NASA Contractor Report 3385

An Efficient Code for the Simulation of Nonhydrostatic Stratified Flow Over Obstacles

Gregory G. Pihos and Morton G. Wurtele

GRANTS NSG-4001 and NSG-4024
APRIL 1981

NASA

NASA Contractor Report 3385

An Efficient Code for the Simulation of Nonhydrostatic Stratified Flow Over Obstacles

Gregory G. Pihos and Morton G. Wurtele
University of California
Los Angeles, California

Prepared for
Dryden Flight Research Center
under Grants NSG-4001 and NSG-4024

National Aeronautics
and Space Administration

**Scientific and Technical
Information Branch**

1981

ACKNOWLEDGEMENTS

The authors wish to thank Mr. L.J. Ehernberger of the NASA Dryden Flight Research Center for his continuing support, and for his comments on earlier versions of this paper. We also express gratitude to Dr. Robert Sharman for useful technical discussions, and to several reviewers for their criticism.

This work was supported under NASA contracts NSG 4001 and NSG 4024. Computations were performed on the facilities of the UCLA Office of Academic Computing.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS

a	Amplitude of flow disturbance at the lower boundary
b	Half the wavelength of the disturbance at the lower boundary
c	Reciprocal of scale height of undisturbed flow, $\frac{1}{u_0}\frac{\partial u}{\partial z}$
c_1, c_2, c_3	Arbitrary constants to be determined by boundary conditions
c_p	Specific heat capacity at constant pressure
c_s	Local sound speed
c_v	Specific heat capacity at constant volume
d	Dimensionless terrain height function
f	Coriolis parameter, $2\Omega \sin\phi$
F	$\frac{gS}{\bar{u}^2} + \frac{s}{\bar{u}}\frac{\partial \bar{u}}{\partial z} - \frac{1}{\bar{u}}\frac{\partial^2 \bar{u}}{\partial z^2} - \frac{1}{4}s^2 + \frac{1}{2}\frac{\partial s}{\partial z}$
$\vec{g}, g$	Acceleration of gravity and its magnitude
G	Dimensionless streamline function
h	Terrain height function
H	Total height of the fluid
$\vec{i}, \vec{j}, \vec{k}$	Unit coordinate vectors in the x,y, and z directions
J	Jacobian operator
J_ν	Bessel function of the first kind
k	Wavenumber in the x direction
$K_{i\mu}$	Bessel function of the third kind of imaginary order
k_1	$\frac{1}{c^2}(k^2 + \frac{1}{4}s^2)$
k_s	Stationary wave number, $\frac{gS}{u_0^2}$
ℓ_n	$(\sigma^2 - n^2\pi^2)^{1/2}$

m	$1 - \frac{\bar{u}^2}{c_s^2}$
N	Brunt-Vaisala frequency, $(\frac{g}{\bar{\theta}} \frac{\partial \bar{\theta}}{\partial z})^{1/2}$
p, p_o	Pressure, reference pressure
q	An arbitrary variable
Q	Non-adiabatic heating
r	Distance from barrier, $(x^2 + z^2)^{1/2}$
R	Gas constant for dry air
Ri	Richardson number, $\frac{g}{\theta} \frac{\partial \theta}{\partial z} \Big/ (\frac{\partial u}{\partial z})^2$
Ri_o	$gS/u_o^2 c^2$
s	Static stability of the incompressible atmosphere, $\frac{-1}{\bar{\rho}} \frac{\partial \bar{\rho}}{\partial z}$
S	Static stability of the compressible atmosphere, $\frac{1}{\bar{\theta}} \frac{\partial \bar{\theta}}{\partial z}$
s_o	$-\frac{1}{\rho_o} \frac{\partial \bar{\rho}}{\partial z}$
t	Time
T, T_o	Temperature, reference temperature
u, v, w	Components of wind in the x, y, and z directions
u_o	Reference wind component in the x direction
$\vec{v}$	Velocity vector
x, y, z	Cartesian coordinates (z is vertical)
Y_o	Bessel function of the second kind
z_o	Height associated with a streamline far upstream of barrier
z_1	$1 + cz$
z_2	$Ri_o^{1/2} e^{-cz}$
α	$\tan^{-1} (\frac{x}{z})$
γ, γ_d	Lapse rate, dry adiabatic lapse rate

γ_n	$\frac{n\pi}{\ell_n 2} - n\pi/(\ell_n^2 - \frac{\pi^2}{b^2})$
δ	$z_o - z$
ξ	Vorticity, $\frac{\partial w}{\partial x} - \frac{\partial u}{\partial z}$
θ	Potential temperature, $T(\frac{p}{p_o})^k$
κ	R/c_p
λ	Wavelength
λ_s	Stationary wavelength, $\frac{2\pi u_o}{N}$
μ	$(Ri_o - \frac{1}{4})^{1/2}$
ν	$(\frac{k^2}{c^2} - 1)^{1/2}$
π	3.14...
Π	Exner function, $c_p(\frac{p}{p_o})^k$
ρ, ρ_o	Density, reference density
σ^2	$\lvert\frac{g}{\rho}\frac{d\rho}{dz_o}\rvert / u^2$
ϕ	Latitude
ψ	Streamfunction
ω	Density weighted vertical velocity, $(\frac{\rho}{\rho_o})^{1/2} w$
Ω	Angular rotational frequency of the earth
∇	$\frac{\partial}{\partial x}\vec{i} + \frac{\partial}{\partial z}\vec{k}$
∇^2	$\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2}$
$\hat{}$	Fourier transform of a variable
$\bar{}$	Unperturbed part of a quantity
$'$	Perturbed part of a quantity
Δ	Increment of a quantity
$\frac{d}{dt}$	Total derivative, $\frac{\partial}{\partial t} + u\frac{\partial}{\partial x} + w\frac{\partial}{\partial z}$
$\frac{\partial}{\partial t}, \frac{\partial}{\partial x}, \frac{\partial}{\partial z}$	Partial derivatives with respect to t, x, and z

AN EFFICIENT CODE FOR THE SIMULATION OF NONHYDROSTATIC STRATIFIED FLOW OVER OBSTACLES

Gregory G. Pihos and Morton G. Wurtele
Department of Atmospheric Sciences
University of California, Los Angeles

INTRODUCTION

Background

The gravity wave is the subject of a voluminous literature containing theoretical and/or observational studies. However, the numerical simulation of this phenomenon has received much less attention than has climate modeling or numerical weather prediction. A two-dimensional, nonlinear, nonhydrostatic model (Foldvik and Wurtele, 1967) produced realistic results, but was too expensive for operational use. Various linear models (Danielsen and Bleck, 1970; Vergeiner, 1971) avoided this difficulty, but could be considered reliable only when reproducing wave-like features and not when simulating turbulence-generating, wave-breaking patterns. Some highly successful computations are those of Klemp and Lilly (1978), which are applicable when the disturbance generated satisfies the quasi-hydrostatic approximation.

When the prediction of areas of clear-air turbulence (CAT) is the chief emphasis of a study, it is essential to retain both nonlinear and nonhydrostatic effects in any numerical model. Since 1966, computers and

computational techniques have been developed to an extent permitting the formulation of a model like that of Foldvik and Wurtele, but efficient enough to run at low cost. As a consequence, such a model can be used (1) to study sensitivity of results to input data; (2) to test the implications of a great variety of idealized initial and boundary conditions; and (3) to simulate easily and cheaply in real time the gravity-wave and CAT patterns associated with any operationally analyzed or predicted synoptic situation. The model and the code of the present study have been developed with all three of these purposes in mind.

Gravity Waves and CAT

Before describing the model in detail, it may be advisable to clarify some ideas and concepts involved in the relation between gravity waves and clear air turbulence. Although a number of subtle dynamic considerations are involved in the stability of stratified shear flow, for the purposes of this study we shall proceed from the assumption that a Richardson number of 0.25 is the marginally critical value for the stability of an incompressible Boussinesq flow to small perturbations. Normally the initial conditions assumed for the model will be characterized by Richardson numbers many times larger than the critical. By one means or another--represented in the model by flow over an obstacle--this flow is disturbed, and disturbed flow will contain areas in which the Richardson number is reduced from its initial value and areas in which it is increased. The dynamic/kinematic mechanism of this Richardson number modification-by-deformation is subject to various semi-quantitative explanations. Some

interpretations are reviewed by Pao and Goldburg (1969). The most widely accepted explanation of CAT generation is that large amplitude gravity waves resulting from flow over mountain ranges can and do generate local regions in which the Richardson number falls below the critical value, and moderate to severe CAT results.

Steady state linear or nonlinear solutions have been constructed for a number of highly idealized conditions. However, mathematical analysis cannot predict with any precision whether and where areas of subcritical Richardson number will occur from an arbitrary disturbance in an arbitrary flow field. Thus, for the purposes of the present work, we rely upon the numerical model exclusively to make these predictions. No attempt is made here to verify any particular theoretical interpretations of CAT formation; this must be the goal of further study. It should also be emphasized that qualitative features, such as wavelength, rotor formation, trapping, and upward propagation, are well simulated by various gravity wave models in the literature, but that there exist few quantitative comparisons with observational data. Further comparison to actual measurements will be required to measure the reliability of this model, or any other, to simulate nature. To this end, use of this model is welcome, and we have endeavored to make the code as understandable and as versatile as possible. In addition to the description of the model in the following section, a documentation of the code is given in the appendix.

THE TWO-DIMENSIONAL, BOUSSINESQ MODEL

The model must include buoyancy as the primary restoring force for any disturbance of the free stream. However, dynamic compressibility--the effect resulting in acoustic waves--is not significant in the study of CAT. Thus, incompressibility is assumed, but in a manner that retains the static effect of compressibility. Thus temperature, potential temperature, density, and pressure in the undisturbed atmosphere may be realistically represented in the initial conditions of the model. Further, the Boussinesq assumption is made, neglecting the kinematic effect of density variation (that is, where density multiplies velocity) while retaining the full dynamic effect of buoyancy (where density multiplies gravity).

The two-dimensional, Boussinesq model greatly facilitates computational ease and speed. Only two variables, vorticity and density, are directly advanced in time. The third variable, the streamfunction, is obtained at each time step by solving a Poisson equation (eq. (7c) below). The method of solution consists of applying a fast Fourier transform in the horizontal, then utilizing a one-dimensional marching solution in the vertical. This noniterative procedure is at least an order of magnitude faster than iterative relaxation techniques. Another advantage of this model is the existence of an energy integral for arbitrary mean density profiles, such as upstream inversions. (In contrast, non-Boussinesq models present computational difficulties when the stability profile is varying rapidly.) These factors will permit the program to be used frequently with sounding data, or in theoretical profiles. A description of the input/output

options is given in the appendix.

Equations Solved by the Model

We begin with the following set of equations:

The equation of motion:

$$\frac{d\vec{v}}{dt} + f\vec{k} \times \vec{v} = \frac{-1}{\rho} \nabla p + \vec{g} \qquad (1a)$$

The equation of state:

$$p=\rho RT \qquad (1b)$$

The equation of continuity:

$$\frac{d\rho}{dt} + \rho\nabla\cdot\vec{v}=0 \qquad (1c)$$

and the thermodynamic equation:

$$\frac{dQ}{dt} = c_p \frac{d}{dt}(\ln T) - R \frac{d}{dt}(\ln p) \qquad (1d)$$

Molecular viscosity and thermal conductivity will be considered to be unimportant.

This set of four equations in four unknowns may be simplified by making several assumptions. First, we will concentrate on mountain-induced gravity waves, and will limit our consideration to length scales of motion which are small compared to cyclone scale motions, so the effect of rotation will be ignored. Second, the time scale of the motion is small compared to the time scale of radiative heating, so the adiabatic assumption, $\frac{dQ}{dt} = 0$, may be made. The system of equations then becomes:

$$\frac{d\vec{v}}{dt} = \frac{-1}{\rho} \nabla \rho + \vec{g} \qquad (2a)$$

$$p=\rho RT \tag{2b}$$

$$\frac{d\rho}{dt} + \rho\nabla\cdot\vec{v}=0 \tag{2c}$$

$$c_p \frac{d}{dt} (\ln T) - R \frac{d}{dt} (\ln P)=0 \tag{2d}$$

A further simplification results from assuming the fluid density to be incompressible but not homogeneous. This permits us to allow density gradients in the vertical and buoyant restoring force without the unnecessary computation of acoustic motions. Both terms in equation (2c) then become equal to zero. This permits us to express the system of equations (2) as a system of three equations in three unknowns:

$$\frac{d\vec{v}}{dt} = \frac{-1}{\rho} \nabla p + \vec{g} \tag{3a}$$

$$\frac{d\rho}{dt} = 0 \tag{3b}$$

$$\nabla\cdot\vec{v}=0 \tag{3c}$$

The Boussinesq approximation states that the kinematic effects of density gradients are negligible compared to the buoyancy effects of density gradients in the equation of motion. The nonlinear equation (3a) then becomes:

$$\rho_o \frac{d\vec{v}}{dt} = -\nabla p + \rho\vec{g} \tag{4}$$

where ρ_o is an undisturbed reference density.

Although significant airflow is deflected around long mountain ranges such as the Sierra Nevada, even when the wind is normal to the ridgeline (Holmboe and Klieforth, 1957), we will concentrate on the flow passing directly over the mountain. For this purpose, the cross section of the range is sufficiently uniform such that the flow may be assumed to be two dimensional. Taking the curl of equation (4) with the operator $(\vec{j}\cdot\nabla x)$ in a left-handed x-z coordinate system yields the vorticity equation:

$$\frac{\partial \zeta}{\partial t} = -\nabla\cdot(\zeta\vec{v}) - \frac{g}{\rho_o}\frac{\partial \rho}{\partial x} \tag{5a}$$

Similarly, $\frac{\partial \rho}{\partial t} = -\nabla\cdot(\rho\vec{v})$ (5b)

where $\zeta \equiv \vec{j}\cdot\nabla x\vec{v} = \frac{\partial w}{\partial x} - \frac{\partial u}{\partial z}$, and u and w are the horizontal and vertical components of $\vec{v}$.

The streamfunction ψ for two-dimensional incompressible flow may be defined by $\frac{\partial \psi}{\partial x} \equiv w$ and $\frac{-\partial \psi}{\partial z} \equiv u$. Then, $\zeta \equiv \nabla^2\psi$, where $\nabla^2 \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2}$, and the system of equations (5) becomes:

$$\frac{\partial \zeta}{\partial t} = \frac{\partial \zeta}{\partial x}\frac{\partial \psi}{\partial z} - \frac{\partial \zeta}{\partial z}\frac{\partial \psi}{\partial x} - \frac{g}{\rho_o}\frac{\partial \rho}{\partial x} \tag{6a}$$

$$\frac{\partial \rho}{\partial t} = \frac{\partial \rho}{\partial x}\frac{\partial \psi}{\partial z} - \frac{\partial \rho}{\partial z}\frac{\partial \psi}{\partial x} \tag{6b}$$

$$\nabla^2\psi = \zeta \tag{6c}$$

or introducing the Jacobian operator, $J(a,b) \equiv \frac{\partial a}{\partial x}\frac{\partial b}{\partial z} - \frac{\partial a}{\partial z}\frac{\partial b}{\partial x}$,

$$\frac{\partial \zeta}{\partial t} = J(\zeta,\psi) - \frac{g}{\rho_o}\frac{\partial \rho}{\partial x} \tag{7a}$$

$$\frac{\partial \rho}{\partial t} = J(\rho,\psi) \tag{7b}$$

$$\nabla^2\psi = \zeta \tag{7c}$$

The lower boundary condition is that the surface is a streamline:

$$\psi(x,h(x)) = \text{constant} \tag{7d}$$

where $h(x)$ is the height of the barrier. The upper boundary condition is that the kinetic energy becomes vanishingly small with elevation:

$$\frac{1}{2}\rho(u^2 + w^2) = \frac{1}{2}\rho(\nabla\psi)^2 \to 0 \text{ as } z \to \infty \quad . \tag{7e}$$

The system of equations (7) constitutes the model on which the program is based. The scheme for their numerical solution and the associated code are fully described in the appendix. The assumptions made are justified and discussed below. In subsequent sections, numerical simulations with the model are compared with selected special cases for which analytical solutions can be obtained.

A Justification of the Incompressible and Boussinesq Approximations

The effect of these two assumptions may be seen by examining the vertical velocity profile of the steady state, linearized perturbation equations. Defining $\theta \equiv T(\frac{p_o}{p})^{\kappa}$ and $\Pi \equiv c_p(\frac{p}{p_o})^{\kappa}$, where $\kappa \equiv R/c_p$, the compressible system of equations (2) may be reexpressed as:

$$\frac{d\vec{v}}{dt} = -\theta\nabla\Pi + \vec{g} \tag{8a}$$

$$\theta \frac{d\Pi}{dt} = -c_s^{\,2} \nabla\cdot\vec{v} \tag{8b}$$

$$\frac{d\theta}{dt} = 0 \tag{8c}$$

where $c_s^{\,2} \equiv c_p\kappa\theta\Pi/c_v = c_pRT/c_v$. In a two-dimensional steady-state system (where $\frac{\partial}{\partial t} = 0$), these become:

$$u \frac{\partial u}{\partial x} + w \frac{\partial u}{\partial z} = -\theta \frac{\partial \Pi}{\partial x} \tag{9a}$$

$$u \frac{\partial w}{\partial x} + w \frac{\partial w}{\partial z} = -\theta \frac{\partial \Pi}{\partial z} - g \tag{9b}$$

$$\theta(u \frac{\partial \Pi}{\partial x} + w \frac{\partial \Pi}{\partial z}) = -c_s^{\,2}(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z}) \tag{9c}$$

$$u \frac{\partial \theta}{\partial x} + w \frac{\partial \theta}{\partial z} = 0 \tag{9d}$$

Now, assume that each variable $q(x,z)$ in the system may be expressed as the sum of a perturbed part $q'(x,z)$, and an unperturbed part $\bar{q}(z)$; that is, $q(x,z) = q'(x,z) + \bar{q}(x,z)$. Further assume that $\bar{w}(z) = 0$, and that the unperturbed state is in hydrostatic balance, that is, $g = -\bar{\theta} \frac{\partial \bar{\Pi}}{\partial z}$. The linearized perturbation equations for the compressible system are then:

$$\bar{u} \frac{\partial u'}{\partial x} + w' \frac{\partial \bar{u}}{\partial z} = -\bar{\theta} \frac{\partial \Pi'}{\partial x} \tag{10a}$$

$$\bar{u}\frac{\partial w'}{\partial x} = -\bar{\theta}\frac{\partial \Pi'}{\partial z} - \theta'\frac{\partial \bar{\Pi}}{\partial z} \tag{10b}$$

$$\bar{u}\frac{\partial \theta'}{\partial x} + w'\frac{\partial \bar{\theta}}{\partial z} = 0 \tag{10c}$$

$$\bar{\theta}(\bar{u}\frac{\partial \Pi'}{\partial x} + w'\frac{\partial \bar{\Pi}}{\partial z}) = -c_s^2(\frac{\partial u'}{\partial x} + \frac{\partial w'}{\partial z}) \tag{10d}$$

This system is then solved for w':

$$\frac{\partial^2 w'}{\partial x^2} + \frac{\partial^2 w'}{\partial z^2} - \frac{\partial}{\partial z}\ln(\frac{m}{\bar{\rho}})\frac{\partial w'}{\partial z} + [\frac{g}{\bar{u}^2}\frac{\partial}{\partial z}\ln m\bar{\theta} + \frac{1}{\bar{u}}\frac{\partial}{\partial z}\ln(\frac{m}{\bar{\rho}})\frac{\partial \bar{u}}{\partial z} - \frac{1}{\bar{u}}\frac{\partial^2 \bar{u}}{\partial z^2}]w' = 0 \tag{11}$$

where $m \equiv 1 - \frac{\bar{u}^2}{c_s^2}$. Usually, $\bar{u} << c_s$, which means $m \approx 1$, so the equation for w' in the compressible system becomes:

$$\frac{\partial^2 w'}{\partial x^2} + \frac{\partial^2 w'}{\partial z^2} - s\frac{\partial w'}{\partial z} + (\frac{gS}{\bar{u}^2} + \frac{s}{\bar{u}}\frac{\partial \bar{u}}{\partial z} - \frac{1}{\bar{u}}\frac{\partial^2 \bar{u}}{\partial z^2})w' = 0 \tag{12}$$

where $s \equiv -\frac{1}{\bar{\rho}}\frac{\partial \bar{\rho}}{\partial z}$, $S \equiv \frac{1}{\bar{\theta}}\frac{\partial \bar{\theta}}{\partial z}$, and $s = S + \frac{g}{c_s^2}$.

To analyze the effect of the incompressible assumption, we will derive a similar equation for w' from the incompressible system of equations (3). In a two-dimensional steady-state system, these equations have the form:

$$u\frac{\partial u}{\partial x} + w\frac{\partial u}{\partial z} = \frac{-1}{\rho}\frac{\partial p}{\partial x} \tag{13a}$$

$$u\frac{\partial w}{\partial x} + w\frac{\partial w}{\partial z} = \frac{-1}{\rho}\frac{\partial p}{\partial z} - g \tag{13b}$$

$$u \frac{\partial \rho}{\partial x} + w \frac{\partial \rho}{\partial z} = 0 \tag{13c}$$

$$\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} = 0 \tag{13d}$$

As before, assume that each variable $q(x,z)$ may be expressed as $q(x,z) = q'(x,z) + \bar{q}(x,z)$, and assume that $g\bar{\rho} = \frac{-\partial \bar{p}}{\partial z}$. Then the linearized perturbation equations for the incompressible system are:

$$\bar{u} \frac{\partial u'}{\partial x} + w' \frac{\partial \bar{u}}{\partial z} = \frac{-1}{\bar{\rho}} \frac{\partial p'}{\partial x} \tag{14a}$$

$$\bar{u} \frac{\partial u'}{\partial x} = \frac{-1}{\bar{\rho}} \frac{\partial p'}{\partial z} - \frac{\rho' g}{\bar{\rho}} \tag{14b}$$

$$\bar{u} \frac{\partial \rho'}{\partial x} + w' \frac{\partial \bar{\rho}}{\partial z} = 0 \tag{14c}$$

$$\frac{\partial u'}{\partial x} + \frac{\partial w'}{\partial z} = 0 \tag{14d}$$

The equation for w' in the incompressible system is then:

$$\frac{\partial^2 w'}{\partial x^2} + \frac{\partial^2 w'}{\partial z^2} - s \frac{\partial w'}{\partial z} + \left(\frac{gs}{\bar{u}^2} + \frac{s}{\bar{u}} \frac{\partial \bar{u}}{\partial z} - \frac{1}{\bar{u}} \frac{\partial^2 \bar{u}}{\partial z^2}\right) w' = 0 \tag{15}$$

The only difference between the steady-state compressible and incompressible equations appears in the terms $\frac{gSw'}{\bar{u}^2}$ and $\frac{gsw'}{\bar{u}^2}$. So atmospheric motions may still be modeled by the heterogeneous incompressible model by replacing the frequency of oscillation of the incompressible fluid $(gs)^{1/2}$ by the Brunt-Vaisala frequency $N = (gS)^{1/2}$ of atmospheric

buoyancy oscillations.

Now, to discuss the effect of the Boussinesq approximation, we will derive an equation for w' from the system of equations (4), (3b), and (3c). The two-dimensional steady-state equations are:

$$u \frac{\partial u}{\partial x} + w \frac{\partial u}{\partial z} = \frac{-1}{\rho_o} \frac{\partial p}{\partial x} \tag{16a}$$

$$u \frac{\partial w}{\partial x} + w \frac{\partial w}{\partial z} = \frac{-1}{\rho_o} \frac{\partial p}{\partial z} - \frac{\rho g}{\rho_o} \tag{16b}$$

The complete set includes equations (13c) and (13d). The linearized perturbation equations are:

$$\bar{u} \frac{\partial u'}{\partial x} + w' \frac{\partial \bar{u}}{\partial z} = \frac{-1}{\rho_o} \frac{\partial p'}{\partial x} \tag{17a}$$

$$\bar{u} \frac{\partial w'}{\partial x} = \frac{-1}{\rho_o} \frac{\partial p'}{\partial x} - \frac{\rho' g}{\rho_o} \tag{17b}$$

The complete set of linearized equations includes (14c) and (14d). These result in the following equation for w' in the Boussinesq system:

$$\frac{\partial^2 w'}{\partial x^2} + \frac{\partial^2 w'}{\partial z^2} + \left(\frac{g s_o}{\bar{u}^2} - \frac{1}{\bar{u}} \frac{\partial^2 \bar{u}}{\partial z^2}\right) w' = 0 \tag{18}$$

where $s_o \equiv \frac{-1}{\rho_o} \frac{\partial \bar{\rho}}{\partial z}$.

The incompressible Boussinesq and non-Boussinesq w' equations may be made similar to each other by transforming the non-Boussinesq equation into the form:

$$\frac{\partial^2\omega'}{\partial x^2} + \frac{\partial^2\omega'}{\partial z^2} + \left(\frac{gs}{\bar{u}^2} + \frac{s}{\bar{u}}\frac{\partial\bar{u}}{\partial z} - \frac{1}{\bar{u}}\frac{\partial^2\bar{u}}{\partial z^2} - \frac{1}{4}s^2 + \frac{1}{2}\frac{\partial s}{\partial z}\right)\omega' = 0 \qquad (19)$$

where $\omega' \equiv \left(\frac{\bar{\rho}}{\rho_o}\right)^{1/2} w'$. In most atmospheric profiles, the $\frac{gs}{\bar{u}^2}$ and $\frac{1}{\bar{u}}\frac{\partial^2\bar{u}}{\partial z^2}$ terms dominate over the $\frac{s}{\bar{u}}\frac{\partial\bar{u}}{\partial z}$, $\frac{1}{4}s^2$, and $\frac{1}{2}\frac{\partial s}{\partial z}$ terms, so the dynamics of w' in the Boussinesq system are similar to the dynamics of ω' in the non-Boussinesq system. Since $w' = \left(\frac{\rho_o}{\bar{\rho}}\right)^{1/2}\omega'$, the dynamics of the two systems are similar to the degree that $\bar{\rho}(z)$ remains constant. This is true essentially for shallow atmospheric systems (Ogura and Phillips, 1962).

To sum up, the reason for using the incompressible assumption and Boussinesq approximation is to simplify the system of equations to be solved. In order to retain the dynamics of the compressible atmosphere, the stability S will be used, and vertical density gradients will be retained everywhere. The total percentage variation of density in the fluid will be kept small, since ρ will then have the same scale height as the potential temperature. Then the compressible system w' equation (12) applies, and may be similarly transformed using $w' = \left(\frac{\rho_o}{\bar{\rho}}\right)^{1/2}\omega'$ to yield:

$$\frac{\partial^2\omega'}{\partial x^2} + \frac{\partial^2\omega'}{\partial z^2} + F(z)\omega' = 0 \qquad (20)$$

where $F(z) \equiv \frac{gS}{\bar{u}^2} + \frac{s}{\bar{u}}\frac{\partial\bar{u}}{\partial z} - \frac{1}{\bar{u}}\frac{\partial^2\bar{u}}{\partial z^2} - \frac{1}{4}s^2 + \frac{1}{2}\frac{\partial s}{\partial z}$.

Techniques for Analytical Solution of the Linear Problem

The mountain wave problem consists of solving equation (20) with the appropriate boundary conditions in the upper half plane $z \geq 0$. The lower

boundary condition consists of tangential flow to the barrier,

$$w'(x,z) = \frac{dh(x)}{dx}\,[\bar{u}(z) + u'(x,z)] \text{ at } z = h(x) \tag{21}$$

where h(x) is the height of the barrier. Assuming that h is small, the linearized version of equation (21) is written as:

$$w'(x,0) = \bar{u}(0)\,\frac{dh(x)}{dx} \tag{22}$$

The upper boundary condition is that the kinetic energy, $\bar{\rho}w^2/2$, vanish at $r = \infty$ where $r = (x^2 + z^2)^{1/2}$. In terms of $\omega'(x,z)$, these conditions become:

$$\omega'(x,0) = \bar{u}(0)\,\frac{dh(x)}{dx}\ , \text{ and } \lim_{r\to\infty} \omega' = 0\ . \tag{23}$$

Now, assume that $\omega'(x,z)$ may be expressed as a sum of individual wave components, $\omega'(x,z) = \int_{-\infty}^{\infty} \hat{\omega}(k,z)e^{ikx}dk$, and express the ground terrain as a sum of Fourier components, $h(x) = \int_{-\infty}^{\infty}\hat{h}(k)e^{ikx}dk$. Since the system is linear, the behavior of a single wave component may now be examined. The $\hat{\omega}$ system becomes:

$$\frac{d^2\hat{\omega}(k,z)}{dz^2} + [F(z) - k^2]\hat{\omega}(k,z) = 0 \tag{24a}$$

with boundary conditions:

$$\hat{\omega}(k,0) = ik\bar{u}(0)\hat{h}(k), \text{ and } \lim_{z\to\infty} \hat{\omega}(k,z) = 0. \tag{24b}$$

The general solution is $\hat{\omega}(k,z) = c_1(k)\hat{\omega}_1(k,z) + c_2(k)\hat{\omega}_2(k,z)$, where $c_1(k)$ and $c_2(k)$ are arbitrary constants to be determined by $\lim_{z\to\infty}[\hat{\omega}(k,z)] = 0$. Then $\hat{\omega}(k,z) = c_3(k)\hat{\omega}_3(k,z)$, where $\hat{\omega}_3(k,z)$ is a linear combination of $\hat{\omega}_1(k,z)$ and $\hat{\omega}_2(k,z)$, and $c_3(k)$ is determined by another boundary condition, $c_3(k) = \hat{\omega}(k,0)/\hat{\omega}_3(k,0)$. Then

$$\hat{\omega}(k,z) = iku(0)\hat{h}(k)\frac{\hat{\omega}_3(k,z)}{\hat{\omega}_3(k,0)} \tag{25}$$

$$\omega'(x,z) = \int_{-\infty}^{\infty} ik\bar{u}(0)\hat{h}(k)\frac{\hat{\omega}_3(k,z)}{\hat{\omega}_3(k,0)}e^{ikx}\,dk \tag{26}$$

This integral is improperly defined for any value of k where $\hat{\omega}_3(k,0) = 0$. If the integral is evaluated at these singularities by taking Cauchy's principal value, the primary contribution to the integral comes from the neighborhood of the singularities. These discrete values of k, if any exist, correspond to free waves or resonance waves of the system, and represent eigensolutions of equation (24) which dominate other waves in the system.

For a given velocity and stability profile, the boundary conditions either do not uniquely determine the steady-state solution, or do not uniquely determine the amplitudes of the free waves. Mathematical uniqueness may be established, however, by some physical argument, such as requiring that all waves vanish far upstream of the barrier, or by considering a time-dependent system which asymptotically approaches the steady-state solution.

COMPARISON OF MODEL WITH ANALYTICAL SOLUTIONS

To establish confidence in the consistency of the numerical model, the computations will be compared to known analytical solutions. In general, these analytical solutions exist only for the linearized steady-state equations with simple idealized meteorological profiles. Although the model also incorporates nonlinear and transient motions, it should qualitatively and quantitatively resemble the linear, steady-state solutions after a period of model time, assuming that the barrier height and the density and velocity profiles have been selected to preclude highly non-linear effects.

Constant Velocity, Constant Stability Case

This is basically the simplest case, since F(z) in equation (20) has the constant value $F = \frac{gS}{u_o^2} - \frac{1}{4} s^2$, where $\bar{u} = u_o$. Usually, $\frac{1}{4} s^2 << \frac{gS}{u_o^2}$, so that the ω' equation becomes:

$$\frac{\partial^2 \omega'}{\partial x^2} + \frac{\partial^2 \omega'}{\partial z^2} + k_s^2 \omega' = 0 \qquad (27)$$

where $k_s \equiv (\frac{gS}{u_o^2})$ is the stationary wavenumber. This wave equation specifies a disturbance with wavelength $\lambda_s \equiv \frac{2\pi}{k_s} = \frac{2\pi u_o}{N} = 2\pi u_o \left[\frac{T_o}{g(\gamma_d - \gamma)}\right]^{1/2}$ (where T_o is the reference temperature, γ is the lapse rate, and γ_d is the dry adiabatic lapse rate), which is independent of the barrier and is approached asymptotically at large distances by the actual solution since the boundary conditions have not yet been taken into account. Using the previously

stated boundary conditions, Lyra (1943) expressed the vertical velocity field for an arbitrary obstacle $z = h(x)$ as an infinite series of Bessel functions of the first and second kind (J_ν and Y_0):

$$\omega'(x,z) = 2u_0 \int_{-\infty}^{\infty} \frac{dh(x)}{dx} \frac{\partial}{\partial z} \left[\frac{1}{4} Y_0(k_s r) + \frac{1}{\pi} \sum_{\nu=0}^{\infty} J_{2\nu+1}(k_s r) \frac{\cos(2\nu+1)\alpha}{2\nu+1}\right] dx \tag{28}$$

where $\alpha = \tan^{-1}(\frac{x}{z})$. Since the Bessel functions J_ν are eigensolutions of the system, an infinite number of free waves exist, due to the zeros in the J_ν. For many barrier shapes, including the rectangular shape in the numerical model, the free waves add up to form an infinite series of backward tilting lee waves with $\lambda \to \lambda_s$ as $r \to \infty$.

Figures 1 through 3 show the transient development of the streamlines for the Lyra problem when u_0 = 25 m/sec, T_0 = 273K, γ = 0, $\Delta x = \Delta z$ = 1000 m, and Δt = 20 sec (referred to as case 1) at times 50 Δt, 75 Δt, and 100 Δt. The qualitative appearance of the waves agrees with the results of Lyra, except for the influence of the top boundary in the model. This boundary will not be as important in most other velocity profiles, since more energy will be trapped at lower levels. The theory predicts that λ_s = 8.4 km. The most reliable wavelength measurements for comparison to the analytical solutions are taken in the area of a well-developed wave pattern and as far downstream as is feasible to avoid distortions caused by the assumption of no upstream perturbations. At 7 km elevation, figure 3 exhibits 8.5 km separation between the second and third crests, and 8 km separation between the third and fourth crests.

As the model approaches a steady state, $\frac{\partial \rho}{\partial x}\frac{\partial \psi}{\partial z} - \frac{\partial \rho}{\partial z}\frac{\partial \psi}{\partial x} = \frac{\partial \rho}{\partial t} \to 0$ or $\frac{\partial \psi}{\partial z} / \frac{\partial \psi}{\partial x} = \frac{\partial \rho}{\partial z} / \frac{\partial \rho}{\partial x}$; that is, the slopes of the isolines of streamfunction and density are everywhere equal, so that the isolines of these variables should coincide. Figures 4 and 5 show the general resemblance of the density field to the streamline field at time 50 Δt and 100 Δt, except in the vicinity of the barrier, where most transient development is still taking place.

Assuming that $\bar{\rho}(z) \approx \rho_o$ in the model, then $\omega' \approx (\frac{\rho_o}{\bar{\rho}})^{1/2}\omega' \equiv w' \equiv \frac{\partial \psi'}{\partial x}$, and equation (20) becomes:

$$\frac{\partial^2 \psi'}{\partial x^2} + \frac{\partial^2 \psi'}{\partial z^2} + F(z)\psi' = 0 \qquad (29)$$

In the Lyra case, $F(z) = k_s^2$, and $\bar{\zeta} = 0$, so then $\zeta' = \nabla^2\psi'$, which implies that $\zeta' = -k_s^2\psi'$; that is, the isolines of vorticity should coincide with the isolines of streamfunction displacement. Figures 6 and 7 show the resemblance of the vorticity field to the troughs and crests of the streamline field, except in the vicinity of the barrier, where the assumption of linearity is not valid. It should also be noted that the vorticity field shows small scale perturbations which are computational in nature. These short wavelength perturbations occur mainly as a result of aliasing, that is, the inability of any numerical model to resolve disturbances with wavelengths less than two grid intervals. As is discussed in the appendix, the finite differencing scheme used retards the unstable growth of these perturbations, and we have not found it necessary to use filtering, smoothing, or damping operators in order to run physically meaningful computations for a sufficiently long period of time. Since the vorticity is the second derivative of the streamfunction, the streamfunction field should remain smoother than the vorticity field.

The local Richardson number, defined as $Ri \equiv -\frac{g}{\rho}\frac{\partial \rho}{\partial z} \Big/ (\frac{\partial^2 \psi}{\partial z^2})^2$ in the model corresponding to figures 3, 5, and 7 at time 100 Δt, is shown in figure 8. It should be noted that Richardson number tends to vary rapidly over several orders of magnitude in disturbed sections of the flow. Thus, $\log_{10}(Ri)$ is actually plotted, and values of $Ri \leq .16$ or $Ri > 10$ are set to 0.16 or 10 respectively, in order to highlight areas where $Ri < 0.25$. Also, since the Richardson number is the quotient of first and second derivatives, the finite-difference analog for Ri is not dependable within one grid interval of the ground terrain.

Linear Shear, Constant Stability Case

For this case, $F(z) = \frac{gS}{\bar{u}^2} + \frac{s}{\bar{u}}\frac{\partial \bar{u}}{\partial z} - \frac{1}{4}s^2$, where $\bar{u} = u_o(1 + cz)$. Assuming that $\frac{gS}{\bar{u}^2} - \frac{1}{4}s^2 >> \frac{s}{\bar{u}}\frac{\partial \bar{u}}{\partial z}$, equation (29) becomes:

$$\frac{\partial^2 \psi'}{\partial x^2} + \frac{\partial^2 \psi'}{\partial z^2} + (\frac{gS}{\bar{u}^2} - \frac{s^2}{4})\psi' = 0 \qquad (30)$$

Assume that $\psi'(x,z) = \int_{-\infty}^{\infty} e^{ikx}\hat{\psi}(k,z)dk$, and let $z_1 = 1 + cz$. Then, for a single wave component:

$$\frac{d^2\hat{\psi}}{dz_1^2} + (-k_1^2 + \frac{Ri_o}{z_1^2})\hat{\psi} = 0 \qquad (31)$$

where $k_1^2 = \frac{k^2 + \frac{s^2}{4}}{c^2}$, and $Ri_o = \frac{gS}{u_o^2 c^2}$. The solution to this equation satisfying the upper boundary condition is a modified Bessel function of the third kind of imaginary order $K_{i\mu}(k_1 z_1)$, where $\mu = (Ri_o - \frac{1}{4})^{1/2}$.

Using the lower boundary condition, the solution can be expressed as (Wurtele, 1953):

$$\psi'(x,z) = \left(\frac{1+cz}{2\pi}\right)^{1/2} \int_{-\infty}^{\infty} e^{ixk} \hat{h}(k) \frac{K_{i\mu}\left[k_1(1+cz)\right]}{K_{i\mu}(k_1)} dk \qquad (32)$$

The free waves of the system correspond to discrete values of k_n (or $(k_1)_n$) for which $K_{i\mu}[(k_1)_n] = 0$, with wavelength $\lambda_n \equiv \frac{2\pi}{k_n} = 2\pi[(k_1)_n c^2 - \frac{1}{4} s^2]^{-1/2}$ and no tilt with height, and exist only for $\mu > 0$ or $Ri_0 > \frac{1}{4}$. The number of free waves with wavelengths in the mesoscale range increases with decreasing shear, approaching an infinite number in the Lyra case.

When $u_0 = 10$ m/sec, $c = 2.7 \times 10^{-4}$/m, $T_0 = 250$K, and $\gamma = 6.76$K/km (referred to as case 2), then $Ri_0 = 16.0$, $\mu = 3.97$, and the theory predicts two free wave modes with wavelengths 13.7 km and 31.0 km. Figure 9 shows the streamfunction field at 3750 seconds. Only the first wave has developed, with two crests separated by 13 km. At 7500 seconds, figure 10 reveals the shorter wave dominating below 5 km, and a longer wave with two crests separated by 29 km prevailing at higher elevations, in accordance with the theory.

