

THE NUMERICAL SOLUTION OF A SYSTEM OF DIFFUSION/CONVECTION  
EQUATIONS DESCRIBING COASTAL CIRCULATION

by

THOMAS O. NICOL

B.Sc., The University Of British Columbia, 1976

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF  
THE REQUIREMENTS FOR THE DEGREE OF  
MASTER OF SCIENCE

in

THE FACULTY OF GRADUATE STUDIES  
Institute Of Applied Mathematics And Statistics

We accept this thesis as conforming  
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

October 1983

© Thomas O. Nicol, 1983

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the Head of my Department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Institute Of Applied Mathematics And Statistics

The University of British Columbia  
2075 Wesbrook Place  
Vancouver, Canada  
V6T 1W5

Date: October 14, 1983

Abstract

A mathematical model describing nearshore ocean currents is examined. The motivation for the problem is discussed and a derivation of the model equations presented. The model equations consist of a pair of coupled, nonlinear, parabolic partial differential equations. A numerical method, the Crank-Nicolson finite difference scheme, is presented for solving the corresponding linear equations, and the convergence and stability of the method discussed. Methods for dealing with the nonlinear terms, and their effects on accuracy and stability, are examined. The initial and boundary conditions of the model equations present special problems, and we describe methods to solve them. A more efficient finite difference scheme for our equations, based on the Crank-Nicolson formula, is introduced, and its advantages discussed. A computer program incorporating these methods is developed to solve the model equations, and results for test data presented. We conclude with recommendations for additions to the program to model the currents more accurately.

## Table of Contents

Abstract .....	ii
List of Figures .....	iv
Acknowledgements .....	v
I. INTRODUCTION .....	1
II. CHAPTER 1 .....	4
2.1 Derivation Of The Model .....	4
2.2 The Domain Of The Model .....	8
2.3 Boundary Conditions .....	8
2.4 Initial Conditions .....	9
2.5 How The Model Will Be Used .....	10
III. CHAPTER 2 .....	13
3.1 Notation .....	13
3.2 Finite Differences .....	14
3.3 Consistency, Stability And Convergence .....	19
3.4 Variable Coefficients .....	24
3.5 Stability Of Boundary Conditions .....	25
3.6 Nonlinear Terms .....	27
3.7 Solving The Equations .....	29
IV. CHAPTER 3 .....	32
4.1 Introduction To ADI .....	33
4.2 Accuracy And Stability .....	35
4.3 Computational Details And Sample Problems .....	37
4.4 Applying ADI To The Model Equations .....	42
V. CHAPTER 4 .....	44
5.1 Preliminary Tests .....	44
5.2 Initial Conditions At The Land Boundary .....	46
5.3 Finding H On The Boundaries .....	49
5.4 Boundary Condition For S .....	52
5.5 Tests Of The Model Equations .....	53
5.6 Interpretation Of Results .....	54
5.7 Recommendations For Proceeding .....	57
BIBLIOGRAPHY .....	78

List of Figures

1. Domain of the Coastal Circulation Model .....	11
2. Domain of the Numerical Model .....	12
3. Test 1: S at time step 2 .....	60
4. Test 1: S at time step 5 .....	61
5. Test 1: S at time step 10 .....	61
6. Test 1: S at time step 15 .....	62
7. Test 1: S at time step 20 .....	62
8. Test 1: S at time step 30 .....	63
9. Test 1: S at time step 40 .....	63
10. Test 1: S at time step 50 .....	64
11. Test 1: H at time step 2 .....	64
12. Test 1: H at time step 5 .....	65
13. Test 1: H at time step 10 .....	65
14. Test 1: H at time step 15 .....	66
15. Test 1: H at time step 20 .....	66
16. Test 1: H at time step 30 .....	67
17. Test 1: H at time step 40 .....	67
18. Test 1: H at time step 50 .....	68
19. Test 1: H <sup>2</sup> S at time step 2 .....	68
20. Test 1: H <sup>2</sup> S at time step 5 .....	69
21. Test 1: H <sup>2</sup> S at time step 10 .....	69
22. Test 1: H <sup>2</sup> S at time step 15 .....	70
23. Test 1: H <sup>2</sup> S at time step 20 .....	70
24. Test 1: H <sup>2</sup> S at time step 30 .....	71
25. Test 1: H <sup>2</sup> S at time step 40 .....	71
26. Test 1: H <sup>2</sup> S at time step 50 .....	72
27. Test 2: S at time step 20 .....	72
28. Test 2: S at time step 30 .....	73
29. Test 2: S at time step 40 .....	73
30. Test 2: H at time step 20 .....	74
31. Test 2: H at time step 30 .....	75
32. Test 2: H at time step 40 .....	75
33. Test 2: H <sup>2</sup> S at time step 20 .....	76
34. Test 2: H <sup>2</sup> S at time step 30 .....	77
35. Test 2: H <sup>2</sup> S at time step 40 .....	77

### Acknowledgement

The completion of this thesis has depended on the good advice, given over many hours, from Jim Varah, the multitude of suggestions from Paul Leblond, and the original idea of Bill Emery; and from all, a great deal of patience.

