QL,RAIN)                    STAT    1
C* * *    COMPUTES HYDROSTATIC TERMS ON RHS OF BALANCE EQUATION    * * *STAT    2
      DIMENSION RAIN(16,16,7),Q1(16,16,7),TH8(16,16,7),THS(7),DELP(7),   STAT    3
     1          STAT(16,16,7),QL(16,16,7),FACTOR(6),   QST(7),  P0(7)    STAT    4
      DATA      QST/0.,0.,0.,1.1,3.6,7.8,15.5/                           STAT    5
      DATA      P0/150.,200.,300.,500.,700.,850.,1000./                  STAT    6

      DO 22 I=2,15          $DO 22 J=2,15          $DO 22 K=1,7          STAT    7
   22 QL(I,J,K)=QL(I,J,K)+RAIN(I,J,K)                                    STAT    8

      DO 11 K=2,6                                                       STAT    9
      TT2=THS(K+1)+THS(K)                                               STAT   10
      FACTOR(K)=1.E-4*P0(K)*96.236/(1.435*TT2*(P0(K)*0.001)**0.286)      STAT   11

      DO 11 I=2,15          $DO 11 J=2,15                               STAT   12
      TT1=TH8(I,J,K)+TH8(I,J,K+1)                                       STAT   13
      TT1=TT1+0.00061*(THS(K)*(Q1(I,J,K)-QST(K))+THS(K+1)*(Q1(I,J,K+1)  STAT   14
     1          -QST(K+1)))                                             STAT   15
      TP=0.25*(DELP(K)+DELP(K+1))*TT2                                   STAT   16
      THP1=(TH8(I,J,K+1)-TH8(I,J,K))/TP                                 STAT   17
      THP1=THP1+0.00061*(THS(K+1)*(Q1(I,J,K+1)-QST(K+1))-THS(K)*(Q1     STAT   18
     1          (I,J,K)-QST(K)))/TP                                     STAT   19
      THP2=TT1*0.7143/(TT2*P0(K)*0.01)                                  STAT   20
      THP3=8.*TT1*(THS(K+1)-THS(K))/(TT2*TT2*(DELP(K)+DELP(K+1)))       STAT   21
      QL1=1.E-3*(QL(I,J,K+1)-QL(I,J,K))*TT2/(TP*2.)                     STAT   22
      QQ=(QL(I,J,K+1)+QL(I,J,K) )*0.001                                 STAT   23
      QL2=0.3571*QQ/(P0(K)*0.01)                                        STAT   24
      QL3=2.*QQ*(THS(K+1)-THS(K))/(TT2*(DELP(K)+DELP(K+1)))             STAT   25
      STAT(I,J,K)=(THP1+THP2-THP3)*FACTOR(K)                            STAT   26
   11 STAT(I,J,K)=STAT(I,J,K)+(QL3-QL1-QL2)*FACTOR(K)                   STAT   27

      DO 33 I=2,15          $DO 33 J=2,15          $DO 33 K=1,7          STAT   28
   33 QL(I,J,K)=QL(I,J,K)-RAIN(I,J,K)                                   STAT   29
      RETURN                                                            STAT   30
         END                                                            STAT   31
```

```
      SUBROUTINE SUBRHS(U,V,W,AVJP,DELT,RHS,RO)                                 SUBRHS   1
C* * *    FORMS PART OF RHS OF BALANCE EQUATION                              * * *SUBRHS   2
      DIMENSION U(15,15,7),AVJP(16,16,7),RHS(14,14,7),SP(15),RO(7),          SUBRHS   3
     1          V(15,15,7),   F(15,15,7),   W(14,14,7),  X(15)               SUBRHS   4
      DO 22 I=1,14          $DO 22 J=1,14          $DO 22 K=1,7               SUBRHS   5
   22 W(I,J,K)=W(I,J,K)/RO(K)                                                 SUBRHS   6
      DO 99 I=1,14          $DO 99 J=1,14          $DO  5 K=1,7               SUBRHS   7
    5 SP(K)=W(I,J,K)                                                          SUBRHS   8
        CALL DDP(SP,X)                                                        SUBRHS   9
      DO 99 K=1,7                                                             SUBRHS  10
      IF(I.EQ.1) GO TO 20                                                     SUBRHS  11
      IF(I.EQ.14) GO TO 50                                                    SUBRHS  12
      GO TO 80                                                                SUBRHS  13
   20 IF(J.EQ.1.OR.J.EQ.14) GO TO 30                                          SUBRHS  14
      F(I,J,K)=(AV4(U,I,J,K,7,7,7)*(W(I+1,J,K)-W(I,J,K)))/DELT+(AV4(V,I,SUBRHS  15
     1          J,K,7,7,7)*(W(I,J-1,K)-W(I,J+1,K)))/(2.*DELT)+W(I,J,K)*X(SUBRHS  16
     2          K)                                                            SUBRHS  17
      GO TO 99                                                                SUBRHS  18
   30 IF(J.EQ.14) GO TO 40                                                    SUBRHS  19
      F(I,J,K)=(AV4(U,I,J,K,7,7,7)*(W(I+1,J,K)-W(I,J,K))+AV4(V,I,J,K,7,7SUBRHS  20
     1          ,7)*(W(I,J,K)-W(I,J+1,K)))/DELT+W(I,J,K)*X(K)                 SUBRHS  21
      GO TO 99                                                                SUBRHS  22
   40 F(I,J,K)=(AV4(U,I,J,K,7,7,7)*(W(I+1,J,K)-W(I,J,K))+AV4(V,I,J,K,7,7SUBRHS  23
     1          ,7)*(W(I,J-1,K)-W(I,J,K)))/DELT+W(I,J,K)*X(K)                 SUBRHS  24
      GO TO 99                                                                SUBRHS  25
   50 IF(J.EQ.1.OR.J.EQ.14) GO TO 60                                          SUBRHS  26
      F(I,J,K)=(AV4(U,I,J,K,7,7,7)*(W(I,J,K)-W(I-1,J,K)))/DELT+(AV4(V,I,SUBRHS  27
     1          J,K,7,7,7)*(W(I,J-1,K)-W(I,J+1,K)))/(2.*DELT)+W(I,J,K)*X(SUBRHS  28
     2          K)                                                            SUBRHS  29
      GO TO 99                                                                SUBRHS  30
   60 IF(J.EQ.14) GO TO 70                                                    SUBRHS  31
      F(I,J,K)=(AV4(U,I,J,K,7,7,7)*(W(I,J,K)-W(I-1,J,K))+AV4(V,I,J,K,7,7SUBRHS  32
     1          ,7)*(W(I,J,K)-W(I,J+1,K)))/DELT+W(I,J,K)*X(K)                 SUBRHS  33
      GO TO 99                                                                SUBRHS  34
   70 F(I,J,K)=(AV4(U,I,J,K,7,7,7)*(W(I,J,K)-W(I-1,J,K))+AV4(V,I,J,K,7,7SUBRHS  35
     1          ,7)*(W(I,J-1,K)-W(I,J,K)))/DELT+W(I,J,K)*X(K)                 SUBRHS  36
      GO TO 99                                                                SUBRHS  37
```