Exponential Shear, Constant Stability Case

In this case, $\bar{u} = u_0 e^{cz}$, so $F(z) = \frac{gS}{u_0^2} e^{-2cz} + sc - c^2 - \frac{1}{4} s^2 + \frac{1}{2} \frac{\partial s}{\partial z}$. Assuming that the total percentage variation of $\bar{\rho}$ is small compared to the total percentage variation of $\bar{u}$, then the terms s^2, $\frac{\partial s}{\partial z}$, and sc are all much smaller than c^2, and equation (29) becomes:

$$\frac{\partial^2 \psi'}{\partial x^2} + \frac{\partial^2 \psi'}{\partial z^2} + \left(\frac{gS}{u_0^2} e^{-2cz} - c^2\right)\psi' = 0 \qquad (33)$$

Assume that $\psi'(x,z) = \int_{-\infty}^{\infty} e^{ikx}\hat{\psi}(k,z)dk$, and let $z_2 = Ri_o^{1/2}e^{-cz}$, where $Ri_o \equiv \frac{gS}{u_o^2 c^2}$. Then, for a single wave component:

$$\frac{d^2\hat{\psi}}{dz_2^2} + \frac{1}{z_2}\frac{d\hat{\psi}}{dz_2} + \left(1 - \frac{\frac{k^2}{c^2} + 1}{z_2^2}\right)\psi = 0 \qquad (34)$$

The solution to this equation satisfying the upper boundary condition is a Bessel function of the first kind, $J_\nu(z_2)$, where $\nu \equiv (\frac{k^2}{c^2} + 1)^{1/2} \geq 1$. Using the lower boundary condition, the solution can be expressed as (Palm and Foldvik, 1960):

$$\psi'(x,z) = \bar{u}(z)\int_{-\infty}^{\infty} e^{ikx}\hat{h}(k)\,\frac{J_\nu(Ri_o^{1/2}e^{-cz})}{J_\nu(Ri_o^{1/2})}\,dk \qquad (35)$$

The free waves, if any, result from discrete values of k_n(or ν_n) for which $J_\nu(Ri_o^{1/2}) = 0$, and have wavelength $\lambda_n \equiv \frac{2\pi}{k_n} = \frac{2\pi}{c(\nu_n^2-1)^{1/2}}$ with maximum amplitude development at $z_n = -\frac{1}{c}\ln[(z_2)_n/Ri_o]$, where $(z_2)_n$ is the value of z_2 at which $J_\nu(z_2)$ attains its maximum value between the nth and (n+1)th zeros of $J_\nu(z_2)$.

It can be seen graphically (Jahnke and Emde, 1945) that no waves exist if $Ri_o^{1/2} \lesssim 3.8$, one wave exists if $3.8 \lesssim Ri_o^{1/2} \lesssim 7.0$, and two waves exist if $7.0 \lesssim Ri_o^{1/2} \lesssim 10.2$. When u_o = 20 m/sec, $c = 1.8\times10^{-4}$/m, and $N = 1.2\times10^{-2}$/sec (referred to as case 3), then $Ri_o^{1/2} = 3.3$, and the theory predicts no waves. Figure 11 shows no waves after 1200 seconds. When u_o = 10 m/sec, $c = 2\times10^{-4}$/m, and $N = 1.2\times10^{-2}$/sec (referred to as case 4), then $Ri_o^{1/2} = 6.0$, and the theory predicts one wave mode with λ = 12.8 km, and with maximum amplitude development at z = 2.24 km. At 4500 sec, Figure 12 shows two crests separated by 13 km and maximum amplitude occurs at z = 2 km.

Nonlinear Case with Constant ρu^2

This case, developed by Long (1955), solves the nonlinear equations with a nonlinear barrier in a fluid with a rigid top and bottom. The incompressible, steady-state equations of motion (13a,b) may be rewritten as follows:

$$\rho \frac{\partial}{\partial x}\left(\frac{u^2+w^2}{2}\right) - \zeta\rho w = -\frac{\partial p}{\partial x} \tag{36a}$$

$$\rho \frac{\partial}{\partial z}\left(\frac{u^2+w^2}{2}\right) + \zeta\rho u = -\frac{\partial p}{\partial z} - \rho g \tag{36b}$$

where $\rho = \rho(\psi)$ and $\zeta = \zeta(\psi)$. Eliminating p, equation (36) becomes:

$$\left(u\frac{\partial}{\partial x} + w\frac{\partial}{\partial z}\right)\left[\zeta + \frac{1}{\rho}\frac{d\rho}{d\psi}\left(\frac{u^2+w^2}{2} + gz\right)\right] \equiv \frac{d}{dt}\left[\zeta + \frac{1}{\rho}\frac{d\rho}{d\psi}\left(\frac{u^2+w^2}{2} + gz\right)\right] = 0 \tag{37}$$

This is then integrated to yield:

$$\nabla^2\psi + \frac{1}{\rho}\frac{d\rho}{d\psi}\frac{(\nabla\psi)^2}{2} = \zeta + \frac{1}{\rho}\frac{d\rho}{d\psi}\left[\frac{u^2}{2} + g(z_o - z)\right] \tag{38}$$

where $u(\psi)$ and $z_o(\psi)$ are the horizontal velocity and height associated with the streamline for constant ψ far upstream of the barrier. Noting that $\frac{d}{d\psi} = -\frac{1}{u}\frac{d}{dz_o}$, and substituting $\delta = z_o - z$ yields an equation in δ:

$$\nabla^2\delta + \left[\frac{(\nabla\delta)^2}{2} + \frac{\partial\delta}{\partial z}\right]\frac{d}{dz_o}(\ln \rho u^2) = \frac{g}{\rho u^2}\frac{d\rho}{dz_o}\delta \tag{39}$$

Now, if ρu^2 is constant, then $\left|\frac{g}{\rho}\frac{d\rho}{dz_o}\right|/u^2$ is constant, and equation (39) becomes the linear equation:

$$\nabla^2\delta + \sigma^2\delta = 0 \tag{40}$$

where $\sigma^2 \equiv \left|\frac{g}{\rho}\frac{d\rho}{dz_o}\right| / u^2$. The constant ρu^2 criterion is satisfied approximately by the model by setting u constant and keeping density gradients small.

Specifying the boundary conditions as $\delta(x,0) = \frac{a}{2}(1 + \cos\frac{\pi x}{b})$ for $-b \leq x \leq b$, with $\delta(x,0) = 0$ elsewhere, and $\delta(x,H) = 0$ at the top boundary $z = H$, Long expressed the solution of (38) as:

$$\delta(x > b, z) = -a \sum_{n=1}^{n_1} [2\gamma_n \sin(\ell_n b)\sin(\ell_n x)\sin(n\pi z)] - a \sum_{n=n_1+1}^{\infty} \{\frac{\gamma_n}{2}[e^{\ell_n(x-b)} - e^{-\ell_n(x+b)}]\sin(n\pi z)\} \quad (41)$$

where $\ell_n^2 = \sigma^2 - n^2\pi^2$, $\gamma_n = \frac{n\pi}{\ell_n^2} - n\pi/(\ell_n^2 - \frac{\pi^2}{b^2})$ and n_1 is the largest integer for which $\ell_n^2 > 0$.

When $u_o = 30$ m/sec, $N = 1.2\text{x}10^{-2}$/sec, $H = 10$ km, $a = 0.3H$, and $b = 0.4H$ (referred to as case 5), the theory predicts a single wave mode with wavelength $\lambda_1 \equiv \frac{2\pi H}{\ell_1} = 2\pi H/(\frac{N^2H^2}{u_o^2} - \pi^2) = 25.4$ km with maximum vertical velocity $|w_{max}| \approx 2au_o\gamma_1\ell_1 \sin \ell_1 b = 21.2$ m/sec. In order to obtain numerical computations corresponding to Long's solution, the model simulates the lower boundary condition by specifying the nonrigid flow boundary $w(x,0) = -\frac{u_o a\pi}{2b}\sin(\frac{\pi x}{b})$ for $-b \leq x \leq b$, and $w(x,0) = 0$ elsewhere. Values of a and b have been chosen to preclude highly nonlinear disturbances downstream. Figures 13 and 14 show the streamline and vertical velocity fields at time 3000 sec. The separation between the two crests is approximately 25 km, and the maximum vertical velocity is approximately 18 m/sec. It should be noted that the vertical velocity is defined to have a greater value than this at the inflow, and that the measurements must be taken at least beyond the first crest.

Comparison of Nonlinear Effects

In the cases above, the height of the barrier has been chosen to be sufficiently small so that the model will produce linear effects. However, features such as the reversed flow or rotors often observed in association with mountain lee waves arise from the nonlinearity of the barrier. This concept may be explained as follows. According to equation (22), the linearized surface boundary condition for the streamfunction may be written:

$$\psi'(x,0) = u_o h(x) \equiv u_o h d(x) \tag{42}$$

where d(x) is a dimensionless profile function of order unity, and h is the maximum height of the barrier. In the constant velocity, constant stability case, for example, $\psi'(x,z)$ is a function of $k_s h$, so that ψ may be expressed in the form:

$$\psi = \bar{\psi} + \psi' = -u_o z \cdot G(k_s x, k_s z) \tag{43}$$

where G is of order unity, and $G(k_s x, 0) = d(x)$. The horizontal velocity is then:

$$u = -\frac{\partial \psi}{\partial z} = u_o - u_o h \frac{\partial G(k_s x, k_s z)}{\partial z} = u_o \left[1 - k_s h \frac{\partial G(k_s x, k_s z)}{\partial (k_s z)}\right] \tag{44}$$

where $\partial G/\partial(k_s z)$ is dimensionless and of order unity. The condition for reversed flow, or $u \leq 0$, is then approximately $k_s h \geq 1$. Miles (1969) calculated the critical values of $k_s h$ for flow over semi-elliptical barriers to be between 0.67 and 1.73, depending on the ellipticity of the barrier.

Figure 15 displays the streamline field at 1500 seconds for the Lyra problem when $k_s h = 1.17$ (referred to as case 6), showing that the

critical limit has just been exceeded, with reverse flow at some points in the field. This case is similar to case 1, except that $\Delta x = \Delta z = 750$ m, and $\Delta t = 15$ sec. The corresponding density field is displayed in figure 16. A more highly nonlinear case with $k_s h = 1.95$, $T_o = 250K$, $\Delta x = \Delta z = 625$ m, $\Delta t = 10$ sec (referred to as case 7) is portrayed in figures 17 through 19, at times 600 seconds, 800 seconds, and 1000 seconds, respectively. This sequence reveals the development of highly unstable configurations which break down realistically into rotorlike formations. It should be noted that the turbulence associated with the instability of the breaking wave is not simulated. The density field corresponding to streamline figure 19 is shown in figure 20. The Richardson number fields corresponding to streamline figures 15 and 19 are shown in figures 21, and 22, respectively.

COMPARISON OF MODEL WITH OBSERVATIONS

Some of the most detailed observations of the atmospheric structure associated with mountain-induced waves have been taken over the eastern slope of the Rocky Mountains near Boulder and Denver. Boulder is located at the immediate base of the north-south range, and is susceptible to occasional downslope windstorms. On January 11, 1972, a particularly violent windstorm with peak mean wind velocities of 30 m/sec, and gusts to 55 m/sec swept through the area. Lilly and Zipser (1972) derived cross sections of the potential temperature (figure 23) and horizontal wind velocity (figure 24) from aircraft observations taken during this event. The figures reveal a severe downslope windstorm and extensive mid-tropospheric clear air turbulence induced by a wave of large amplitude and wavelength.

Nonlinear numerical simulations of this case have been performed by Klemp and Lilly (1978), and by Peltier and Clark (1979).

The wind and stability profiles are initialized for our model from the Grand Junction, Colorado sounding at 000Z on 12 January 1972. Grand Junction is approximately 300 km upwind and at approximately the same elevation as Boulder. The lee slope of the Rocky Mountains in the vicinity of Boulder is reproduced as closely as is possible with a resolution of Δx = 2000 m and Δz = 500 m. The upwind terrain is quite complex, but its model representation was not found to have an appreciable effect in the computation owing to partial upstream blocking. Since the potential temperature often varies on the surface of a high mountain, the density has been allowed to vary on the surface of the barrier for this computation.

Figures 25 and 26 show the streamline field and the horizontal velocity field at 4250 seconds. The model reproduces many of the observed features of the mountain wave. The computed trough of the wave is located almost directly over Boulder, which is situated within one grid point of the lee slope. The computation shows the first crest of the wave to be 38 km downstream from the crest of the mountain, compared to an observed distance of 37 km. The wave shows no tilt with height up to the tropopause at approximately 11 km, then tilts back sharply into the stratosphere.

In comparing the locations of the maximum and minimum wind velocities, it should be noted that figure 23 contains two sets of observations taken several hours apart. The winds in figure 24 are derived from the data taken during the time when the trough of the wave had moved over the lee slope of the mountain, presumably due to variation in the upstream wind or stability profiles. A study of numerous windstorms in the Boulder area by Brinkmann

(1974) reveals that the surface wind speed maximum is localized beneath the trough of the wave, and the output of the model is in agreement with this finding.

SUMMARY AND CONCLUSIONS

This report has described the development of a numerical model for the simulation of nonlinear, nonhydrostatic stratified flow over obstacles. This type of model is appropriate for the investigation of clear air turbulence associated with gravity waves resulting from flow over mountain ranges. To simplify the equations to be solved, the flow has been assumed to be two dimensional and incompressible, and the Boussinesq approximation has been made. These features have made possible a code which is sufficiently versatile and efficient to accommodate case studies using either idealized profiles or actual sounding data.

Simplicity has also been retained in the boundary conditions. Disturbances are generated by a rigid barrier, which is part of the lower boundary. The top boundary is also rigid, with periodic boundary conditions at the sides. Although these conditions require a sufficiently large computational field to produce physically useful results before contamination occurs, the computational speed of the program has always made this feasible.

The consistency of the numerical model has been established by comparing the computations to known analytic solutions, and by comparison with mountain wave observations. The model reproduced qualitative and quantitative features of the steady-state solutions, and also realistically simulated breaking wave patterns associated with highly nonlinear obstacles.

These tests provide confidence that the model may now be applied to observational data for further comparison.

APPENDIX: DESCRIPTION OF THE COMPUTER PROGRAM

DESCRIPTION OF CALCULATIONS

The System to be Solved

The program solves the time-dependent system of equations:

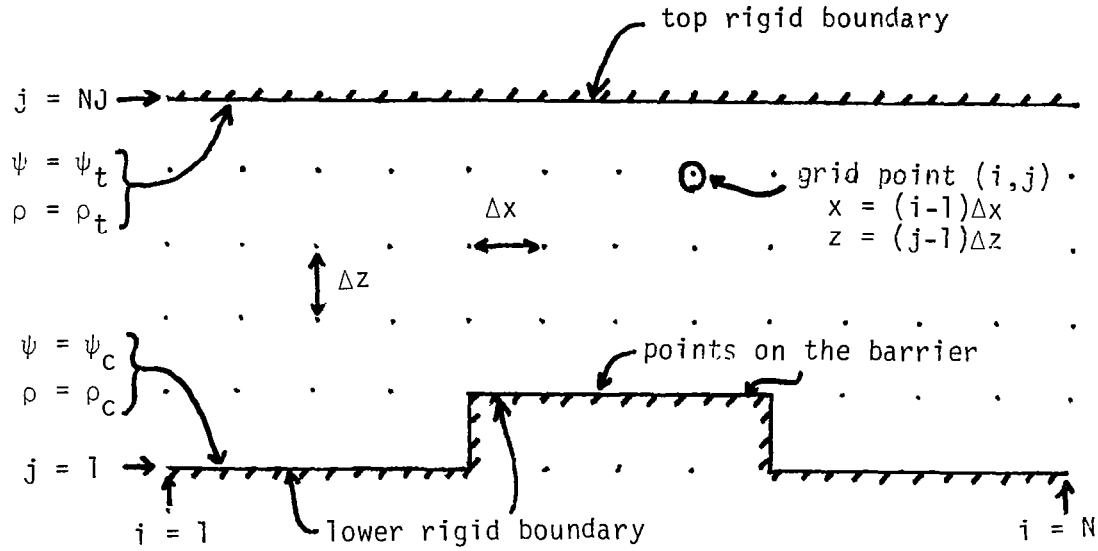
$$\frac{\partial \zeta}{\partial t} = J(\zeta,\psi) - \frac{g}{\rho_o}\frac{\partial \rho}{\partial x} \qquad \text{(A1a)}$$

$$\frac{\partial \rho}{\partial t} = J(\rho,\psi) \qquad \text{(A1b)}$$

$$\nabla^2\psi = \zeta \qquad \text{(A1c)}$$

on a rectangular grid in the x-z plane, with rigid slip boundaries (tangential flow) at the top and bottom (where ψ and ρ have constant, fixed values), and periodic boundary conditions at the sides such that the inflow at one side matches the outflow at the other. To facilitate the finite difference calculation of horizontal derivatives in the program, the second to last column is a duplicate of the first, and the last column is a duplicate of the second. Disturbances are generated in the flow by a rigid barrier of user-specified shape and size, which becomes part of the lower boundary. This system is pictured in the diagram below. Grid points in the x-z plane are indexed with the letters (i,j) beginning with (i,j) = (1,1). In this system, all variables will be assumed to have values only at the same discrete grid points (i,j). All finite difference

expressions are valid for all unique (i,j) in the grid unless otherwise stated.



Several comments need to be made regarding the boundary conditions. First, since the streamfunction is held constant at all times along the lower surface and the barrier, the program is applying a nonlinear boundary condition in every problem. Thus, the potential exists for nonlinear features to form even when simulating a linear analytic solution. The distinction between "linear" and "nonlinear" cases is determined by the degree of nonlinear behavior in the solution.

Second, it should be noted that a periodic boundary condition in the x-direction allows disturbances which propagate to either lateral boundary to reenter through the opposite side, eventually contaminating the solution. The computational field must be given sufficient horizontal extent so that useful results are obtained before contamination occurs. This disadvantage is offset by the absence of reflection at the lateral boundaries, and by the flexibility of the model to simulate a wide variety of nonlinear flow prob-

lems without the necessity of devising a suitable set of open boundary conditions for each problem.

Finally, we mention that the top boundary condition constitutes a rigid lid. Since this is highly reflective, the computational field must be given sufficient vertical extent so that useful results are obtained before significant reflection occurs. Due to the computational speed of the model, it has always been feasible to utilize a sufficiently large array for these purposes.

Scheme for Solving the Equations

The program uses an explicit, centered-time (leapfrog), centered-space finite differencing scheme with fixed boundary conditions on ψ and ρ to solve the system of equations (1). Assuming that ψ, ρ, and ζ are all known at time steps m-1 and m, then equations (1a) and (1b) may be integrated to yield ρ and ζ at time step m+1:

$$\zeta^{m+1} = \left[J(\zeta^m, \psi^m) - \frac{g}{\rho_0} \frac{\partial \rho}{\partial x}\right] \cdot 2\Delta t + \zeta^{m-1} \qquad \text{(A2a)}$$

$$\rho^{m+1} = J(\rho^m, \zeta^m) \cdot 2\Delta t + \rho^{m-1} \qquad \text{(A2b)}$$

where ρ^m and ζ^m are the values of ρ and ζ at time step m. Then ψ^{m+1} is found from ζ^{m+1} by relaxing $\nabla^2\psi^{m+1} = \zeta^{m+1}$.

To begin the procedure, ψ, ρ, and ζ are initialized at all points at time step 0. Then, in order to start the three level time differencing scheme, it is necessary to obtain the values of the variables at time step

1 by another method. The model uses a Matsuno-type scheme, which involves taking a half time step forward, then using a centered time difference to proceed to time step 1, as follows:

$$\zeta^{1/2} = [J(\zeta^o, \psi^o) - \frac{g}{\rho_o} \frac{\partial \rho^o}{\partial x}] \cdot \frac{\Delta t}{2} + \zeta^o \qquad \text{(A3a)}$$

$$\rho^{1/2} = J(\rho^o, \psi^o) \cdot \frac{\Delta t}{2} + \rho^o \qquad \text{(A3b)}$$

$$\text{relax} \quad \nabla^2 \psi^{1/2} = \zeta^{1/2} \text{ for } \psi^{1/2} \qquad \text{(A3c)}$$

$$\zeta^1 = [J(\zeta^{1/2}, \psi^{1/2}) - \frac{g}{\rho_o} \frac{\partial \rho^{1/2}}{\partial x}] \cdot \Delta t + \zeta^o \qquad \text{(A4a)}$$

$$\rho^1 = J(\rho^{1/2}, \psi^{1/2}) \cdot \Delta t + \rho^o \qquad \text{(A4b)}$$

$$\text{relax} \quad \nabla^2 \psi^1 = \zeta^1 \quad \text{for} \quad \psi^1. \qquad \text{(A4c)}$$

This scheme results in less computational error, and is less destabilizing than taking a full forward time step.

After hundreds of time steps, the leapfrog scheme may introduce a time-splitting instability into the solution of this nonlinear model. This instability may be suppressed by the occasional insertion of a time step made by a two level scheme (Mesinger and Arakawa, 1976). Thus, the program restarts the procedure every 25 time steps with the scheme described above. This is not the only method which may be used, but it is a standard numerical procedure (Arakawa and Lamb, 1977).

Initialization of $\bar{\psi}^0$, $\bar{\rho}^0$, and $\bar{\zeta}^0$

The user determines the unperturbed $\bar{\psi}^0(z)$, $\bar{\rho}^0(z)$, and $\bar{\zeta}^0(z)$ by specifying the initial horizontal velocity profile, $u^0(z)$, and either the initial stability profile, $S^0(z)$, or the initial temperature and pressure profiles, $T^0(z)$ and $p^0(z)$. In two dimensions, $\bar{\psi}^0(x,z) \equiv -\int[u^0(x,z)dz - w^0(x,z)dx]$. When $w^0(x,z) = 0$, and $u^0(x,z)$ is a function of z only, this becomes:

$$\bar{\psi}^0(z) = -\int_0^z u^0(z')dz' + \psi_c \qquad \text{(A5)}$$

where ψ_c is an arbitrary constant which has been set to zero by the program. Denoting the value of $\bar{\psi}^0$ at grid point (i,j) as $\bar{\psi}^0_{ij}$, the finite difference form of equation (A5), using the trapezoidal rule, is:

$$\bar{\psi}^0_{i,1} = 0 \qquad \text{(A6a)}$$

$$\bar{\psi}^0_{i,j} = \bar{\psi}^0_{i,j-1} - \left(\frac{u^0_{i,j-1} + u^0_{i,j}}{2} \right)\Delta z \quad \text{for all } j=2,\ldots,NJ \qquad \text{(A6b)}$$

where $\bar{\psi}^0_{ij} \equiv \bar{\psi}^0[(j-1)\Delta z]$.

For the compressible atmosphere, the stability is $S^0(z) = \frac{1}{\theta^0(z)} \frac{\partial \theta^0(z)}{\partial z}$. The density profile in this incompressible model is defined to correspond to the stability of the compressible atmosphere by setting $\frac{-1}{\bar{\rho}^0(z)} \frac{\partial \bar{\rho}^0(z)}{\partial z}$ $\equiv S^0(z)$. The expression for $\bar{\rho}^0(z)$ is then:

$$\bar{\rho}^0(z) = \rho_c \exp[-\int_0^z S^0(z')dz'] \qquad \text{(A7)}$$

where ρ_c is an arbitrary constant which has been set to 1.25 kg/m^3. Using the trapezoidal rule, the finite difference form of equation (A7) is:

$$\bar{\rho}^o_{i,1} = \rho_c \tag{A8a}$$

$$\bar{\rho}^o_{i,j} = \bar{\rho}^o_{i,j-1} \exp[-\frac{\Delta z}{2}(S^o_{i,j-1} + S^o_{i,j})] \text{ for all } j=2, \ldots, NJ \tag{A8b}$$

If the user specifies the temperature and pressure profiles instead of the stability, then using the definition $\theta^o \equiv T^o(\frac{p_c}{\bar{o}})^\kappa$, the density profile is defined by setting $\frac{-1}{\bar{\rho}^o}\frac{\partial \bar{\rho}^o}{\partial z} \equiv \frac{1}{T^o}\frac{\partial T^o}{\partial z} - \frac{p_R}{c_p p}\frac{\partial p^o}{z}$. The expression for $\bar{\rho}^o(z)$ becomes:

$$\bar{\rho}^o(z) = \rho_c \frac{T}{T^o(z)} (\frac{p^o(z)}{p_c})^\kappa \tag{A9}$$

where ρ_c, T_c, and p_c are arbitrary constants which have been set to 1.25 kg/m^3, 273K, and 10^5 kg/m-sec^2 respectively, and $\kappa = 2/7$. The finite difference form of equation (A9) is:

$$\bar{\rho}^o_{i,j} = \frac{\rho_c T_c}{T^o_{i,j}} (\frac{p^o_{i,j}}{p_c})^{2/7} \tag{A10}$$

The vorticity is calculated initially from the streamfunction by the expression $\bar{\zeta}^o = \nabla^2\bar{\psi}^o$. Since $\bar{\psi}^o$ is a function of z only, the expression for $\bar{\zeta}^o$ becomes:

$$\bar{\zeta}^o(z) = \frac{\partial^2\bar{\psi}^o(z)}{\partial z^2} \tag{A11}$$

The finite difference form of equation (A11) is:

$$\overline{\zeta}^{o}_{i,j} = \frac{1}{(\Delta z)^2}\left(\overline{\psi}^{o}_{i,j-1} + \overline{\psi}^{o}_{i,j+1} - 2\overline{\psi}^{o}_{i,j}\right) \text{ for } j=2,\ldots,NJ-1 \qquad \text{(A12a)}$$

$$\overline{\zeta}^{o}_{i,1} = \overline{\zeta}^{o}_{i,2} \qquad \text{(A12b)}$$

$$\overline{\zeta}^{o}_{i,NJ} = \overline{\zeta}^{o}_{i,NJ-1} \qquad \text{(A12c)}$$

Influence of the Barrier

Since Fourier transform methods are used to relax $\nabla^2\psi = \zeta$ for the streamfunction at each time step, all interior grid points in the field, including those on or inside the barrier, must be considered to be part of the flow. Values of ψ at these points are thus subject to change as a result of the transformations. To preserve the desired boundary condition on the barrier, the effect of the vorticity generated by each separate point on the barrier is superposed with the effect of the vorticity generated by all the internal points in the grid. Then, the streamfunction at each point on the barrier is expressed as a linear combination of the relaxation solutions associated with these vorticities (Roache, 1972):

$$\psi^{\ell} = \psi^{\ell}_{o} + \sum_{k=1}^{NPB} \alpha_k \psi^{\ell}_{k} \quad \text{for all } \ell = 1, \text{ NPB} \qquad \text{(A13)}$$

where ψ_o is the solution of $\nabla^2\psi_o = \zeta$, ψ_k is the solution of $\nabla^2\psi_k = \zeta_k$ due to a unit vorticity ζ_k at barrier point k, the superscript ℓ represents the values of ψ, ψ_o, and ψ_k at barrier point ℓ, and NPB is the number of

points on the barrier. The α_k's are determined at each time step from the linear system (A13). Then, at each grid point (i,j) in the system,

$$\psi_{i,j} = (\psi_o)_{i,j} + \sum_{k=1}^{NPB} \alpha_k (\psi_k)_{i,j} \qquad \text{(A14)}$$

Although the superposed solution ψ results from additional vorticity on the barrier, the solution is a valid one, since it satisfies $\nabla^2\psi = \zeta$ at all internal points in the flow, and satisfies all boundary conditions, including the barrier. The additional vorticity on the barrier introduces no perturbations in the vorticity of the flow at any time.

As an example in computing the α_k's, consider the case with three points on the barrier. The system of equations (A13) then becomes:

$$\begin{aligned}
\psi^1 &= \psi_o^1 + \alpha_1\psi_1^1 + \alpha_2\psi_2^1 + \alpha_3\psi_3^1 \\
\psi^2 &= \psi_o^2 + \alpha_1\psi_1^2 + \alpha_2\psi_2^2 + \alpha_3\psi_3^2 \qquad \text{(A15)} \\
\psi^3 &= \psi_0^3 + \alpha_1\psi_1^3 + \alpha_2\psi_2^3 + \alpha_3\psi_3^3
\end{aligned}$$

Applying Cramer's rule,

$$\alpha_1 = \frac{1}{D}\begin{vmatrix} (\psi^1 - \psi_o^1) & \psi_2^1 & \psi_3^1 \\ (\psi^2 - \psi_o^2) & \psi_2^2 & \psi_3^2 \\ (\psi^3 - \psi_o^3) & \psi_2^3 & \psi_3^3 \end{vmatrix} \qquad \text{(A16)}$$

$$= \beta_{11}(\psi^1 - \psi_o^1) + \beta_{12}(\psi^2 - \psi_o^2) + \beta_{13}(\psi^3 - \psi_o^3),$$

where D is the determinant of the matrix $\begin{bmatrix} \psi_1^1 & \psi_2^1 & \psi_3^1 \\ \psi_1^2 & \psi_2^2 & \psi_3^2 \\ \psi_1^3 & \psi_2^3 & \psi_3^3 \end{bmatrix}$

and $\beta_{11} = \frac{1}{D}\begin{vmatrix} \psi_2^2 & \psi_3^2 \\ \psi_2^3 & \psi_3^3 \end{vmatrix}$, $\beta_{12} = \frac{1}{D}\begin{vmatrix} \psi_2^1 & \psi_3^1 \\ \psi_2^3 & \psi_3^3 \end{vmatrix}$, $\beta_{13} = \frac{1}{D}\begin{vmatrix} \psi_2^1 & \psi_3^1 \\ \psi_2^2 & \psi_3^2 \end{vmatrix}$, (A17)

with similar expressions for α_2 and α_3. For the general case with NPB points on the barrier,

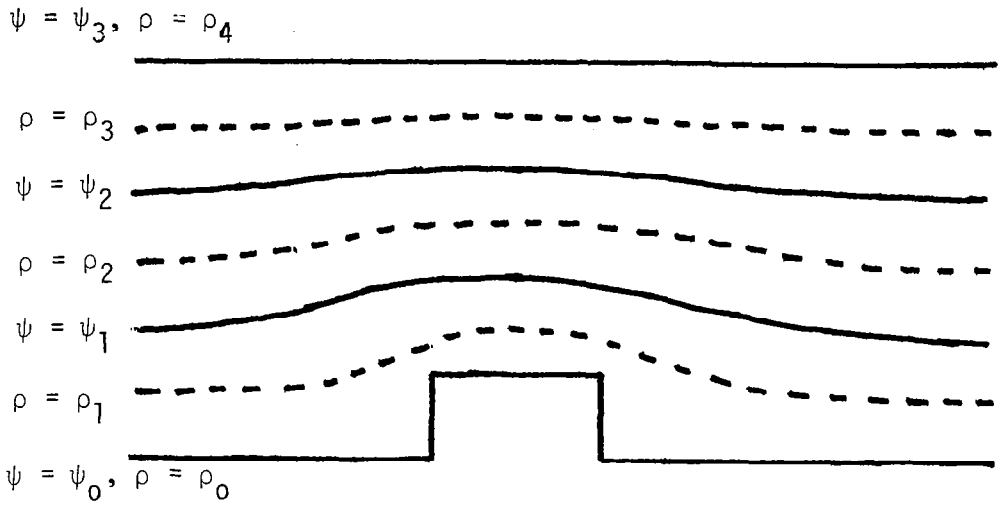
$$\alpha_k = \sum_{\ell=1}^{NPB} (\psi^\ell - \psi_o^\ell)\beta_{k\ell} \text{ , for all } k=1, \text{ NPB} \qquad \text{(A18)}$$

where the $\beta_{k\ell}$ need to be calculated only once. A Gaussian elimination scheme is used to calculate the determinants for the $\beta_{k\ell}$'s.

Specifying the Initial Conditions

At time step 0, the barrier is suddenly introduced into the flow by defining the bottom topography to be a line of constant ψ and ρ. To reduce the physical shock of introducing the barrier, a solution due to the barrier may be added to $\bar{\psi}^o$ without perturbing the vorticity in the flow. This is expressed as $\psi^o = \bar{\psi}^o + \psi_{barrier}$, where ψ^o is the streamfunction at time step 0, and $\nabla^2\psi_{barrier} = 0$. Since the isolines of ψ and ρ coincide everywhere as the model approaches a steady state, a similar perturbation is added to $\bar{\rho}^o$ so that the isolines of ψ^o and ρ^o approximately coincide, as

shown below.



The finite difference expressions for ψ^0, ρ^0, and ζ^0 are then:

$$\psi^0_{i,j} = \overline{\psi}^0_{i,j} + \sum_{k=1}^{NPB} \{(\overline{\psi_k})^0_{i,j} \cdot \sum_{\ell=1}^{NPB} [\psi_c - (\overline{\psi^\ell})^0]\} \qquad \text{(A19a)}$$

$$\rho^0_{i,j} = \overline{\rho}^0_{i,j'-1} + (\psi^0_{i,j} - \overline{\psi}^0_{i,j'-1}) \cdot (\overline{\rho}^0_{i,j'} - \overline{\rho}^0_{i,j'-1}) \Big/ (\overline{\psi}^0_{i,j'} - \overline{\psi}^0_{i,j'-1}) \qquad \text{(A19b)}$$

$$\zeta^0_{i,j} = \overline{\zeta}^0_{i,j} \qquad \text{(A19c)}$$

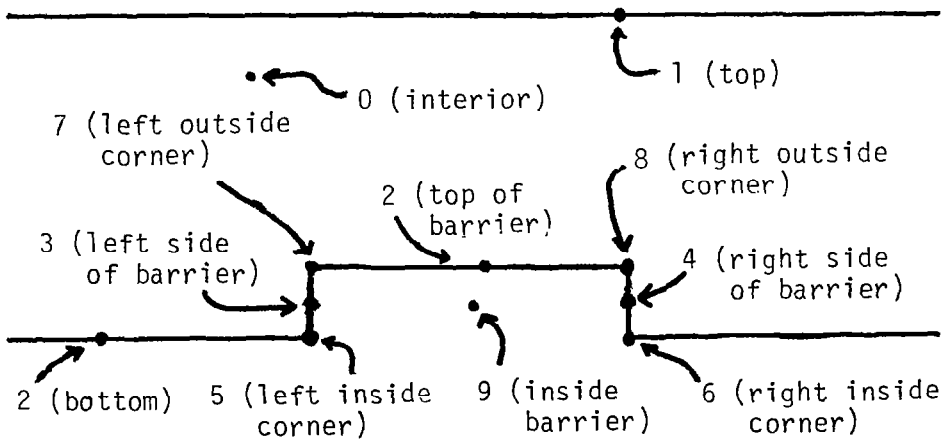
where, in equation (19b), $j' \geq 2$ is the lowest row number for which $|\overline{\psi}^0_{i,j'}| \geq |\psi^0_{ij}|$.

It should be noted that the potential flow scheme described above does not have a beneficial effect for every possible initial velocity profile $u^0(z)$. Specifically, if the velocity changes direction at upper levels, or is generally decreasing with height, then the code should be modified to

set ψ^0 and ρ^0 to $\bar{\psi}^0$ and $\bar{\rho}^0$ (except on the barrier, where $\psi^0 \equiv \psi_c$ and $\rho^0 \equiv \rho_c$).

Calculation of $J(\zeta,\psi)$, $J(\rho,\psi)$, and $\frac{\partial \rho}{\partial x}$

The long-term computational stability of the model depends on the finite difference form of the equations to be integrated. Arakawa (1966) devised a method to retard nonlinear computational instability in the equation $\frac{\partial \zeta}{\partial t} = J(\zeta,\psi)$ by conserving mean vorticity, mean kinetic energy, and mean square vorticity. This scheme is used by the program to calculate $J(\zeta,\psi)$ on the boundaries, and $J(\zeta,\psi)$ and $J(\rho,\psi)$ at interior grid points. The finite difference form, $J_{ij}(\zeta,\psi)$, depends on the location of the grid point, which may be one of ten types, as shown below.



After applying boundary conditions, and using the fact that ψ is constant on the upper boundary and $\psi \equiv 0$ on the lower boundary, the finite difference expressions for $J(\zeta,\psi)$ are as follows:

for type 0, $J_{ij}(\zeta,\psi) = \frac{-1}{12\Delta x\Delta z}[(\psi_{i,j-1} + \psi_{i+1,j-1} - \psi_{i,j+1} - \psi_{i+1,j+1})\zeta_{i+1,j}$

$+ (-\psi_{i-1,j-1} - \psi_{i,j-1} + \psi_{i-1,j+1} + \psi_{i,j+1})\zeta_{i-1,j}$

$+ (\psi_{i+1,j} + \psi_{i+1,j+1} - \psi_{i-1,j} - \psi_{i-1,j+1})\zeta_{i,j+1}$

$+ (-\psi_{i+1,j-1} - \psi_{i+1,j} + \psi_{i-1,j-1} + \psi_{i-1,j})\zeta_{i,j-1}$

$+ (\psi_{i+1,j} - \psi_{i,j+1})\zeta_{i+1,j+1} + (-\psi_{i,j-1} + \psi_{i-1,j})\zeta_{i-1,j-1}$

$+ (\psi_{i,j+1} - \psi_{i-1,j})\zeta_{i-1,j+1} + (-\psi_{i+1,j} + \psi_{i,j-1})\zeta_{i+1,j-1}]$ (A20a)

for type 1, $J_{ij}(\zeta,\psi) = \frac{-1}{6\Delta x\Delta z}[(\psi_{i,j-1} + \psi_{i+1,j-1} - 2\psi_{i,j})\zeta_{i+1,j}$

$+ (-\psi_{i-1,j-1} - \psi_{i,j-1} + 2\psi_{i,j})\zeta_{i-1,j}$

$+ (-\psi_{i+1,j-1} + \psi_{i-1,j-1})\zeta_{i,j-1}$

$+ (-\psi_{i,j-1} + \psi_{i,j})(\zeta_{i-1,j-1} - \zeta_{i+1,j-1})]$ (A20b)

for type 2, $J_{ij}(\zeta,\psi) = \frac{-1}{6\Delta x\Delta z}[(-\psi_{i,j+1} - \psi_{i+1,j+1})\zeta_{i+1,j}$

$+ (-\psi_{i-1,j+1} + \psi_{i,j+1})\zeta_{i-1,j}$

$+ (\psi_{i+1,j+1} - \psi_{i-1,j+1})\zeta_{i,j+1} + \psi_{i,j+1}(\zeta_{i-1,j+1} - \zeta_{i+1,j+1})]$ (A20c)

for type 3, $J_{ij}(\zeta,\psi) = \frac{-1}{6\Delta x\Delta z}[(-\psi_{i-1,j-1} + \psi_{i-1,j+1})\zeta_{i-1,j}$
$+ (-\psi_{i-1,j} - \psi_{i-1,j+1})\zeta_{i,j+1} + (\psi_{i-1,j-1} + \psi_{i-1,j})\zeta_{i,j-1}$ (A20d)
$+ \psi_{i-1,j}(\zeta_{i-1,j-1} - \zeta_{i-1,j+1})]$

for type 4, $J_{ij}(\zeta,\psi) = \frac{-1}{6\Delta x\Delta z}[(\psi_{i+1,j-1} - \psi_{i+1,j+1})\zeta_{i+1,j}$
$+ (\psi_{i+1,j} + \psi_{i+1,j+1})\zeta_{i,j+1} + (-\psi_{i+1,j-1} - \psi_{i+1,j})\zeta_{i,j-1}$
$+ \psi_{i+1,j}(\zeta_{i+1,j+1} - \zeta_{i+1,j-1})]$ (A20e)

for type 5, $J_{ij}(\zeta,\psi) = \frac{-1}{3\Delta x\Delta z}[\psi_{i-1,j+1}(\zeta_{i-1,j} - \zeta_{i,j+1})]$ (A20f)

for type 6, $J_{ij}(\zeta,\psi) = \frac{-1}{3\Delta x\Delta z}[\psi_{i+1,j+1}(\zeta_{i,j+1} - \zeta_{i+1,j})]$ (A20g)

for type 7, $J_{ij}(\zeta,\psi) = \frac{-1}{9\Delta x\Delta z}[(-\psi_{i,j+1} - \psi_{i+1,j+1})\zeta_{i+1,j}$
$+ (-\psi_{i-1,j-1} + \psi_{i-1,j+1} + \psi_{i,j+1})\zeta_{i-1,j} + (\psi_{i+1,j+1} -$
$\psi_{i-1,j} - \psi_{i-1,j+1})\zeta_{i,j+1} + (\psi_{i-1,j-1} + \psi_{i-1,j})\zeta_{i,j-1}$
$+ \psi_{i,j+1}(\zeta_{i-1,j+1} - \zeta_{i+1,j+1}) + \psi_{i-1,j}(\zeta_{i-1,j-1} - \zeta_{i-1,j+1})]$ (A20h)

for type 8, $J_{ij}(\zeta,\psi) = \frac{-1}{9\Delta x\Delta z}[(\psi_{i+1,j-1} - \psi_{i,j+1} - \psi_{i+1,j+1})\zeta_{i+1,j}$
$+ (\psi_{i-1,j+1} + \psi_{i,j+1})\zeta_{i-1,j} + (\psi_{i+1,j} + \psi_{i+1,j+1} - \psi_{i-1,j+1})\zeta_{i,j+1}$
$+ (-\psi_{i+1,j-1} - \psi_{i+1,j})\zeta_{i,j-1} + \psi_{i+1,j}(\zeta_{i-1,j+1} - \zeta_{i+1,j-1})$
$+ \psi_{i,j+1}(\zeta_{i-1,j+1} - \zeta_{i+1,j+1})]$ (A20i)

for type 9, $J_{ij}(\zeta,\psi) \equiv 0$ (A20j)

For point type 0, $J_{ij}(\rho,\psi)$ has a similar expression to equation (A20a).