## I. INTRODUCTION

Studies of ocean currents in coastal regions have shown that fresh water run-off is an important factor in large-scale circulation (see [6] for several examples). While these studies consider the effect of large rivers on the coastal ocean, the influence of run-off over a large coastal region has not been closely examined. Dr. Paul LeBlond and Dr. William Emery of the Department of Oceanography at the University of British Columbia have undertaken a study of river run-off and salinity distribution off the coast of southern British Columbia and northern Washington State in an effort to understand their effects on alongshore currents.

One aspect of the study has been the development of a mathematical model to describe those coastal currents created by river run-off. The model consists of a set of equations that describe nearshore hydrodynamics in terms of changes in salinity, and thus can be used to measure the impact of coastal run-off. The main purpose of this thesis is to determine a numerical method to solve the equations defining the model, and to develop a computer program implementing the method.

As well as being of interest from a physical oceanographic point of view, it is hoped that the results from the model may be used to gain an understanding of offshore fisheries. In particular, it has been observed that in most years a large percentage of Fraser River salmon migrate via the Strait of Juan de Fuca, while the remainder return through Johnstone Strait at the northern end of Vancouver Island. In some years, however, a

majority return through Johnstone Strait. The reasons for this are not known, but it is likely that the large-scale coastal circulation plays a large part in which route is chosen. An understanding of these mechanisms may lead to an explanation for this behavior.

Chapter One contains a discussion of the derivation of the problem, leading to the equations to be solved. Definitions of the variables and parameters follow, and the boundary and initial conditions are specified. A description of how the model is to be used concludes the chapter.

In Chapter Two, the notation to be used to describe finite difference schemes is introduced. The concepts of consistency, accuracy and stability, as they relate to application of the Crank-Nicolson differencing of a linearized form of the equations, are discussed. The boundary conditions are then specified in terms of finite differences and their effects on stability examined. The nonlinear terms in the model equations are discussed and methods of dealing with them presented.

In Chapter Three, we examine the systems of linear equations produced by the Crank-Nicolson scheme, and discuss methods to solve them. More efficient difference schemes for the equations are the alternating direction implicit, or ADI, methods. They are described and the concepts of Chapter Two are extended to one of these, the Peaceman-Rachford ADI scheme, for a linear equation. Methods of treating the additional problems presented by the model equations, including lower order space derivatives, variable coefficients and nonlinear terms, are



examined, and tested by solving sample problems with known solutions. Results from computer runs are shown and the accuracy and stability of the solutions discussed. Finally, we discretize the full equations in terms of the Peaceman-Rachford scheme.

In Chapter Four, the ADI method is applied to the model equations with simplified boundaries to ensure the difference scheme satisfies basic physical properties. The discontinuous nature of the initial conditions is discussed, and we show how this problem is dealt with. The land boundary condition for the thickness layer is examined and the problems and methods we use to handle them are explained. Next the formula required by the ADI method for calculating the salinity defect boundary conditions is presented. Sample runs and their results for the full model equations are given, with brief physical interpretations. Recommendations for proceeding with the model conclude the chapter.

## II. CHAPTER 1

### 2.1 Derivation Of The Model

The primary concern of this thesis is the set of numerical problems presented by the mathematical model of coastal circulation. The derivation that follows is brief and includes little justification, from a physical oceanography viewpoint, for the assumptions and accommodations made in formulating the final equations. A complete derivation and discussion of the implications of the model is available in [6]. Our derivation is a condensation of that information.

The model considers the nearshore ocean as having two layers; a lower layer of constant density, and hence salinity, and an upper layer of varying density and thickness. The motion of this upper layer is described by the two-dimensional momentum and conservation equations. If we consider a time scale greater than five days, local accelerations can be ignored, and the motion described is primarily geostrophic. In addition, we want only waves with a Rossby number much less than one; that is,  $U_0 f / \Delta x \ll 1$ , where  $U_0$  is the speed scale and  $f$  the Coriolis force. The space scale,  $\Delta x$ , must therefore be much greater than 1 km. We assume at this point that a value of  $\Delta x > 10$  km. is sufficient.

Under these assumptions, the momentum equations for the upper layer are

$$(1) -fv = (-1/\rho_1) \partial p_1 / \partial x - F_x$$

$$(2) fu = (-1/\rho_1) \partial p_1 / \partial y - F_y$$

where:

$(u,v) = (u(x,y,t),v(x,y,t))$  = components of velocity;

$p_1$  = pressure in the upper layer;

$\rho_1$  = density of the upper layer;

$(F_x,F_y)$  = components of friction.