```
 80 IF(J.EQ.1.OR.J.EQ.14) GO TO 90                                              SMRHS 38
    F(I,J,K)=(AV4(U,I,J,K,7,7,7)*(W(I+1,J,K)-W(I-1,J,K))+AV4(V,I,J,K,7SMRHS 39
   1          ,7,7)*(W(I,J-1,K)-W(I,J+1,K)))/(2.*DELT)+W(I,J,K)*X(K)    SMRHS 40
    GO TO 99                                                                    SMRHS 41
 90 IF(J.EQ.14) GO TO 100                                                       SMRHS 42
    F(I,J,K)=(AV4(U,I,J,K,7,7,7)*(W(I+1,J,K)-W(I-1,J,K)))/(2.*DELT)+(  SMRHS 43
   1          AV4(V,I,J,K,7,7,7)*(W(I,J,K)-W(I,J+1,K)))/DELT+W(I,J,K)*XSMRHS 44
   2          (K)                                                               SMRHS 45
    GO TO 99                                                                    SMRHS 46
100 F(I,J,K)=(AV4(U,I,J,K,7,7,7)*(W(I+1,J,K)-W(I-1,J,K)))/(2.*DELT)+   SMRHS 47
   1       (AV4(V,I,J,K,7,7,7)*(W(I,J-1,K)-W(I,J,K)))/DELT+W(I,J,K)*X(KSMRHS 48
   2          )                                                                 SMRHS 49
 99 CONTINUE                                                                    SMRHS 50

    DO  7 I=1,14          $DO  7 J=1,14          $DO  8 K=1,7              SMRHS 51
    W(I,J,K)=W(I,J,K)*RO(K)                                                     SMRHS 52
  8 SP(K)=F(I,J,K)*RO(K)                                                        SMRHS 53
      CALL DDP(SP,X)                                                            SMRHS 54

    DO 7 K=1,7                                                                  SMRHS 55
  7 RHS(I,J,K)=-X(K)                                                            SMRHS 56
    N=0                                                                         SMRHS 57
      CALL EFF(U,V,AVVP,DELT,N,F,SP,X,W)                                        SMRHS 58

    DO  9 I=1,14          $DO  9 J=1,14          $DO  9 K=1,7              SMRHS 59
  9 RHS(I,J,K)=RHS(I,J,K)-((F(I+1,J,K)+F(I+1,J+1,K)-F(I,J,K)-F(I,J+1,KSMRHS 60
   1          ))/(2.*DELT))                                                     SMRHS 61
    N=1                                                                         SMRHS 62
      CALL EFF(U,V,AVVP,DELT,N,F,SP,X,W)                                        SMRHS 63

    DO 11 I=1,14          $DO 11 J=1,14          $DO 11 K=1,7              SMRHS 64
 11 RHS(I,J,K)=RHS(I,J,K)-((F(I+1,J,K)+F(I,J,K)-F(I+1,J+1,K)-F(I,J+1,KSMRHS 65
   1          ))/(2.*DELT))                                                     SMRHS 66
    RETURN                                                                      SMRHS 67
      END                                                                       SMRHS 68
```

```
      SUBROUTINE EFF(U,V,AVSP,DELT,N,F,SP,X,SP)                          EFF     1
C* * *     CALLED FROM SUBRHS                                          * * *EFF     2
      DIMENSION U(15,15,7),SP(14,14,7),SP(15),AVSP(16,16,7),           EFF     3
     1          V(15,15,7), F(15,15,7), X(15)                           EFF     4

      DO 51 K=1,7            SDO 47 I=1,13          SDO 47 J=1,13      EFF     5
   47 AVSP(I+1,J+1,K)=AV4(SP,I,J,K,14,14,7)                            EFF     6
      DO 48 I=2,14                                                     EFF     7
      AVSP(I,1,K)=AVSP(I,2,K)*2.-AVSP(I,3,K)                           EFF     8
      AVSP(I,15,K)=AVSP(I,14,K)*2.-AVSP(I,13,K)                        EFF     9
   48 CONTINUE                                                         EFF    10
      DO 49 J=2,14                                                     EFF    11
      AVSP(1,J,K)=AVSP(2,J,K)*2.-AVSP(3,J,K)                           EFF    12
      AVSP(15,J,K)=AVSP(14,J,K)*2.-AVSP(13,J,K)                        EFF    13
   49 CONTINUE                                                         EFF    14
      AVSP(1,1,K)=AVSP(1,2,K)*2.-AVSP(1,3,K)                           EFF    15
      AVSP(15,15,K)=AVSP(15,14,K)*2.-AVSP(15,13,K)                     EFF    16
      AVSP(1,15,K)=AVSP(1,14,K)*2.-AVSP(1,13,K)                        EFF    17
      AVSP(15,1,K)=AVSP(15,2,K)*2.-AVSP(15,3,K)                        EFF    18
   51 CONTINUE                                                         EFF    19
      IF(N.EQ.1) GO TO 1                                               EFF    20

      DO 99 I=1,15           SDO 99 J=1,15          SDO  5 K=1,7       EFF    21
    5 SP(K)=U(I,J,K)                                                   EFF    22
        CALL DDP(SP,X)                                                 EFF    23
      DO 99 K=1,7                                                      EFF    24
      IF(I.EQ.1) GO TO 20                                              EFF    25
      IF(I.EQ.15)GO TO 30                                              EFF    26
      GO TO 80                                                         EFF    27
   20 IF(J.EQ.1.OR.J.EQ.15)GO TO 30                                    EFF    28
      F(I,J,K)=(U(I,J,K)*(U(I+1,J,K)-U(I,J,K)))/DELT+(V(I,J,K)*(U(I,J-1,EFF    29
     1          K)-U(I,J+1,K)))/(2.*DELT)+AVSP(I,J,K)*X(K)             EFF    30
      GO TO 99                                                         EFF    31
```

```
 30 IF(J.EQ.15)GO TO 40                                                    EFF   32
    F(I,J,K)=(U(I,J,K)*(U(I+1,J,K)-U(I,J,K))+V(I,J,K)*(U(I,J,K)-U(I,J+EFF   33
   1          1,K)))/DELT+AVWP(I,J,K)*X(K)                                EFF   34
    GO TO 99                                                              EFF   35
 40 F(I,J,K)=(U(I,J,K)*(U(I+1,J,K)-U(I,J,K))+V(I,J,K)*(U(I,J-1,K)-U(I,EFF   36
   1          J,K)))/DELT+AVWP(I,J,K)*X(K)                                EFF   37
    GO TO 99                                                              EFF   38
 50 IF(J.EQ.1.OR.J.EQ.15)GO TO 60                                         EFF   39
    F(I,J,K)=(U(I,J,K)*(U(I,J,K)-U(I-1,J,K)))/DELT+(V(I,J,K)*(U(I,J-1,EFF   40
   1          K)-U(I,J+1,K)))/(2.*DELT)+AVWP(I,J,K)*X(K)                  EFF   41
    GO TO 99                                                              EFF   42
 60 IF(J.EQ.15)GO TO 70                                                   EFF   43
    F(I,J,K)=(U(I,J,K)*(U(I,J,K)-U(I-1,J,K))+V(I,J,K)*(U(I,J,K)-U(I,J+EFF   44
   1          1,K)))/DELT+AVWP(I,J,K)*X(K)                                EFF   45
    GO TO 99                                                              EFF   46
 70 F(I,J,K)=(U(I,J,K)*(U(I,J,K)-U(I-1,J,K))+V(I,J,K)*(U(I,J-1,K)-U(I,EFF   47
   1          J,K)))/DELT+AVWP(I,J,K)*X(K)                                EFF   48
    GO TO 99                                                              EFF   49
 80 IF(J.EQ.1.OR.J.EQ.15)GO TO 90                                         EFF   50
    F(I,J,K)=(U(I,J,K)*(U(I+1,J,K)-U(I-1,J,K))+V(I,J,K)*(U(I,J-1,K)-U(EFF   51
   1          I,J+1,K)))/(2.*DELT)+AVWP(I,J,K)*X(K)                       EFF   52
    GO TO 99                                                              EFF   53
 90 IF(J.EQ.15)GO TO 100                                                  EFF   54
    F(I,J,K)=(U(I,J,K)*(U(I+1,J,K)-U(I-1,J,K)))/(2.*DELT)+(V(I,J,K)*(UEFF   55
   1          (I,J,K)-U(I,J+1,K)))/DELT+AVWP(I,J,K)*X(K)                  EFF   56
    GO TO 99                                                              EFF   57
100 F(I,J,K)=(U(I,J,K)*(U(I+1,J,K)-U(I-1,J,K)))/(2.*DELT)+(V(I,J,K)*(UFFF   58
   1          (I,J-1,K)-U(I,J,K)))/DELT+AVWP(I,J,K)*X(K)                  EFF   59
 99 CONTINUE                                                              EFF   60
    GO TO 4                                                               EFF   61