For all other types, $J_{ij}(\rho,\psi) = 0$, which is equivalent to fixing the value of ρ on the boundaries. This method appears to be the most suitable for modeling physically significant transient motions in the system. When the Arakawa form of the Jacobian was used for $J(\rho,\psi)$ on the boundaries, the system remained stable, but often approached a steady state at an unacceptably slow rate, since ρ was free to vary on the boundaries while ψ was fixed.

The finite difference expressions for $\frac{\partial \rho}{\partial x}$ are:

$$\left(\frac{\partial \rho}{\partial x}\right)_{i,j} = \frac{\rho_{i+1,j} - \rho_{i-1,j}}{2\Delta x} \qquad \text{for point types } 0,7,8 \qquad \text{(A21a)}$$

$$= 0 \qquad \text{for types } 1,2,5,6,9 \qquad \text{(A21b)}$$

$$= \frac{\rho_{i,j} - \rho_{i-1,j}}{\Delta x} \qquad \text{for type } 3 \qquad \text{(A21c)}$$

$$= \frac{\rho_{i+1,j} - \rho_{i,j}}{\Delta x} \qquad \text{for type } 4. \qquad \text{(A21d)}$$

Relaxation of $\nabla^2\psi = \zeta$ for ψ

The finite difference form of the equation $\nabla^2\psi = \zeta$ is:

$$\zeta_{i,j} = \frac{\psi_{i-1,j} + \psi_{i+1,j} - 2\psi_{i,j}}{(\Delta x)^2} + \frac{\psi_{i,j-1} + \psi_{i,j+1} - 2\psi_{i,j}}{(\Delta z)^2} \qquad \text{(A22)}$$

The exact, noniterative solution of equation (A22), with periodic boundary conditions in x, may be expressed as the sum of discrete Fourier components $c_n(z)$ in x at each level of z:

$$\psi(x,z) = \frac{1}{N_x} \sum_{n=0}^{N_x-1} c_n(z) e^{inkx} \tag{A23}$$

where $k = \frac{2\pi}{N_x \Delta x}$; $x = p\Delta x$ for $p=0,\ldots,N_x-1$; $z = q\Delta z$ for $q=1,\ldots,N_z$;

$N_x = NI - 2$, the number of unique grid points in the x-direction; and

$N_z = NJ - 2$, the number of interior grid points in the z-direction.

The $c_n(z)$ are calculated from the boundary values of ψ, and from the Fourier components $d_n(z)$ of ζ (where $\zeta(x,z) = \frac{1}{N_x} \sum_{n=0}^{N_x-1} d_n(z) e^{inkx}$), through a system of finite difference equations which result from marching in the z-direction for each component n.

The relationship between the Fourier components c_n and d_n is derived as follows:

$$\zeta(p\Delta x, q\Delta z) = \frac{1}{N_x} \sum_{n=0}^{N_x-1} d_n(q\Delta z) e^{inkp\Delta x} = \frac{\partial^2 (p\Delta x, q\Delta z)}{\partial (p\Delta x)^2} + \frac{\partial^2 \psi(p\Delta x, q\Delta z)}{\partial (q\Delta z)^2}$$

$$= \frac{1}{N_x} \left(\frac{1}{(\Delta x)^2} \frac{\partial^2}{\partial p^2} + \frac{1}{(\Delta z)^2} \frac{\partial^2}{\partial q^2} \right) \sum_{n=0}^{N_x-1} c_n(q\Delta z) e^{inkp\Delta x} \tag{A24}$$

$$= \frac{1}{N_x} \sum_{n=0}^{N_x-1} \left[c_n(q\Delta z) \frac{1}{(\Delta x)^2} \frac{d^2}{dp^2} e^{inkp\Delta x} + e^{inkp\Delta x} \frac{1}{(\Delta z)^2} \frac{d^2}{dq^2} c_n(q\Delta z) \right].$$

Using a centered finite difference approximation for the second derivative,

$$\frac{1}{(\Delta x)^2} \frac{d^2}{dp^2} e^{inkp\Delta x} = \frac{e^{ink(p+1)\Delta x} + e^{ink(p-1)\Delta x} - 2e^{inkp\Delta x}}{(\Delta x)^2} \tag{A25}$$

$$= e^{inkp\Delta x} \frac{2}{(\Delta x)^2} [\cos(nk\Delta x) - 1]$$

and $$\frac{1}{(\Delta z)^2} \frac{d^2}{dq^2} c_n(q\Delta z) = \frac{c_n[(q+1)\Delta z] + c_n[(q-1)\Delta z] - 2c_n(q\Delta z)}{(\Delta z)^2} \tag{A26}$$

Equation (A24) then becomes:

$$\sum_{n=0}^{N_x-1} d_{n,q} e^{inkp\Delta x} = \sum_{n=0}^{N_x-1} \{c_{n,q} \frac{2}{(\Delta x)^2} [\cos(\frac{2\pi n}{N_x})-1] e^{inkp\Delta x}$$

$$+ \frac{c_{n,q+1} + c_{n,q-1} - 2c_{n,q}}{(\Delta z)^2} e^{inkp\Delta x}\} \qquad \text{(A27)}$$

where $c_{n,q} \equiv c_n(q\Delta z)$. Since each Fourier component responds independently of the others, the relationship under the summation holds separately for each n:

$$d_{n,q} = c_{n,q} \cdot \frac{2}{(\Delta x)^2} [\cos(\frac{2\pi n}{N_x}) - 1] + \frac{c_{n,q+1} + c_{n,q-1} - 2c_{n,q}}{(\Delta z)^2} \qquad \text{(A28)}$$

which becomes:

$$-c_{n,q-1} + [2 + (k')^2]c_{n,q} - c_{n,q+1} = (\Delta z)^2 d_{n,q} \qquad \text{(A29)}$$

where $(k')^2 \equiv 2 \frac{(\Delta z)^2}{(\Delta x)^2} [1 - \cos(\frac{2\pi n}{N_x})]$,

For each n=0,. . ., $N_x - 1$, equation (A29) comprises a set of N_z linear equations in N_z unknowns $c_{n,q}$, for q = 1,. . .. N_z, and is equivalent to the matrix equation $Ac_n = b$:

$$\begin{pmatrix} 2+k'^2 & -1 & & & \\ -1 & 2+k'^2 & -1 & & \\ & -1 & \ddots & \ddots & \\ & & \ddots & & -1 \\ & & & -1 & 2+k'^2 \end{pmatrix}_{N_z,N_z} \begin{pmatrix} c_{n,1} \\ \vdots \\ c_{n,N_z} \end{pmatrix}_{N_z,1} =$$

$$\begin{pmatrix} c_{n,o}-(\Delta z)^2 d_{n,1} \\ -(\Delta z)^2 d_{n,2} \\ \vdots \\ -(\Delta z)^2 d_{n,N_z-1} \\ c_{n,N_z+1} - (\Delta z)^2 d_{n,N_z} \end{pmatrix}_{N_z,1} \qquad \text{(A30)}$$

The $d_{n,q}$ terms are found by taking the Fourier series of ζ at each interior level $z = q\Delta z$, $q = 1,\ldots, N_z$, and the $c_{n,o}$ and c_{n,N_z+1} terms are found by taking the Fourier series of ψ at the upper and lower boundaries, as follows:

$$d_{n,q} = \sum_{p=0}^{N_x-1} \zeta(p\Delta x, q\Delta z) e^{-inkp\Delta x} \qquad \text{(A31a)}$$

$$c_{n,o} = \sum_{p=0}^{N_x-1} \psi(p\Delta x, 0) e^{-inkp\Delta x} \qquad \text{(A31b)}$$

$$c_{n,N_z+1} = \sum_{p=0}^{N_x-1} \psi[p\Delta x, (N_z+1)\Delta z] e^{-inkp\Delta x} \qquad \text{(A31c)}$$

From this, the $c_{n,q}$ terms for each $n = 0,\ldots, N_x - 1$, for all $q=1,\ldots,N_z$

are found by inverting the tridiagonal matrix A:

$$c_n = A^{-1} b, \text{ for each } n=0,\ldots,N_x-1 \ . \tag{A32}$$

Then, ψ is obtained by taking the inverse Fourier transform of the $c_{n,q}$:

$$\psi(p\Delta x, q\Delta z) = \frac{1}{N_x} \sum_{n=0}^{N_x-1} c_{n,q} e^{inkp\Delta x} \ . \tag{A33}$$

The scheme for calculating the $c_{n,q}$ for each n is the following:

$$a_d \equiv 2\{1+ \frac{(\Delta z)^2}{(\Delta x)^2} [1-\cos(\frac{2\pi n}{N_x})]\}$$

$$u_p(1) = a_d, \ f(1) = b(1)$$

$$\left.\begin{aligned} &\text{then } u_\ell = \frac{-1}{u_p(q-1)} \\ &u_p(q) = a_d + u_\ell \\ &f(q) = b(q) - u_\ell \cdot f(q-1) \end{aligned}\right\} \text{ for } q=2,\ldots,N_z \tag{A34}$$

$$\text{then } c_{n,N_z} = \frac{f(N_z)}{u_p(N_z)}$$

$$\text{and } c_{n,q} = \frac{f(q) + c_{n,q+1}}{u_p(q)} \qquad \text{for } q=N_z-1,\ldots,1.$$

Calculation of u, Ri, and w

These quantities are calculated from ψ, ρ, and ζ at any time step specified by the user. Since the values are not used for any subsequent time step, the finite difference equations for u are smoothed by using cubic spline function interpolation to calculate continuous first and second derivatives of ψ and ρ in z.

For each column of x, the procedure is to match the first and second derivatives of subsequent pairs of NJ-1 cubic polynomials $q_j(z)$ (j = 2,..., NJ), at NJ-2 internal grid points, z_j(j=2,...,NJ-1), given boundary conditions at z_1 and z_{NJ}. Let these polynomials have the form:

$$q_j(z) = tv_j + (1-t)v_{j-1} + t\Delta z(1-t)[(k_{j-1} - d_j)(1-t) - (k_j-d_j)t], \quad (A35)$$

where $t = \frac{z-z_{j-1}}{\Delta z}$, $d_j = \frac{v_j-v_{j-1}}{\Delta z}$, v_j = the value of ψ or ρ at point j, and $k_j = \frac{dq_j(z_j)}{dz}$ (Dahlquist and Bjorck, 1974). Then the expressions for the first and second derivatives in z become:

$$\frac{dq_j(z)}{dz} = \frac{v_j-v_{j-1}}{\Delta z} + (3t^2-4t+1)(k_{j-1}-d_j)+(3t^2-2t)(k_j-d_j) \quad (A36)$$

and $$\frac{d^2q_j(z)}{dz^2} = \frac{1}{\Delta z}[(6t-4)(k_{j-1}-d_j)+(6t-2)(k_j-d_j)] . \quad (A37)$$

These polynomials satisfy the relationships:

$$\frac{dq_{j+1}(z_j)}{dz} = \frac{dq_j(z_j)}{dz} = k_j \qquad \text{for } j=2,...,NJ-1 \quad (A38)$$

$$\text{and } \frac{d^2q_j(z_j)}{dz^2} = \frac{d^2q_{j+1}(z_j)}{dz^2} = \frac{1}{\Delta z}\left[-4k_j+6d_{j+1}-2k_{j+1}\right] \text{ for } j=2,\ldots,NJ-1 \quad (A39)$$

$$\text{provided that } k_{j-1}+4k_j+k_{j+1}=3(d_{j+1}+d_j)\ . \quad (A40)$$

The boundary conditions are:

$$\frac{d^2q_2(z_1)}{dz^2} = \frac{\partial^2 v_1}{\partial z^2} = a' = \frac{1}{\Delta z}\left[-4k_1 + 6d_2 - 2k_2\right] \text{ or } 2k_1+k_2=3d_2-\frac{a'\Delta z}{2} \quad (A41a)$$

$$\frac{d^2q_{NJ}(z_{NJ})}{dz^2} = \frac{\partial^2 v_{NJ}}{\partial z^2} = b' = \frac{1}{\Delta z}\left[2k_{NJ-1} - 6d_{NJ} + 4k_{NJ}\right] \text{ or}$$

$$k_{NJ-1} + 2k_{NJ} = 3d_{NJ} + \frac{b'\Delta z}{2} \quad (A41b)$$

Equations (A41a), (A40), and (A41b) comprise a set of NJ linear equations in NJ unknowns k_j, j=1,...,NJ, and are equivalent to the matrix equation $Ac_n=b$:

$$\begin{pmatrix} 2 & 1 & 0 & & & & \\ 1 & 4 & 1 & & & & \\ 0 & 1 & 4 & \ddots & & & \\ & \ddots & \ddots & \ddots & \ddots & & \\ & & \ddots & 4 & 1 & 0 & \\ & & & 1 & 4 & 1 & \\ & & & 0 & 1 & 2 & \end{pmatrix}_{NJ,NJ} \begin{pmatrix} k_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ k_{NJ} \end{pmatrix}_{NJ,1} = \begin{pmatrix} 3d_2 - \frac{a'\Delta z}{2} \\ 3(d_2+d_3) \\ \cdot \\ \cdot \\ \cdot \\ 3(d_{NJ-1}-d_{NJ}) \\ \frac{b'\Delta z}{2} + 3d_{NJ} \end{pmatrix}_{NJ,1} \quad (A42)$$

The $\frac{\partial v_j}{\partial z} \equiv k_j$ are found by inverting the tridiagonal matrix A according to the following scheme:

$$u_p(1) = a_d(1)$$

$$f(1) = b(1)$$

$$\left.\begin{aligned} u_\ell &= \frac{-1}{u_p(j-1)} \\ u_p(j) &= a_d(j) + u_\ell \\ f(j) &= b(j) + u_\ell \cdot f(j-1) \end{aligned}\right\} \quad \text{for } j = 2,\ldots, NJ \qquad \text{(A43)}$$

$$\text{then } k_{NJ} = \frac{f(NJ)}{u_p(NJ)}$$

$$\text{and } k_j = \frac{f(j) - k_{j+1}}{u_p(j)} \quad \text{for } j = NJ - 1,\ldots,1$$

where $a_d(t)=2$, $a_d(j)=4$ for $j=2,\ldots,NJ-1$, and $a_d(NJ)=2$.

Then, by equation (39) the second derivative of v_j becomes:

$$\frac{\partial^2 v_j}{\partial z^2} = \frac{1}{\Delta z}\left[-4\,\frac{\partial v_j}{\partial z} + 6\left(\frac{v_{j+1}-v_j}{\Delta z}\right) - 2\,\frac{\partial v_{j+1}}{\partial z}\right] \quad \text{for } j=2,\ldots,NJ-1 \qquad \text{(A44)}$$

The finite difference expressions for $u = \frac{-\partial\psi}{\partial z}$ and $Ri = \frac{-g}{\rho}\frac{\partial\rho}{\partial z} \Big/ \left(\frac{\partial^2\psi}{\partial z^2}\right)^2$ are as follows:

$$u_{i,j} = \frac{-\partial\psi_{i,j}}{\partial z} \qquad \text{(A45)}$$

$$Ri_{i,j} = \frac{-g}{\rho_{i,j}}\,\frac{\partial\rho_{i,j}}{\partial z} \Big/ \left(\frac{\partial^2\psi_{i,j}}{\partial z^2}\right)^2 \qquad \text{(A46)}$$

The first derivative of ψ is obtained for each column ($i=1,\ldots,NI$) by applying equation (A42) with $d_{i,j} \equiv \frac{\psi_{i,j}-\psi_{i,j-1}}{\Delta z}$, $a' \equiv \overline{\zeta}^{o}_{i,1}$, and $b' \equiv \overline{\zeta}^{o}_{i,NJ}$. Then the second derivative of ψ may be expressed as:

$$\frac{\partial^2\psi_{i,j}}{\partial z^2} = \frac{1}{\Delta z}\left[-4\,\frac{\partial\psi_{i,j}}{\partial z} + 6\left(\frac{\psi_{i,j+1}-\psi_{i,j}}{\Delta z}\right) - 2\,\frac{\partial\psi_{i,j+1}}{\partial z}\right] \tag{A47}$$

The first derivative of ρ is obtained for each column (i=1,...,NI) by applying equation (A42) with $d_{i,j} \equiv \frac{\rho_{i,j} - \rho_{i,j-1}}{\Delta z}$, $a' \equiv 0$, and $b' \equiv 0$.

The finite difference expression for $w = \frac{\partial\psi}{\partial x}$ is:

$$w_{i,j} = \frac{\psi_{i+1,j} - \psi_{i-1,j}}{2\Delta x} \tag{A48}$$

This equation may be solved by expressing $\psi(x,z)$ and $w(x,z)$ as a sum of discrete Fourier components $c_n(z)$ and $d_n(z)$ in x at each level of z:

$$\psi(x,z) = \frac{1}{N_x}\sum_{n=0}^{N_x-1} c_n(z)e^{inkx} \tag{A49}$$

$$w(x,z) = \frac{1}{N_x}\sum_{n=0}^{N_x-1} d_n(z)e^{inkx} \tag{A50}$$

where $k = \frac{2\pi}{N_x\Delta x}$, $x = p\Delta x$ for $p=0,\ldots,N_x-1$, and $N_x=NI-2$. The relationship between the Fourier components c_n and d_n is derived as follows:

$$w(p\Delta x,z) = \frac{1}{N_x}\sum_{n=0}^{N_x-1} d_n(z)e^{inkp\Delta x} = \frac{\partial\psi(p\Delta x,z)}{\partial(p\Delta x)}$$

$$= \frac{1}{N_x\Delta x}\frac{\partial}{\partial p}\sum_{n=0}^{N_x-1} c_n(z)e^{inkp\Delta x} = \frac{1}{N_x}\sum_{n=0}^{N_x-1} c_n(z)\,\frac{1}{\Delta x}\frac{d}{dp}\,e^{inkp\Delta x} \tag{A51}$$

$$= \frac{1}{N_x}\sum_{n=0}^{N_x-1} c_n(z)\left[\frac{e^{ink(p+1)\Delta x} - e^{ink(p-1)\Delta x}}{2\Delta x}\right]$$

$$= \frac{1}{N_x}\sum_{n=0}^{N_x-1} c_n(z)\,\frac{i}{\Delta x}\sin(nk\Delta x)e^{inkp\Delta x}$$

Since each Fourier component responds independently of the others, the terms under the summation are equal for each n:

$$d_n(z) = c_n(z) \frac{i}{\Delta x} \sin(nk\Delta x) \tag{A52}$$

The $c_n(z)$ terms are found by taking the Fourier series of ψ at each level of z:

$$c_n(z) = \sum_{n=0}^{N_x-1} \psi(p\Delta x, z) e^{-inkp\Delta x} \tag{A53}$$

Then equation (A52) yields $d_n(z)$, and w is obtained by taking the inverse Fourier transform of the $d_n(z)$:

$$w(p\Delta x, z) = \frac{1}{N_x} \sum_{n=0}^{N_x-1} d_n(z) e^{inkp\Delta x} \tag{A54}$$

DESCRIPTION OF CARD INPUT DATA

Space and Time Data

Space and Time Card: NI,NJ,NT,NPB,IBT,ISAVE,DELTAX,DELTAZ,DELTAT

(format: 6I4,1P3D10.2)

NI — Number of grid points in the x-direction (i.e., number of columns). Due to restrictions imposed by the fast Fourier transform routine, NI must equal $2^n + 2$, where n is a positive integer.

NJ — Number of grid points in the z-direction (i.e., number of rows).

NT — Number of time steps.

NPB — Number of grid points on the surface of the barrier, excluding points on the lowest row. Refer to explanation of barrier data below.

IBT — Beginning time step. If IBT = 0, then read wind velocity and stability data. If IBT.NE.0, then the data saved at time step IBT from a previous run is to be input from a tape or permanent file.

ISAVE — Data retention indicator. If ISAVE = 1, then data at the last time step is to be saved on a tape or permanent file. Otherwise, set ISAVE = 0.

DELTAX — Grid spacing in the x-direction (in meters).

DELTAZ — Grid spacing in the z-direction (in meters).

DELTAT — Time step interval (in seconds).

Barrier Data

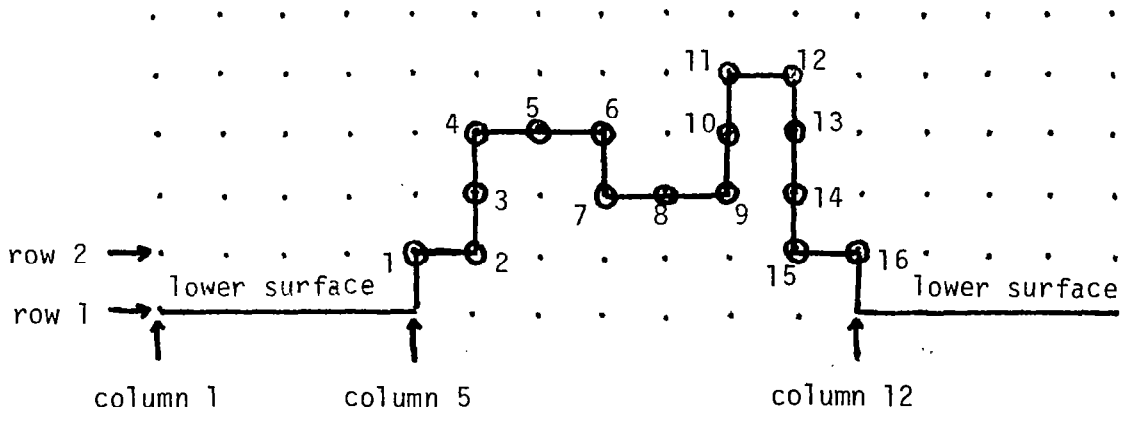
Barrier Card(s): (IB(I),JB(I),I=1,NPB)

(format: 20I4)

IB(I) Column number of the Ith point on the barrier.

JB(I) Row number of the Ith point on the barrier.

As shown in the example below, IB(I) and JB(I) are to be specified in a continuous fashion along the surface of the barrier, beginning with the leftmost grid point in the second row, and ending with the rightmost grid point in the second row. The surface of the barrier must connect adjacent grid points only horizontally or vertically, never diagonally. No barrier grid point may ever be specified in the first row. The position and shape of the barrier may be arbitrary, but only one barrier is permitted, and its width may not exceed (NI-4) horizontal grid intervals. On the barrier pictured below, NPB = 16, with IB (1) = 5 JB(1) = 2, IB(2) = 6, JB(2) = 2, etc. The width of the barrier is 7 horizontal grid intervals.



Graphical Output Data

The output of the program consists of shaded, line printer graphical displays of the streamfunction (ψ), density (ρ), vorticity (ζ), horizontal velocity (u), vertical velocity (w), or Richardson number (Ri) at any time step specified by the user. The shading consists of alternating clear areas and printed areas. To emphasize aspects of the flow, the user may output any segment of the entire grid, vary the vertical scale of the printout, or vary the resolution of the shading.

Grid Plot Card: IS,JS,IN,JN,NJSM1

(format: 5I4)

IS	First grid point in the x-direction to be plotted.
JS	First grid point in the z-direction to be plotted.
IN	Number of grid points in the x-direction to be plotted.
JN	Number of grid points in the z-direction to be plotted.
NJSM1	Number of lines of print between two vertical grid points.

Shading Level Card: NPSILV,NRHOLV,NZTALV,NULV,NWLV,NRILV

(format: 6I4)

NPSILV	Maximum number of levels of shading on ψ graphs.
NRHOLV	Maximum number of levels of shading on ρ graphs.
NZTALV	Maximum number of levels of shading on ζ graphs.
NULV	Maximum number of levels of shading on u graphs.
NWLV	Maximum number of levels of shading on w graphs.
NRILV	Maximum number of levels of shading on Ri graphs.

ψ Plot Card: IPSIGR(I) ($I \leq 20$)

(format: 20I4)

IPSIGR Sequence of time steps at which ψ is to be printed.

ρ Plot Card: IRHOGR(I) ($I \leq 20$)

(format: 20I4)

IRHOGR Sequence of time steps at which ρ is to be printed.

ζ Plot Card: IZTAGR(I) ($I \leq 20$)

(format: 20I4)

IZTAGR Sequence of time steps at which ζ is to be printed.

u Plot Card: IUGR(I) ($I \leq 20$)

(format: 20I4)

IUGR Sequence of time steps at which u is to be printed.

w Plot Card: IWGR(I) ($I \leq 20$)

(format: 20I4)

IWGR Sequence of time steps at which w is to be printed.

Ri Plot Card: IRIGR(I) ($I \leq 20$)

(format: 20I4)

IRIGR Sequence of time steps at which $\log_{10}(Ri)$ is to be printed.

Note that all variables need not be output at the same time steps, but that all time steps must be in order for any one variable. If a particular variable is not to be printed at all during a job, then a "-1" must be punched in columns 3 and 4 of the appropriate card. Time step 0 is a legitimate time step at which any field may be printed.

If IBT.NE.0, there are no more cards to be read beyond this point.

Wind Velocity Data

This data determines the initial values of $\overline{\psi}^0(z)$ and $\overline{\zeta}^0(z)$.

Wind Card: ICASE

(format: I4)

ICASE Wind velocity profile indicator.

If ICASE = 1, this is a sounding data case, and temperature and pressure data are to be read in addition to wind data.

Sounding Cards: U(J),T(J),P(J) for J = 1,...,NJ

(format: 1P3D10.2)

U(J) Horizontal wind at row J (in knots)

T(J) Temperature at row J (in ^{0}C).

P(J) Pressure at row J (in mb).

If ICASE = 1, there are no more cards to be read beyond this point.

If ICASE = 2, this is a constant velocity profile case.

Velocity Card: U1

(format: 1PD10.2)

U1 Horizontal velocity at all levels (in m/sec).

Then, U(J) = U1 for J = 1,...,NJ.

If ICASE = 3, this is a constant shear profile case.

Velocity and Shear Card: U1,C

(format: 1P2D10.2)

U1 Horizontal velocity at row 1 (in m/sec).

C Vertical shear (in sec^{-1}).

Then, U(J) = U1 + C * DELTAZ * (J - L) for J = 1,...,NJ.

If ICASE = 4, this is an exponential profile case.

Velocity and Shear Card: U1,U2,C

(format: 1P3D10.2)

U1 Constant velocity to be added to the profile at all levels (in m/sec).

U2 Base horizontal velocity (in m/sec).

C Vertical shear divided by U2 (in m^{-1})

Then, U(J) = U1 + U2 * EXP(C*DELTAZ*(J-L)) for J = 1,...,NJ.

If ICASE = 5, this is a hyperbolic tangential profile case.

Velocity Card: U1,U2,LI,MDZ

(format: 1P2D10.2,2I4)

U1 Horizontal velocity to be added to the profile at all levels (in m/sec).

U2 Base horizontal velocity (in m/sec).

LI Row number at which the profile has its inflection point.

MDZ Number of rows away from the inflection point at which U1 is deflected by the amount U2. The sign of MDZ determines the sign of (U(NJ)-U(1)).

Then, U(J) = U1 + U2 * DTANH((J-LI)/MDZ) for J = 1,...,NJ.

If ICASE = 6, this is a case, other than a sounding data case, for which the velocity is to be specified at each row.

Velocity Card(s): U(J) for J = 1,...,NJ

(format: 1P8D10.2)

U(J) Horizontal velocity at row J (in m/sec).

Stability Data

This data determines the initial values of $\overline{\rho}^{o}(z)$.

Stability Card: JCASE

(format: I4)

JCASE = Stability profile indicator.

IF JCASE = 1, this is a constant lapse rate case.

Lapse Rate Card: GAMMA, TO

(format: 1P2D10.2)

GAMMA Lapse rate (in ^{o}C/m).

TO Reference temperature (in ^{o}C)

Then, S(J) = (DALR - GAMMA) / TO for J = 1,...,NJ,

where DALR = dry adiabatic lapse rate.

If JCASE = 2, this is a constant Brunt-Vaisala frequency case.

Frequency Card: BV

(format: 1PD10.2)

BV Brunt-Vaisala frequency (in sec^{-1}).

Then, S(J) = BV ** 2 / G for J = 1,...,NJ; where G =

acceleration of gravity.

If JCASE = 3, this is a constant Richardson number case.

Ri Card: RI

(format: 1PD10.2)

RI Richardson number.

Then, S(1) = R * (U(1) - U(0)) ** 2, where

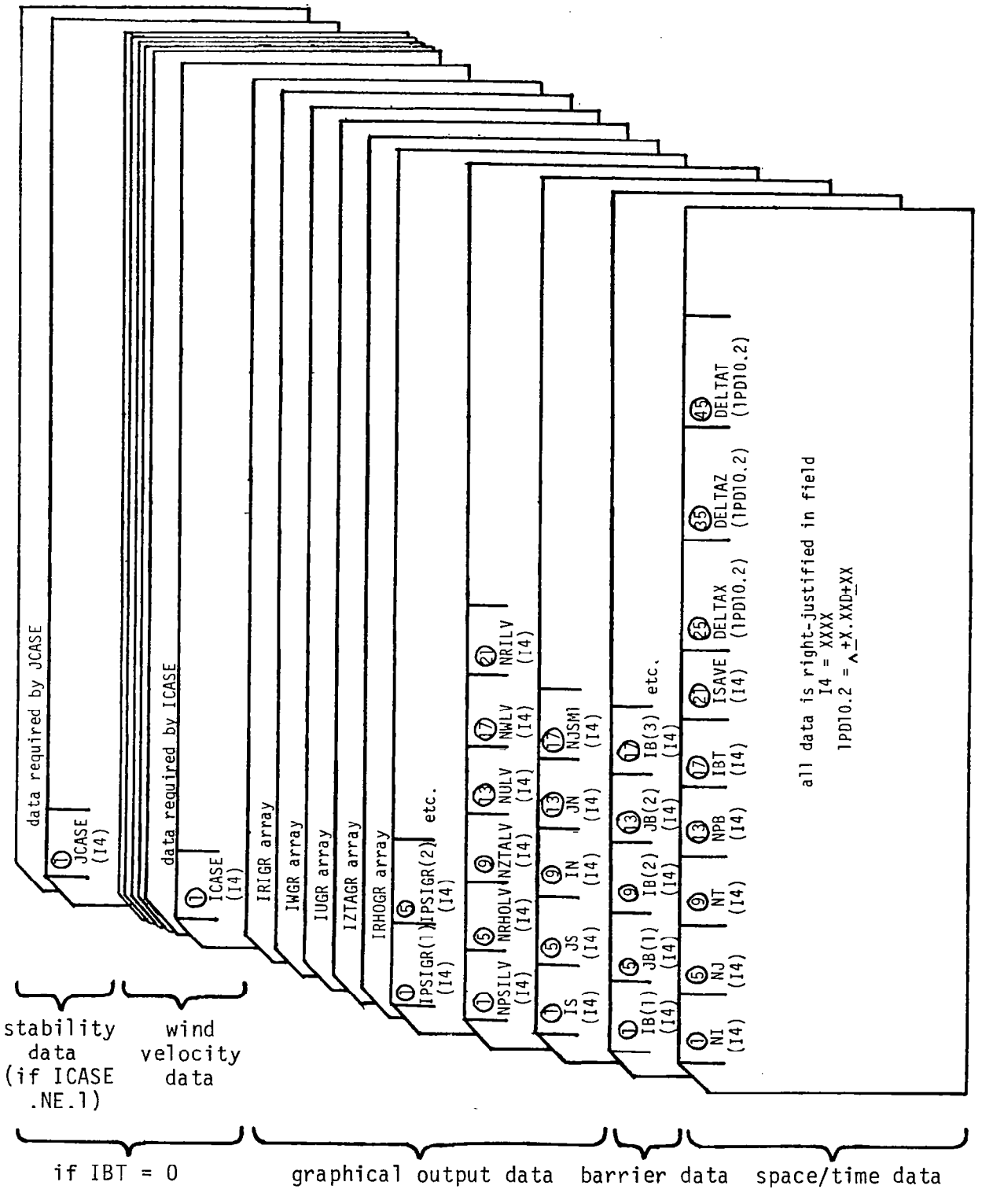
R = RI / (G * DELTAZ ** 2),

S(J) = R * (U(J+1) - U(J-1)) ** 2 / 4

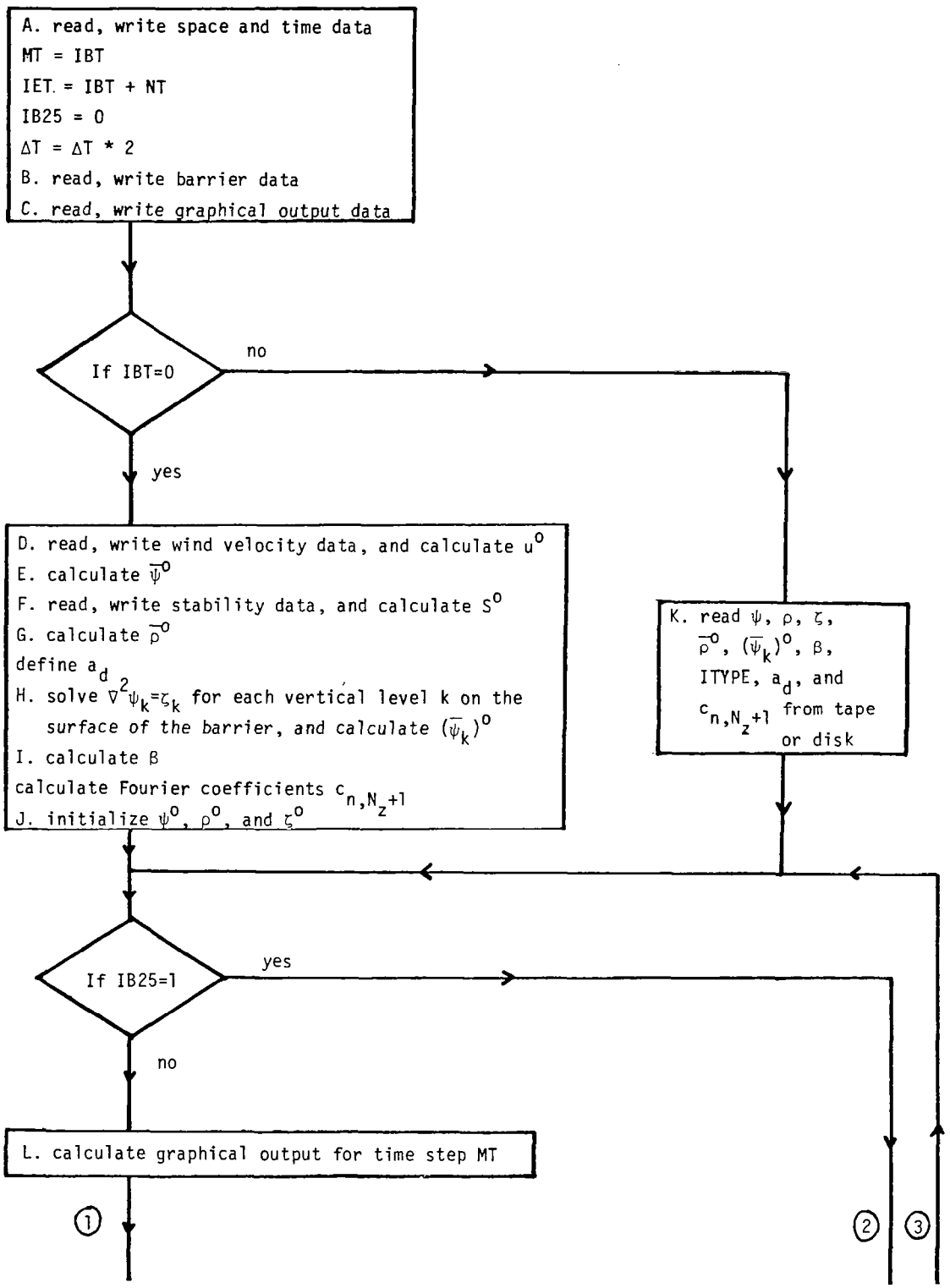
for J = 2, ...,NJ-1, and

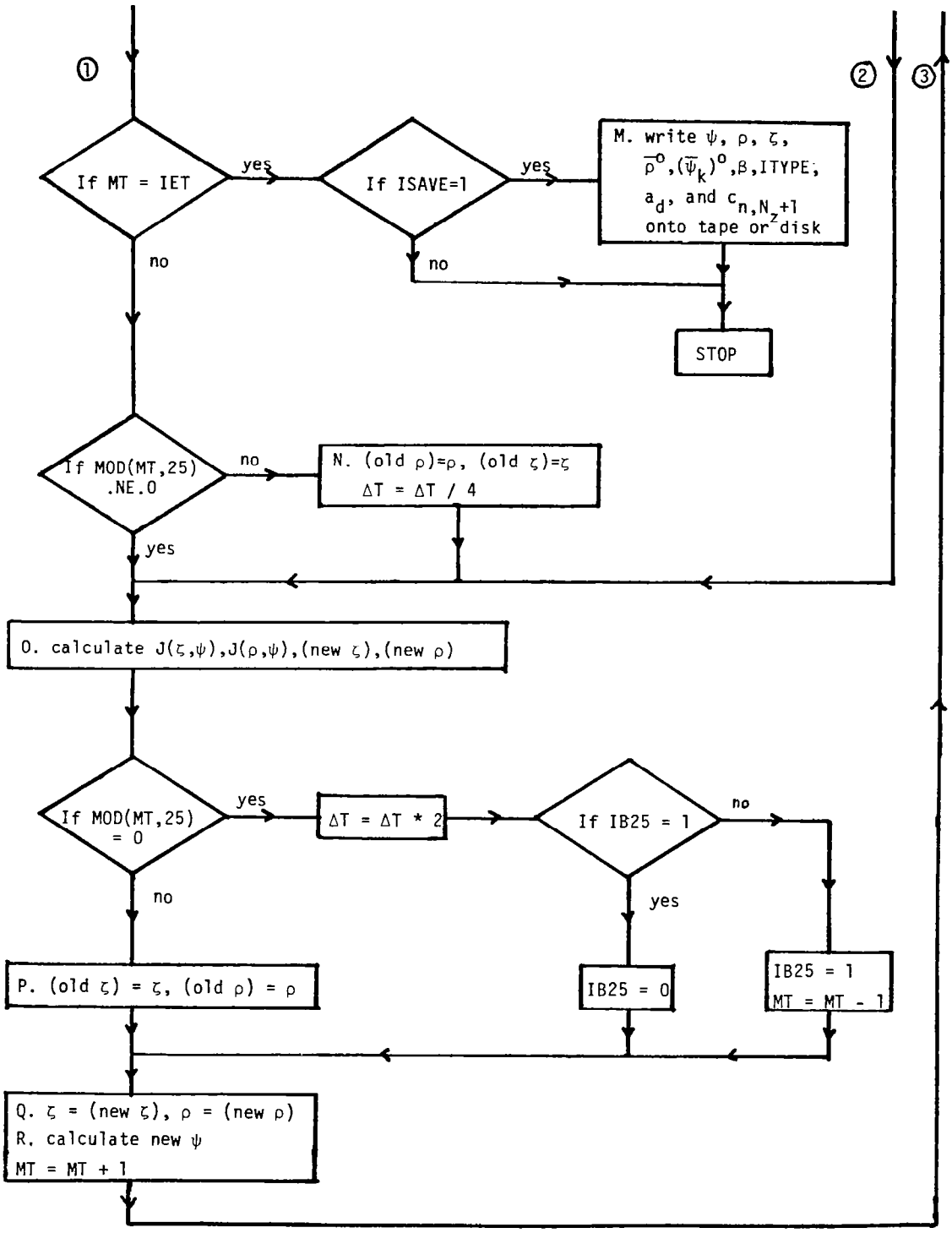
S(NJ) = R * (U(NJ) - U(NJ-1)) **2.

Summary of input card data



FLOW OUTLINE OF THE PROGRAM




①
②
③
If MT = IET
yes
If ISAVE=1
yes
M. write ψ, ρ, ζ, $\overline{\rho}^{o}$,$(\overline{\psi}_k)^{o}$,β,ITYPE, a_d, and c_{n,N_z+1} onto tape or disk
no
no
STOP
If MOD(MT,25) .NE.0
no
N. (old ρ)=ρ, (old ζ)=ζ
ΔT = ΔT / 4
yes
O. calculate J(ζ,ψ),J(ρ,ψ),(new ζ),(new ρ)
If MOD(MT,25) = 0
yes
ΔT = ΔT * 2
If IB25 = 1
no
yes
no
P. (old ζ) = ζ, (old ρ) = ρ
IB25 = 0
IB25 = 1
MT = MT - 1
Q. ζ = (new ζ), ρ = (new ρ)
R. calculate new ψ
MT = MT + 1

LIST OF VARIABLES IN THE PROGRAM

For reference, the variables are briefly described in alphabetical order, with array dimensions, and symbolic references, if any.

Variables in the Main Routine

A(NI,NJ)	Temporary array for storage of fields to be plotted.
AD(NI)	Diagonal elements of the matrix used to relax $\nabla^2\psi = \zeta$ for x. (a_d)
AJZP	Jacobian for ζ and ψ, $J(\zeta,\psi)$
AK(NI)	Coefficients used to calculate vertical velocity from ψ.
ARGI	Argument used for sinusoidal functions in AD and AK.
AS(NJ)	Diagonal elements of the spline matrix. (a_d)
A1 through A9	Temporary storage for values of ψ at selected grid points.
BETA(NPB,NPB)	Coefficients used to calculate α. (β)
BLANK4	4 blank print symbols.
BV	Brunt-Vaisala frequency. (N)
B1 through B8	Temporary terms for the calculation of Jacobian terms.
C	Shear in horizontal velocity. (c)
CDZ	C divided by DELTAZ.
CFAC	Cofactor sign divided by the determinant of the α matrix.
CKTMPS	Conversion factor from knots to meters/sec.
CRI	Constant Richardson number.
DALR	Dry adiabatic lapse rate. (γ_d)
DELTAT	Time step interval. (Δt)
DELTAX	Grid interval in the horizontal direction. (Δx)
DELTAZ	Grid interval in the vertical direction. (Δz)
DMGDTO	(DALR - GAMMA) / STMP.
DRHO	Density differential in the horizontal.

DXM2	DELTAX multiplied by 2.
DXSQ	DELTAX squared.
DZD2	DELTAZ divided by 2.
DZD6	DELTAZ multiplied by 6.
DZM2	DELTAZ multiplied by 2.
DZSQ	DELTAZ squared.
DZ2DX2	DZSQ divided by DXSQ.
FAC	-1 / (12 * DELTAX * DELTAZ)
FACM2	FAC multiplied by 2.
FACM4	FAC multiplied by 4.
FACM43	FAC multiplied by 4/3.
FTI(NI,NJ)	Imaginary part of the Fourier transform of ψ. ($Im(C_{n,q})$)
FTR(NI,NJ)	Real part of the Fourier transform of ψ. ($Re(c_{n,q})$)
F1(NI)	Utility array used by subrouting CALCW.
F2(NI)	Utility array used by subroutine CALCW.
G	Acceleration of gravity. (g)
GAMMA	Lapse rate. (γ)
I	An index.
IB(NPBP2)	Column numbers of points on the barrier.
IBT	Number of the first time step.
IB25	Indicator of an even multiple of 25 time steps.
ICASE	Wind velocity profile indicator.
IDUM(8)	Filler array for tape or disk read/write.
IE	Number of the last column to be plotted.
IEB	Rightmost column number on the barrier.
IET	Number of the last time step.
IEW	Number of the last column at which w is calculated.
IEXP2	Power of two of the number of unique horizontal grid points, N_x.
IL	Number of print symbols to be plotted in a line of graphic output.

IN	Number of columns to be plotted.
INM1	IN minus 1.
IPSIGR(20)	Time step numbers at which ψ is plotted.
IRHOGR(20)	Time step numbers at which ρ is plotted.
IRIGR(20)	Time step numbers at which $\log_{10}(Ri)$ is plotted.
IS	Number of the first column to be plotted.
ISAVE	Tape save parameter.
ISB	Leftmost column number on the barrier.
ISW	Number of the first column at which w is calculated.
ITYPE(NI,NJ)	Grid point type.
IUGR(20)	Time step numbers at which u is plotted.
IWB	Horizontal grid point width of the barrier.
IWGR(20)	Time step numbers at which w is plotted.
IZTAGR(20)	Time step numbers at which ζ is plotted.
J	An index
JB(NPBP2)	Row numbers of points on the barrier.
JCASE	Stability profile indicator.
JE	Number of the highest row to be plotted.
JMAX	Highest row number on the barrier.
JMAXM1	JMAX minus 1.
JN	Number of rows to be plotted.
JS	Number of the lowest row to be plotted.
K	An index
KM2	K minus 2.
L	An index.
LI	Row number at which the hyperbolic tangential profile has its inflection point.
LINE14(33)	Part of annotation on graph of field.
LINE24(33)	Line of print symbols associated with a level of the grid.
LINE3(132,NJSM1)	Logical symbol equivalent of LINE34.
LINE34(33,NJSM1)	Lines of print symbols between two levels of the grid.

LINE48(14)	Part of annotation on shading value scale.
LINE58(14)	Part of annotation on shading value scale.
LINE68(14)	Part of annotation on shading value scale.
M	An index.
MDZ	Number of rows away from the inflection point at which U1 is deflected by the amount U2.
MT	Time step number.
N	An index.
NAME(2,6)	Titles of quantities on graphical output.
NEWRHO(NI,NJ)	Temporary array for newly calculated density.
NEWZTA(NI,NJ)	Temporary array for newly calculated vorticity.
NI	Number of grid points in the horizontal direction, or number of columns.
NIM1	NI minus 1.
NIM2	Number of unique columns = NI minus 2.
NIPA	Number of unique columns, plus the horizontal grid point width of the barrier.
NIS	Number of print symbols from one horizontal grid point to the next.
NISM1	NIS minus 1.
NJ	Number of grid points in the vertical direction, or number of rows.
NJM1	NJ minus 1.
NJM2	NJ minus 2. (N_z)
NJSM1	Number of print lines between two vertical grid points.
NPB	Number of grid points on the surface of the barrier
NPBM1	NPB minus 1.
NPBP1	NPB plus 1.
NPBP2	NPB plus 2.
NPSIGR	Next time step number at which ψ is to be plotted.
NPSILV	Maximum number of levels of shading for ψ graphs.
NRHOGR	Next time step number at which ρ is plotted.

NRHOLV	Maximum number of levels of shading for ρ graphs.
NRILV	Maximum number of levels of shading for $\log_{10}(Ri)$ graphs.
NT	Number of time steps.
NUGR	Next time step number of which u is to be plotted.
NULV	Maximum number of levels of shading for u graphs.
NWGR	Next time step number at which w is to be plotted.
NWLV	Maximum number of levels of shading for w graphs.
NZTAGR	Next time step number at which ζ is to be plotted.
NZTALV	Maximum number of levels of shading for ζ graphs.
OLDRHO(NI,NJ)	Density from the previous timestep. (old ρ)
OLDZTA(NI,NJ)	Vorticity from the previous timestep. (old ζ)
P	Pressure. (p)
PIM2	π multiplied by 2.
PSI(NI,NJ)	Streamfunction. (ψ)
PSIAUX(NIPA, NJM2,JMAXM1)	The solution to $\nabla^2\psi_k = \zeta_k$ for each vertical level on the surface of the varrier. ($(\bar{\psi}_k)_o$)
PSIBFT(NIM2,2)	Fourier transform of the upper boundary of ψ. ($c_n(N_z+1)$)
PSIM(NPB,NPB)	Barrier influence coefficient matrix. (α)
PSIO(NJ)	Initial unperturbed streamfunction. ($\bar{\psi}^o(z)$)
RDCP	$R/c_p = 2/7$
RHO(NI,NJ)	Density. (ρ)
RHOO(NJ)	Initial unperturbed density. ($\bar{\rho}^o(z)$)
RI(NI,NJ)	Richardson number. (Ri)
RIDZSG	CRI / (DELTAZ ** 2 * G).
S(NJ)	Initial stability. ($S^o(z)$)
SBV	Stability in the constant Brunt-Vaisala frequency case.
SPR	Pressure at the lower boundary. (p_c)
SPSI	Streamfunction on the lower boundary. (ψ_c)
SRHO	Density on the lower boundary. (ρ_c)
SRTP	SRHO * STMP / SPR ** RDCP.

STMP	Temperature at the lower boundary. (T_c)
SYM(53)	Symbols used in printout of fields.
T	Temperature. (T)
TM(NPB,NPB)	Temporary array formed from PSIM.
TO	Base temperature for constant lapse rate atmosphere. (T_o)
U(NI,NJ)	Horizontal velocity. (u)
UA1(NJ) through UA4(NJ)	Utility arrays used by subroutines FA, CALCRI, and CALCU.
UA5(NJ)	Utility array used by subroutines FA and CALCRI.
UA6(NJ)	Utility array used by subroutine CALCRI.
UO(NJ)	Initial horizontal velocity. ($u^o(z)$)
U1	Horizontal velocity to be added to u at all levels. (u_1)
U2	Base horizontal velocity for profiles. (u_2)
W(NI,NJ)	Vertical velocity. (w)
ZETA(NI,NJ)	Vorticity. (ζ)

Variables Used Uniquely in Subroutine CALCRI

AD(NJ)	Diagonal elements of the spline matrix. (a_d)
B(NJ)	Elements of the b matrix. (b)
D(NJ)	Backward-difference derivatives of ψ or ρ with respect to z. (d)
DB	Term which incorporates vorticity into bottom boundary condition. ($a'\Delta z/2$)
DT	Term which incorporates vorticity into top boundary condition. ($b'\Delta z/2$)
DPSI1	Temporary storage variable for first derivative of ψ.
DPSI2	Temporary storage variable for first derivative of ψ.
DRHO(NJ)	First derivative of ρ in the vertical. ($\partial\rho/\partial z$)
D2PSI(NJ)	Second derivative of ψ in the vertical. ($\partial^2\psi/\partial z^2$)

R	Temporary storage variable for the Richardson number.
UL	Temporary variable used in inverting the spline matrix. (u_ℓ)
UP(NJ)	Temporary array used in inverting the spline matrix. (u_p)
Z(NJ)	Temporary array used in inverting the spline matrix. (f)

Variables Used Uniquely in Subroutine CALCU

AD(NJ)	Diagonal elements of the spline matrix. (a_d)
B(NJ)	Elements of the b matrix. (b)
D(NJ)	Backward-difference derivative of ψ with respect to z. (d)
UL	Temporary variable used in inverting the spline matrix. (u_ℓ)
Z(NJ)	Temporary array used in inverting the spline matrix. (f)

Variables Used Uniquely in Subroutine CALCW

FTI(NIM2)	Imaginary part of the Fourier transform of ψ. ($Im(c_n(z))$)
FTR(NIM2)	Real part of the Fourier transform of ψ. ($Re(c_n(z))$)
T	Temporary variable used to store Fourier coefficients.

Variables Used Uniquely in Subroutine FA

BI(NJ)	Imaginary part of the elements of the b matrix. (Im(b))
BR(NJ)	Real part of the elements of the b matrix. (Re(b))
UL	Temporary variable used in inversion of the relaxation matrix. (u_ℓ)
UP(NJ)	Temporary array used in inversion of the relaxation matrix. (u_p)
ZI(NJ)	Imaginary part of the temporary array used in inversion of the relaxation matrix (Im(f))
ZR(NJ)	Real part of the temporary array used in inversion of the relaxation matrix. (Re(f))

Variables Used Uniquely in Subroutine FAST

Except for the addition of the calling parameter INV, this subroutine is the fast Fourier transform subroutine from the UCLA BMD library. The calling parameters are:

INV	+1, the Fourier transform is calculated. -1, the inverse Fourier transform is calculated.
M	Power of two of N.
N	Dimension of the complex array (X,Y).
X(N)	Real part of the array to be transformed, and also the real part of the transform.
Y(N)	Imaginary part of the array to be transformed, and also the imaginary part of the transform.

Variables Used Uniquely in Subroutine GE

D	Ratio of the element beneath the pivotal element to the pivotal element.
IP1	Lowest row number for which an element beneath the pivotal element is to be zeroed out.
N1	Dimensions of the array X.
N2	Dimensions of the square matrix for which the determinant is calculated.
N2M1	N2 minus 1.
T	Temporary variable for the exchange of column elements.
X(N1,N1)	Array from which the matrix is selected for calculation of the determinant.
XMAX	Largest element in a row of the matrix.
Y	Temporary variable for the storage of XMAX.

Variables Used Uniquely in Subroutine GRAFIC

D	Temporary value increment between levels of shading.
DIFI	Difference in value from one print character in a line to the next.
DIFJ	Number of levels of shading from one vertical grid point to the next.
DL	LOG10(D).
DV	Number of print lines from one vertical grid point to the next multiplied by XINC.
E	Largest absolute value to be plotted.
EL	LOG10(E).
IDL	LOG10(XINC).
IEL	Base part of the values of the lines separating levels of shading.
IMAX	Number of levels of shading between zero and XMAX.
IMIN	Number of levels of shading between zero and XMIN.
LINE2(132)	Logical symbol equivalent of LINE24.
NEW(132)	Minimum-adjusted value at each print character in the line corresponding to the present vertical grid point.
NQ	Number corresponding to variable to be plotted.
OLD(132)	Minimum-adjusted value at each print character in the line corresponding to the previous vertical grid point.
SIGPRT	Significant part of XINC.
TEMP	Temporary variable for storage of the symbol index.
VINC	Increment between the values of V1 and V2.
V1(13)	Significant part of the values of the lines separating levels of shading.
V2(13)	Significant part of the values of the lines separating levels of shading.
X(NI,NJ)	Array to be plotted.
XINC	Value increment between levels of shading.

XMAX	Maximum value in the field; also, maximum value to be plotted.
XMIN	Minimum value in the field; also, minimum value to be plotted.
Y	Temporary storage variable for X array elements.

Variables Used Uniquely in Subroutine GRAFIN

BLANK	1 blank symbol.
BLANK4	4 blank symbols.
BLANK8	8 blank symbols.
DASH	Symbol used in annotation of shading value scale.
DOT	Symbol used in annotation of graph.
LINE1(132)	Logical symbol equivalent of LINE14.
LINE2(132)	Logical symbol equivalent of LINE24.
LINE4(112)	Logical symbol equivalent of LINE48.
LINE5(112)	Logical symbol equivalent of LINE58.
LINE6(112)	Logical symbol equivalent of LINE68.
NAME1(2,6)	Titles of quantities on graphical output.
NAME2(2,6)	Titles of quantities on graphical output.
NFB	First word in LINE24 and LINE34 which is to be filled with blank symbols.
SYM1(53)	Symbols used in graph of field.
SYM2(53)	Symbols used in graph of field.

Variables Used Uniquely in Subroutine PTSPEC

None.

Variables Used Uniquely in Subroutine SP

ALPHA	Barrier influence coefficient. (α)
X(NI,NJ)	Variable for which a superposition solution is required. (ψ or ρ)

XO	Value of the array X on the lower boundary. ($\overline{\psi}^{o}_{11}$ or $\overline{\rho}^{o}_{11}$)

CONSIDERATIONS IN RUNNING THE PROGRAM

The user must specify all pertinent dimensions at the beginning of the first routine. The variables whose dimensions are subject to change are:

PSI(NI,NJ)	AD(NIM2)
RHO(NI,NJ)	AK(NIM2)
ZETA(NI,NJ)	F1(NIM2)
OLDRHO(NI,NJ)	F2(NIM2)
OLDZTA(NI,NJ)	AS(NJ)
NEWRHO(NI,NJ)	UA1(NJ)
NEWZTA(NI,NJ)	UA2(NJ)
U(NI,NJ)	UA3(NJ)
W(NI,NJ)	UA4(NJ)
RI(NI,NJ)	UA5(NJ)
A(NI,NJ)	UA6(NJ)
PSIAUX(NIPA,NJM2,JMAXM1)	BETA(NPB,NPB)
FTR(NI,NJ)	PSIM(NPB,NPB)
FTI(NI,NJ)	TM(NPB,NPB)
PSIBFT(NIM2, 2)	LINE34(33,NJSM1)
UO(NJ)	ITYPE(NI,NJ)
S(NJ)	IB(NPBP2)
RHOO(NJ)	JB(NPBP2)
PSIO(NJ)	LINE3(132,NJSM1)

where NI, NJ, NPB, NPBP2, NIM2, NJM2, NIPA, JMAXM1, and NJSM1 are defined in the previous section.

On the IBM 360/91, the approximate execution space is: 60 * NI * NJ + 8 * NIPA * NJM2 * JMAXM1 + 6 * NI + 11 * NJ + 80K IBM bytes, or roughly, 8 * (JMAXM1 + 8) * NI * NJ + 80K IBM bytes. Approximately 7K bytes of additional space are required for each tape used in conjunction with the program, for buffering purposes.

On the IBM 360/91, the approximate execution time is: .00013 * NI * NJ * NT * (1 + .0017 * NPB ** 2) + 2 seconds. The running time is not appreciably increased by moderate increases in the amount of graphical output.

A version of this program exists which is compatible with CDC systems.

A SAMPLE CASE

Consider the Lyra case for u_0 = 25 m/sec, T_0 = 250K, and γ = 0 on a 64 x 24 grid for 80 timesteps, with $\Delta x = \Delta z$ = 625 m, Δt = 10 sec, and a 2500 m high by 1875 m wide rectangular barrier. Suppose that the entire field of ψ is to be output on a vertically exaggerated scale at time step 60, and ψ, ρ, ζ, u, w, and $\log_{10}(Ri)$ are to be output at time step 80, with no data to be saved. The source program deck, the data card deck, and the actual output from this case are displayed on pages 77 through 97. The estimated execution space and time on the IBM 360/91 are 237K and 17 seconds.