  1 DO 199 I=1,15          $DO 199 J=1,15          $DO 15 K=1,7           EFF   62
 15 SP(K)=V(I,J,K)                                                        EFF   63
       CALL DDP(SP,X)                                                     EFF   64
    DO 199 K=1,7                                                          EFF   65
    IF(I.EQ.1) GO TO 120                                                  EFF   66
    IF(I.EQ.15)GO TO 150                                                  EFF   67
```

```
      GO TO 180                                                                 EFF    68
  120 IF(J.EQ.1.OR.J.EQ.15)GO TO 130                                            EFF    69
      F(I,J,K)=(U(I,J,K)*(V(I+1,J,K)-V(I,J,K)))/DELT+(V(I,J,K)*(V(I,J-1,EFF    70
     1          K)-V(I,J+1,K)))/(2.*DELT)+AVWP(I,J,K)*X(K)                      EFF    71
      GO TO 199                                                                 EFF    72
  130 IF(J.EQ.15)GO TO 140                                                      EFF    73
      F(I,J,K)=(U(I,J,K)*(V(I+1,J,K)-V(I,J,K))+V(I,J,K)*(V(I,J,K)-V(I,J+EFF    74
     1          1,K)))/DELT+AVWP(I,J,K)*X(K)                                    EFF    75
      GO TO 199                                                                 EFF    76
  140 F(I,J,K)=(U(I,J,K)*(V(I+1,J,K)-V(I,J,K))+V(I,J,K)*(V(I,J-1,K)-V(I,EFF    77
     1          J,K)))/DELT+AVWP(I,J,K)*X(K)                                    EFF    78
      GO TO 199                                                                 EFF    79
  150 IF(J.EQ.1.OR.J.EQ.15)GO TO 160                                            EFF    80
      F(I,J,K)=(U(I,J,K)*(V(I,J,K)-V(I-1,J,K)))/DELT+(V(I,J,K)*(V(I,J-1,EFF    81
     1          K)-V(I,J+1,K)))/(2.*DELT)+AVWP(I,J,K)*X(K)                      EFF    82
      GO TO 199                                                                 EFF    83
  160 IF(J.EQ.15)GO TO 170                                                      EFF    84
      F(I,J,K)=(U(I,J,K)*(V(1,J,K)-V(I-1,J,K))+V(I,J,K)*(V(I,J,K)-V(I,J+EFF    85
     1          1,K)))/DELT+AVWP(I,J,K)*X(K)                                    EFF    86
      GO TO 199                                                                 EFF    87
  170 F(I,J,K)=(U(I,J,K)*(V(I,J,K)-V(I-1,J,K))+V(I,J,K)*(V(I,J-1,K)-V(I,EFF    88
     1          J,K)))/DELT+AVWP(I,J,K)*X(K)                                    EFF    89
      GO TO 199                                                                 EFF    90
  180 IF(J.EQ.1.OR.J.EQ.15)GO TO 190                                            EFF    91
      F(I,J,K)=(U(I,J,K)*(V(I+1,J,K)-V(I-1,J,K))+V(I,J,K)*(V(I,J-1,K)-V(EFF    92
     1          I,J+1,K)))/(2.*DELT)+AVWP(I,J,K)*X(K)                           EFF    93
      GO TO 199                                                                 EFF    94
  190 IF(J.EQ.15)GO TO 1100                                                     EFF    95
      F(I,J,K)=(U(I,J,K)*(V(I+1,J,K)-V(I-1,J,K)))/(2.*DELT)+(V(I,J,K)*(VEFF    96
     1          (I,J,K)-V(I,J+1,K)))/DELT+AVWP(I,J,K)*X(K)                      EFF    97
      GO TO 199                                                                 EFF    98
 1100 F(I,J,K)=(U(I,J,K)*(V(I+1,J,K)-V(I-1,J,K)))/(2.*DELT)+(V(I,J,K)*(VEFF    99
     1          (I,J-1,K)-V(I,J,K)))/DELT+AVWP(I,J,K)*X(K)                      EFF   100
  199 CONTINUE                                                                  EFF   101
    4 RETURN                                                                    EFF   102
      END                                                                       EFF   103
```

```
      SUBROUTINE ADVRAIN(UR,VR,WP,RAIN,DELT,OUTFLO,VINCR,HINCR)          ADVRN   1
C* * *     COMPUTES ADVECTION OF RAINWATER BY THE WINDFIELD AND      * * *ADVRN   2
C* * *                  RAINFALL TO GROUND                           * * *ADVRN   3
      DIMENSION UR(15,15,7),HINCR(16,16,7),RAIN(16,16,7),OUTFLO(56,7),   ADVRN   4
     1          VR(15,15,7),VINCR(15,15,7),  WP(14,14,7),  DELP(7)       ADVRN   5
      LOGICAL   MARK,MARK1,OUTFLO                                        ADVRN   6
      COMMON    CCX,DM,KOUNT2,MARK                                       ADVRN   7
      DATA      DELP/0.5,0.5,1.,2.,2.,1.5,1.5/                           ADVRN   8
      MARK1=.TRUE.                                                       ADVRN   9

  206 DO 11 I=1,14          $DO 11 J=1,14          $DO 11 K=1,7          ADVRN  10
      IP1=I+1               $JP1=J+1               $IP2=I+2      $JP2=J+2 ADVRN  11
      KM1=K-1               $KP1=K+1                                     ADVRN  12
      IF (K.EQ.1) GO TO 202                                              ADVRN  13
      W1=AVPY4(UR,IP1,J,KM1,15,15,7)                                     ADVRN  14
      W5=AVPY4(UR,I,J,KM1,15,15,7)                                       ADVRN  15
      W2=AVPX4(VR,I,J,KM1,15,15,7)                                       ADVRN  16
      W6=AVPX4(VR,I,JP1,KM1,15,15,7)                                     ADVRN  17
      GO TO 203                                                          ADVRN  18
  202 W1=(UR(IP1,J,1)+UR(IP1,JP1,1))*0.25                                ADVRN  19
      W5=(UR(I,J,1)+UR(I,JP1,1))*0.25                                    ADVRN  20
      W2=(VR(I,J,1)+VR(IP1,J,1))*0.25                                    ADVRN  21
      W6=(VR(I,JP1,1)+VR(IP1,JP1,1))*0.25                                ADVRN  22
  203 CONTINUE                                                           ADVRN  23
```

```
      VINCR(I,J,K)=((W1+W5)*(RAIN(IP2,JP1,K)-RAIN(I,JP1,K))+(W2+W6)      ADVRN 24
     1          *(RAIN(IP1,J,K)-RAIN(IP1,JP2,K)))*0.25/DM                ADVRN 25
      IF(K.NE.1.AND.K.NE.7)VINCR(I,J,K)=VINCR(I,J,K)+WP(I,J,KM1)*(RAIN(IADVRN 26
     1          P1,JP1,K)-RAIN(IP1,JP1,KM1))/(DELP(KM1)+DELP(K))+WP(I,J,KADVRN 27
     2          )*(RAIN(IP1,JP1,KP1)-RAIN(IP1,JP1,K))/(DELP(KP1)+DELP(K))ADVRN 28
  11  IF(K.EQ.7)  VINCR(I,J,K)= VINCR(I,J,K)+(WP(I,J,7)+WP(I,J,6))*(RAINADVRN 29
     1          (IP1,JP1,7)-RAIN(IP1,JP1,6))/(DELP(6)+DELP(7))           ADVRN 30
      MARK1=.NOT.MARK1                                                   ADVRN 31
      IF(MARK1) GO TO 208                                                ADVRN 32
      IF(.NOT.MARK ) GO TO 208                                           ADVRN 33