Listing of source program

```
C
C     THIS IS THE MAIN ROUTINE, WHICH READS THE DATA, INITIALIZES THE
C     FIELDS, AND INTEGRATES THE DENSITY AND VORTICITY EQUATIONS
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 PSI( 66,24),RHO( 66,24),ZETA( 66,24),
     1       CLDRHO( 66,24),OLDZTA( 66,24),NEWRHO( 66,24),
     2       NEWZTA( 66,24),F2( 64),
     2       U( 66,24),W( 66,24),RI( 66,24),A( 66,24),
     3       PSIAUX( 67,22,4),FTR( 66,24),FTI( 66,24),PSIBFT( 64,2),
     4       U0(24),S(24),RHO0(24),PSI0(24),AD( 64),AK( 64),F1( 64),
     5       AS(24),UA1(24),UA2(24),UA3(24),UA4(24),UA5(24),UA6(24),
     6       BETA(10,10),PSIM(10,10),TM(10,10),
     7       LINE48(14),LINE58(14),LINE68(14)
      REAL*4 LINE14(33),LINE24(33),LINE34(33,2),NAME(2,6),
     1       BLANK4
      INTEGER*4 IPSIGR(20),IRHOGR(20),IZTAGR(20),IUGR(20),IWGR(20),
     1          IRIGR(20),ITYPE( 66,24),IB(12),JB(12),IDUM(8)
      LOGICAL*1 LINE3(132,2),SYM(53)
      EQUIVALENCE (LINE3(1),LINE34(1)),(NEWZTA(1),U(1),W(1),RI(1)),
     1            (PSI(1),FTR(1)),(NEWRHO(1),FTI(1),A(1))
      COMMON/C1/DELTAX,DXM2,DELTAZ,DZM2,DZSQ,DZD6,J,NI41,NJM1,IEXP2,
     1          IS,JS,IE,JE,ISW,IEW,NPBP1,JMAX,ISB,IEB
      COMMON/C2/LINE48,LINE58,LINE68,LINE14,LINE24,NAME,
     1          MT,IN,JN,INM1,NIS,NISM1,IL,SYM
C     SPECIFY PHYSICAL FACTORS
      G = 9.8
      CKTMPS = 1852. / 3600.
      SPSI = 0.
      SRHO = 1.25
      STMP = 273.
      SPR = 1.0D+05
      SRTP = SRHO * STMP / SPR ** (2./7.)
      DALR = 9.76D-03
      RDCP = 2. / 7.
C
C     A. READ, WRITE SPACE AND TIME DATA
C
      READ(5,1001) NI,NJ,NT,NPB,IBT,ISAVE,DELTAX,DELTAZ,DELTAT
 1001 FORMAT(6I4,1PD10.2)
      WRITE(6,2001) NI,NJ,NT,NPB,IBT,ISAVE,DELTAX,DELTAZ,DELTAT
 2001 FORMAT(1H1,'SUMMARY OF INPUT DATA FOR NON-ITERATIVE, SOLID ',
     1       'BARRIER PROGRAM'////' NI =',I4,',  NJ =',I4,',  NT =',I4,
     2       ',  NPB =',I4,',  IBT =',I4,',  ISAVE =',I4,',  DX =',
     3       1PD10.2,',  DZ =',1PD10.2,',  DT =',1PD10.2)
C     CALCULATE QUANTITIES WHICH DEPEND ON SPACE AND TIME DATA
      NIM1 = NI - 1
      NIM2 = NI - 2
C     CALCULATE POWER OF TWO OF NUMBER OF UNIQUE HORIZONTAL GRID POINTS
      IEXP2 = 1
      I = NIM2
    5 I = I / 2
      IF(I.EQ.1) GO TO 10
      IEXP2 = IEXP2 + 1
      GO TO 5
   10 NJM1 = NJ - 1
      NJM2 = NJ - 2
      MT = IBT
      IET = IBT + NT
      IB25 = 0
      NPBM1 = NPB - 1
      NPBP1 = NPB + 1
      NPBP2 = NPB + 2
      DXM2 = DELTAX * 2.
      DXSQ = DELTAX ** 2
      DZM2 = DELTAZ * 2.
      DZD2 = DELTAZ / 2.
      DZSQ = DELTAZ ** 2
      DZD6 = DELTAZ / 6.
      DZ2DX2 = 2. * DZSQ / DXSQ
      FAC = -1. / (12. * DELTAX * DELTAZ)
      FACM2 = FAC * 2.
      FACM4 = FAC * 4.
      FACM43 = FACM4 / 3.
      DELTAT = DELTAT * 2.
C
C     B. READ, WRITE BARRIER DATA
C
      READ(5,1002) (IB(I),JB(I),I=2,NPBP1)
 1002 FORMAT(20I4)
```

```
      WRITE(6,2002) (IB(I),JB(I),I=2,NPBP1)
 2002 FORMAT(1H0,'COORDINATES OF BARRIER =',10(2X,'(',I2,',',I2,')'))
C     CALCULATE QUANTITIES WHICH DEPEND ON BARRIER DATA
      IWB = IB(NPBP1) - IB(2)
      NIPA = NIM2 + IWB
C     CALCULATE TYPE OF EACH GRID POINT
      CALL PTSPEC(ITYPE,IB,JB,NI,NJ,NPBP2)
      JMAXM1 = JMAX - 1
      NDUM = 2 * MOD(NJ*((NI*7)/2+1)+NIM2*3+NIPA*NJM2*JMAXM1+NPB**2,4)
      DO 15 I = 1,NDUM
   15 IDUM(I) = 0
C
C     C. READ, WRITE GRAPHICAL OUTPUT DATA
C
      READ(5,1003) IS,JS,IN,JN,NJSM1,NPSILV,NRHOLV,NZTALV,NULV,NWLV,
     1             NRILV,IPSIGR,IRHOGR,IZTAGR,IUGR,IWGR,IRIGR
 1003 FORMAT(514/614,6(/2014))
      WRITE(6,2003) IS,JS,IN,JN,NJSM1,NPSILV,NRHOLV,NZTALV,NULV,NWLV,
     1              NRILV,IPSIGR,IRHOGR,IZTAGR,IUGR,IWGR,IRIGR
 2003 FORMAT(1H0,'IS =',I4,',  JS =',I4,',  IN =',I4,',  JN =',I4,
     1       ',  NJSM1 =',I4  //' NPSILV =',I4,',  NRHOLV =',I4,
     2       ',  NZTALV =',I4,',  NULV =',I4,',  NWLV =',I4,
     3       ',  NRILV =',I4//' IPSIGR =',2014//' IRHOGR =',2014//
     4       ' IZTAGR =',2014//' IUGR =  ',2014//,' IWGR =  ',2014//
     5       ' IRIGR = ',2014)
C     CALCULATE QUANTITIES WHICH DEPEND ON GRAPHICAL DATA
      NPSIGR = 1
      NRHOGR = 1
      NZTAGR = 1
      NUGR = 1
      NWGR = 1
      NRIGR = 1
      IE = IS + IN - 1
      JE = JS + JN - 1
C     CALCULATE HORIZONTAL EXTENT OF GRID POINTS AT WHICH W IS TO BE
C     CALCULATED
      ISW = IS
      IF(IS.EQ.1) ISW = 2
      IEW = IE
      IF(IE.EQ.NI) IEW = NIM1
      INM1 = IN - 1
C     SET UP ANNOTATION FOR GRAPHICAL OUTPUT
      CALL GRAFIN(LINE34,LINE3,NJSM1)
      IF(IBT.NE.0) GO TO 230
C
C     D. READ WIND VELOCITY DATA, AND CALCULATE U0
C
      READ(5,1004) ICASE
 1004 FORMAT(I4)
      GO TO (60,70,80,90,100,110),ICASE
C     SOUNDING CASE (CALCULATE RHO0 IN ADDITION TO U0)
   60 WRITE(6,2004)
 2004 FORMAT(1H0,'SOUNDING DATA CASE'//6X,'U',11X,'T',11X,'P'/)
      DO 65 J = 1,NJ
      READ(5,1005) U0(J),T,P
 1005 FORMAT(1P3D10.2)
      WRITE(6,2005) U0(J),T,P
 2005 FORMAT(1X,1PD10.2,1P2D12.2)
      U0(J) = U0(J) * CKTMPS
      T = T + 273.
      P = P * 1.0D+02
   65 RHO0(J) = SRTP * P ** (2./7.) / T
      GO TO 120
C     CONSTANT VELOCITY CASE
   70 READ(5,1006) U1
 1006 FORMAT(1PD10.2)
      WRITE(6,2006) U1
 2006 FORMAT(1H0,'CONSTANT VELOCITY CASE,  U1 =',1PD10.2)
      DO 75 J = 1,NJ
   75 U0(J) = U1
      GO TO 120
C     CONSTANT SHEAR CASE
   80 READ(5,1007) U1,C
 1007 FORMAT(1P2D10.2)
      WRITE(6,2007) U1,C
 2007 FORMAT(1H0,'CONSTANT SHEAR CASE,  U1 =',1PD10.2,',  C =',1PD10.2//
     1       6X,'U'/)
      COZ = C * DELTAZ
      DO 85 J = 1,NJ
      U0(J) = U1 + COZ * (J - 1)
   85 WRITE(6,2008) U0(J)
 2008 FORMAT(1X,1PD10.2)
      GO TO 120
```

```
C       EXPONENTIAL PROFILE CASE
   90 READ(5,1005) U1,U2,C
      WRITE(6,2009) U1,U2,C
 2009 FORMAT(1H0,'EXPONENTIAL PROFILE CASE.  U1 =',1PD10.2,',  U2 =',
     1       1PD10.2,',   C =',1PD10.2//6X,'U'/)
      CDZ = C * DELTAZ
      DO 95 J = 1,NJ
      U0(J) = U1 + U2 * DEXP(CDZ*(J-1))
   95 WRITE(6,2008) U0(J)
      GO TO 120
C       HYPERBOLIC TANGENTIAL PROFILE CASE
  100 READ(5,1008) U1,U2,LI,MDZ
 1008 FORMAT(1P2D10.2,2I4)
      WRITE(6,2010) U1,U2,LI,MDZ
 2010 FORMAT(1H0,'HYPERBOLIC TANGENTIAL PROFILE CASE.  U1 =',1PD10.2,
     1       ',  U2 =',1PD10.2,',   LI =',I4,',   MDZ =',I4//6X,'U'/)
      DO 105 J = 1,NJ
      U0(J) = U1 + U2 * DTANH(DFLOAT((J-LI)/MDZ))
  105 WRITE(6,2008) U0(J)
      GO TO 120
C       OTHER VELOCITY CASE
  110 WRITE(6,2011)
 2011 FORMAT(1H0,'SPECIFIED VELOCITY CASE'//6X,'U'/)
      READ(5,1009) (U0(J),J=1,NJ)
 1009 FORMAT(1P8D10.2)
      DO 115 J = 1,NJ
  115 WRITE(6,2008) U0(J)
C
C       E. CALCULATE PSI0
C
  120 PSI0(1) = 0.
      DO 125 J = 2,NJ
  125 PSI0(J) = PSI0(J-1) - DZD2 * (U0(J-1) + U0(J))
      IF(ICASE.EQ.1) GO TO 170
C
C       F. READ, WRITE STABILITY DATA, AND CALCULATE STABILITY
C
      READ(5,1004) JCASE
      GO TO (130,140,150),JCASE
C       CONSTANT LAPSE RATE CASE
  130 READ(5,1007) GAMMA,T0
      WRITE(6,2012) GAMMA,T0
 2012 FORMAT(1H0,'CONSTANT LAPSE RATE CASE.  GAMMA =',1PD10.2,',  T0 =',
     1       1PD10.2)
      DMGDT0 = (DALR - GAMMA) / T0
      DO 135 J = 1,NJ
  135 S(J) = DMGDT0
      GO TO 160
C       CONSTANT BRUNT-VAISALA FREQUENCY CASE
  140 READ(5,1006) BV
      WRITE(6,2013) BV
 2013 FORMAT(1H0,'CONSTANT BRUNT-VAISALA FREQUENCY CASE.  BV =',1PD10.2)
      SBV = BV ** 2 / G
      DO 145 J = 1,NJ
  145 S(J) = SBV
      GO TO 160
C       CONSTANT RICHARDSON NUMBER CASE
  150 READ(5,1006) CRI
      WRITE(6,2014) CRI
 2014 FORMAT(1H0,'CONSTANT RICHARDSON NUMBER CASE.  RI =',1PD10.2)
      RIDZSG = CRI / (DZSQ * G)
      S(1) = RIDZSG * (U0(2) - U0(1)) ** 2
      DO 155 J = 2,NJM1
  155 S(J) = RIDZSG * (U0(J+1) - U0(J-1)) ** 2 / 4.
      S(NJ) = RIDZSG * (U0(NJ) - U0(NJM1)) ** 2
C
C       G. CALCULATE RHO0
C
  160 RHO0(1) = SRHO
      DO 165 J = 2,NJ
  165 RHO0(J) = RHO0(J-1) * DEXP(-DZD2*(S(J-1)+S(J)))
C       INITIALIZE VORTICITY, ANGULAR ARGUMENTS, MATRIX DIAGONAL
C       ELEMENTS, AND FOURIER TRANSFORM PSI = 0 ON TOP BOUNDARY FOR
C       CALCULATION OF BARRIER INFLUENCE COEFFICIENTS
  170 DO 175 I = 1,NI
      DO 175 J = 1,NJ
  175 ZETA(I,J) = 0.
      PIM2 = 6.28318530718D+00
      DO 180 I = 1,NIM2
      ARGI = PIM2 * (I - 1) / NIM2
      AD(I) = 2. + DZ2DX2 * (1. - DCOS(ARGI))
      AK(I) = DSIN(ARGI) / DELTAX
```

```
      DO 180 J = 1,2
  180 PSIBFT(I,J) = 0.
C     DEFINE SPLINE MATRIX DIAGONAL ELEMENTS
      AS(1) = 2.
      DO 182 J = 2,NJM1
  182 AS(J) = 4.
      AS(NJ) = 2.
C
C     H. SOLVE POISSON EQUATION WITH UNIT VORTICITY DISTURBANCE AT EACH
C     VERTICAL LEVEL FOR WHICH THERE IS A POINT ON THE SURFACE OF THE
C     BARRIER, AND STORE RESULTS INTO PSIAUX
C
      DO 190 K = 1,JMAXM1
      ZETA(IB(NPBP1),K+1) = 1.
      CALL FA(PSI,ZETA,FTR,TI,PSIBFT,AD,UA1,UA2,UA3,UA4,UA5,NI,NJ,NIM2,
     1        NJM2)
      ZETA(IB(NPBP1),K+1) = 0.
      DO 187 J = 1,NJM2
      DO 185 I = 1,NIM2
  185 PSIAUX(I,J,K) = PSI(I,J+1)
      DO 187 L = 1,IWB
  187 PSIAUX(NIM2+L,J,K) = PSIAUX(L,J,K)
      DO 189 J = 1,NPB
      IF(JB(J+1).NE.K+1) GO TO 189
      L = IB(NPBP1) - IB(J+1)
      DO 188 I = 1,NPB
      PSIM(I,J) = PSI(IB(I+1)+L,JB(I+1))
  188 TM(I,J) = PSIM(I,J)
  189 CONTINUE
  190 CONTINUE
C     CALCULATE DETERMINANT OF ALPHA MATRIX
      CALL GE(TM,DET,NPB,NPB,IACC1)
C
C     I. CALCULATE BETA
C
      DO 205 I = 1,NPB
      DO 205 J = 1,NPB
      CFAC = (-1) ** (I+J) / DET
C     SELECT COFACTOR MATRIX ELEMENTS
      M = 0
      DO 200 K = 1,NPB
      IF(I.EQ.K) GO TO 200
      M = M + 1
      N = 0
      DO 195 L = 1,NPB
      IF(J.EQ.L) GO TO 195
      N = N + 1
      TM(M,N) = PSIM(K,L)
  195 CONTINUE
  200 CONTINUE
C     CALCULATE DETERMINANT OF COFACTOR MATRIX
      CALL GE(TM,CFDET,NPB,NPBM1,IACC2)
  205 BETA(J,I) = CFAC * CFDET * 10. ** ((IACC2-IACC1)*50)
C     CALCULATE FOURIER TRANSFORM OF PSI ON TOP BOUNDARY
      DO 209 I = 1,NIM2
  209 PSIBFT(I,1) = PSI0(NJ)
      CALL FAST(PSIBFT(1,1),PSIBFT(1,2),NIM2,IEXP2,1)
C
C     J. INITIALIZE PSI, RHO, AND ZETA
C
      DO 210 I = 1,NI
      DO 210 J = 1,NJ
  210 PSI(I,J) = PSI0(J)
C     SUPERPOSITION SOLUTION FOR PSI FOR TIME STEP 0
      CALL SP(PSI,PSIAUX,ITYPE,BETA,IB,JB,SPSI,NI,NJ,NIM2,NJM2,NIPA,
     1        JMAXM1,NPB,NPBP2)
      DO 215 I = 1,NI
      DO 215 J = 1,NJ
      DO 214 K = 2,NI
  214 IF(DABS(PSI(I,J)).LE.DABS(PSI0(K))) GO TO 215
  215 RHO(I,J) = (PSI(I,J) - PSI0(K-1)) * (RHO0(K) - RHO0(K-1)) /
     1           (PSI0(K) - PSI0(K-1)) + RHO0(K-1)
C     CALCULATE ZETA FROM PSI0
      DO 220 J = 2,NJM1
  220 ZETA(1,J) = (PSI0(J-1) + PSI0(J+1) - 2. * PSI0(J)) / DZSQ
      ZETA(1,1) = ZETA(1,2)
      ZETA(1,NJ) = ZETA(1,NJM1)
      DO 225 J = 1,NJ
      DO 225 I = 2,NI
  225 ZETA(I,J) = ZETA(1,J)
C     ZERO OUT VORTICITY INSIDE BARRIER
      DO 227 I = ISB,IEB
      DO 227 J = 1,JMAXM1
```

```
  227 IF(ITYPE(I,J).EQ.9) ZETA(I,J) = 0.
      GO TO 235
C
C     K. READ PSI, RHO, ZETA, RHO0, PSIAUX, BETA, ITYPE, AD, AND PSIBFT
C     FROM TAPE OR DISK
C
  230 READ(1) PSI,RHO,ZETA,RHO0,PSIAUX,BETA,ITYPE,AD,PSIBFT,
     1        (IDUM(I),I=1,NDUM)
  235 IF(IB25.EQ.1) GO TO 280
C
C     L. CALCULATE GRAPHICAL OUTPUT FOR TIME STEP MT
C
      IF(MT.NE.IPSIGR(NPSIGR)) GO TO 240
      CALL GRAFIC(PSI,A,LINE34,LINE3,1,NPSILV,NI,NJ,NJSM1)
      NPSIGR = NPSIGR + 1
  240 IF(MT.NE.IRHOGR(NRHOGR)) GO TO 245
      CALL GRAFIC(RHO,A,LINE34,LINE3,2,NRHOLV,NI,NJ,NJSM1)
      NRHOGR = NRHOGR + 1
  245 IF(MT.NE.IZTAGR(NZTAGR)) GO TO 250
      CALL GRAFIC(ZETA,A,LINE34,LINE3,3,NZTALV,NI,NJ,NJSM1)
      NZTAGR = NZTAGR + 1
  250 IF(MT.NE.IUGR(NUGR)) GO TO 255
      CALL CALCU(PSI,ZETA,U,AS,UA1,UA2,UA3,UA4,NI,NJ)
      CALL GRAFIC(U,A,LINE34,LINE3,4,NULV,NI,NJ,NJSM1)
      NUGR = NUGR + 1
  255 IF(MT.NE.IWGR(NWGR)) GO TO 260
      CALL CALCW(PSI,W,F1,E2,AK,NI,NJ,NIM2)
      CALL GRAFIC(W,A,LINE34,LINE3,5,NWLV,NI,NJ,NJSM1)
      NWGR = NWGR + 1
  260 IF(MT.NE.IRIGR(NRIGR)) GO TO 265
      CALL CALCRI(PSI,RHO,ZETA,RI,AS,UA1,UA2,UA3,UA4,UA5,UA6,NI,NJ)
      CALL GRAFIC(RI,A,LINE34,LINE3,6,NRILV,NI,NJ,NJSM1)
      NRIGR = NRIGR + 1
  265 IF(MT.LT.IET) GO TO 270
C
C     M. WRITE PSI, RHO, ZETA, RHO0, PSIAUX, BETA, ITYPE, AD, AND PSIBFT
C     ONTO TAPE OR DISK
C
      IF(ISAVE.EQ.1) WRITE(2) PSI,RHO,ZETA,RHO0,PSIAUX,BETA,ITYPE,AD,
     1                        PSIBFT,(IDUM(I),I=1,NDUM)
      STOP
  270 IF(MOD(MT,25).NE.0) GO TO 280
      DELTAT = DELTAT / 4.
C
C     N. SET OLDRHO = RHO, OLDZTA = ZETA
C
      DO 275 I = 2,NIM1
      DO 275 J = 1,NJ
      OLDRHO(I,J) = RHO(I,J)
  275 OLDZTA(I,J) = ZETA(I,J)
C
C     O. CALCULATE JACOBIANS J(ZETA,PSI), J(RHO,PSI)
C
  280 DO 345 I = 2,NIM1
      DO 340 J = 1,NJ
      K = ITYPE(I,J)
C     NEWRHO FOR POINT TYPES 1 THROUGH 9
      NEWRHO(I,J) = OLDRHO(I,J)
      IF(K.NE.0) GO TO 285
C     JACOBIANS FOR POINT TYPE 0
      A1 = PSI(I-1,J+1)
      A2 = PSI(I,J+1)
      A3 = PSI(I+1,J+1)
      A4 = PSI(I-1,J)
      A6 = PSI(I+1,J)
      A7 = PSI(I-1,J-1)
      A8 = PSI(I,J-1)
      A9 = PSI(I+1,J-1)
      B1 = A8 + A9 - A2 - A3
      B2 = - A7 - A8 + A1 + A2
      B3 = A6 + A3 - A4 - A1
      B4 = - A9 - A6 + A7 + A4
      B5 = A6 - A2
      B6 = - A8 + A4
      B7 = A2 - A4
      B8 = - A6 + A3
      AJZP = FAC * (B1 * ZETA(I+1,J) + B2 * ZETA(I-1,J) + B3 *
     1       ZETA(I,J+1) + B4 * ZETA(I,J-1) + B5 * ZETA(I+1,J+1) + B6 *
     2       ZETA(I-1,J-1) + B7 * ZETA(I-1,J+1) + B8 * ZETA(I+1,J-1))
      DRHO = (RHO(I+1,J) - RHO(I-1,J)) / 2.
C     CALCULATE NEWRHO FOR POINT TYPE 0
      NEWRHO(I,J) = FAC * (B1 * RHO(I+1,J) + B2 * RHO(I-1,J) + B3 *
```

```
     1              RHO(I,J+1) + B4 * RHO(I,J-1) + B5 * RHO(I+1,J+1) +
     2              B6 * RHO(I-1,J-1) + B7 * RHO(I-1,J+1) + B8 *
     3              RHO(I+1,J-1)) * DELTAT + OLDRHO(I,J)
      GO TO 335
C     JACOBIAN FOR POINT TYPE 1
  285 IF(K.NE.1) GO TO 290
      A5 = PSI(I,J)
      A7 = PSI(I-1,J-1)
      A8 = PSI(I,J-1)
      A9 = PSI(I+1,J-1)
      AJZP = FACM2 * (ZETA(I+1,J) * (A8 + A9 - 2. * A5) + ZETA(I-1,J) *
     1       (- A7 - A8 + 2. * A5) + ZETA(I,J-1) * (- A9 + A7) +
     2       (ZETA(I-1,J-1) - ZETA(I+1,J-1)) * (- A8 + A5))
      DRHO = 0.
      GO TO 335
C     JACOBIAN FOR POINT TYPE 2
  290 IF(K.NE.2) GO TO 295
      A1 = PSI(I-1,J+1)
      A2 = PSI(I,J+1)
      A3 = PSI(I+1,J+1)
      AJZP = FACM2 * (ZETA(I+1,J) * (- A2 - A3) + ZETA(I-1,J) *
     1       (A1 + A2) + ZETA(I,J+1) * (A3 - A1) + A2 * (ZETA(I-1,J+1) -
     2       ZETA(I+1,J+1)))
      DRHO = 0.
      GO TO 335
  295 KM2 = K - 2
      GO TO (300,305,310,315,320,325,330),KM2
C     JACOBIAN FOR POINT TYPE 3
  300 A1 = PSI(I-1,J+1)
      A4 = PSI(I-1,J)
      A7 = PSI(I-1,J-1)
      AJZP = FACM2 * (ZETA(I-1,J) * (- A7 + A1) + ZETA(I,J+1) *
     1       (- A4 - A1) + ZETA(I,J-1) * (A7 + A4) + A4 *
     2       (ZETA(I-1,J-1) - ZETA(I-1,J+1)))
      DRHO = RHO(I,J) - RHO(I-1,J)
      GO TO 335
C     JACOBIAN FOR POINT TYPE 4
  305 A3 = PSI(I+1,J+1)
      A6 = PSI(I+1,J)
      A9 = PSI(I+1,J-1)
      AJZP = FACM2 * (ZETA(I+1,J) * (A9 - A3) + ZETA(I,J+1) * (A6 + A3)
     1       + ZETA(I,J-1) * (- A9 - A6) + A6 * (ZETA(I+1,J+1) -
     2       ZETA(I+1,J-1)))
      DRHO = RHO(I+1,J) - RHO(I,J)
      GO TO 335
C     JACOBIAN FOR POINT TYPE 5
  310 AJZP = FACM4 * PSI(I-1,J+1) * (ZETA(I-1,J) - ZETA(I,J+1))
      DRHO = 0.
      GO TO 335
C     JACOBIAN FOR POINT TYPE 6
  315 AJZP = FACM4 * PSI(I+1,J+1) * (ZETA(I,J+1) - ZETA(I+1,J))
      DRHO = 0.
      GO TO 335
C     JACOBIAN FOR POINT TYPE 7
  320 A1 = PSI(I-1,J+1)
      A2 = PSI(I,J+1)
      A3 = PSI(I+1,J+1)
      A4 = PSI(I-1,J)
      A7 = PSI(I-1,J-1)
      AJZP = FACM43 * (ZETA(I+1,J) * (- A2 - A3) + ZETA(I-1,J) *
     1       (- A7 + A1 + A2) + ZETA(I,J+1) * (A3 - A4 - A1) +
     2       ZETA(I,J-1) * (A7 + A4) + A2 * (ZETA(I-1,J+1) -
     3       ZETA(I+1,J+1)) + A4 * (ZETA(I-1,J-1) - ZETA(I-1,J+1)))
      DRHO = (RHO(I+1,J) - RHO(I-1,J)) / 2.
      GO TO 335
C     JACOBIAN FOR POINT TYPE 8
  325 A1 = PSI(I-1,J+1)
      A2 = PSI(I,J+1)
      A3 = PSI(I+1,J+1)
      A6 = PSI(I+1,J)
      A9 = PSI(I+1,J-1)
      AJZP = FACM43 * (ZETA(I+1,J) * (A9 - A2 - A3) + ZETA(I-1,J) *
     1       (A1 + A2) + ZETA(I,J+1) * (A6 + A3 - A1) + ZETA(I,J-1) *
     2       (- A9 - A6) + A5 * (ZETA(I+1,J+1) - ZETA(I+1,J-1)) + A2 *
     3       (ZETA(I-1,J+1) - ZETA(I+1,J+1)))
      DRHO = (RHO(I+1,J) - RHO(I-1,J))/ 2.
      GO TO 335
C     NEWZTA FOR POINT TYPE 9
  330 NEWZTA(I,J) = OLDZTA(I,J)
      GO TO 340
C     CALCULATE NEWZTA FOR POINT TYPES 0 THROUGH 8
  335 NEWZTA(I,J) = (AJZP - (G * DRHO) / (DELTAX * RHOJ(J))) * DELTAT
     1              + OLDZTA(I,J)
```

```
  340 CONTINUE
  345 CONTINUE
      IF(MOD(MT,25).EQ.0) GO TO 355
C-----------------------------------------------------------------------
C     P. SET OLDZTA = ZETA, OLDRHO = RHO
C
      DO 350 I = 2,NIM1
      DO 350 J = 1,NJ
      OLDZTA(I,J) = ZETA(I,J)
  350 OLDRHO(I,J) = RHO(I,J)
      GO TO 365
  355 DELTAT = DELTAT * 2.
      IF(IB25.EQ.1) GO TO 360
      IB25 = 1
      MT = MT - 1
      GO TO 365
  360 IB25 = 0
C
C     Q. SET ZETA = NEWZTA, RHO = NEWRHO
C
  365 DO 370 I = 2,NIM1
      DO 370 J = 1,NJ
      ZETA(I,J) = NEWZTA(I,J)
  370 RHO(I,J) = NEWRHO(I,J)
      DO 375 J = 1,NJ
      ZETA(1,J) = ZETA(NIM1,J)
      ZETA(NI,J) = ZETA(2,J)
      RHO(1,J) = RHO(NIM1,J)
  375 RHO(NI,J) = RHO(2,J)
C
C     R. CALCULATE PSI FOR NEXT TIME STEP
C
C     SOLVE POISSON EQUATION FOR PSI
      CALL FA(PSI,ZETA,FTR,FTI,PSIBFT,AD,UA1,UA2,UA3,UA4,UA5,NI,NJ,NIM2,
     1        NJM2)
C     SUPERPOSITION SOLUTION FOR PSI AT TIME STEP MT
      CALL SP(PSI,PSIAUX,ITYPE,BETA,IB,JB,SPSI,NI,NJ,NIM2,NJM2,NIPA,
     1        JMAXM1,NPB,NPBP2)
      MT = MT + 1
      GO TO 235
      END
      SUBROUTINE CALCRI(PSI,RHO,ZETA,RI,AD,UP,B,Z,D,D2PSI,DRHO,NI,NJ)
C
C     THIS ROUTINE CALCULATES THE RICHARDSON NUMBER FIELD RI
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 PSI(NI,NJ),RHO(NI,NJ),ZETA(NI,NJ),RI(NI,NJ),UP(NJ),
     1       AD(NJ),B(NJ),Z(NJ),D(NJ),D2PSI(NJ),DRHO(NJ)
      COMMON/C1/DELTAX,DXM2,DELTAZ,DZM2,DZSQ,DZ06,G,NI1,NJM1,IEXP2,
     1          IS,JS,IE,JE,ISA,IEW,NPBP1,JMAX,ISB,IEB
C-----------------------------------------------------------------------
C     A. CALCULATE SECOND DERIVATIVE OF PSI
C
      UP(1) = AD(1)
      D(1) = 0.
      DO 10 I = IS,IE
C     BOUNDARY CONDITIONS ON SECOND DERIVATIVE OF PSI
      D2PSI(1) = ZETA(I,1)
      D2PSI(NJ) = ZETA(I,NJ)
      DB = D2PSI(1) * DZ06
      DT = D2PSI(NJ) * DZ06
C     CALCULATE B MATRIX ELEMENTS
      DO 1 J = 2,NJ
      D(J) = (PSI(I,J) - PSI(I,J-1)) / DELTAZ
    1 B(J-1) = 3. * (D(J) + D(J-1))
      B(1) = B(1) - 3. * DB
      B(NJ) = 3. * (D(NJ) + DT)
C     TRIDIAGONAL SPLINE MATRIX INVERSION SCHEME FOR FIRST DERIVATIVE OF
C     PSI
      Z(1) = B(1)
      DO 2 J = 2,NJ
      UL = -1. / UP(J-1)
      UP(J) = AD(J) + UL
    2 Z(J) = B(J) + UL * Z(J-1)
      DPSI1 = Z(NJ) / UP(NJ)
C     CALCULATE SECOND DERIVATIVE OF PSI FROM FIRST DERIVATIVE OF PSI
      DO 3 J = 2,NJM1
      K = NJ - J + 1
      DPSI2 = (Z(K) - DPSI1) / UP(K)
      D2PSI(K) = (-4. * DPSI2 + 6. * D(K+1) - 2. * DPSI1) / DELTAZ
    3 DPSI1 = DPSI2
C
C     B. CALCULATE FIRST DERIVATIVE OF RHO
```

```
C
      DO 4 J = 2,NJ
      D(J) = (RHO(I,J) - RHO(I,J-1)) / DELTAZ
C     CALCULATE B MATRIX ELEMENTS
    4 B(J-1) = 3. * (D(J) + D(J-1))
      B(NJ) = 3. * D(NJ)
C     TRIDIAGONAL SPLINE MATRIX INVERSION SCHEME
      Z(1) = B(1)
      DO 5 J = 2,NJ
      UL = -1. / UP(J-1)
      UP(J) = AD(J) + UL
    5 Z(J) = B(J) + UL * Z(J-1)
      DRHO(NJ) = Z(NJ) / UP(NJ)
      DO 6 J = 2,NJ
      K = NJ - J + 1
    6 DRHO(K) = (Z(K) - DRHO(K+1)) / UP(K)
C
C     C. CALCULATE RI FROM DRHO, D2PSI
C
      DO 9 J = JS,JE
      IF(D2PSI(J).EQ.0.) GO TO 8
      R = - G * DRHO(J) / (RHO(I,J) * D2PSI(J)**2)
      IF(R.GT..25) GO TO 7
      R = .25
      GO TO 9
    7 IF(R.LT.9.99) GO TO 9
    8 R = 9.99
    9 RI(I,J) = DLOG10(R)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE CALCU(PSI,ZETA,J,AD,UP,B,Z,D,NI,NJ)
C
C     THIS ROUTINE CALCULATES THE HORIZONTAL VELOCITY FIELD U
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 PSI(NI,NJ),ZETA(NI,NJ),U(NI,NJ),UP(NJ),AD(NJ),B(NJ),Z(NJ),
     1       D(NJ)
      COMMON/C1/DELTAX,DXM2,DELTAZ,DZM2,DZSQ,DZD6,S,NIM1,NJM1,IEXP2,
     1          IS,JS,IE,JE,ISM,IEM,NPLP1,JMAX,ISB,IEB
      UP(1) = AD(1)
      D(1) = 0.
      DO 3 I = IS,IE
C     BOUNDARY CONDITIONS ON SECOND DERIVATIVE OF PSI
      DB = ZETA(I,1) * DZD6
      DT = ZETA(I,NJ) * DZD6
C     CALCULATE B MATRIX ELEMENTS
      DO 1 J = 2,NJ
      D(J) = (PSI(I,J) - PSI(I,J-1)) / DELTAZ
    1 B(J-1) = 3. * (D(J) + D(J-1))
      B(1) = B(1) - 3. * DB
      B(NJ) = 3. * (D(NJ) + DT)
C     TRIDIAGONAL SPLINE MATRIX INVERSION SCHEME FOR J
      Z(1) = B(1)
      DO 2 J = 2,NJ
      UL = -1. / UP(J-1)
      UP(J) = AD(J) + UL
    2 Z(J) = B(J) + UL * Z(J-1)
      U(I,NJ) = - Z(NJ) / UP(NJ)
      DO 3 J = 2,NJ
      K = NJ - J + 1
    3 U(I,K) = - (Z(K) + U(I,K+1)) / UP(K)
      RETURN
      END
      SUBROUTINE CALCW(PSI,W,FTR,FTI,AK,NI,NJ,NIM2)
C
C     THIS ROUTINE CALCULATES THE VERTICAL VELOCITY FIELD W
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 PSI(NI,NJ),W(NI,NJ),FTR(NIM2),FTI(NIM2),AK(NIM2)
      COMMON/C1/DELTAX,DXM2,DELTAZ,DZM2,DZSQ,DZD6,S,NIM1,NJM1,IEXP2,
     1          IS,JS,IE,JE,ISM,IEM,NPLP1,JMAX,ISB,IEB
C     CALCULATE FOURIER TRANSFORM OF PSI
      DO 3 J = JS,JE
      DO 1 I = 1,NIM2
      FTR(I) = PSI(I,J)
    1 FTI(I) = 0.
      CALL FAST(FTR,FTI,NIM2,IEXP2,-1)
C     CALCULATE FOURIER COEFFICIENTS OF W
      DO 2 I = 1,NIM2
      T = FTR(I)
      FTR(I) = AK(I) * FTI(I)
```

```
    2 FTI(I) = AK(I) * T
C     CALCULATE W BY TAKING INVERSE TRANSFORM
      CALL FAST(FTR,FTI,NIM2,IEXP2,1)
      W(NIM,J) = FTR(1)
      W(NI,J) = FTR(2)
      DO 3 I = IS,NIM2
    3 W(I,J) = FTR(I)
      RETURN
      END
      SUBROUTINE FA(PSI,ZETA,FTR,FTI,PSIBFT,AD,BR,BI,ZR,ZI,UP,NI,NJ,
C
C     THIS ROUTINE SOLVES THE POISSON EQUATION FOR THE STREAMFUNCTION
C     USING A FOURIER TRANSFORM IN X, MARCHING SOLUTION IN Z TECHNIQUE
C
     1                NIM2,NJM2)
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 PSI(NI,NJ),ZETA(NI,NJ),FTR(NI,NJ),FTI(NI,NJ),
     1       PSIBFT(NIM2,2),AD(NIM2),BR(NJ),BI(NJ),ZR(NJ),ZI(NJ),UP(NJ)
      COMMON/C1/DELTAX,DXM2,DELTAZ,DZM2,DZSQ,DZD6,S,NI41,NJM1,IEXP2,
     1          IS,JS,IE,JE,IS4,IEW,NPBP1,JMAX,ISB,IEB
C     CALCULATE FOURIER TRANSFORM OF ZETA
      DO 2 J = 2,NJM1
      DO 1 I = 1,NIM2
      FTR(I,J) = ZETA(I,J)
    1 FTI(I,J) = 0.
    2 CALL FAST(FTR(1,J),FTI(1,J),NIM2,IEXP2,1)
C     CALCULATE B MATRIX ELEMENTS
      DO 5 I = 1,NIM2
      DO 3 J = 1,NJM2
      BR(J) = - DZSQ * FTR(I,J+1)
    3 BI(J) = - DZSQ * FTI(I,J+1)
      BR(NJM2) = BR(NJM2) + PSIBFT(I,1)
      BI(NJM2) = BI(NJM2) + PSIBFT(I,2)
C     CALCULATE FOURIER COEFFICIENTS OF PSI BY TRIDIAGONAL MATRIX
C     INVERSION SCHEME
      UP(1) = AD(1)
      ZR(1) = BR(1)
      ZI(1) = BI(1)
      DO 4 J = 2,NJM2
      UL = -1. / UP(J-1)
      UP(J) = AD(I) + UL
      ZR(J) = BR(J) - UL * ZR(J-1)
    4 ZI(J) = BI(J) - UL * ZI(J-1)
      FTR(I,NJM1) = ZR(NJM2) / UP(NJM2)
      FTI(I,NJM1) = ZI(NJM2) / UP(NJM2)
      DO 5 J = 2,NJM2
      K = NJM2 - J + 1
      FTR(I,K+1) = (ZR(K) + FTR(I,K+2)) / UP(K)
    5 FTI(I,K+1) = (ZI(K) + FTI(I,K+2)) / UP(K)
C     CALCULATE PSI BY TAKING INVERSE TRANSFORM
      DO 6 J = 2,NJM1
    6 CALL FAST(FTR(1,J),FTI(1,J),NIM2,IEXP2,-1)
      RETURN
      END
      SUBROUTINE FAST(X,Y,N,M,INV)
C
C     THIS ROUTINE CALCULATES THE FAST FOURIER TRANSFORM OF THE COMPLEX
C     ARRAY (X,Y) IN PLACE, WHERE THE LENGTH OF X AND Y IS A POWER OF
C     TWO
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 X(N),Y(N)
      IMAX = N
      PIM2 = 6.28318530718D+00
      DO 2 L = 1,M
      JDELT = IMAX
      IMAX = IMAX / 2
      FJ = JDELT
      ARG = INV * PIM2 / FJ
      C = DCOS(ARG)
      S = DSIN(ARG)
      U = 1.
      V = 0.
      DO 2 I = 1,IMAX
      DO 1 J = I,N,JDELT
      K = J + IMAX
      XJ = X(J) + X(K)
      YJ = Y(J) + Y(K)
      XK = X(J) - X(K)
      YK = Y(J) - Y(K)
      X(K) = U * XK - V * YK
      Y(K) = U * YK + V * XK
    1 X(J) = XJ
```

```
    1 Y(J) = YJ
      T = C * U - S * V
      V = C * V + S * U
    2 U = T
      J = 1
      NT = N / 2
      IMAX = N - 1
      DO 5 I = 1,IMAX
      IF(I.GE.J) GO TO 3
      T = X(J)
      X(J) = X(I)
      X(I) = T
      T = Y(J)
      Y(J) = Y(I)
      Y(I) = T
    3 K = NT
    4 IF(K.GE.J) GO TO 5
      J = J - K
      K = K / 2
      GO TO 4
    5 J = J + K
      IF(INV.EQ.1) RETURN
      FN = N
      DO 6 I = 1,N
      X(I) = X(I) / FN
    6 Y(I) = Y(I) / FN
      RETURN
      END
      SUBROUTINE GE(X,DET,N1,N2,IACC)
C
C     THIS ROUTINE CALCULATES THE DETERMINANT OF THE MATRIX X, IN PLACE,
C     USING GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 X(N1,N1)
      N2M1 = N2 - 1
      IACC = 0
      DET = 1.
      DO 7 I = 1,N2M1
C     FOR EACH ROW, DETERMINE LARGEST MATRIX ELEMENT
      XMAX = 0.
      K = 0
      DO 1 J = I,N2
      Y = DABS(X(I,J))
      IF(Y.LE.XMAX) GO TO 1
      XMAX = Y
      K = J
    1 CONTINUE
      IF(K.NE.0) GO TO 2
      DET = 0.
      RETURN
    2 IF(K.EQ.I) GO TO 4
C     INTERCHANGE COLUMNS IF NECESSARY
      DO 3 J = I,N2
      T = X(J,I)
      X(J,I) = X(J,K)
    3 X(J,K) = T
      DET = - DET
    4 IP1 = I + 1
C     MULTIPLY ROW CONTAINING PIVOTAL ELEMENT BY APPROPRIATE CONSTANTS,
C     AND ADD TO ROWS BENEATH, SO AS TO ZERO OUT ALL ELEMENTS BENEATH
C     PIVOTAL ELEMENT
      DO 6 J = IP1,N2
      IF(X(J,I).EQ.0.) GO TO 6
      D = X(J,I) / X(I,I)
      DO 5 K = IP1,N2
    5 X(J,K) = X(J,K) - D * X(I,K)
    6 CONTINUE
    7 CONTINUE
C     CALCULATE DETERMINANT FROM DIAGONAL ELEMENTS
      DO 8 I = 1,N2
      IF(DABS(DET).LT.1.D+50) GO TO 8
      DET = DET * 1.D-50
      IACC = IACC + 1
    8 DET = DET * X(I,I)
      RETURN
      END
      SUBROUTINE GRAFIC(X,A,LINE34,LINE3,NG,NLEV,N1,NJ,NJSM1)
C
C     THIS ROUTINE CREATES A SHADED, LINE PRINTER GRAPHICAL DISPLAY
C     OF THE FIELD X WITHIN THE DEFINED LIMITS OF INTEREST
C
```

```
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 X(NI,NJ),A(NI,NJ),LINE48(14),LINE58(14),LINE68(14),
     1       OLD(132),NEW(132),V1(13),V2(13)
      REAL*4 LINE14(33),LINE24(33),LINE34(33,NJSM1),NAME(2,6)
      LOGICAL*1 LINE2(132),LINE3(132,NJSM1),SYM(53)
      EQUIVALENCE (LINE2(1),LINE24(1))
      COMMON/C1/DELTAX,DX42,DELTAZ,DZM2,DZSQ,DZD6,S,NI41,NJ41,IEXP2,
     1          IS,JS,IE,JE,ISW,IEW,NPBP1,JMAX,ISO,IEO
      COMMON/C2/LINE48,LINE58,LINE68,LINE14,LINE24,NAME,
     1          MT,IN,JN,INM1,NIS,NISM1,IL,SYM
C     DETERMINE MINIMUM AND MAXIMUM VALUES IN THE FIELD, AND STORE FIELD
C     UPSIDE DOWN INTO A
      XMIN = X(IS,JS)
      XMAX = X(IS,JS)
      DO 5 J = JS,JE
      K = JE - J + 1
      DO 5 I = IS,IE
      Y = X(I,J)
      A(I-IS+1,K) = Y
      IF(Y.LT.XMIN) XMIN = Y
    5 IF(Y.GT.XMAX) XMAX = Y
C     CALCULATE INCREMENT BETWEEN SUCCESSIVE LEVELS OF SHADING
      D = (XMAX - XMIN) / NLEV
      IF(D.GT.1.D-12) GO TO 10
      WRITE(6,2001) (NAME(I,NQ),I=1,2),MT,XMIN
 2001 FORMAT('1H1,2A4,' FIELD, TIME STEP =',I4,'. IS VIRTUALLY CONSTANT',
     1       10X,'CONSTANT =',1PD10.2)
      RETURN
   10 DL = DLOG10(D)
      IDL = DL
      IF(DL.LT.0.) IDL = IDL - 1
      SIGPRT = 10. ** (DL-IDL)
      IF(SIGPRT.GT.2.) GO TO 15
      SIGPRT = 2.
      GO TO 25
   15 IF(SIGPRT.GT.5.) GO TO 20
      SIGPRT = 5.
      GO TO 25
   20 SIGPRT = 10.
   25 XINC = SIGPRT * 10. ** IDL
C     CALCULATE MINIMUM AND MAXIMUM VALUES TO BE PLOTTED
      DV = (NJSM1 + 1) * XINC
      IMIN = XMIN / XINC
      IF(XMIN.LT.0.) IMIN = IMIN - 1
      XMIN = IMIN * XINC
      IMAX = XMAX / XINC
      IF(XMAX.GT.0.) IMAX = IMAX + 1
      XMAX = IMAX * XINC
C     WRITE HEADING
      WRITE(6,2002) (NAME(I,NQ),I=1,2),MT,XMIN,XMAX,XINC,IS,JS,LINE14
 2002 FORMAT('1H1,4X,2A4,' FIELD, TIME STEP =',I4,', MINIMUM =',1PD10.2,
     1       ', MAXIMUM =',1PD10.2,', INTERVAL =',1PD10.2,
     2       ', BASE POINT = (',I2,',',I2,')'//1X,33A4)
      DO 50 J = 1,JN
C     LINEAR INTERPOLATION TO DETERMINE THE VALUE OF THE FIELD AT EVERY
C     PRINT CHARACTER IN A LINE
      DO 30 I = 1,INM1
      DIFI = (A(I+1,J) - A(I,J)) / NIS
      L = NIS * (I - 1) + 2
      NEW(L) = A(I,J) - XMIN
      DO 30 K = 1,NISM1
   30 NEW(K+L) = NEW(K+L-1) + DIFI
      NEW(IL) = A(IN,J) - XMIN
      IF(J.EQ.1) GO TO 40
C     LINEAR INTERPOLATION TO DETERMINE VALUE OF FIELD ON LINES BETWEEN
C     VERTICAL GRID POINTS
      DO 35 I = 2,IL
      DIFJ = (NEW(I) - OLD(I)) / DV
      TEMP = OLD(I) / XINC + 2.
      DO 35 K = 1,NJSM1
      TEMP = TEMP + DIFJ
      L = TEMP
C     DETERMINE SYMBOLS ASSOCIATED WITH THE VALUES OF THE FIELD AT EACH
C     PRINT CHARACTER, AND PRINT LINES
   35 LINE3(I,K) = SYM(L)
      WRITE(6,2003) LINE34
 2003 FORMAT(1X,33A4)
   40 DO 45 I = 2,IL
      L = NEW(I) / XINC + 2.
   45 LINE2(I) = SYM(L)
      WRITE(6,2003) LINE24
      DO 50 I = 2,IL
   50 OLD(I) = NEW(I)
```

```
      WRITE(6,2003) LINE14
C     CALCULATE THE VALUE OF EACH LINE SEPARATING LEVELS OF SHADING
      E = DMAX1(DABS(XMIN),DABS(XMAX))
      EL = DLOG10(E)
      IEL = EL
      IF(EL.LT.0.) IEL = IEL - 1
      V2(1) = XMIN / 10. ** IEL
      XINC = 2. * XINC / 10. ** IEL
      V1(1) = V2(1) + XINC
      XINC = 2. * XINC
      DO 55 I = 2,13
      V1(I) = V1(I-1) + XINC
   55 V2(I) = V2(I-1) + XINC
C     WRITE SHADING VALUE SCALE
      WRITE(6,2004) V1,LINE48,LINE58,IEL,LINE68,V2
 2004 FORMAT(1H0,15X,13F8.2/1X,'SCALE (TO BE ',14A8/1X,'MULTIPLIED ',
     1       14A8/1X,'BY 10**',I3,2X,14A8/12X,13F8.2)
      RETURN
      END
      SUBROUTINE GRAFIN(LINE34,LINE3,NJSM1)
C
C     THIS ROUTINE INITIALIZES ANNOTATION FOR GRAPHICAL OUTPUT
C
      REAL*8 LINE48(14),LINE58(14),LINE68(14),BLANK8
      REAL*4 LINE14(33),LINE24(33),LINE34(33,NJSM1),NAME1(2,6),
     1       NAME2(2,6)
      LOGICAL*1 LINE1(132),LINE2(132),LINE3(132,NJSM1),LINE4(112),
     1          LINE5(112),LINE6(112),SYM1(53),SYM2(53),DOT,DASH,BLANK
      EQUIVALENCE (LINE1(1),LINE14(1)),(LINE2(1),LINE24(1)),
     1            (LINE4(1),LINE48(1)),(LINE5(1),LINE58(1)),
     2            (LINE6(1),LINE68(1))
      COMMON/C1/DELTAX,DXM2,DELTAZ,DZM2,DZSQ,DZD5,U,NM1,NJM1,IEXP2,
     1          IS,JS,IE,JE,ISM,IEM,NPBP1,JMAX,ISB,IEB
      COMMON/C2/LINE48,LINE58,LINE68,LINE14,LINE24,NAME2,
     1          MT,IN,JN,INM1,NIS,NISM1,IL,SYM2
      DATA SYM1/' ','A',' ','B',' ','C',' ','D',' ','E',' ','F',' ','G',
     1          ' ','H',' ','I',' ','J',' ','K',' ','L',' ','M',' ','N',
     2          ' ','O',' ','P',' ','Q',' ','R',' ','S',' ','T',' ','U',
     3          ' ','V',' ','W',' ','X',' ','Y',' ','Z',' '/
      DATA NAME1/'    ',' PSI','    ',' RHO','    ','ZETA',
     1           '    ','    ','  U','    ','    ','  W',' LOG','(R1)'/
      DATA BLANK8/'        '/,BLANK4/'    '/,BLANK/' '/,DASH/'|'/,
     1     DOT/'.'/
C     CALCULATE NUMBER OF PRINT CHARACTERS FROM ONE HORIZONTAL GRID
C     POINT TO THE NEXT
      NIS = 129 / INM1
      NFB = 12 + NIS * INM1 / 4 + 1
      NISM1 = NIS - 1
      IL = NIS * INM1 + 2
C     INITIALIZE TITLES AND SYMBOLS
      DO 1 I = 1,2
      DO 1 J = 1,6
    1 NAME2(I,J) = NAME1(I,J)
      DO 2 I = 1,53
    2 SYM2(I) = SYM1(I)
C     INITIALIZE ARRAYS USED IN PRINT LINES
      DO 3 I = 1,33
    3 LINE14(I) = BLANK4
      DO 4 I = 1,IN
    4 LINE1(NIS*(I-1)+2) = DOT
      DO 5 I = NFB,33
    5 LINE24(I) = BLANK4
      LINE2(1) = DOT
      LINE2(IL+1) = DOT
      DO 6 I = 1,NJSM1
      LINE3(1,I) = BLANK
      DO 6 J = NFB,33
    6 LINE34(J,I) = BLANK4
      DO 7 I = 1,14
      LINE48(I) = BLANK8
      LINE58(I) = BLANK8
    7 LINE68(I) = BLANK8
      DO 8 I = 1,13
      LINE4(8*I+1) = DASH
    8 LINE6(8*I-3) = DASH
      DO 9 I = 1,26
    9 LINE5(4*I+2) = SYM1(2*I)
      DO 10 I = 1,52
   10 LINE5(2*I+3) = DASH
      RETURN
      END
      SUBROUTINE PTSPEC(ITYPE,IB,JB,NI,NJ,NPBP2)
C
C     THIS ROUTINE DETERMINES THE TYPE OF EACH GRID POINT IN THE FIELD
```

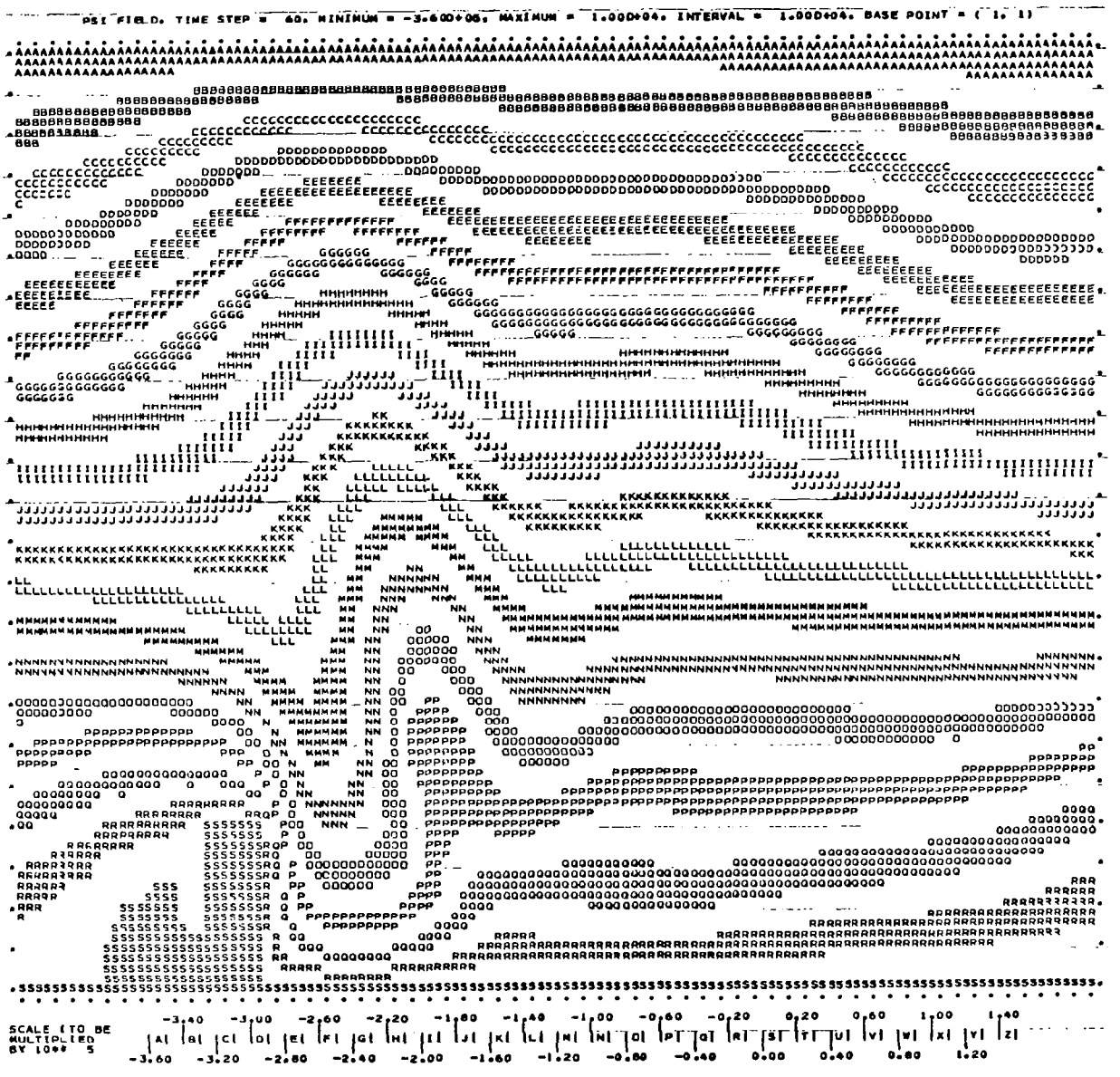
```
C
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER*4 ITYPE(NI,NJ),IB(NPBP2),JB(NPBP2)
      COMMON/C1/DELTAX,DXM2,DELTAZ,DZM2,DZSQ,DZD6,S,NIM1,NJM1,IEXP2,
     1          IS,JS,IE,JE,ISM,IEM,NPBP1,JMAX,ISB,IEB
C     INITIALIZE ALL INTERNAL POINTS, AND UPPER AND LOWER BOUNDARIES
      DO 2 I = 1,NI
      DO 1 J = 2,NJM1
    1 ITYPE(I,J) = 0
      ITYPE(I,NJ) = 1
    2 ITYPE(I,1) = 2
C     DETERMINE TYPE OF ALL POINTS ON THE SURFACE OF THE BARRIER,
C     PROCEEDING FROM LEFT TO RIGHT
      ISB = IB(2)
      IEB = IB(NPBP1)
      IB(1) = ISB
      JB(1) = 1
      IB(NPBP2) = IEB
      JB(NPBP2) = 1
      ITYPE(ISB,1) = 5
      ITYPE(IEB,1) = 6
      JMAX = 1
      DO 9 I = 2,NPBP1
      K = IB(I)
      L = JB(I)
      IF(L.GT.JMAX) JMAX = L
      IF(JB(I-1)-L) 3,4,8
    3 ITYPE(K,L) = 3
      IF(L.EQ.JB(I+1)) ITYPE(K,L) = 7
      GO TO 9
    4 IF(L-JB(I+1)) 5,6,7
    5 ITYPE(K,L) = 5
      GO TO 9
    6 ITYPE(K,L) = 2
      GO TO 9
    7 ITYPE(K,L) = 8
      GO TO 9
    8 ITYPE(K,L) = 4
      IF(L.EQ.JB(I+1)) ITYPE(K,L) = 6
    9 CONTINUE
      JMAXM1 = JMAX - 1
C     DEFINE ALL POINTS INSIDE THE BARRIER TO BE TYPE 9
      DO 10 J = 1,JMAXM1
      K = 0
      DO 10 I = ISB,IEB
      L = ITYPE(I,J)
      IF(L.LT.3.AND.K.EQ.1) ITYPE(I,J) = 9
      IF(L.EQ.3.OR.L.EQ.5) K = 1
   10 IF(L.EQ.4.OR.L.EQ.6) K = 0
      RETURN
      END
      SUBROUTINE SP(X,PSIAUX,ITYPE,BETA,IB,JB,X0,NI,NJ,NIM2,NJM2,NIPA,
     1              JMAXM1,NPB,NPBP2)
C
C     THIS ROUTINE SUPERPOSES THE SOLUTION OF THE POISSON EQUATION WITH
C     THE EFFECT ON PSI GENERATED BY THE VORTICITY FROM EACH POINT ON
C     THE SURFACE OF THE BARRIER
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 X(NI,NJ),PSIAUX(NIPA,NJM2,JMAXM1),BETA(NPB,NPB)
      INTEGER*4 ITYPE(NI,NJ),IB(NPBP2),JB(NPBP2)
      COMMON/C1/DELTAX,DXM2,DELTAZ,DZM2,DZSQ,DZD6,S,NIM1,NJM1,IEXP2,
     1          IS,JS,IE,JE,ISM,IEM,NPBP1,JMAX,ISB,IEB
      DO 2 K = 1,NPB
      ALPHA = 0.
      DO 1 L = 1,NPB
C     CALCULATE ALPHA FROM BETA
    1 ALPHA = ALPHA + BETA(K,L) * (X0 - X(IB(L+1),JB(L+1)))
      L = IB(NPBP1) - IB(K+1)
      M = JB(K+1) - 1
      DO 2 I = 1,NIM2
      DO 2 J = 1,NJM2
    2 X(I,J+1) = X(I,J+1) + ALPHA * PSIAUX(L+I,J,M)
C     DEFINE PSI AT SIDE BOUNDARIES
      DO 3 J = 2,NJM1
      X(NIM1,J) = X(1,J)
    3 X(NI,J) = X(2,J)
C     REDEFINE BOUNDARY CONDITION EXACTLY ON AND INSIDE BARRIER
      DO 4 I = ISB,IEB
      DO 4 J = 2,JMAX
    4 IF(ITYPE(I,J).NE.0) X(I,J) = X0
      RETURN
      END
```

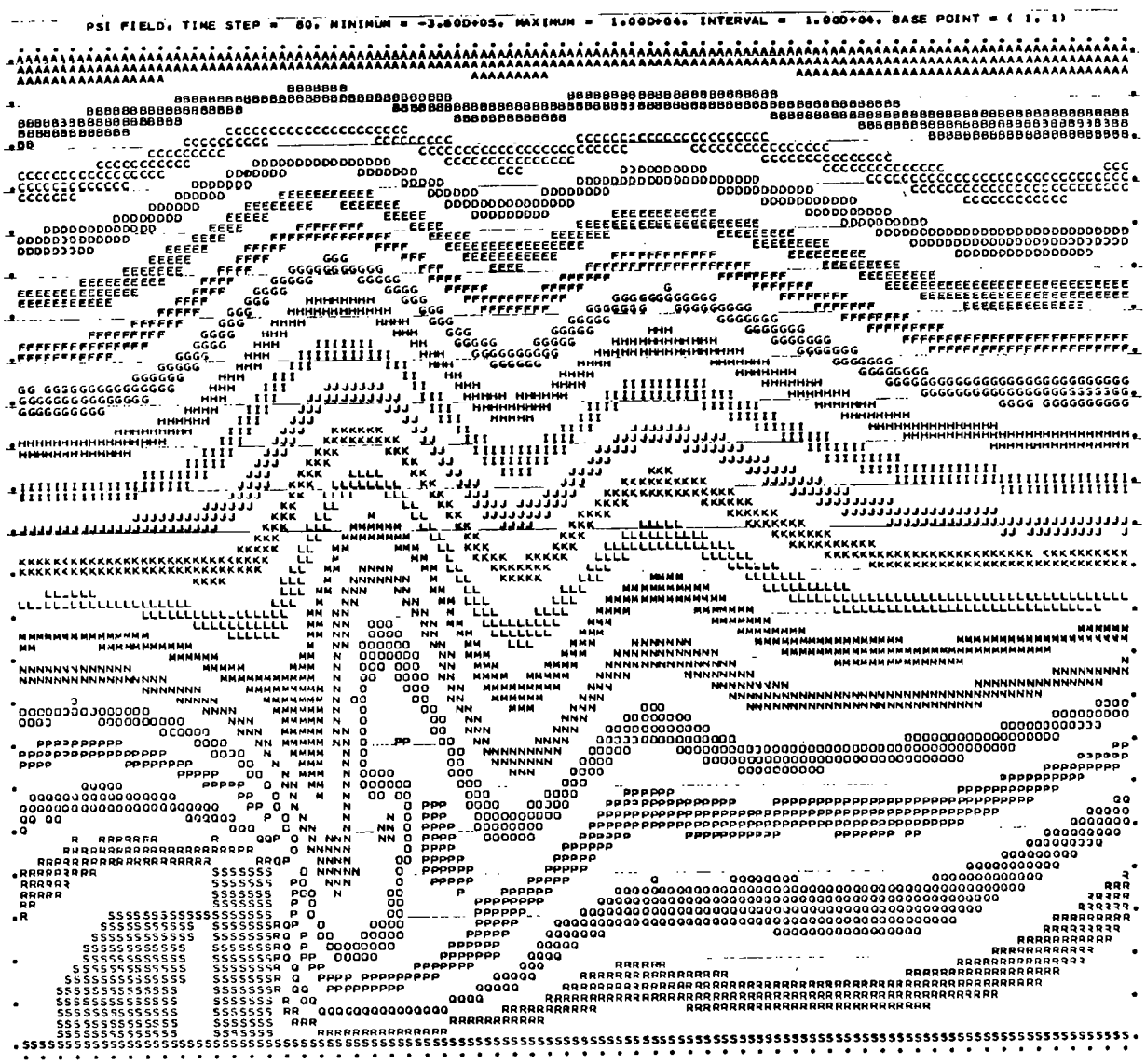
Listing of data cards

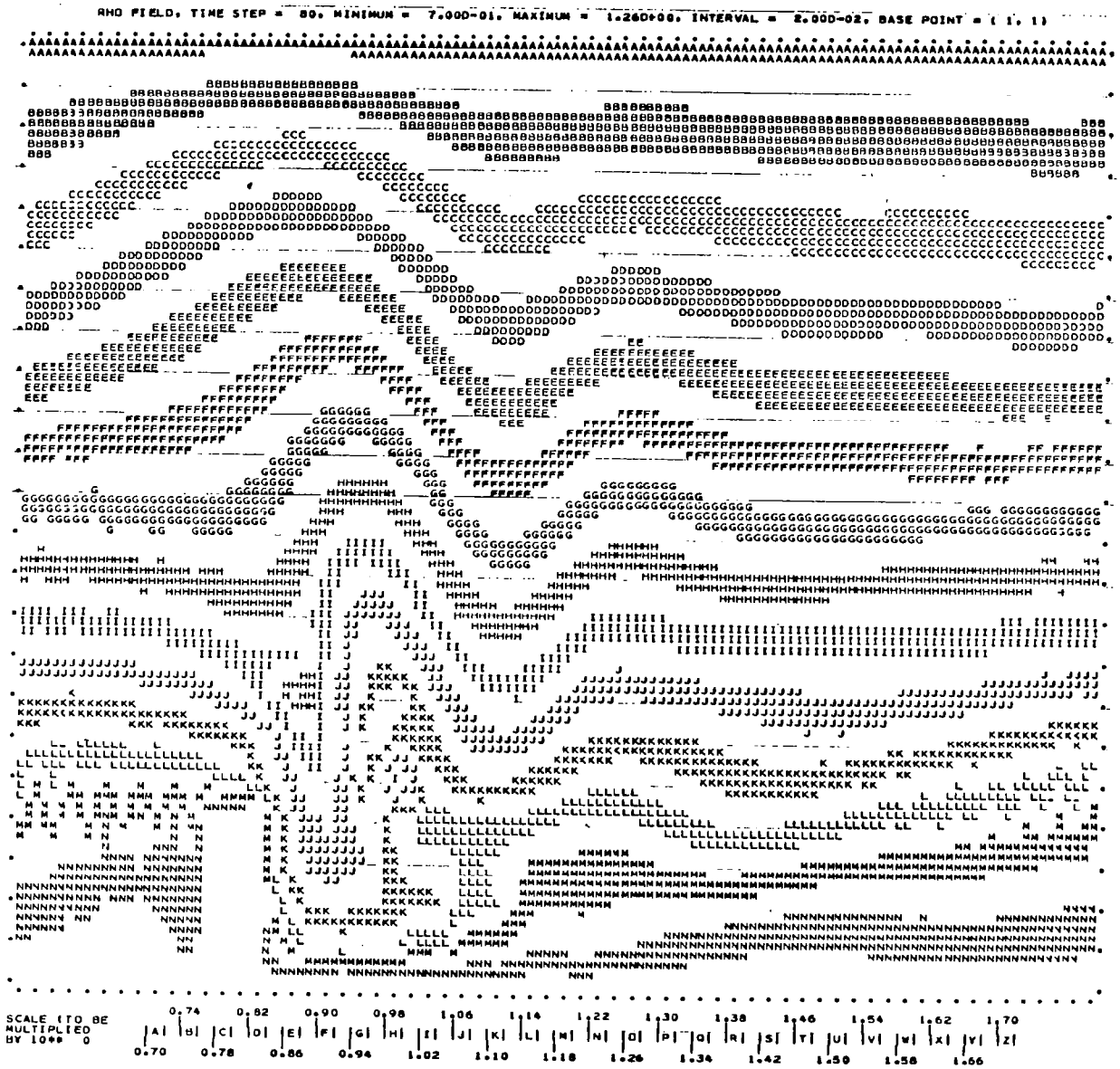
```
65  24  80  10   0   0  6.25D+02  6.25D+02  1.00D+01
12   2  12   3  12   4  12   5  13   5  14   5  15   5  15   4  15   3  15   2
 1   1  65  24   2
40  40  25  25  25   9
60  80
80
80
80
80
80
 2
2.50D+01
 1
0.00D+00   2.50D+02
```

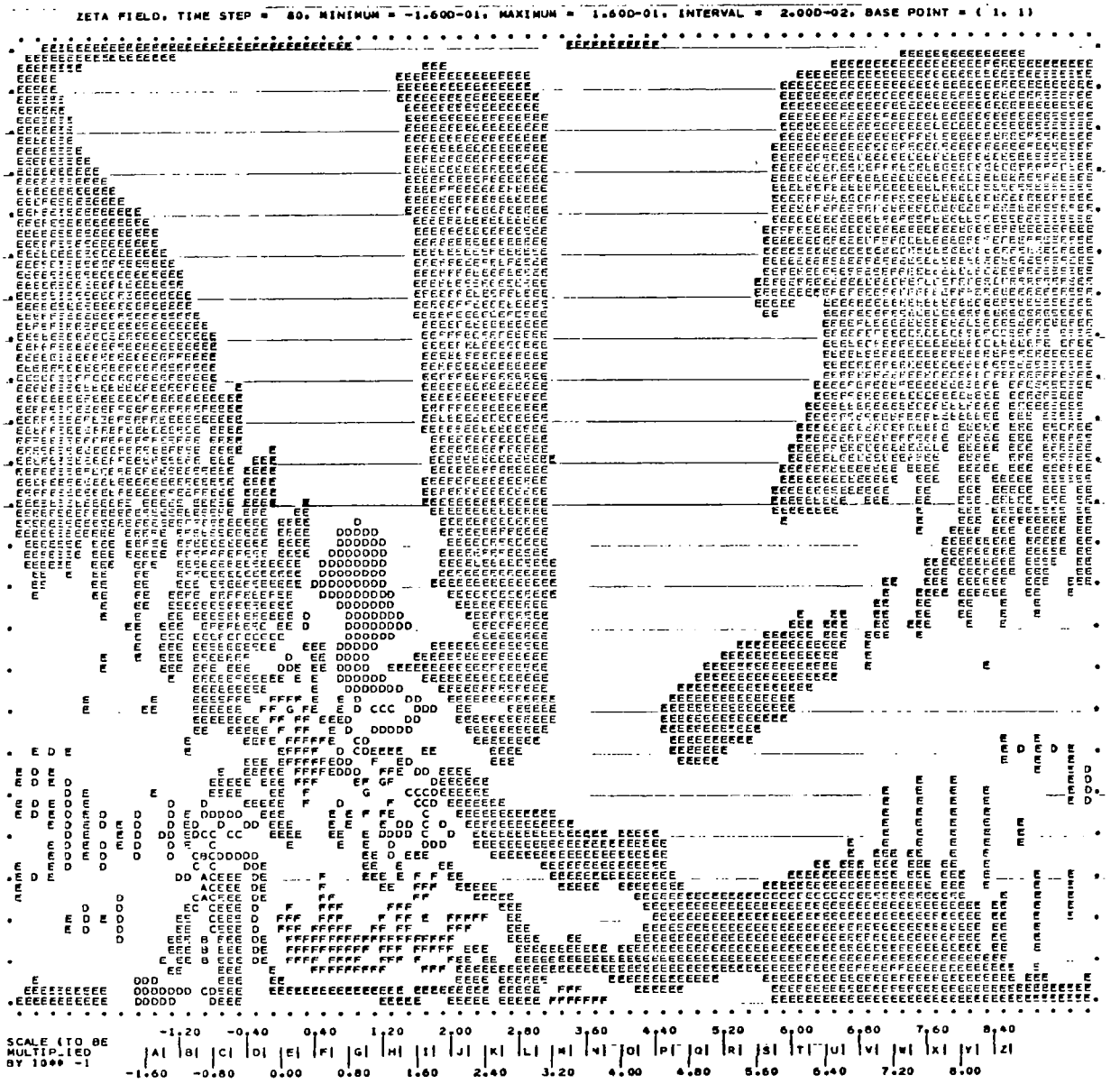
Beginning of output

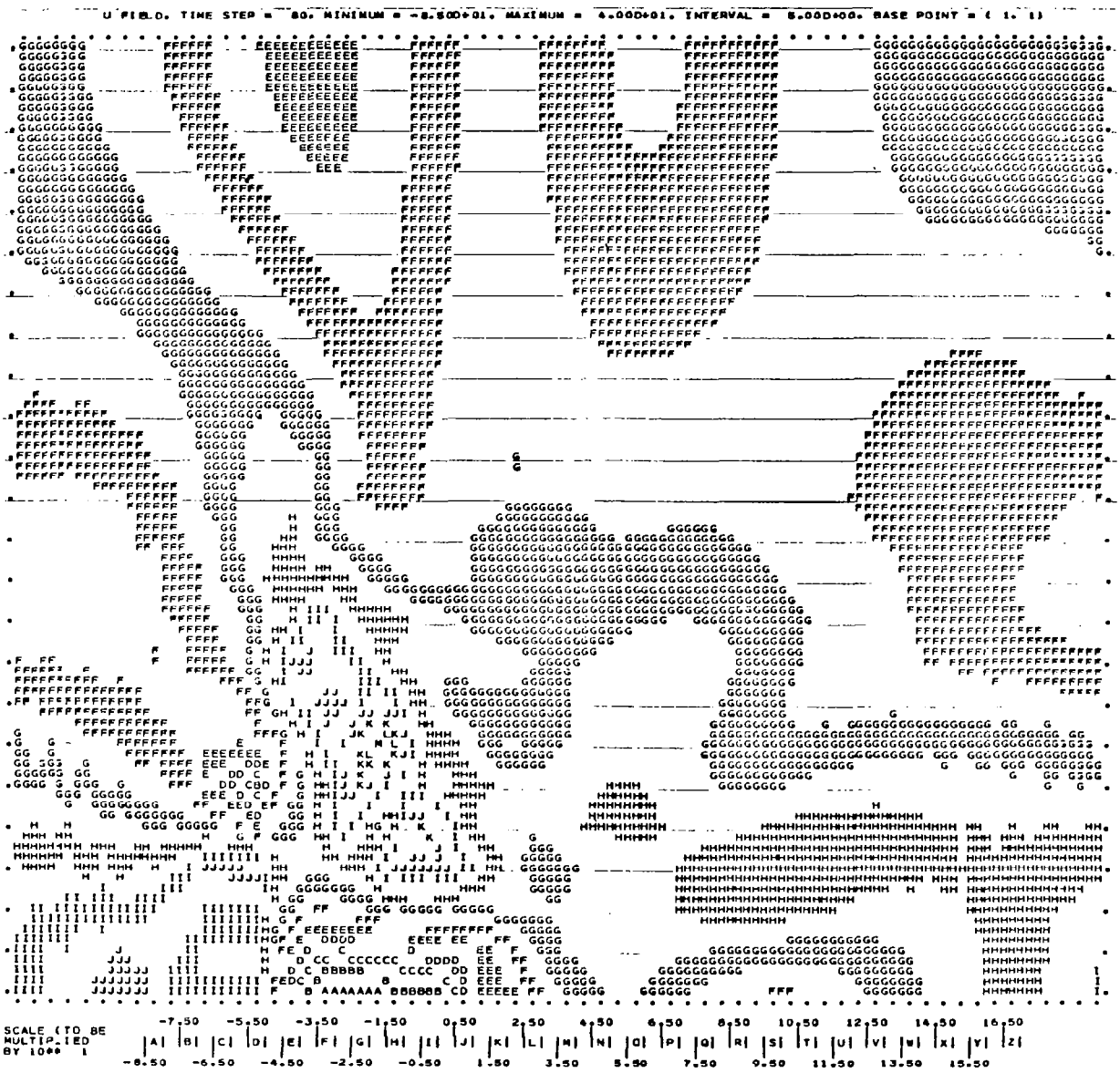
```
SUMMARY OF INPUT DATA FOR NON-ITERATIVE, SOLID BARRIER PROGRAM

NI = 55,  NJ = 24,  NT = 80,  NPB = 10,  IBT = 0,  ISAVE = 0,  DX = 6.25D+02,  DZ = 6.25D+02,  DT = 1.00D+01
COORDINATES OF BARRIER =  (12,2)  (12,3)  (12,4)  (12,5)  (13,5)  (14,5)  (15,5)  (15,4)  (15,3)  (15,2)
IS =  1,  JS =  1,  IN = 65,  JN =  24,  NJSMI =  3
NPSILV =  40,  NRHOLV =  40,  NZTALV =  25,  NULV =  25,  NWLV =  25,  NRILV =  9
IPSIGR =  60  80   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
IRHOGR =  80   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
IZTAGR =  80   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
IUGR   =  80   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
IWGR   =  80   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
IRIGR  =  80   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
CONSTANT VELOCITY CASE,  U1 =  2.50D+01
CONSTANT LAPSE RATE CASE,  GAMMA =  0.0  ,  T0 =  2.50D+02
```

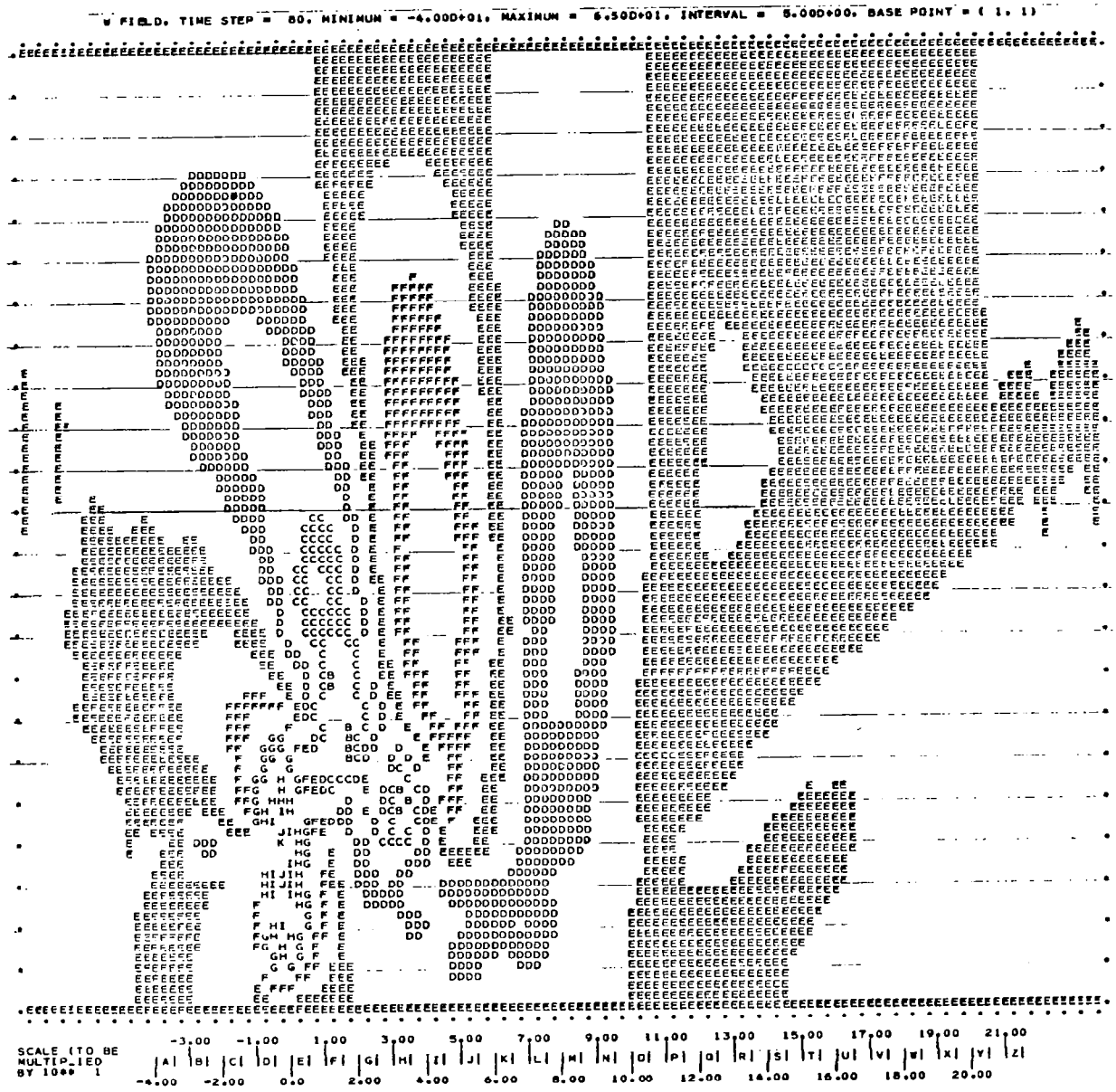

PSI FIELD. TIME STEP = 40. MINIMUM = -3.600+00. MAXIMUM = 1.000+04. INTERVAL = 1.000+04. BASE POINT = (1, 1)
SCALE (TO BE MULTIPLIED BY 10** 5)
-3.40 -3.00 -2.60 -2.20 -1.80 -1.40 -1.00 -0.60 -0.20 0.20 0.60 1.00 1.40
-3.60 -3.20 -2.80 -2.40 -2.00 -1.60 -1.20 -0.80 -0.40 0.00 0.40 0.80 1.20


PSI FIELD, TIME STEP = 60, MINIMUM = -3.60D+05, MAXIMUM = 1.00D+04, INTERVAL = 1.00D+04, BASE POINT = (1, 1)
SCALE (TO BE MULTIPLIED BY 1.0E 5
-3.40 -3.00 -2.60 -2.20 -1.80 -1.40 -1.00 -0.60 -0.20 0.20 0.60 1.00 1.40
-3.60 -3.20 -2.80 -2.40 -2.00 -1.60 -1.20 -0.60 -0.40 0.00 0.40 0.80 1.20


RHO FIELD, TIME STEP = 80, MINIMUM = 7.00D-01, MAXIMUM = 1.26D+00, INTERVAL = 2.00D-02, BASE POINT = (1, 1)
SCALE (TO BE MULTIPLIED BY 10** 0
0.74 0.82 0.90 0.98 1.06 1.14 1.22 1.30 1.38 1.46 1.54 1.62 1.70
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
0.70 0.78 0.86 0.94 1.02 1.10 1.18 1.26 1.34 1.42 1.50 1.58 1.66


ZETA FIELD, TIME STEP = 80, MINIMUM = -1.600D-01, MAXIMUM = 1.600D-01, INTERVAL = 2.000D-02, BASE POINT = (1, 1)
SCALE (TO BE MULTIPLIED BY 1000 -1
-1.20 -0.40 0.40 1.20 2.00 2.80 3.60 4.40 5.20 6.00 6.80 7.60 8.40
-1.60 -0.80 0.00 0.80 1.60 2.40 3.20 4.00 4.80 5.60 6.40 7.20 8.00


U FIELD. TIME STEP = 80. MINIMUM = -8.500+01. MAXIMUM = 4.000+01. INTERVAL = 5.00D+00. BASE POINT = (1. 1)
SCALE (TO BE MULTIPLIED BY 10** 1
-7.50 -5.50 -3.50 -1.50 0.50 2.50 4.50 6.50 8.50 10.50 12.50 14.50 16.50
-8.50 -6.50 -4.50 -2.50 -0.50 1.50 3.50 5.50 7.50 9.50 11.50 13.50 15.50


W FIELD. TIME STEP = 80. MINIMUM = -4.000D+01. MAXIMUM = 6.500D+01. INTERVAL = 5.000D+00. BASE POINT = (1, 1)
SCALE (TO BE MULTIPLIED BY 10** 1
-3.00 -1.00 1.00 3.00 5.00 7.00 9.00 11.00 13.00 15.00 17.00 19.00 21.00
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
-4.00 -2.00 0.0 2.00 4.00 6.00 8.00 10.00 12.00 14.00 16.00 18.00 20.00

LOG(RI) FIELD. TIME STEP = 80. MINIMUM = -8.000-01. MAXIMUM = 1.000+00. INTERVAL = 2.000-01. BASE POINT = (1, 1)
SCALE (TO BE MULTIPLIED BY 10** -1
-4.00 4.00 12.00 20.00 28.00 36.00 44.00 52.00 60.00 68.00 76.00 84.00 92.00
-8.00 0.0 8.00 16.00 24.00 32.00 40.00 48.00 56.00 64.00 72.00 80.00 88.00

REFERENCES

Arakawa, A., 1966: Computational design for long-term numerical integration of the equations of fluid motion: Two dimensional incompressible flow. Part I. J. Comput. Phys., 1, 119-143.

Arakawa, A., and V. R. Lamb, 1977: Computational design of the UCLA general circulation model. Methods in Computational Physics, 17, Academic Press, 173-265.

Brinkmann, W. A. R., 1974: Strong downslope winds at Boulder, Colorado, Mon. Wea. Rev., 102, 592-602.

Dahlquist, G., and A. Bjorck, 1974: Numerical Methods, Prentice-Hall, 131-134.

Danielsen, E. F., and R. Bleck, 1970: Tropospheric and stratospheric ducting of stationary mountain lee waves. J. Atmos. Sci., 27, 758-772.

Foldvik, A. and M. G. Wurtele, 1967: The computation of the transient gravity wave. Geophys. J. R. Astr. Soc., 13, 167-185.

Holmboe, J., and H. Klieforth, 1957: Investigation of mountain lee waves and the air flow over the Sierra Nevada. Final report, Contract No. AF 19(604)-728, Air Force Cambridge Research Center.

Jahnke, E., and F. Emde, 1945: Tables of Functions, Dover, p. 152.

Klemp, J. B., and D. K. Lilly, 1978: Numerical simulation of hydrostatic mountain waves. J. Atmos. Sci., 35, 78-107.

Lilly, D. K., and E. J. Zipser, 1972: The front range windstorm of 11 January 1972--a meteorological narrative. Weatherwise, 25, 56-63.

Long, R. R., 1955: Some aspects of the flow of stratified fluids. III: Continuous density gradients. Tellus, 7, 341-357.

Lyra, G., 1943: Theorie der stationären Leewellenströmung in freier Atmosphäre. Z. für Angew. Math. Mech., 23, 1-28.

Mesinger, F., and A. Arakawa, 1976: Numerical methods used in atmospheric models. WMO GARP Publ. No. 17, 9-16.

Miles, J. W., 1969: Waves and wave drag in stratified flows. Proc. of Twelfth Inst. Congress of Appl. Mechanics, Springer-Verlag.

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

Palm, E., and A. Foldvik, 1960: Contribution to the theory of two-dimensional mountain waves. Geofysiske Publikasjoner, XXI, No. 6.

Pao, Y.-H., and A. Goldburg, eds., 1969: Clear Air Turbulence and its Detection, Plenum Press, 542 pp.

Peltier, W. R., and T. L. Clark, 1979: Evolution and stability of finite-amplitude mountain waves. Part II: Surface wave drag and severe downslope windstorms. J. Atmos. Sci., 36, 1498-1529.

Roache, P. J., 1972: Computational Fluid Dynamics, Hermosa Publ., 133-134.

Vergeiner, I., 1971: An operational linear model for arbitrary basic flow and two-dimensional topography. Quart. J. Roy. Meteorol. Soc., 97, 30-60.

Wurtele, M. G., 1953: Studies of lee waves in atmospheric models with continuously distributed static stability. Scientific Rep. No. 4, Contract No. AF 19(122)-263, Air Force Cambridge Research Center,

LEGEND FOR FIGURES

The figures show the line printer output at fixed times for selected fields in the (x,z) plane, that is, in a cross section normal to the axis of the disturbing obstacle. The top line contains the symbolic name of the scalar field, the time step number, the range of the values plotted, the interval from one band to the next, and the grid point (i,j) values for the lower left hand corner of the output field. Value intervals are represented by alternating bands of letters and clear space. The values of the equi-scalar contours constituted by the boundaries of these bands may be read from the scale at the bottom of the diagram. The scale is subject to change for each diagram, since it is automatically adjusted to the range of output values.

The grid points at which the scalar field is calculated are delineated by dots around the edge of the diagram. Values between the grid points are determined by an interpolation routine. The barrier is indicated by a solid line at the bottom of each diagram, except for figures 13 and 14. For this case only, the lower boundary was specified by a nonrigid flow condition located between 11 and 20 grid points from the left side.

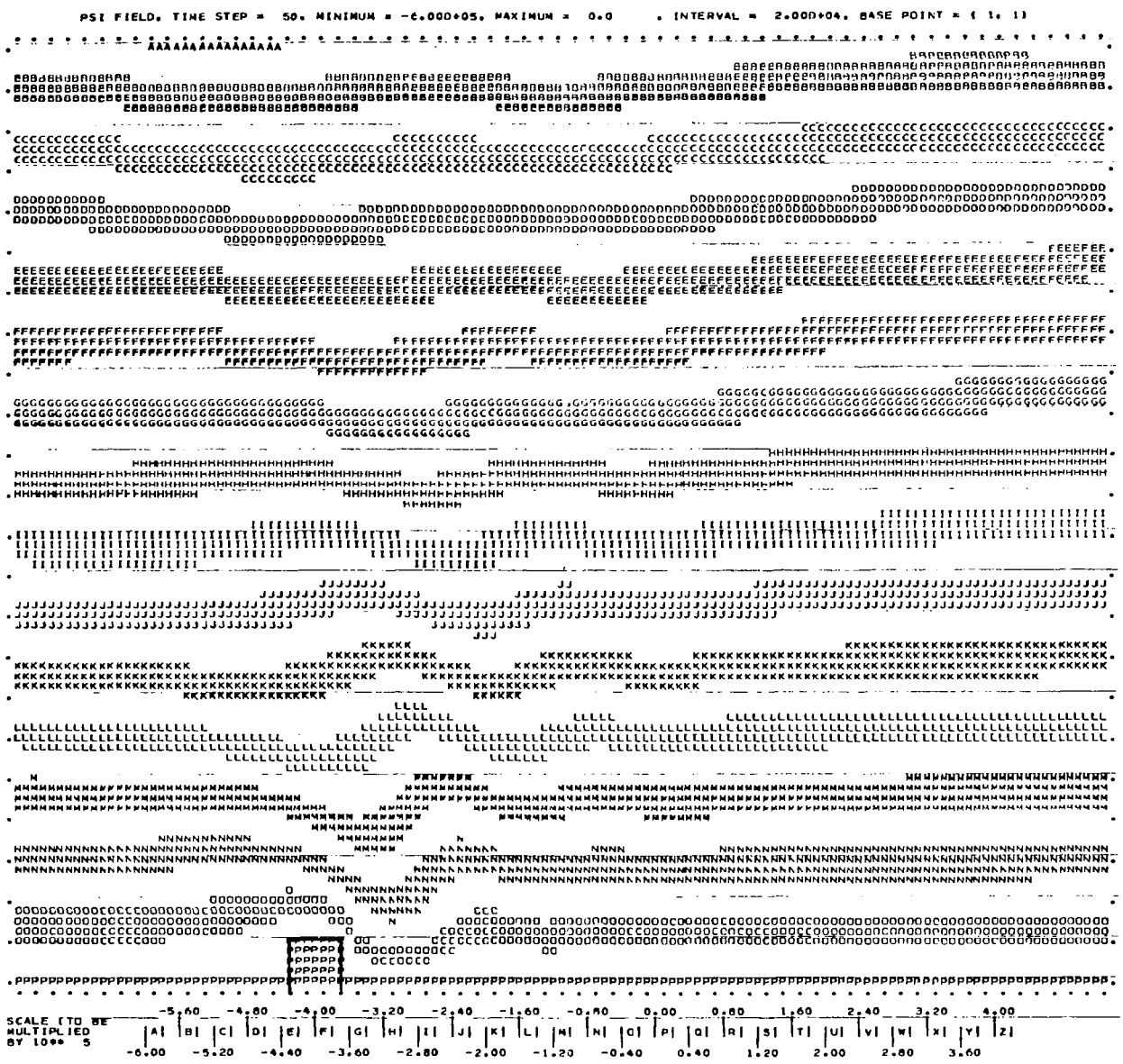


Figure 1. Streamfunction field at 1000 seconds for case 1 (constant velocity, constant stability). Δx = 1000 m, Δz = 1000 m, horizontal extent displayed = 64 km, vertical extent displayed = 23 km.

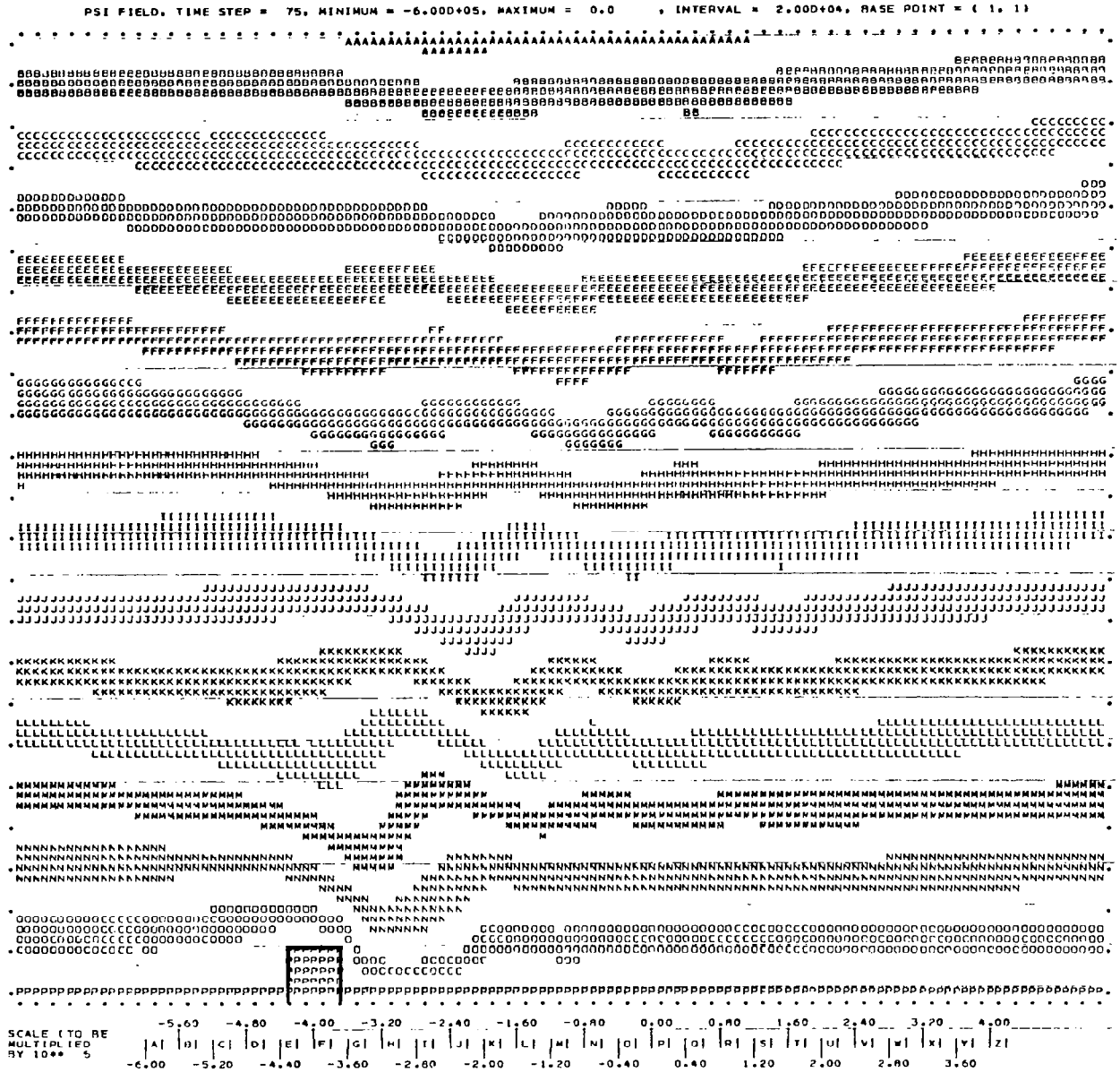


Figure 2. Streamfunction field at 1500 seconds for case 1. Dimensions as in figure 1.

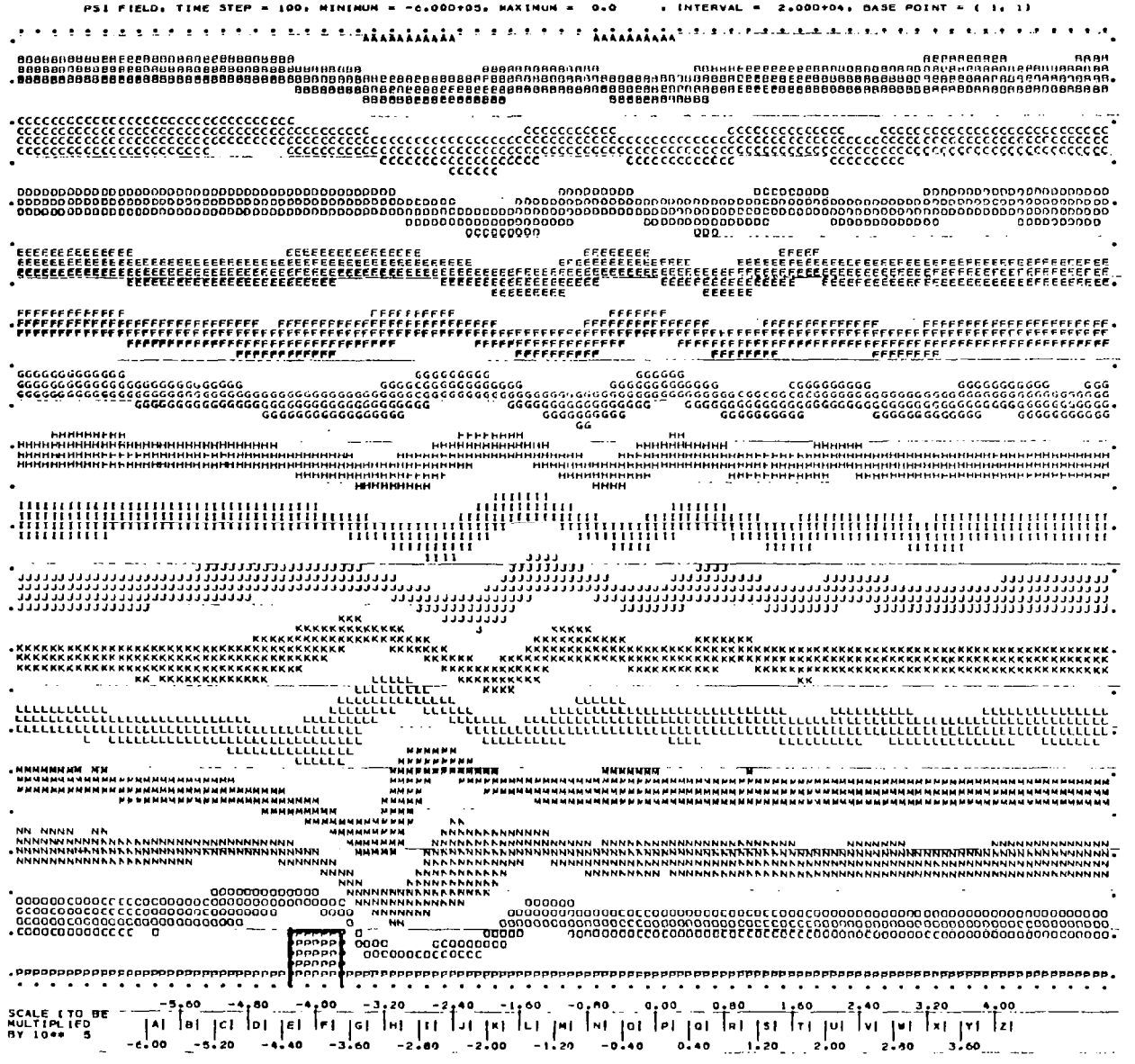


Figure 3. Streamfunction field at 2000 seconds for case 1, Dimensions as in figure 1.

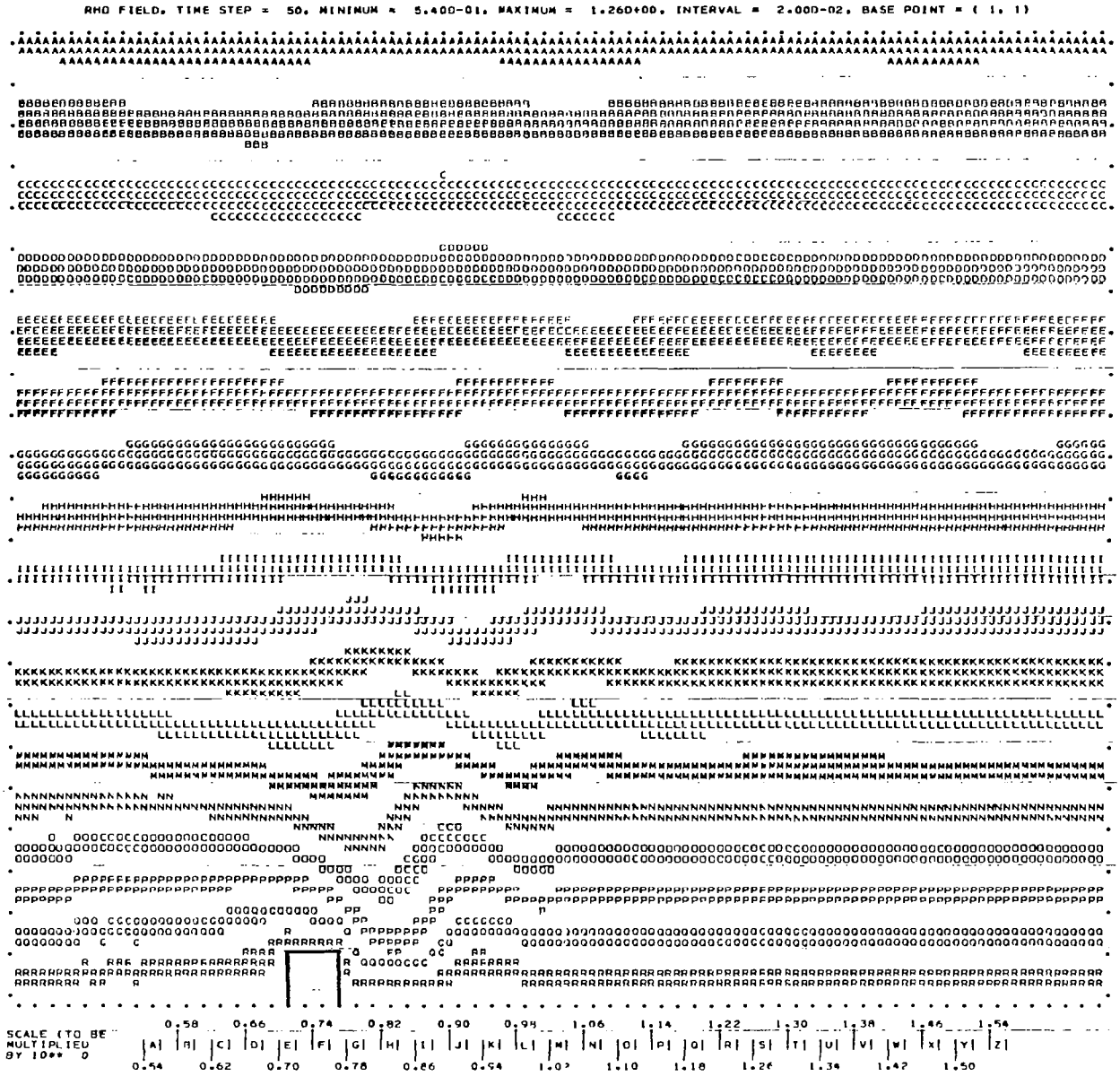


Figure 4. Density field at 1000 seconds for case 1, Dimensions as in figure 1.