      DO 205 I=2,15          $DO 205 J=2,15          $DO 205 K=1,7       ADVRN 34
      HINCR(I,J,K)=RAIN(I,J,K)                                           ADVRN 35
  205 RAIN(I,J,K)=RAIN(I,J,K)-DELT*VINCR(I-1,J-1,K)                      ADVRN 36
      GO TO 206                                                          ADVRN 37

  208 DO 207 I=2,15          $DO 207 J=2,15          $DO 207 K=1,7       ADVRN 38
      IF(.NOT.MARK) HINCR(I,J,K)=RAIN(I,J,K)                             ADVRN 39
      X=DELT*VINCR(I-1,J-1,K)                                            ADVRN 40
      IF(X.GT.HINCR (I,J,K)) X=HINCR(I,J,K)                              ADVRN 41
      RAIN(I,J,K)=HINCR(I,J,K)-X                                         ADVRN 42
  207 IF(K.EQ.7) RAIN(I,J,7)=RAIN(I,J,7)-5.32*RAIN(I,J,7)**              ADVRN 43
     1          1.2*DELT*1.5E-3                                          ADVRN 44
         CALL RFALL(DELT,RAIN)                                           ADVRN 45
      RETURN                                                             ADVRN 46
         END                                                             ADVRN 47
```

```
      SUBROUTINE PHYSIC(TH8,DELT,DQL,Q1,QL,RAIN)                          PHYS    1
C* * *     HANDLES VARIOUS PARTS OF CLOUD PHYSICS PARAMETERISATION  * * *PHYS    2
      DIMENSION Q1(16,16,7),TH8(16,16,7),RHS(7),RAIN(16,16,7),            PHYS    3
     1          QL(16,16,7),DQL(16,16,7),THS(7),    H(6)                 PHYS    4
      DATA       THS/424.,385.,353.,332.,318.,308.,304./                  PHYS    5
      DATA       H /1781.,2714.,3794.,2706.,1635.,1415./                  PHYS    6
      CRIT=0.5                                                            PHYS    7

      DO 11 I=2,15          $DO 11 J=2,15          $DO 16 K=2,7           PHYS    8
      RAIN(I,J,K-1)=ABS(RAIN(I,J,K-1))                                    PHYS    9
      RAIN(I,J,K)=ABS(RAIN(I,J,K))                                        PHYS   10
      P1=P2=P3=0.                                                         PHYS   11
         CALL SOLVQS(I,J,K,TH8,THS,QSAT)                                  PHYS   12
      QSAT=QSAT*1000.                                                     PHYS   13
      IF(Q1(I,J,K).LT.QSAT.AND.RAIN(I,J,K).GT.0..OR.QL(I,J,K).GT.CRIT     PHYS   14
     1          .AND.RAIN(I,J,K).GT.0..OR.QL(I,J,K).GT.CRIT)              PHYS   15
     2          CALL PRODUCT(I,J,K,P1,P2,P3,QSAT,Q1,QL,RAIN,DELT)         PHYS   16
      PROD=P1+P2+P3                                                       PHYS   17
      IF(K.EQ.7) GO TO 12                                                 PHYS   18
      RAIN(I,J,K+1)=ABS(RAIN(I,J,K+1))                                    PHYS   19
      RHS(K)=PROD+5.32*(RAIN(I,J,K-1)**1.2-RAIN(I,J,K+1)**1.2)/(H(K)+     PHYS   20
     1          H(K-1))                                                   PHYS   21
      GO TO 13                                                            PHYS   22
   12 RHS(K)=PROD+(((RAIN(I,J,K-1)+RAIN(I,J,K))*2.66)**1.2)/H(K-1)        PHYS   23
   13 RHS(K)=RHS(K)-5.32*RAIN(I,J,K)**1.2/7500.                           PHYS   24
      QL(I,J,K)=QL(I,J,K)-DELT*(P2+P3)                                    PHYS   25
      DQL(I,J,K)=DQL(I,J,K)-DELT*(P2+P3)                                  PHYS   26
      Q1(I,J,K)=Q1(I,J,K)-DELT*P1                                         PHYS   27
   16 TH8(I,J,K)=TH8(I,J,K)+DELT*P1*.597/0.24                             PHYS   28
      DO 11 K=2,7                                                         PHYS   29
   11 RAIN(I,J,K)=RAIN(I,J,K)+DELT*RHS(K)                                 PHYS   30
      RETURN                                                              PHYS   31
         END                                                              PHYS   32
```

```
      SUBROUTINE RFALL(DELT,RAIN)                                          RFALL   1
C* * *     COMPUTES RAINFALL RATES AND TOTALS                          * * *RFALL   2
      DIMENSION RAIN(16,16,7),RATE(16,16),TOTAL(16,16)                     RFALL   3
      DATA      KT/0/,KTT/0/                                               RFALL   4
      KT=KT+1                 $KTT=KTT+1                                   RFALL   5
      IF(KT.NE.1) GO TO 40                                                 RFALL   6

      DO 33 I=1,16            $DO 33 J=1,16                                RFALL   7
      RATE(I,J)=0.                                                         RFALL   8
   33 TOTAL(I,J)=0.                                                        RFALL   9
   40 CONTINUE                                                             RFALL  10

      DO 22 I=2,15            $DO 22 J=2,15                                RFALL  11
      RATE(I,J)=RAIN(I,J,7)**1.2*19.2                                      RFALL  12
   22 TOTAL(I,J)=TOTAL(I,J)+RATE(I,J)*DELT/3600.                           RFALL  13

      IF(KTT.NE.50) GO TO 88                                               RFALL  14
      WRITE(6,44)                                                          RFALL  15
      WRITE(6,55) ((RATE(I,J),I=1,16),J=1,16)                              RFALL  16
      WRITE(6,66)                                                          RFALL  17
      WRITE(6,55) ((TOTAL(I,J),I=1,16),J=1,16)                             RFALL  18
   88 CONTINUE                                                             RFALL  19
      IF(KTT.EQ.50) KTT=0                                                  RFALL  20
      RETURN                                                               RFALL  21
   44 FORMAT(1HO,*RAINFALL RATES AND TOTALS IN MM/HR AND MM. *,//)         RFALL  22
   55 FORMAT((1X,16(F 7.3,1X)))                                            RFALL  23
   66 FORMAT(1HO,*  TOTAL*,/)                                              RFALL  24
      END                                                                  RFALL  25
```

```
      SUBROUTINE PRODUCT(I,J,K,P1,P2,P3,QSAT,QI,QL,RAIN,DELT)                PROD    1
C* * *    COMPUTES PRODUCTION TERMS IN CLOUD PHYSICS PARAMETERISATION* *PROD    2
      DIMENSION QI(16,16,7),QL(16,16,7),RAIN(16,16,7)                        PROD    3
      AL1=0.002              $AL2=0.001              $BETA=0.001    $CRIT=0.5PROD    4
      IF(QI(I,J,K).LT.QSAT.AND.RAIN(I,J,K).GT.0.) P1=BETA*(QI(I,J,K)-QSAPROD    5
     1          T)                                                           PROD    6
      P=-P1*DELT                                                             PROD    7
      IF(P.GT.RAIN(I,J,K)) P1=-RAIN(I,J,K)/DELT                              PROD    8
      P2=AL1*QL(I,J,K)*RAIN(I,J,K)**0.95                                     PROD    9
      IF(QL(I,J,K).GT.CRIT) P3=AL2*(QL(I,J,K)-CRIT)                          PROD   10
      RETURN                                                                 PROD   11
         END                                                                 PROD   12
```