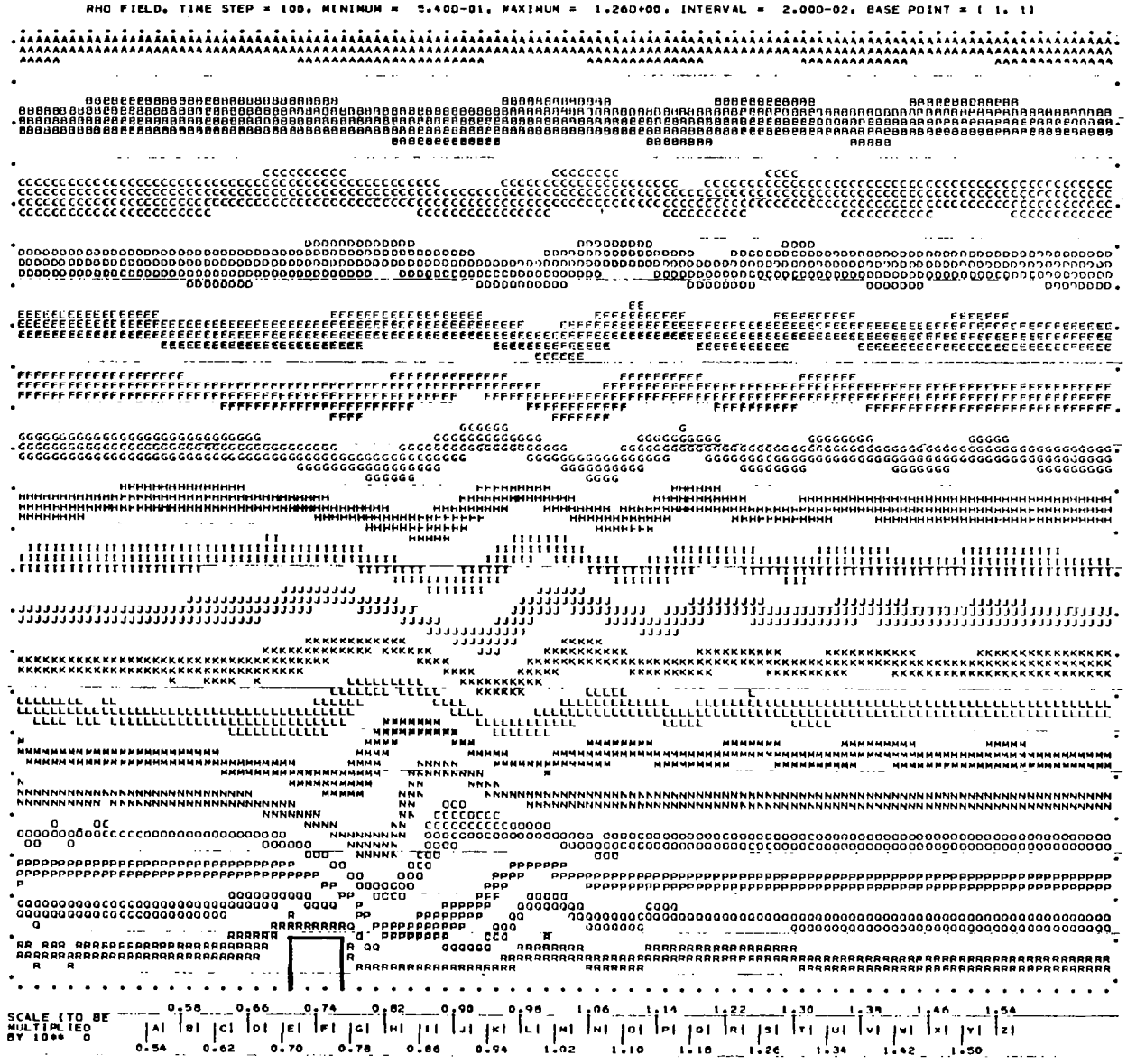


Figure 5. Density field at 2000 seconds for case 1. Dimensions as in figure 1.

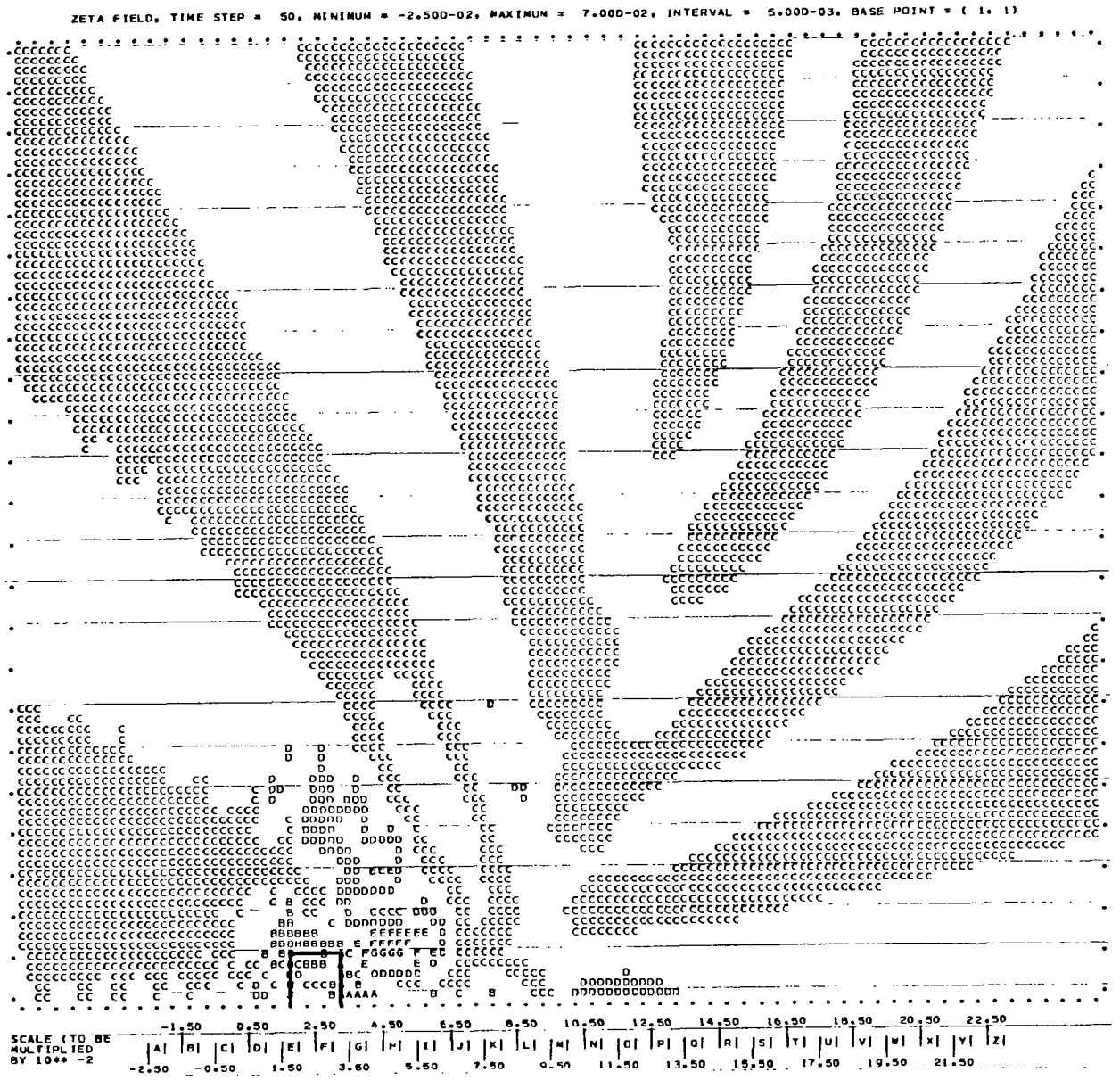


Figure 6. Vorticity field at 1000 seconds for case 1. Dimensions as in figure 1.

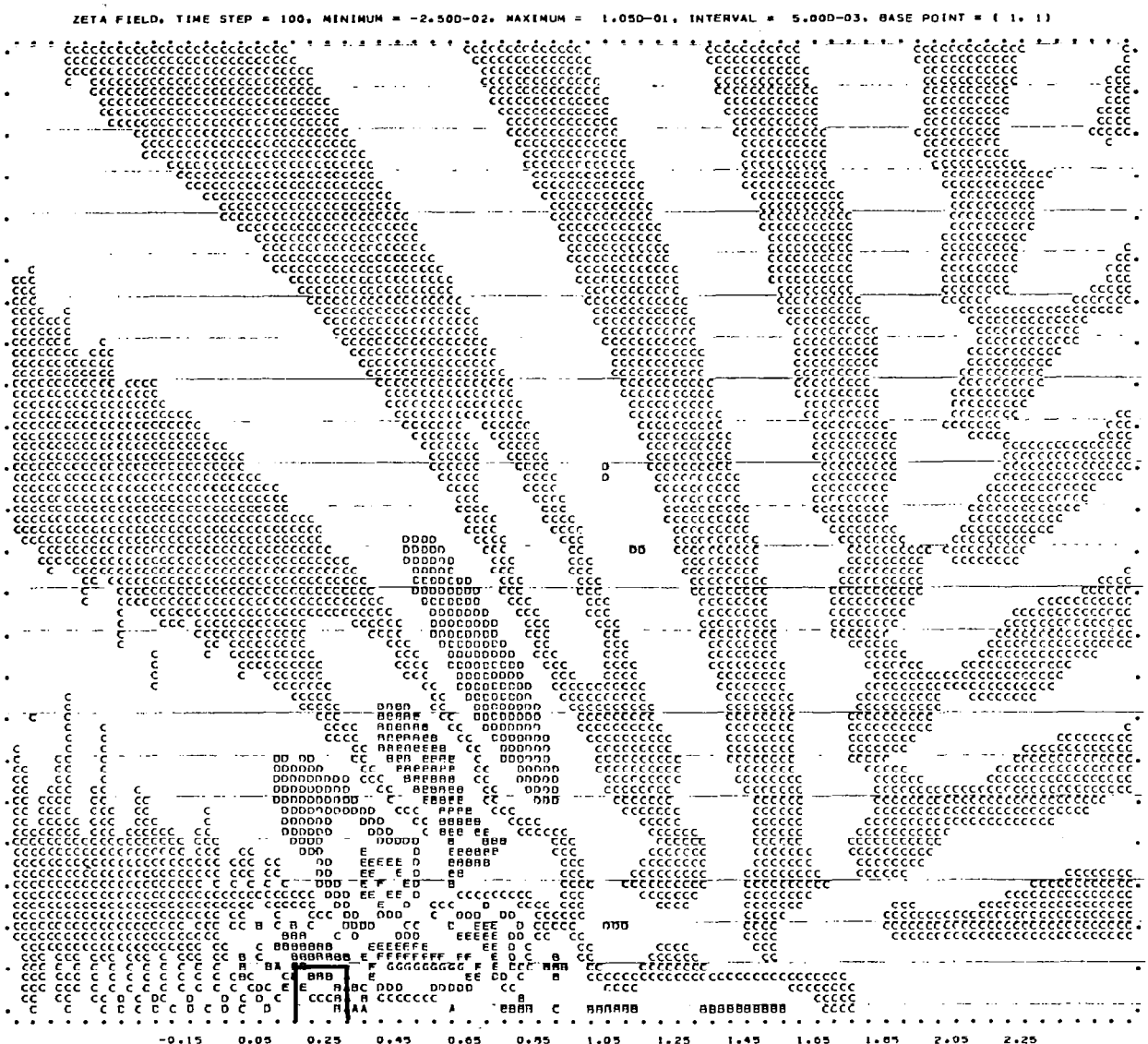


Figure 7. Vorticity field at 2000 seconds for case 1. Dimensions as in figure 1.

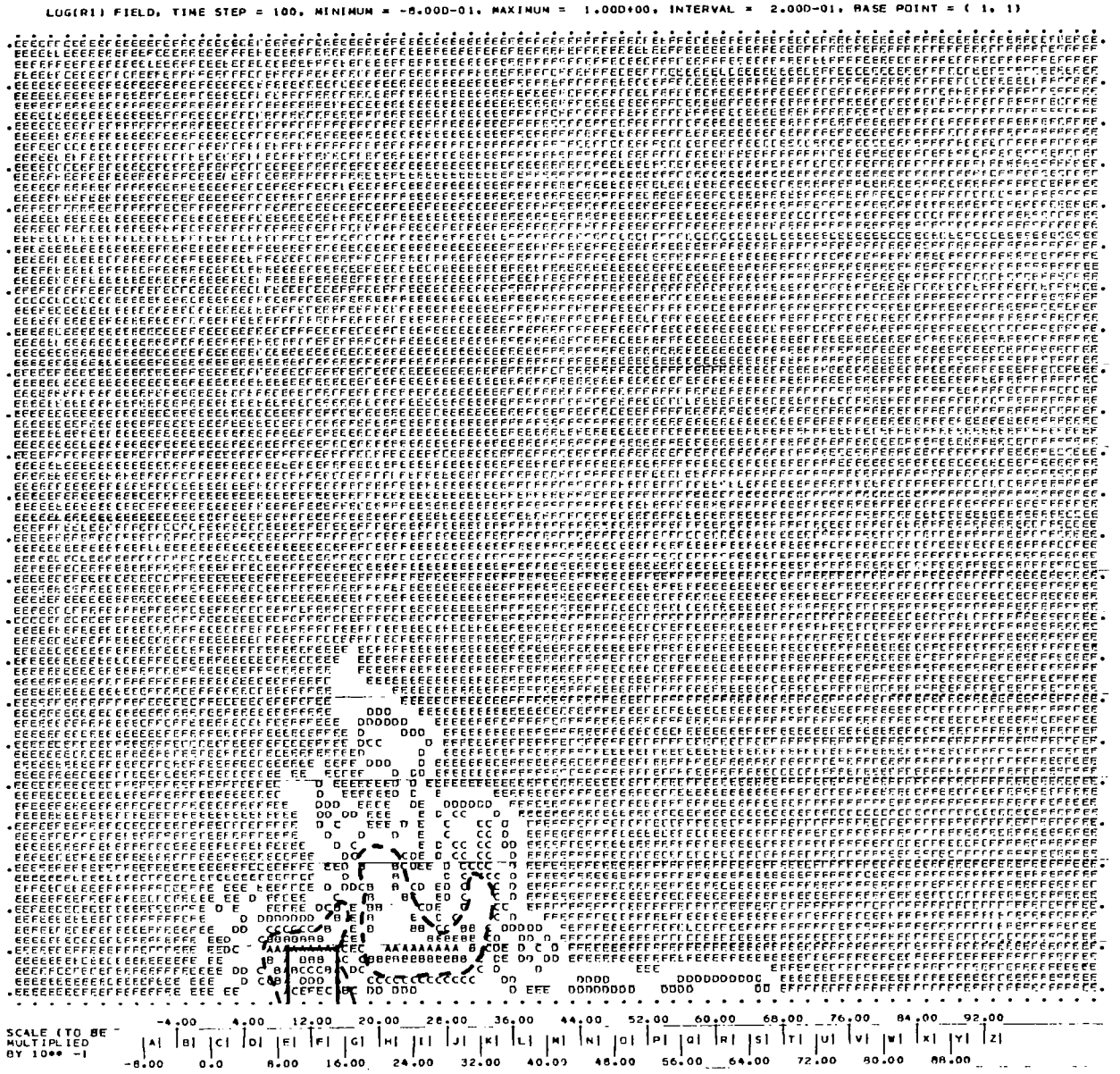


Figure 8. Richardson number field at 2000 seconds for case 1. Dimensions as in figure 1. Areas in which Ri < 1 are indicated by a dashed line. Areas in which Ri < 1/4 are indicated by the symbol "A".

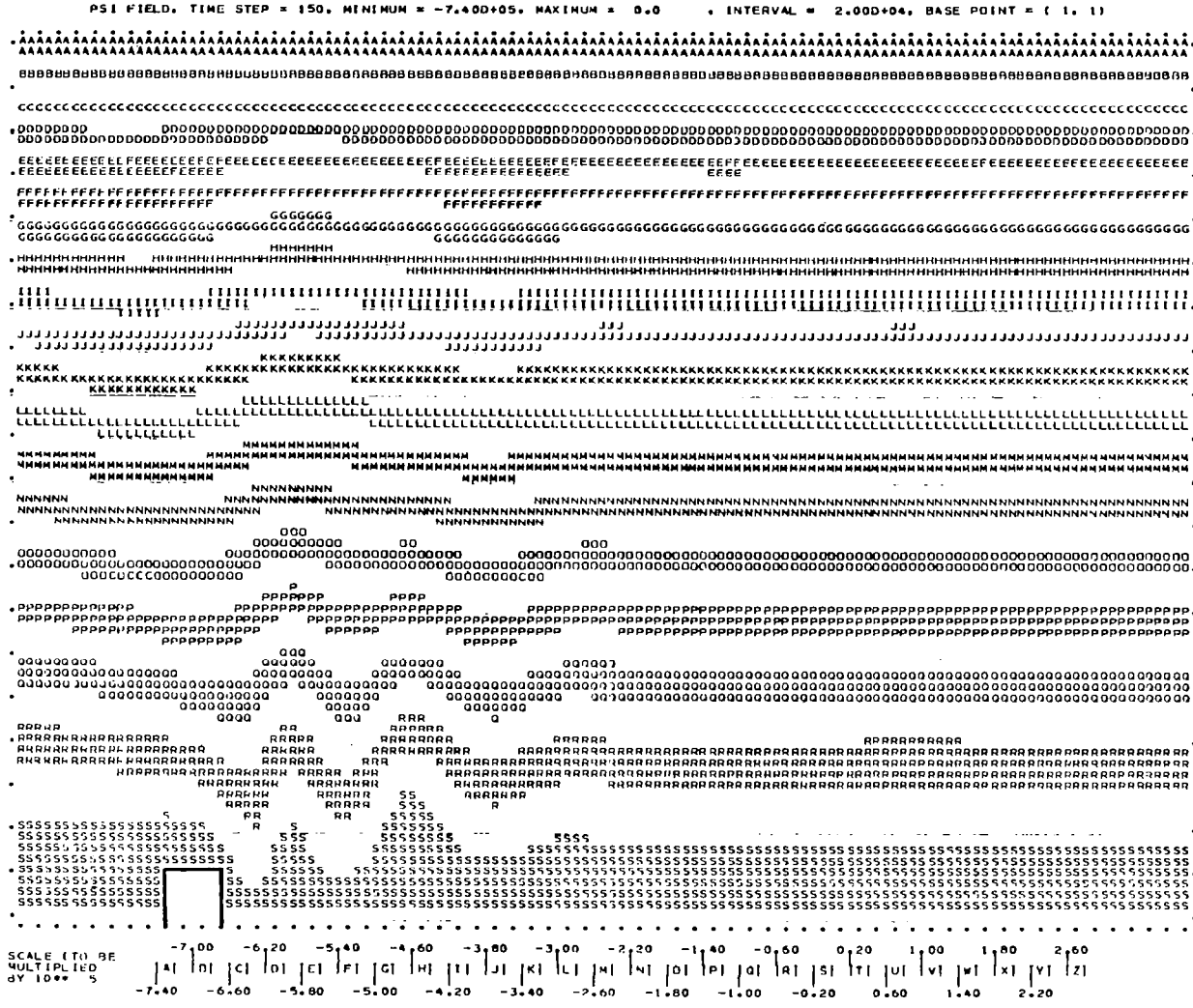


Figure 9. Streamfunction field at 3750 seconds for case 2 (linear shear, constant stability). Δx = 2000 m, Δz = 1000 m, horizontal extent displayed = 128 km, vertical extent displayed = 20 km.

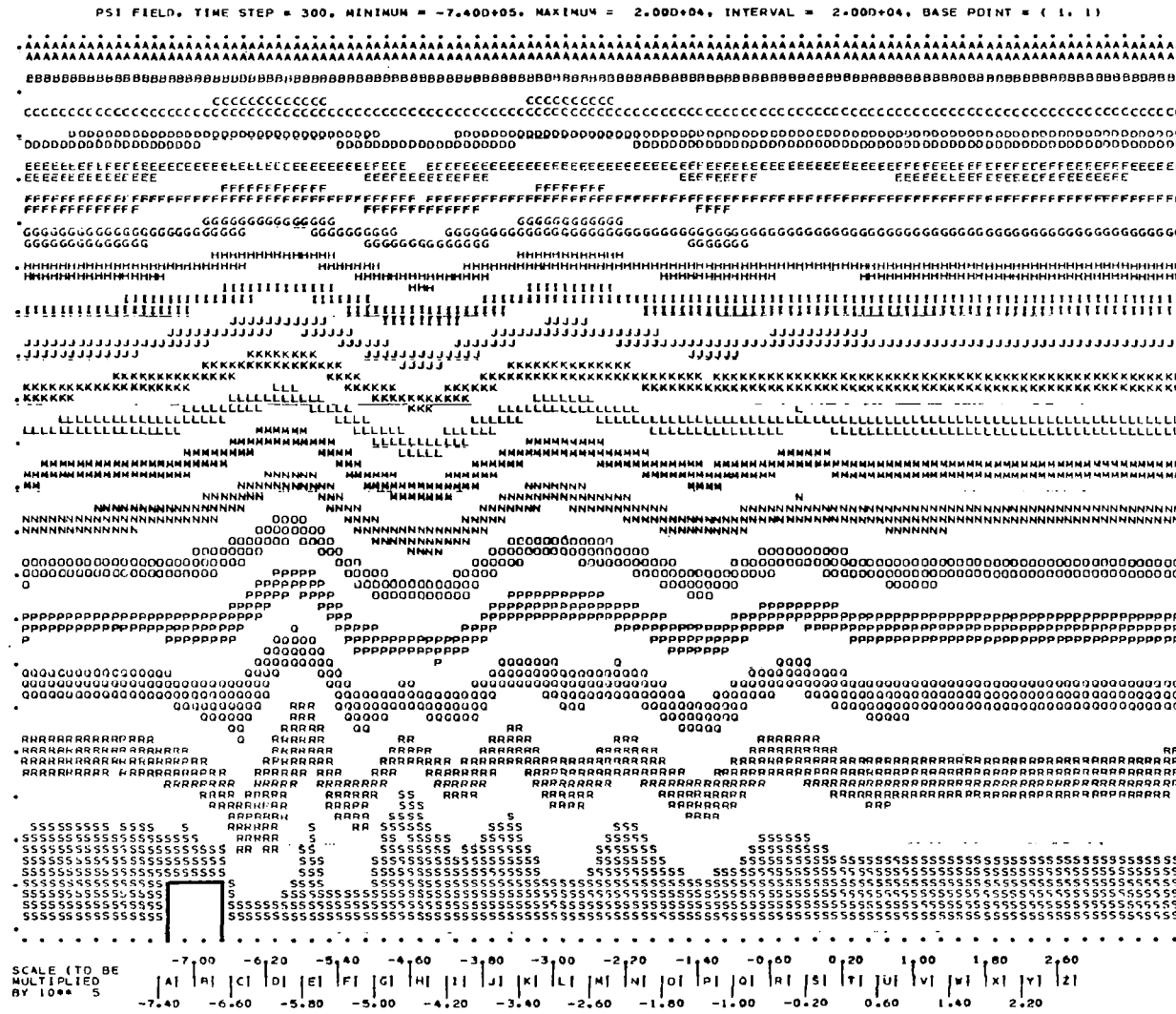


Figure 10. Streamfunction field at 7500 seconds for case 2. Dimensions as in figure 9.

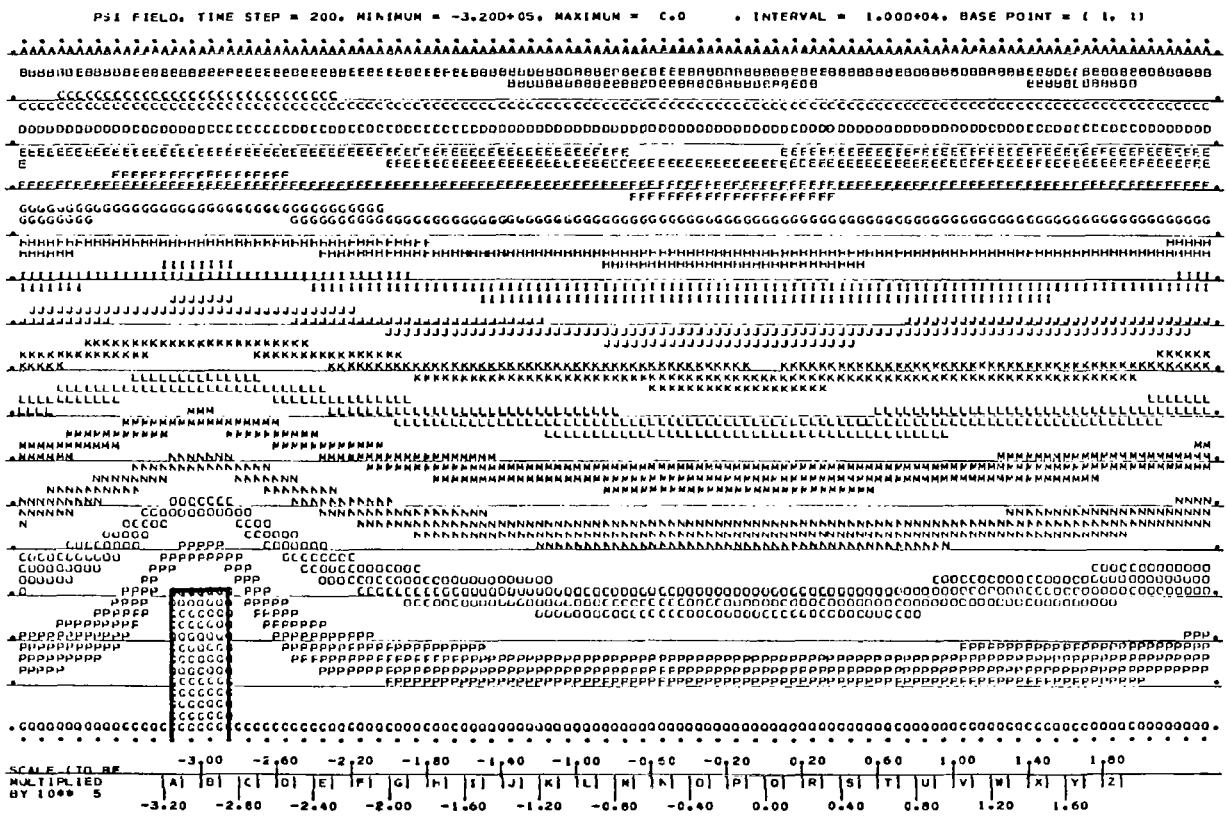


Figure 11. Streamfunction field at 1200 seconds for case 3 (exponential shear, constant stability, $Ri_0^{1/2}$=3,3). Δx = 500 m, Δz = 500 m, horizontal extent displayed = 32 km, vertical extent displayed = 7.5 km.

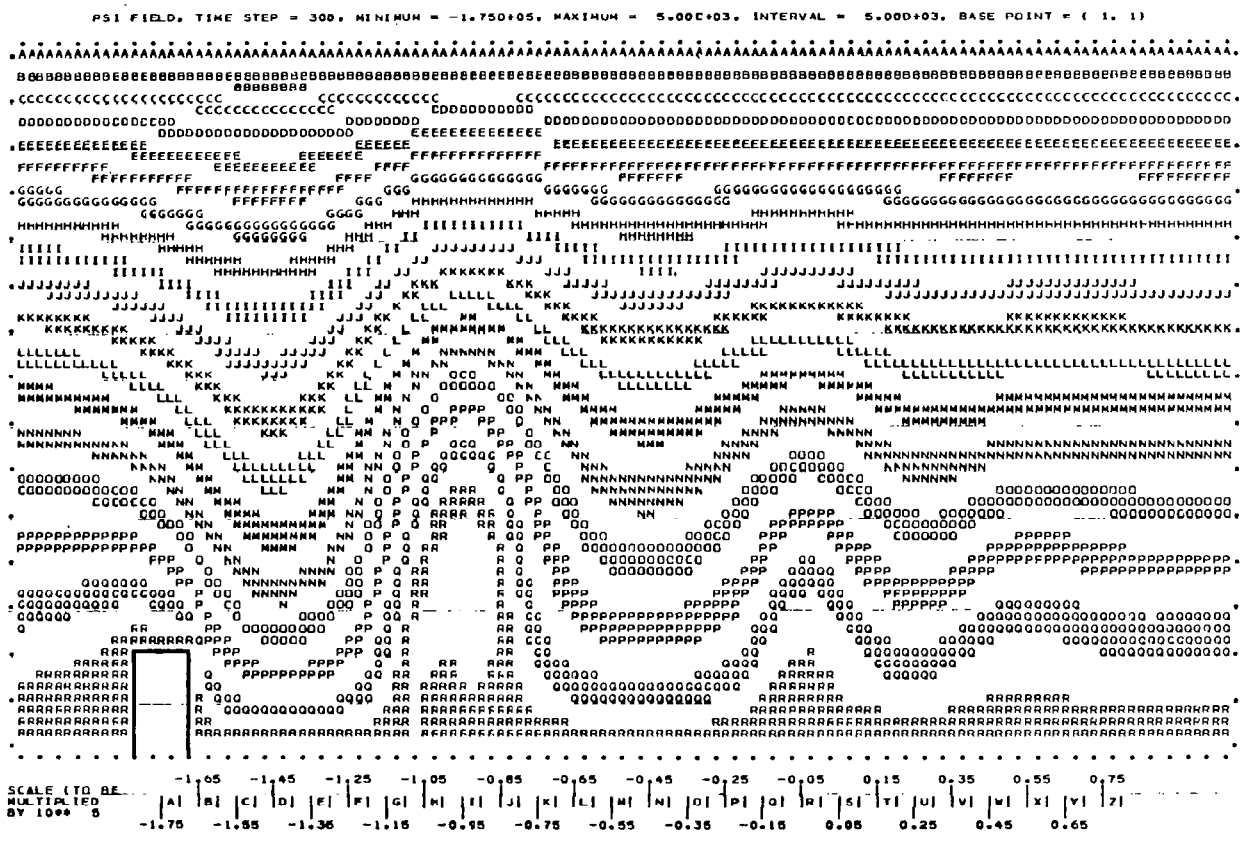


Figure 12. Streamfunction field at 4500 seconds for case 4 (exponential shear, constant stability, $Ri_0^{1/2}$=6.0). Δx = 750 m, Δz = 500 m, horizontal extent displayed = 48 km, vertical extent displayed = 7.5 km.

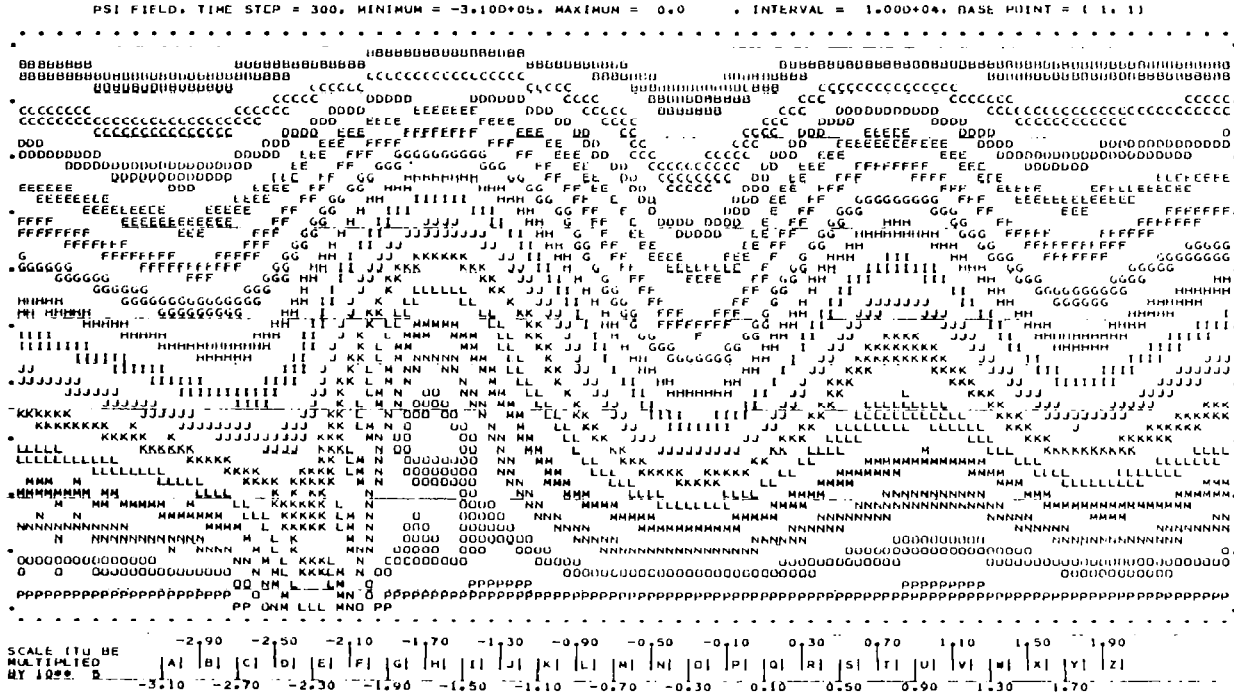


Figure 13. Streamfunction field at 3000 seconds for case 5 (nonlinear, constant ρu^2). Δx = 1000 m, Δz = 1000 m, horizontal extent displayed = 64 km, vertical extent displayed = 10 km.

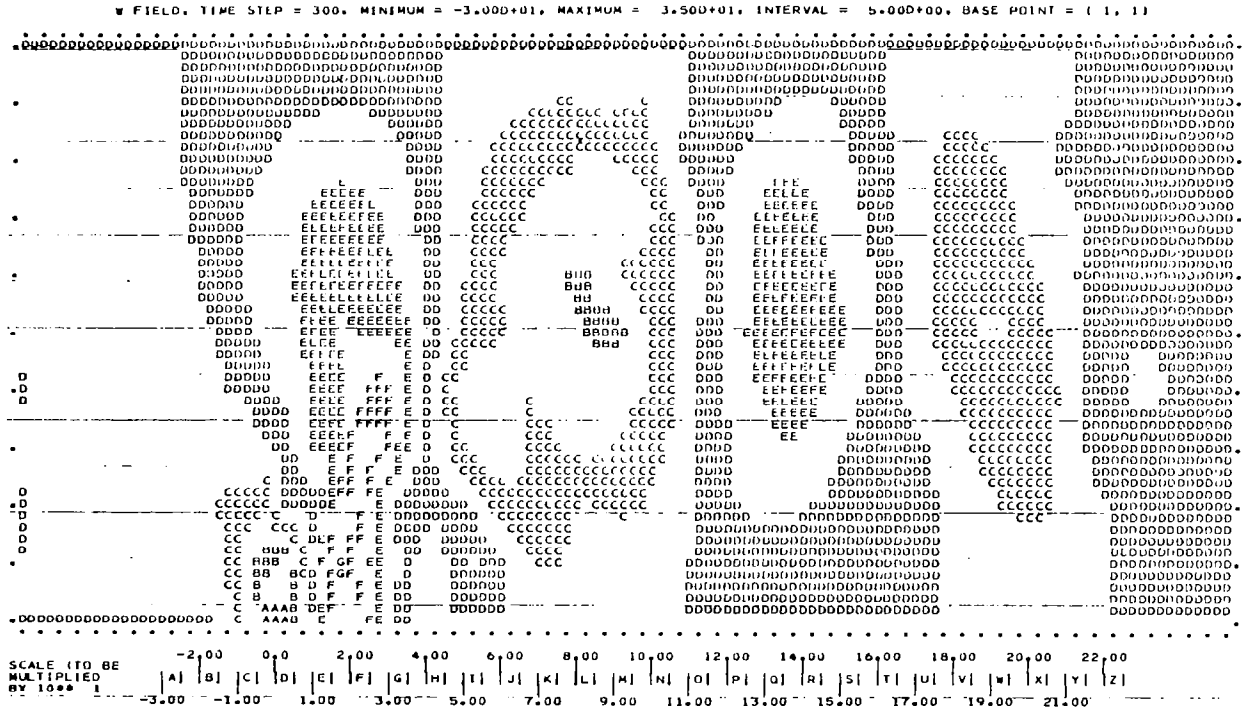


Figure 14. Vertical velocity field at 3000 seconds for case 5, Dimensions as in figure 13.

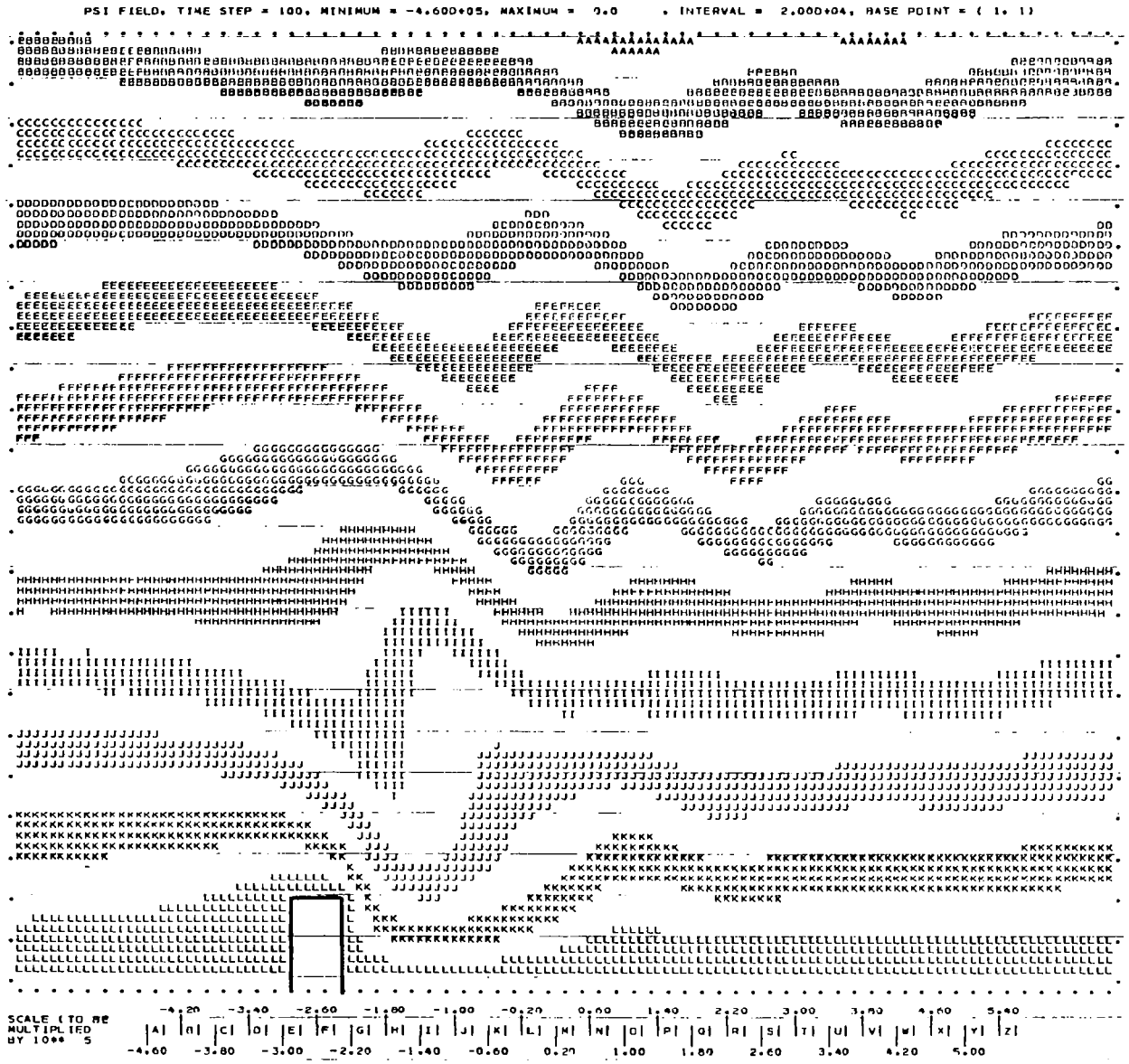


Figure 15. Streamfunction field at 1500 seconds for case 6 (constant velocity, constant stability, $k_s h$=1.17), Δx = 750 m, Δz = 750 m, horizontal extent displayed = 48 km, vertical extent displayed = 17.25 km.

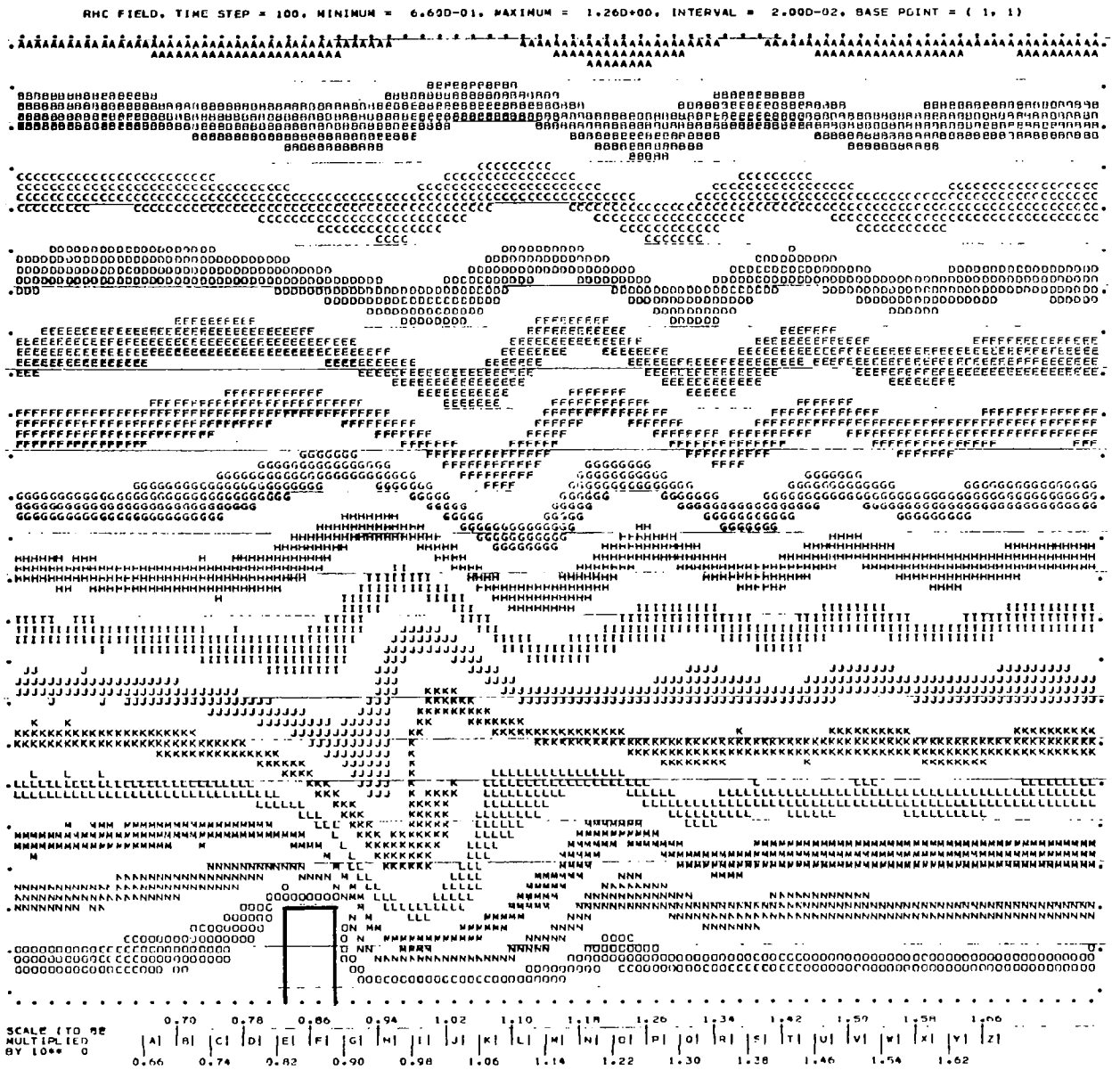


Figure 16. Density field at 1500 seconds for case 6. Dimensions as in figure 15.

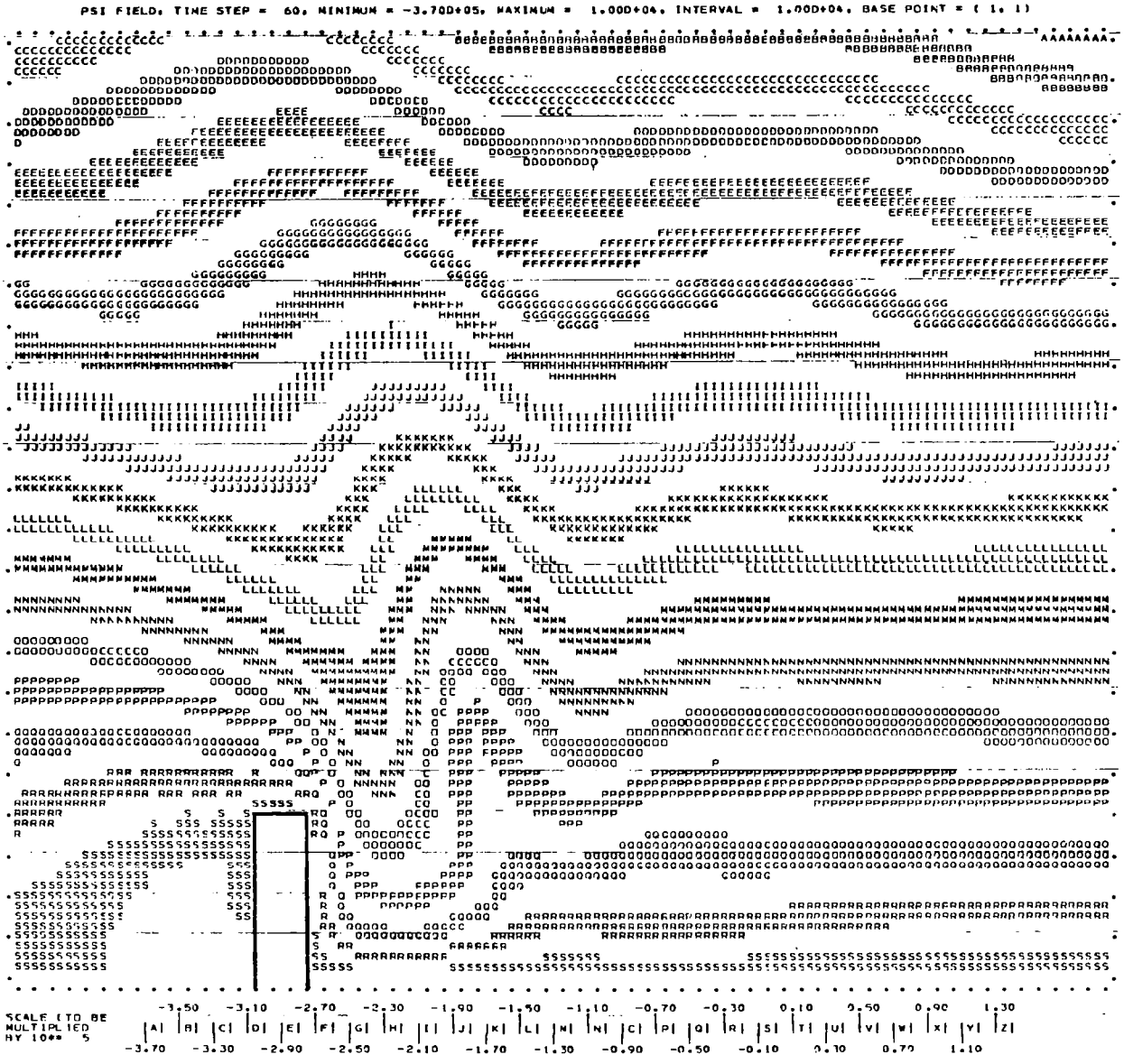


Figure 17. Streamfunction field at 600 seconds for case 7 (constant velocity, constant stability, large obstacle, $k_s h$=1.95). Δx = 625 m, Δz = 625 m, horizontal extent displayed = 40 km, vertical extent displayed = 14.38 km.

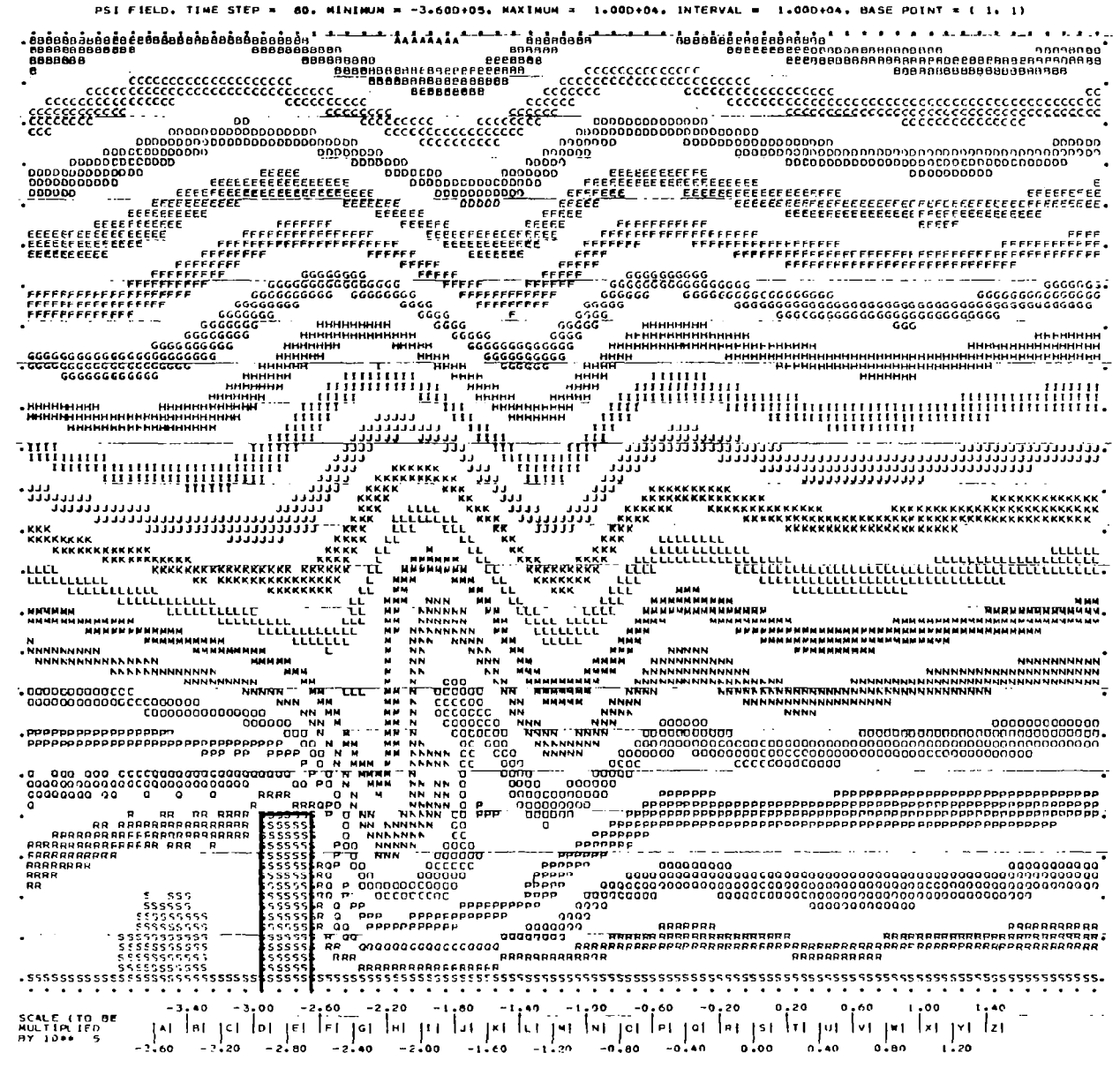


Figure 18. Streamfunction field at 800 seconds for case 7. Dimensions as in figure 17.

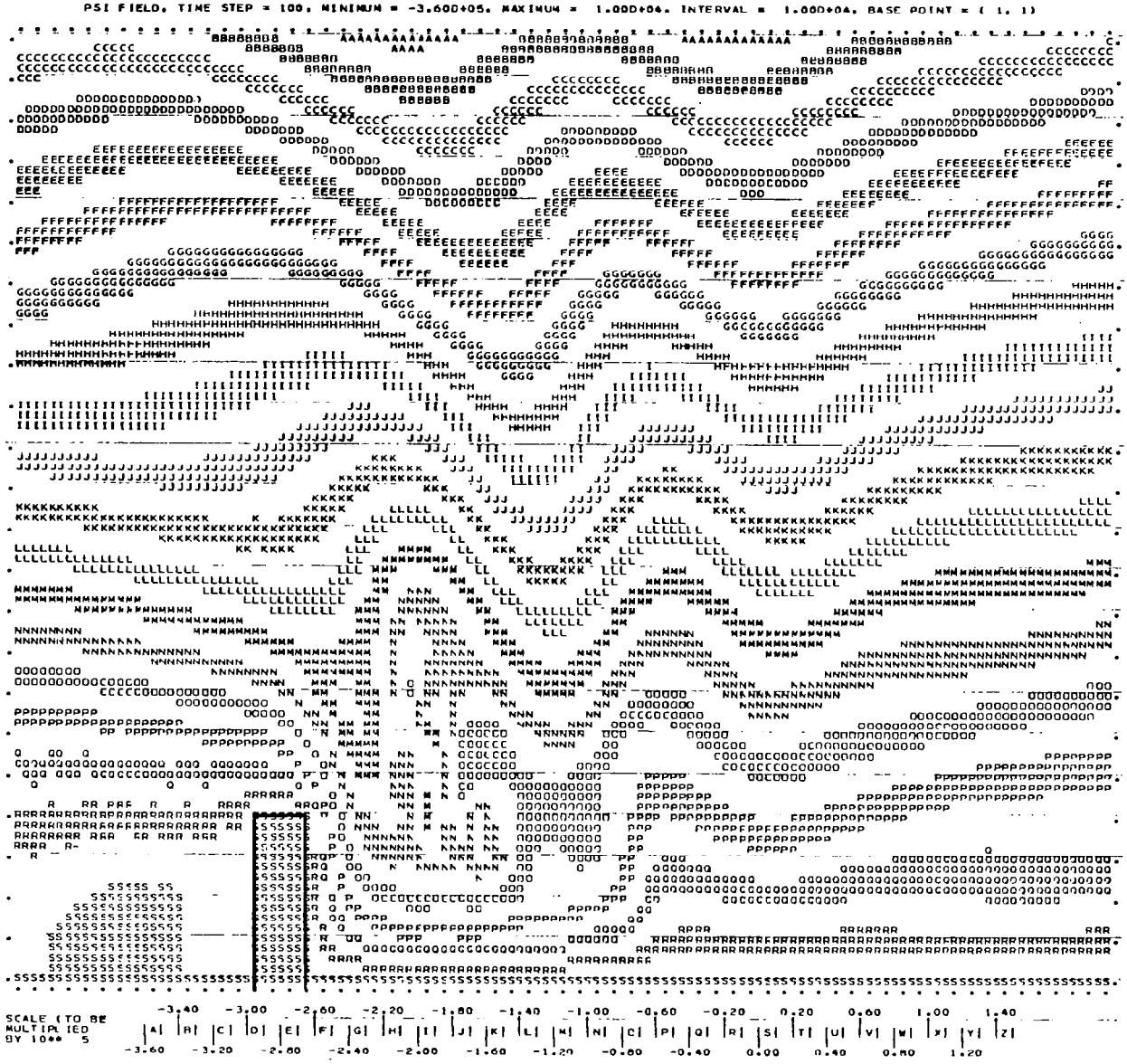


Figure 19. Streamfunction field at 1000 seconds for case 7. Dimensions as in figure 17.

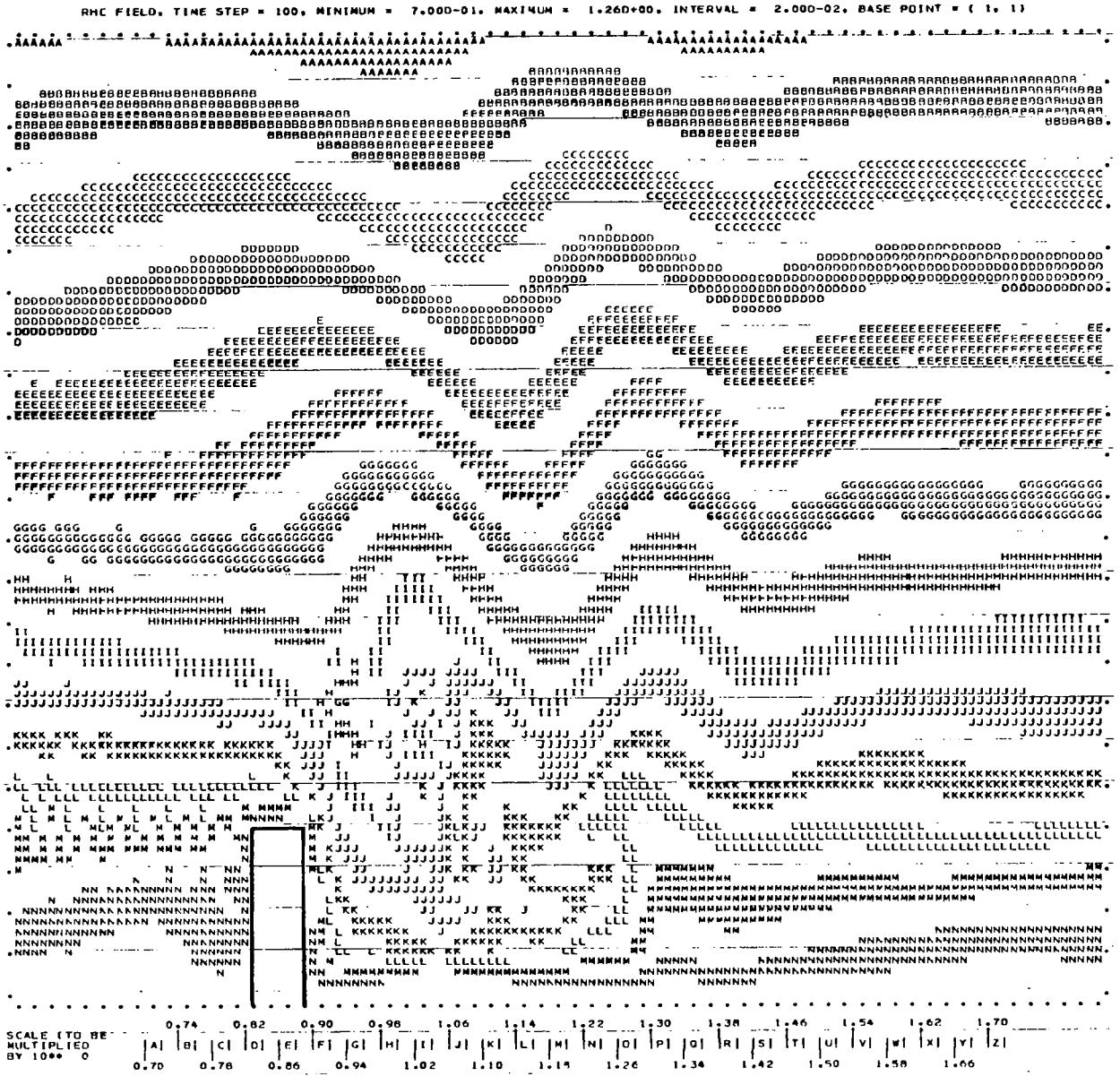


Figure 20. Density field at 1000 seconds for case 7, Dimensions as in figure 17.

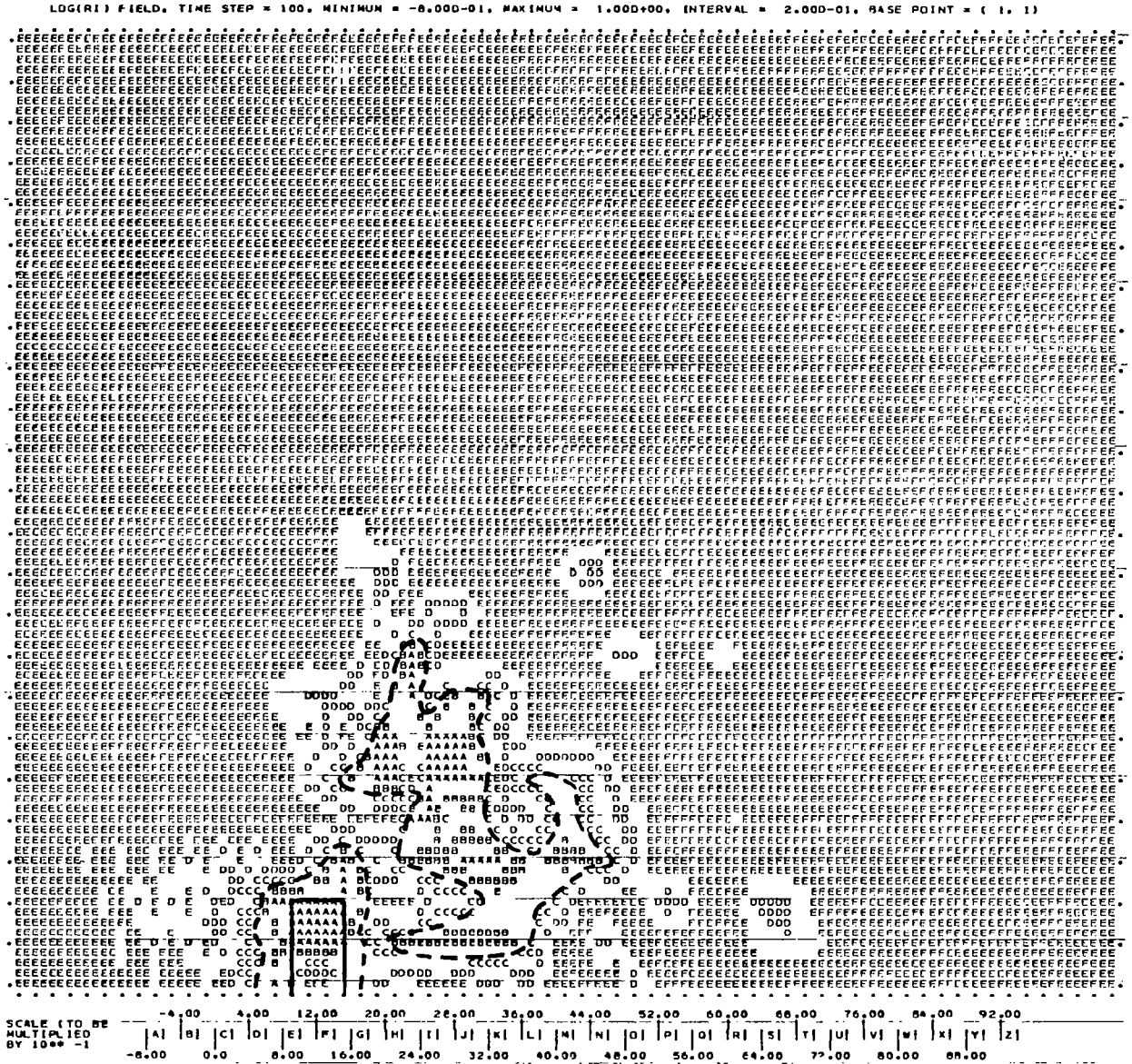


Figure 21. Richardson number field at 1500 seconds for case 6. Dimensions as in figure 15. Areas in which Ri < 1 are indicated by a dashed line. Areas in which Ri < 1/4 are indicated by the symbol "A".

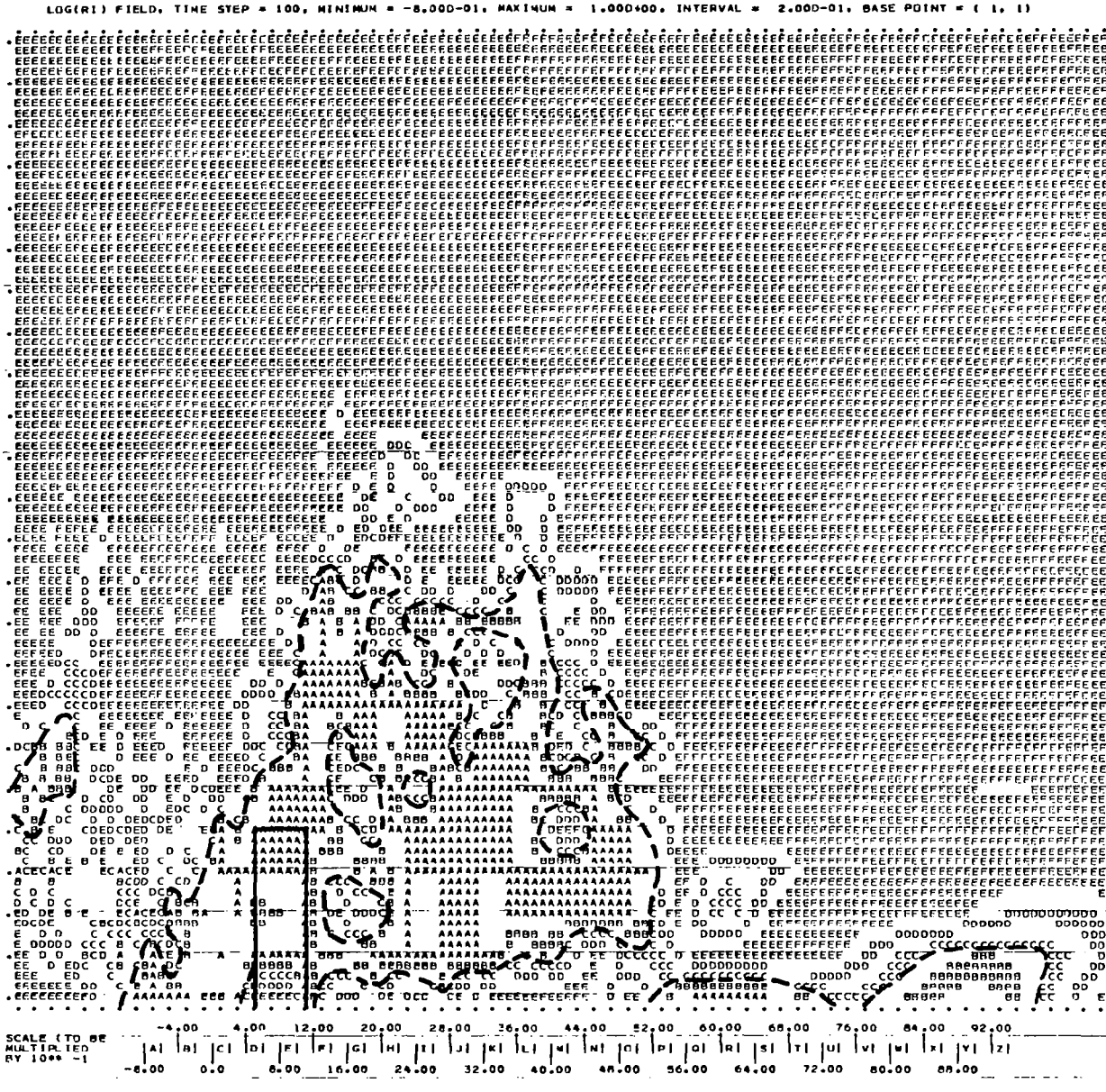


Figure 22. Richardson number field at 1000 seconds for case 7. Dimensions as in figure 17. Areas in which Ri < 1 are indicated by a dashed line. Areas in which Ri < 1/4 are indicated by the symbol "A".

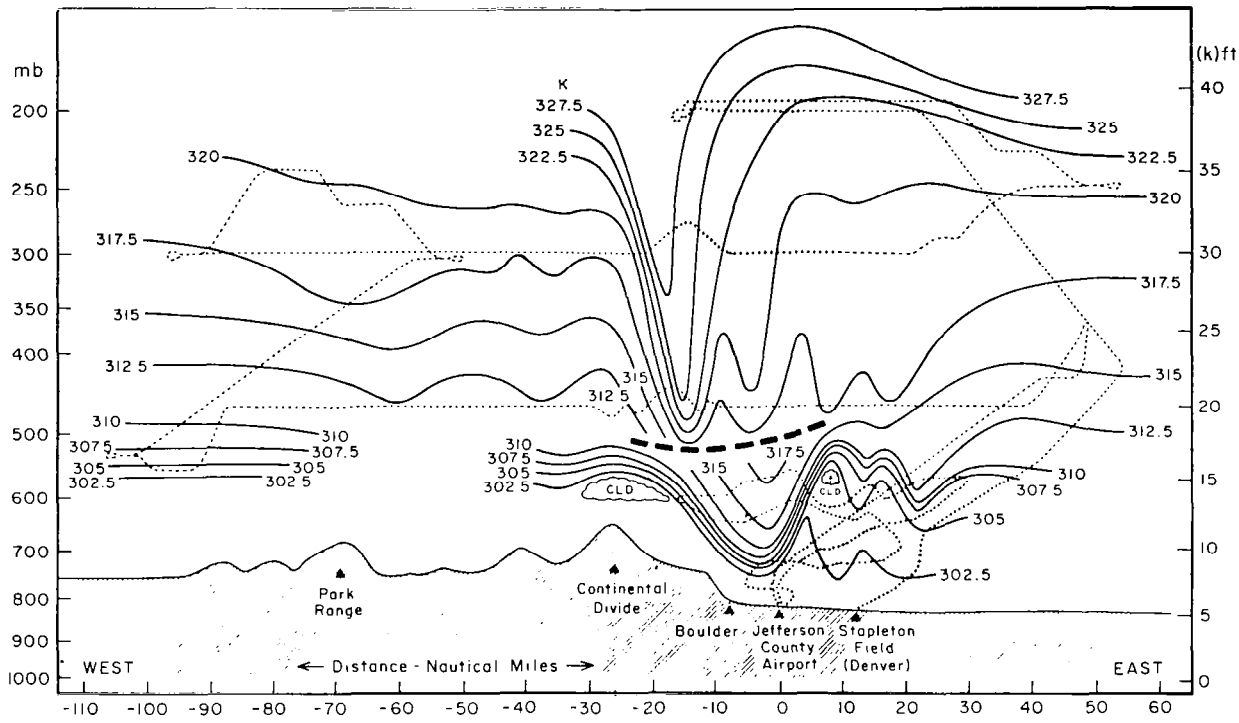


Figure 23. Cross section of the potential temperature (in K) along an east-west line through Boulder on 11 January 1972. Analysis above the heavy dashed line is from the Sabreliner data, taken between 1700 and 2000 MST, and analysis below this line is primarily from the Queen Air data, taken between 1330 and 1500 MST. Flight tracks are indicated by the light dashed lines (from Lilly and Zipser, 1972).

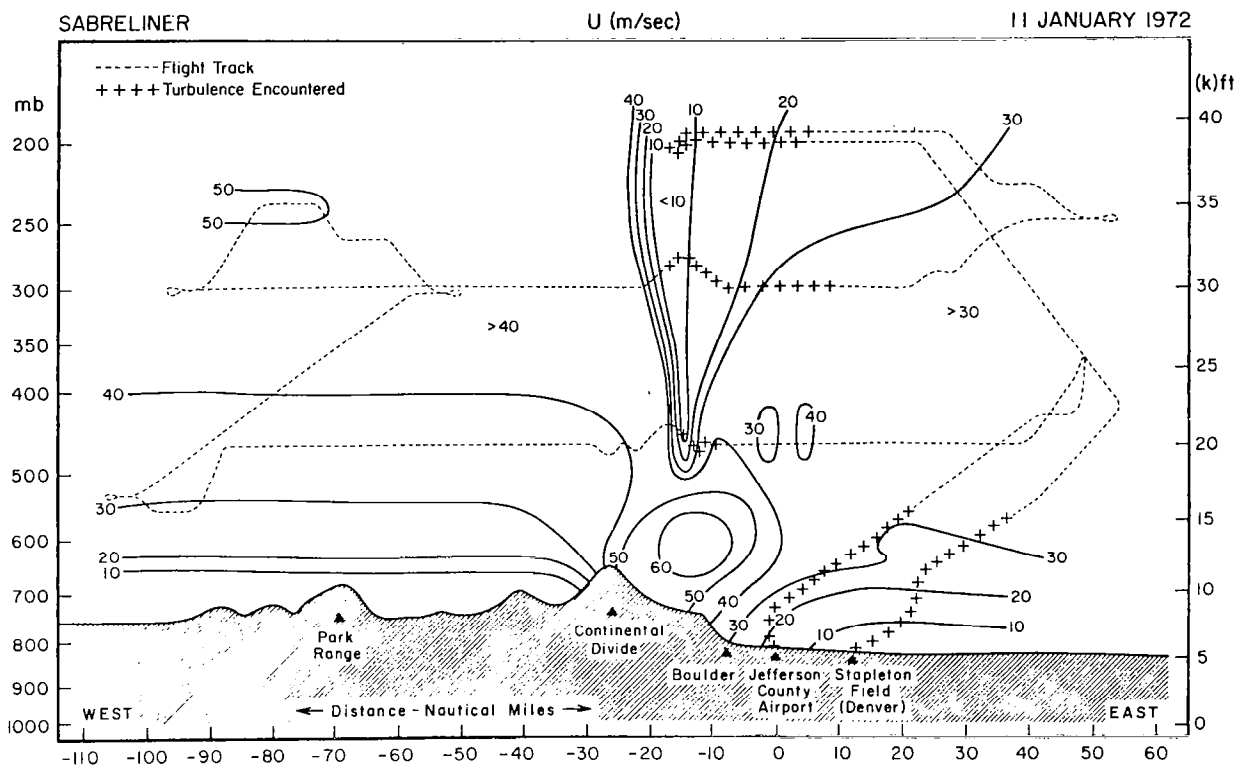


Figure 24. Cross section of horizontal wind velocity (in m/sec) along an east-west line through Boulder on 11 January 1972. This analysis was derived from Sabreliner data only. The analysis below 500 mb was partially obtained from vertical integration of the continuity equation, assuming two-dimensional, steady-state flow. Crosses indicate turbulent portions along the flight track (from Lilly and Zipser, 1972).

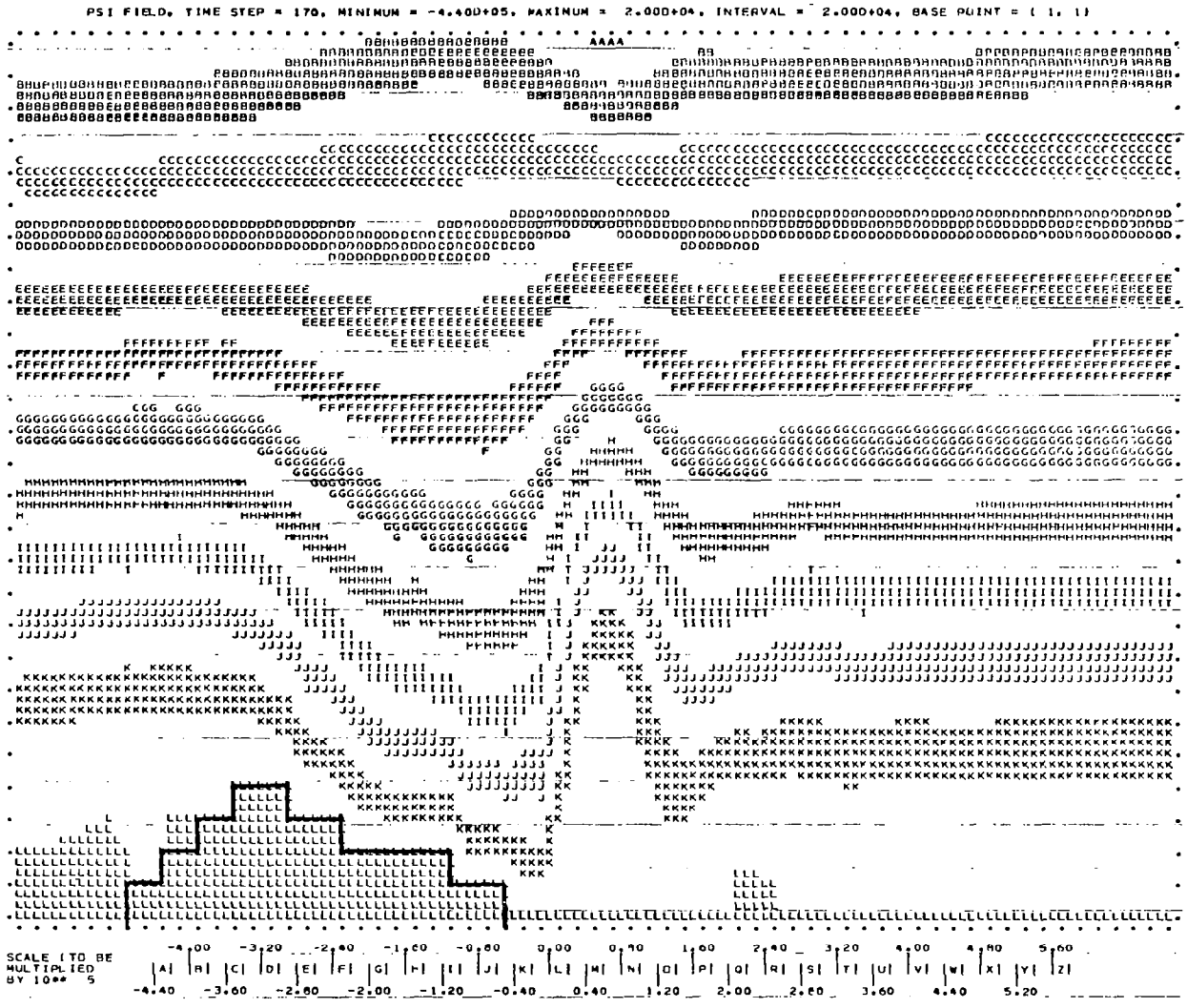


Figure 25. Streamfunction field at 4250 seconds for case 8 (Boulder windstorm). Δx = 2000 m, Δz = 500 m, horizontal extent displayed = 128 km, vertical extent displayed = 11,5 km, elevation at base of mountain = 1,5 km.

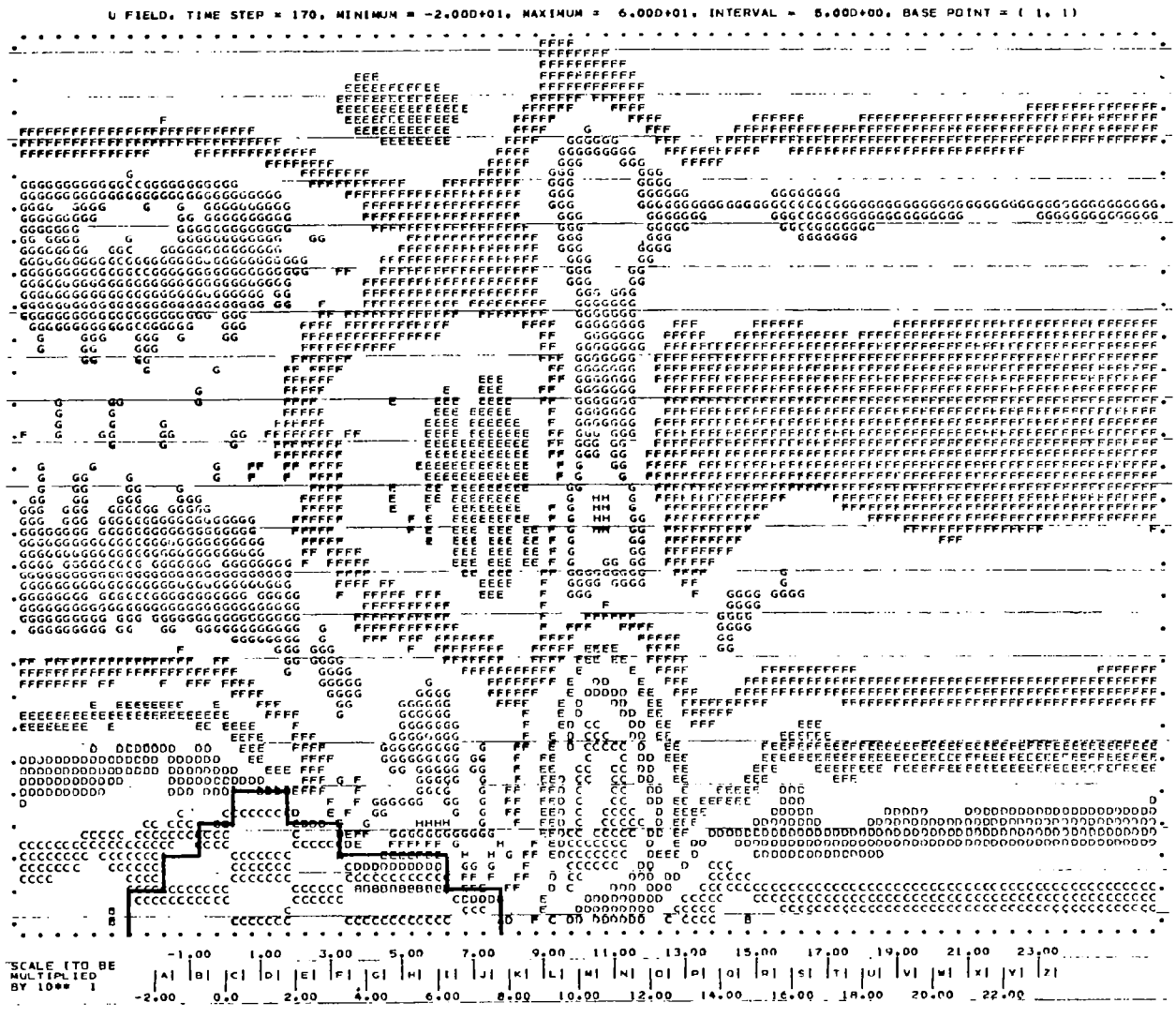


Figure 26. Horizontal velocity field at 4250 seconds for case 8. Dimensions as in figure 25.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-3385		

4. Title and Subtitle	5. Report Date
AN EFFICIENT CODE FOR THE SIMULATION OF NONHYDROSTATIC STRATIFIED FLOW OVER OBSTACLES	6. Performing Organization Code
7. Author(s) Gregory G. Pihos and Morton G. Wurtele	8. Performing Organization Report No.
9. Performing Organization Name and Address Department of Atmospheric Sciences University of California Los Angeles, California 90024	10. Work Unit No. 11. Contract or Grant No. NSG 4001 and NSG 4024
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546	13. Type of Report and Period Covered Contractor Report - Topical 14. Sponsoring Agency Code RTOP 505-44-14

15. Supplementary Notes

NASA Technical Monitor: L. J. Ehernberger, Dryden Flight Research Center

16. Abstract

A number of years have passed since the first nonlinear, non-hydrostatic mountain wave simulation; and a versatile and efficient code for case studies and possible operational use has now been developed. This report describes the physical model and computational procedure of the code in detail. The code is validated in tests against a variety of known analytical solutions from the literature and is also compared against actual mountain wave observations. The code will receive as initial input either mathematically idealized or discrete observational data. The form of the obstacle or mountain is arbitrary.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Mountain wave Gravity wave Numerical model Two-dimensional Nonlinear	Unclassified-Unlimited STAR category: 47

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	138	A07

For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1981