```
      SUBROUTINE SOLVQS(I,J,K,TH8,THS,QSAT)                                  SLVQS   1
C* * *    COMPUTES SATURATED Q USING TETENS FORMULA                  * * *SLVQS   2
      DIMENSION TH8(16,16,7),THS(7),P0(7)                                    SLVQS   3
      DATA      P0/125.,175.,250.,400.,600.,775.,925./                       SLVQS   4
      A1=((P0(K))/1000.)**0.286                                              SLVQS   5
      A2=(TH8(I,J,K)+THS(K))*A1                                              SLVQS   6
      IF(A2.LT.273.2) A=9.5                                                  SLVQS   7
      IF(A2.LT.273.2) B=265.5                                                SLVQS   8
      IF(A2.GE.273.2 ) A=7.5                                                 SLVQS   9
      IF(A2.GE.273.2) B=237.5                                                SLVQS  10
      A4=A*(A2-273.2)/(B+A2-273.2)                                           SLVQS  11
      A5=3.8*(10.**A4)                                                       SLVQS  12
      QSAT=A5/(P0(K)-A5/0.622)                                               SLVQS  13
      RETURN                                                                 SLVQS  14
         END                                                                 SLVQS  15
```

```
      SUBROUTINE SLVTOF(Q1,TH8,QL,THINCR,QINCR,DQL)                          SLVTF   1
C* * *     COMPUTES CONDENSATION AND EVAPORATION OF CLOUD WATER       * * *SLVTF   2
      DIMENSION QINCR(16,16,7),TH2(16,16,7),Q1(16,16,7),THS(7),              SLVTF   3
     1          THINCR(16,16,7),DQL(16,16,7),QL(16,16,7), P0(7)             SLVTF   4
      DATA       THS/424.,385.,353.,332.,318.,308.,304./                     SLVTF   5
      DATA       P0/125.,175.,250.,400.,600.,775.,925./                      SLVTF   6

      DO 99 I=2,15          DO 99 J=2,15          DO 99 K=1,7                 SLVTF   7
      DQL(I,J,K)=0.                                                         SLVTF   8
         CALL SOLVQS(I,J,K,TH2,THS,QSAT)                                    SLVTF   9
      Q1(I,J,K)=Q1(I,J,K)*0.001                                             SLVTF  10
      IF((Q1(I,J,K)-QSAT).LE.0.) GO TO 96                                   SLVTF  11
      Z=TH8(I,J,K)                                                          SLVTF  12
      D= (P0(K)/1000.)**0.286                                               SLVTF  13
      TH8(I,J,K)=TH8(I,J,K)+THS(K)                                          SLVTF  14
      A=597.*QSAT/(D*(TH8(I,J,K)**2)*0.11)                                  SLVTF  15
      B=A+0.24*D/597.                                                       SLVTF  16
      C=A/QSAT-2./TH8(I,J,K)                                                SLVTF  17
      F=SQRT(B**2+2.*A*C*(Q1(I,J,K)-QSAT))                                  SLVTF  18
      TH2(I,J,K)=TH8(I,J,K)-B/(A*C)+F/(A*C)-THS(K)                          SLVTF  19
      THINCR(I,J,K)=TH8(I,J,K)-Z+THINCR(I,J,K)                              SLVTF  20
         CALL SOLVQS(I,J,K,TH8,THS,QSAT)                                    SLVTF  21
      G=Q1(I,J,K)-QSAT                                                      SLVTF  22
      QINCR(I,J,K)=QINCR(I,J,K)-G*1000.                                     SLVTF  23
      IF(Q1(I,J,K).GT.QSAT) Q1(I,J,K)=QSAT                                  SLVTF  24
      QL(I,J,K)=QL(I,J,K)+G*1000.                                           SLVTF  25
      DQL(I,J,K)=G*1000.                                                    SLVTF  26
      Q1(I,J,K)=Q1(I,J,K)*1000.                                             SLVTF  27
      GO TO 99                                                              SLVTF  28
   96 IF(QL(I,J,K).LE.0.) GO TO 98                                          SLVTF  29
```

```
      Z=TH8(I,J,K)                                                      SLVTF  30
      D=(PU(K)/1000.)**0.286                                            SLVTF  31
      TH8(I,J,K)=TH8(I,J,K)+THS(K)                                      SLVTF  32
      A=597.*QSAT/(D*(TH8(I,J,K)**2)*0.11)                              SLVTF  33
      B=A+0.24*D/597.                                                   SLVTF  34
      C=A/QSAT-2./TH8(I,J,K)                                            SLVTF  35
      F=SQRT(B**2+2.*A*C*(Q1(I,J,K)-QSAT))                              SLVTF  36
      TH8(I,J,K)=TH8(I,J,K)-B/(A*C)+F/(A*C)-THS(K)                      SLVTF  37
         CALL SOLVQS(I,J,K,TH8,THS,QSAT)                                SLVTF  38
      G=Q1(I,J,K)-QSAT                                                  SLVTF  39
      IF(ABS(G).GT.QL(I,J,K)*0.001) GO TO 97                            SLVTF  40
      THINCR(I,J,K)=TH8(I,J,K)-Z+THINCR(I,J,K)                          SLVTF  41
      QINCR(I,J,K)=QINCR(I,J,K)-G*1000.                                 SLVTF  42
      Q1(I,J,K)=QSAT                                                    SLVTF  43
      QL(I,J,K)=QL(I,J,K)+G*1000.                                       SLVTF  44
      DQL(I,J,K)=G*1000.                                                SLVTF  45
      Q1(I,J,K)=Q1(I,J,K)*1000.                                         SLVTF  46
      GO TO 99                                                          SLVTF  47
   97 TH8(I,J,K)=Z                                                      SLVTF  48
      THINCR(I,J,K)=THINCR(I,J,K)-2.5*QL(I,J,K)                         SLVTF  49
      TH8(I,J,K)=TH8(I,J,K)-2.5*QL(I,J,K)                               SLVTF  50
      QINCR(I,J,K)=QINCR(I,J,K)+QL(I,J,K)                               SLVTF  51
      Q1(I,J,K)=Q1(I,J,K)*1000.+QL(I,J,K)                               SLVTF  52
      QL(I,J,K)=0.                                                      SLVTF  53
      DQL(I,J,K)=0.                                                     SLVTF  54
      GO TO 99                                                          SLVTF  55
   98 Q1(I,J,K)=Q1(I,J,K)*1000.                                         SLVTF  56
   99 CONTINUE                                                          SLVTF  57
      RETURN                                                            SLVTF  58
         END                                                            SLVTF  59
```

```
      SUBROUTINE ***(***,UR,VR,HINCR,HEIGHT,DP,D**,DELT)                    ***     1
C*  *  *     COMPUTES VERTICAL VELOCITY FROM PRESSURE TENDENCY          *  *  ***     2
      DIMENSION UR(15,15,7),HEIGHT(16,16,7),**(15,15,7),RO(7),              ***     3
     1          VR(15,15,7), HINCR(16,16,7),DP(14,14,7)                    ***     4
      DATA      RO/2.54,,920,,434,,653,,862,,1013,,1152,/                   ***     5

      DO 55 I=1,15          DO 55 J=1,15          DO 55 K=1,7               ***     6
      AV*PK=AV4(DP,I-1,J-1,K,14,14,7)                                       ***     7
      IF(I.EQ.1.AND.J.NE.1.AND.J.NE.15)  AV*PK=(DP(I,J,K)+DP(I,J-1,K))      ***     8
     1      *0.5                                                           ***     9
      IF(I.EQ.15.AND.J.NE.1.AND.J.NE.15)AV*PK=(DP(I-1,J,K)+DP(I-1,J-1,K)    ***    10
     1      )*0.5                                                          ***    11
      IF(J.EQ.1.AND.I.NE.1.AND.I.NE.15)AV*PK=(DP(I,J,K)+DP(I-1,J,K))*0.5    ***    12
      IF(J.EQ.15.AND.I.NE.1.AND.I.NE.15)AV*PK=(DP(I,J-1,K)+DP(I-1,J-1,K)    ***    13
     1      )*0.5                                                          ***    14
      IF(I.EQ.1.AND.J.EQ.1)  AV*PK=DP(I,J,K)                                ***    15
      IF(I.EQ.1.AND.J.EQ.15)  AV*PK=DP(I,J-1,K)                             ***    16
      IF(I.EQ.15.AND.J.EQ.1)  AV*PK=DP(I-1,J,K)                             ***    17
      IF(I.EQ.15.AND.J.EQ.15)  AV*PK=DP(I-1,J-1,K)                          ***    18
      A1=AV4(HINCR,I,J,K,16,16,7)/DELT                                      ***    19
      A2=UR(I,J,K)*(HEIGHT(I+1,J,K)+HEIGHT(I+1,J+1,K)-HEIGHT(I,J,K)         ***    20
     1      -HEIGHT(I,J+1,K))/(2.*DX)                                      ***    21
      A3=VR(I,J,K)*(HEIGHT(I,J,K)+HEIGHT(I+1,J,K)-HEIGHT(I,J+1,K)           ***    22
     1      -HEIGHT(I+1,J+1,K))/(2.*DY)                                    ***    23
      A4=AV*PK*10.**7/(2.51*PC(K))                                          ***    24
   55 **(I,J,K)=A1+A2+A3-A4                                                 ***    25
      RETURN                                                                ***    26
         END                                                                ***    27
```

```
      SUBROUTINE OUT15 (Y,IDIM,JDIM,KDIM,K1)                                  OUT     1
C* * *    PART OF DATA OUTPUT                                          * * *OUT     2
      DIMENSION Y(IDIM,JDIM,KDIM)                                             OUT     3
      KSTART=1              $KFIN=KDIM                                        OUT     4
      IF(K1.NE.0) KSTART=K1                                                   OUT     5
      IF(K1.NE.0) KFIN=K1                                                     OUT     6

      DO 999 K=KSTART,KFIN $DO 146 I=1,IDIM      $DO 146 J=1,JDIM             OUT     7
  146 IF(Y(I,J,K).NE.0.) GO TO 147                                            OUT     8
      WRITE(6,2000) K                                                         OUT     9
      GO TO 999                                                               OUT    10
  147 BIGVAL=ABS(Y(1,1,K))                                                    OUT    11

      DO 1 I=1,IDIM         $DO 1 J=1,JDIM                                    OUT    12
    1 IF (ABS(Y(I,J,K)).GT.BIGVAL) BIGVAL=ABS(Y(I,J,K))                       OUT    13
      IF (BIGVAL.LT.1.0) GO TO 10                                             OUT    14
      IF (BIGVAL.GE.1.0.AND.BIGVAL.LT.10.)GO TO 12                            OUT    15

      DO 2 N=1,20                                                             OUT    16
      BIGVAL=0.1*BIGVAL                                                       OUT    17
      IF (BIGVAL.GE.1.0.AND.BIGVAL.LT.10.)GO TO 20                            OUT    18
    2 CONTINUE                                                                OUT    19
   20 XN=0.1**N                                                               OUT    20

      DO 100 I=1,IDIM       $DO 100 J=1,JDIM                                  OUT    21
  100 Y(I,J,K)=Y(I,J,K)*XN                                                    OUT    22
      N=-N                                                                    OUT    23
      GO TO 1000                                                              OUT    24

   10 DO 3 N=1,20                                                             OUT    25
      BIGVAL=10.*BIGVAL                                                       OUT    26
      IF(BIGVAL.GE.1.0.AND.BIGVAL.LT.10.) GO TO 30                            OUT    27
    3 CONTINUE                                                                OUT    28
   30 XN=10.**N                                                               OUT    29
```

```
      DO 200 I=1,IDIM        $DO 200 J=1,JDIM                    OUT    30
  200 Y(I,J,K)=Y(I,J,K)*XN                                       OUT    31
      GO TO 1000                                                 OUT    32
   12 N=0                                                        OUT    33
 1000 IF (KDIM.EQ.1) GO TO 70                                    OUT    34
      WRITE (6,110) K, N                                         OUT    35
      GO TO 80                                                   OUT    36
   70 WRITE (6,111) N                                            OUT    37
   80 IF (IDIM.EQ.15) GO TO 220                                  OUT    38
      IF (IDIM.EQ.16) GO TO 230                                  OUT    39
      WRITE (6,123) (I,I=1,14)                                   OUT    40

      DO 7 J=1,14                                                OUT    41
      WRITE (6,550) J, (Y(I,J,K),I=1,14), J                      OUT    42
    7 CONTINUE                                                   OUT    43
      WRITE (6,123) (I,I=1,14)                                   OUT    44
      GO TO 998                                                  OUT    45
  220 WRITE (6,122) (I,I=1,15)                                   OUT    46
      DO 6 J=1,15                                                OUT    47
      WRITE (6,500) J, (Y(I,J,K),I=1,15), J                      OUT    48
    6 CONTINUE                                                   OUT    49
      WRITE (6,122) (I,I=1,15)                                   OUT    50
      GO TO 998                                                  OUT    51
  230 WRITE (6,124) (I,I=1,16)                                   OUT    52
      DO 8 J=1,16                                                OUT    53
      WRITE (6,600) J, (Y(I,J,K),I=1,16), J                      OUT    54
    8 CONTINUE                                                   OUT    55
      WRITE (6,124) (I,I=1,16)                                   OUT    56
  998 CONTINUE                                                   OUT    57
```

```
      IF(N.EQ.0) GO TO 999                                              OUT   58
      IF (N.LT.0) GO TO 132                                             OUT   59

      DO 130 I=1,IDIM       $DO 130 J=1,JDIM                            OUT   60
  130 Y(I,J,K)=Y(I,J,K)/XN                                              OUT   61
      GO TO 999                                                         OUT   62
  132 N=-N                                                              OUT   63

      DO 131 I=1,IDIM       $DO 131 J=1,JDIM                            OUT   64
  131 Y(I,J,K)=Y(I,J,K)/XN                                              OUT   65
  999 CONTINUE                                                          OUT   66
      RETURN                                                            OUT   67
  110 FORMAT (10X,5HLEVEL,I3,87X,19HSCALING FACTOR 10**,I3)             OUT   68
  111 FORMAT (105X,19HSCALING FACTOR 10**,I3)                           OUT   69
  122 FORMAT (24X,15(I2,4X))                                            OUT   70
  123 FORMAT (27X,14(I2,4X))                                            OUT   71
  124 FORMAT (21X,16(I2,4X))                                            OUT   72
  500 FORMAT (18X,I2,2X,15F6.2,2X,I2)                                   OUT   73
  550 FORMAT (21X,I2,2X,14F6.2,2X,I2)                                   OUT   74
  600 FORMAT (15X,I2,2X,16F6.2,2X,I2)                                   OUT   75
 2000 FORMAT(10X,5HLEVEL,I3,/20X,8HALL ZERO,/)                          OUT   76
        END                                                             OUT   77
```

```
      SUBROUTINE OUT                                                    OUT/    1
      COMMON /BLK1 / U15(15,15,7),V15(15,15,7),WP14(14,14,7)            OUT/    2
      COMMON /BLK2 / Q16(16,16,7),TH16(16,16,7)                         OUT/    3
      COMMON /BLK3 / ZETA14(14,14,7)                                    OUT/    4
      COMMON /BLK6 / HITE16(16,16,7)                                    OUT/    5
      COMMON /BLK13/ OUTFLO(96,7)                                       OUT/    6
      COMMON /BLK21/ HDIVP(14,14,7),QL(16,16,7)                         OUT/    7
      COMMON CCX,DT,KOUNT2,MARK,KOUNT1,DELT,IND                         OUT/    8
      LOGICAL OUTFLO                                                    OUT/    9
      TIME=DELT*FLOAT(KOUNT2)/60.                                       OUT/   10
      WRITE (6,10) TIME                                                 OUT/   11
      WRITE (6,50)                                                      OUT/   12
         CALL OUT15 (U15,15,15,7,0)                                     OUT/   13
      WRITE (6,52)                                                      OUT/   14
         CALL OUT15 (WP14,14,14,7,0)                                    OUT/   15
      WRITE (6,57)                                                      OUT/   16
         CALL OUT15 (TH16,16,16,7,0)                                    OUT/   17
      WRITE (6,60)                                                      OUT/   18
         CALL OUT15(HITE16,16,16,7,0)                                   OUT/   19
      WRITE(6,54)                                                       OUT/   20
         CALL OUT15(QL,16,16,7,0)                                       OUT/   21
      IF (IND.EQ.0) GO TO 3                                             OUT/   22
    3 WRITE (6,56)                                                      OUT/   23
      DO 58 K=4,7                                                       OUT/   24
   58    CALL OUT15(Q16,16,16,7,K)                                      OUT/   25
         CALL BDYIND (OUTFLO)                                           OUT/   26
      RETURN                                                            OUT/   27
   10 FORMAT (8H1 TIME =,F7.1,6H MINUTES)                               OUT/   28
   50 FORMAT (2H0U)                                                     OUT/   29
   52 FORMAT (3H0WP)                                                    OUT/   30
   54 FORMAT(3H0QL)                                                     OUT/   31
   56 FORMAT (2H0Q)                                                     OUT/   32
   57 FORMAT (3H0TH)                                                    OUT/   33
   60 FORMAT (7H0HEIGHT)                                                OUT/   34
         END                                                            OUT/   35
```

```
      SUBROUTINE EXTONF(B,BY)                                          EXTO    1
      DIMENSION B(16,16),BY(56)                                        EXTO    2
      DO 2 I=2,15                                                      EXTO    3
      L=44-I                                                           EXTO    4
      B(I,1)=-B(I,2)+BY(I-1)+BY(I)                                     EXTO    5
    2 B(I,16)=-B(I,15)+BY(L+1)+BY(L)                                   EXTO    6
      DO 3 J=2,15                                                      EXTO    7
      L=58-J                                                           EXTO    8
      B(16,J)=-B(15,J)+BY(J+13)+BY(J+14)                               EXTO    9
      IF (J.EQ.2) GO TO 4                                              EXTO   10
      B(1,J)=-B(2,J)+BY(L+1)+BY(L)                                     EXTO   11
      GO TO 3                                                          EXTO   12
    4 B(1,J)=-B(2,J)+BY(1)+BY(L)                                       EXTO   13
    3 CONTINUE                                                         EXTO   14
      B(1,1)=-B(2,2)+B(1,2)+B(2,1)                                     EXTO   15
      B(16,16)=-B(15,15)+B(15,16)+B(16,15)                             EXTO   16
      B(1,16)=-B(2,15)+B(1,15)+B(2,16)                                 EXTO   17
      B(16,1)=-B(15,2)+B(15,1)+B(16,2)                                 EXTO   18
      RETURN                                                           EXTO   19
         END                                                           EXTO   20
```

```
      SUBROUTINE DDP(PHI,DPHI)                                         DDP.    1
      DIMENSION PHI(7),DPHI(7)                                         DDP.    2
      DPHI(7)=2.*(PHI(7)-PHI(6))/3.                                    DDP.    3
      DPHI(6)=(PHI(7)-PHI(5))/3.                                       DDP.    4
      DPHI(5)=(16.*PHI(6)-9.*PHI(4)-7.*PHI(5))/42.                     DDP.    5
      DPHI(4)=(PHI(5)-PHI(3))/4.                                       DDP.    6
      DPHI(3)=(PHI(4)-4.*PHI(2)+3.*PHI(3))/6.                          DDP.    7
      DPHI(2)=(PHI(3)-4.*PHI(1)+3.*PHI(2))/3.                          DDP.    8
      DPHI(1)=(PHI(2)-PHI(1))*2.                                       DDP.    9
      RETURN                                                           DDP.   10
         END                                                           DDP.   11
```

```
      SUBROUTINE EXTEND (X,Y,K)                                          EXTE    1
      DIMENSION X(56),Y(16,16,7)                                         EXTE    2
      DO 2 I=2,15                                                        EXTE    3
      L=44-I                                                             EXTE    4
      Y(I,1,K)=-Y(I,2,K)+X(I-1) +X(I)                                    EXTE    5
      Y(I,16,K)=-Y(I,15,K)+X(L+1)+X(L)                                   EXTE    6
      L=58-I                                                             EXTE    7
      Y(16,I,K)=-Y(15,I,K)+X(I+13)+X(I+14)                               EXTE    8
      IF (I.EQ.2) GO TO 3                                                EXTE    9
      Y(1,I,K)=-Y(2,I,K)+X(L+1)+X(L)                                     EXTE   10
      GO TO 2                                                            EXTE   11
    3 Y(1,I,K)=-Y(2,I,K)+X(L)+X(1)                                       EXTE   12
    2 CONTINUE                                                           EXTE   13
      Y(1,1,K)=-Y(2,2,K)+Y(1,2,K)+Y(2,1,K)                               EXTE   14
      Y(16,16,K)=-Y(15,15,K)+Y(15,16,K)+Y(16,15,K)                       EXTE   15
      Y(1,16,K)=-Y(2,15,K)+Y(1,15,K)+Y(2,16,K)                           EXTE   16
      Y(16,1,K)=-Y(15,2,K)+Y(15,1,K)+Y(16,2,K)                           EXTE   17
      RETURN                                                             EXTE   18
         END                                                             EXTE   19
```

```
      SUBROUTINE BDYIND (OUTFLO)                                         BDYN    1
      DIMENSION OUTFLO(56,7), ALINE(56)                                  BDYN    2
      LOGICAL   OUTFLO                                                   BDYN    3
      DATA      AI,AO/1HI,1HO/                                           BDYN    4
      WRITE (6,110)                                                      BDYN    5
      DO 10 K=1,7            $DO 5 I=1,56                                BDYN    6
      ALINE(1)=AI                                                        BDYN    7
    5 IF (OUTFLO(I,K)) ALINE(I)=AO                                       BDYN    8
   10 WRITE (6,100) K,ALINE                                              BDYN    9
      RETURN                                                             BDYN   10
  100 FORMAT (8H  LEVEL ,I1,8X,4(A1,1X,13A1,1X))                         BDYN   11
  110 FORMAT (36H0 BOUNDARY INFLOW/OUTFLOW CONDITIONS,/25X,1HN,15X,1HE,  BDYN   12
     1          15X,1HS,15X,1HW)                                        BDYN   13
         END                                                             BDYN   14
```

```
      FUNCTION AVPX4(X,I,J,K,I1,J1,K1)                                  AVPX    1
      DIMENSION X(I1,J1,K1)                                             AVPX    2
      AVPX4=(X(I,J,K)+X(I+1,J,K)+X(I,J,K+1)+X(I+1,J,K+1))*0.25          AVPX    3
      RETURN                                                            AVPX    4
          END                                                           AVPX    5


      FUNCTION AVPY4(X,I,J,K,I1,J1,K1)                                  AVPY    1
      DIMENSION X(I1,J1,K1)                                             AVPY    2
      AVPY4=(X(I,J,K)+X(I,J+1,K)+X(I,J,K+1)+X(I,J+1,K+1))*0.25          AVPY    3
      RETURN                                                            AVPY    4
          END                                                           AVPY    5


      FUNCTION ALAPL(X,I,J,K,I1,J1,K1)                                  ALAP    1
      DIMENSION X(I1,J1,K1)                                             ALAP    2
      ALAPL=X(I+1,J,K)+X(I-1,J,K)+X(I,J+1,K)+X(I,J-1,K)-4.*X(I,J,K)     ALAP    3
      RETURN                                                            ALAP    4
          END                                                           ALAP    5


      FUNCTION AV4(X,I,J,K,I1,J1,K1)                                    AV4.    1
      DIMENSION X(I1,J1,K1)                                             AV4.    2
      AV4=(X(I,J,K)+X(I,J+1,K)+X(I+1,J,K)+X(I+1,J+1,K))*0.25            AV4.    3
      RETURN                                                            AV4.    4
          END                                                           AV4.    5
```

REFERENCES

Arnasson, G., et al.	1968	'A numerical experiment in dry and moist convection including the rain stage', J.Atmos.Sci. 25 p.404.
Browning, K.A.	1965	'A family outbreak of severe local storms', Air Force Cambridge Research Labs. - Special Report No.32.
Browning, K.A. and Ludlam, F.H.	1962	'Airflow in convective storms', Quart.J.R.Met.Soc. 88 p.117.
Bushby, F.H. and Timpson, M.S.	1967	'A 10-level atmospheric model and frontal rain', Quart.J.R.Met.Soc. 93 p.1.
Byers, H.R. and Braham, R.R.	1949	'The thunderstorm', Washington D.C. Govt. Printing Office.
Charney, J.G.	1955	'The use of the primitive equations of motion in numerical weather prediction', Tellus 7 p.22.
Das, P.	1964	'Role of condensed water in the life-cycle of a convective cloud', J.Atmos.Sci. 21 p.404.
Green, J.S.A.	1969	Advanced dynamics lecture notes: unpublished. Imperial College, London.
Green, J.S.A. and Pearce, R.P.	1962	'Cumulonimbus convection in shear', Tech.Note 12, Dept.Meteor., Imperial College, London.
Hinkelmann, K.	1959	'Ein numerisches Experiment mit den primitiven Gleichungen', "The atmosphere and the sea in motion; Scientific Contributions to the Rossby Memorial Volume", Rockefeller Inst. Press; New York p.486.
Jordan, C.L.	1958	'Mean soundings for the W.Indies area', J.Met. 15 p.91.
Kessler, E.	1969	'On the distribution and continuity of water substance in atmospheric circulations', Met.Mon. Amer.Met.Soc. 10 No.32.
Kwizak, M. and Robert, A.	1971	'A semi-implicit scheme for grid point atmospheric models of the primitive equations', Mon.Weath.Rev. 99 p.32.
Lilly, D.L.	1962	'On the numerical simulation of buoyant convection', Tellus 14 p.148.
Lilly, D.L.	1964	'Numerical solutions for the shape-preserving two-dimensional thermal convective element', J.Atmos.Sci. 21 p.83.

Liu, J.Y. and Orville, H.D. 1969 'Numerical modelling of precipitation and cloud shadow effects on mountain-induced cumuli', J.Atmos.Sci. 26 p.1283.

Ludlam, F.H. 1963 'Severe local storms: a review', Met.Mon. Amer.Met.Soc. 5 No.27.

Malkus, J.S. and Witt, G. 1959 'The evolution of a convective element: a numerical calculation', "The atmosphere and the sea in motion; Rossby Memorial Volume", Rockefeller Inst.Press; New York p.425.

Miyakoda, K. 1960 'Test of convergence speed of iterative methods for solving 2 and 3 dimensional elliptic-type differential equations', J.Met.Soc.Japan 38 p.107.

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

Moncrieff, M.W. and Green, J.S.A. 1972 'The propagation and transfer properties of steady convective overturning in shear', Quart.J.R.Met.Soc. 99

Murray, F.W. 1970 'Numerical models of a tropical cumulus cloud with bilateral and axial symmetry', Mon.Weath.Rev. 98 p.14.

Murray, F.W. and Anderson, C.E. 1965 'Numerical simulation of the evolution of cumulus towers', Report SM-49230 Douglas Aircraft Comp., Inc.

Newton, C.W. and Newton, H.R. 1959 'Dynamical interactions between large convective clouds and the environment with vertical shear', J.Met. 16 p.483.

Nickerson, E.C. 1965 'A numerical experiment in buoyant convection including the use of a heat source', J.Atmos.Sci. 22 p.412.

Ogura, Y. 1962 'Convection of isolated masses of buoyant fluid: a numerical calculation', J.Atmos.Sci. 19 p.492.

Ogura, Y. 1963 'The evolution of a moist convective element in a shallow, conditionally unstable atmosphere: a numerical calculation', J.Atmos.Sci. 20 p.407.

Ogura, Y. and Charney, J.G. 1962 'A numerical model of thermal convection in the atmosphere', Proc.International Symposium on Numerical Weather Prediction, Tokyo, Japan, Nov.7-13 1960. Met.Soc. of Japan, Tokyo, Mar.1962 p.431.

Ogura, Y. and Phillips, N.A. 1962 'Scale analysis of deep and shallow convection in the atmosphere', J.Atmos.Sci. 19 p.173.

Orville, H.D.	1965	'A numerical study of the initiation of cumulus clouds over mountainous terrain', J.Atmos.Sci. 22 p.684.
Orville, H.D. and Sloan, L.J.	1970	'A numerical simulation of the life history of a rainstorm', J.Atmos.Sci. 27 p.1148.
Pearce, R.P.	1972	(To be published).
Shapiro, R.	1970	'Smoothing, filtering and boundary effects', Rev.Geophys.Space Phys. 8 p.359.
Shapiro, R.	1971	'The use of linear filtering as a parameterisation of atmospheric diffusion', J.Atmos.Sci. 28 p.523.
Smagorinsky, J.	1958	'On the numerical integration of the primitive equations of motion for baroclinic flow in a closed region', Mon.Weath.Rev. 86 p.457.
Smagorinsky, J.	1963	'General circulation experiments with the primitive equations', Mon.Weath.Rev. 91 p.99.
Srivastava, R.C.	1967	'A study of the effect of precipitation on cumulus dynamics', J.Atmos.Sci. 24 p.36.
Takeda, T.	1966	'The downdraft in the convective cloud and raindrops: a numerical computation', J.Met.Soc.Japan 41 p.1.
Takeda, T.	1971	'Numerical simulation of a precipitating convective cloud: the formation of a long-lasting cloud', J.Atmos.Sci. 28 p.350.
Weinstein, A.I.	1968	Reports 11, 13: Pennsylvania State University.

ACKNOWLEDGEMENTS

The author is very much indebted to Professor R.P. Pearce (now of Reading University) for his guidance and continual encouragement throughout. Thanks are also expressed to Mr. Robin Drayton and Miss Hilary Cox for valuable assistance with the computer programs, and to Miss Marion Street for her care and patience in typing this thesis.

This work was supported by a grant from the Meteorological Office, Bracknell.