Since the lower layer is assumed to be at rest, and using the assumption that the surface height is much less than the thickness of the upper layer and can therefore be neglected, (1) and (2) can be integrated over the upper layer to yield

$$(3) -fV = -g/2 (H^2\Delta\rho/\rho_1)_x - F_x$$

$$(4) fU = -g/2 (H^2\Delta\rho/\rho_1)_y - F_y$$

where,

$(U,V) = (U(x,y,t),V(x,y,t)) = \underline{U}$  = components of transport, where transport is velocity integrated over the upper layer;

$H = H(x,y,t)$  = the thickness of the upper layer;

$\rho_2$  = density of the lower layer;

$\Delta\rho = \rho_2 - \rho_1$  = density difference between layers;

$g$  = gravity;

In the equations above, we have adopted the usual notation for partial derivatives; they are indicated here and throughout this thesis by  $\partial/\partial x$  (for example) as in equation (1) or by subscripts of the independent variable.

In terms of transports, the volume and salinity conservation equations are

$$(5) H_t + \nabla_h \cdot \underline{U} = w_e$$

$$(6) (HS_u)_t + \nabla_h \cdot \underline{US} = w_e S_b$$

where:

$S_u = S_u(x, y, t)$ , salinity in the upper layer;

$S_b$  = a constant lower layer salinity;

$W_e$  = the entrainment velocity;

$\nabla_h$  = the divergence operator.

Density can be considered a function of salinity, with an equation of state given approximately by

$$(7) \Delta\rho = \rho_1 a(T) S_u$$

where  $a$  is a function of temperature,  $T$ , and is assumed constant for the range of temperatures encountered.

Introducing  $S(x, y, t) = S_b(x, y, t) - S_u(x, y, t)$ , the difference in salinity between lower and upper layers (so  $S(x, y, t) > 0$ ), and using (7) to relate density to salinity, equations (3) through (6) can be rewritten as

$$(8) -fV = -(ga/2\rho_1) (H^2S)_x - kU$$

$$(9) fU = -(ga/2\rho_1) (H^2S)_y - kV$$

$$(10) H_t + \nabla_h \cdot \underline{U} = W_e$$

$$(11) (HS)_t + \nabla_h \cdot \underline{US} = 0$$

Here we have taken friction components  $(F_x, F_y)$  to be proportional to the components of velocity  $(u, v)$ , with  $k$  the friction coefficient, and integrated over the upper layer to produce transports. Multiplying (8) by  $k$  and (9) by  $f$  and combining gives

$$(12) (f^2 + k^2) U = -(ga/2\rho_1) ( f(H^2S)_y + k(H^2S)_x )$$

$$(13) (f^2 + k^2) V = -(ga/2\rho_1) ( -f(H^2S)_x + k(H^2S)_y )$$

Substituting (12) and (13) into (10) yields, with  $C = gak/(2\rho_1(f^2+k^2))$

$$(14) H_t = W_e + C \nabla^2 (H^2 S) .$$

Similarly, substituting (12) and (13) into (11) gives

$$(HS)_t = C (kS \nabla^2 (H^2 S) + fJ(S, H^2 S) + k \nabla S \cdot \nabla (H^2 S))$$

or

$$(15) S_t = C/H (kS \nabla^2 (H^2 S) + fJ(S, H^2 S) + k \nabla S \cdot \nabla (H^2 S)) \\ - (S/H) H_t$$

where  $\nabla^2$  is the Laplacian operator,  $J$  is the Jacobian operator in terms of  $x$  and  $y$  and  $\nabla$  is the gradient operator. The function  $H^2 S$  represents the distribution of the density mass field; we expect fresh water run-off to create gradients in  $H^2 S$ , causing currents in the model domain. Both (14) and (15) are nonlinear, parabolic partial differential equations and are the governing equations for our model of the coastal circulation. Solving (14) yields  $H(x, y, t)$ , the thickness of the upper layer, while solving (15) gives  $S(x, y, t)$ , the salinity defect, the difference between  $S_a$  and  $S_b$ .

The parameters and constants in (14) and (15) are defined and assigned values as follows:

$$g = 9.8 \text{ m/sec}^2$$

$$f = 1.1E-4 \text{ sec}^{-1}$$

$$k = 0.25E-4 \text{ sec}^{-1}$$

$$a = 0.77 \text{ ppt}^{-1}$$

$$\rho_1 = 10^3 \text{ ppt}$$

$$W_e = 6.0E-5 \text{ F}^2 v$$

where

$$F^2 = \text{Froude number} = (U^2 + V^2)/(gaSH^3)/\rho_1$$

$$v = \text{surface velocity} = (U^2 + V^2)^{1/2}/H$$

Further discussion of parameters  $k$ ,  $a$ ,  $W$  and justification for the values chosen can be found in [6]. The components of transport,  $(U,V)$ , and hence the velocity  $v$ , can be found algebraically from (12) and (13).

## 2.2 The Domain Of The Model

The domain of the equations is an area bounded by open ocean to the north, west and south, and on the east by the coast from the Olympic Peninsula to the Queen Charlotte Islands (see Figure 1). The coast is broken between the Olympic Peninsula and Vancouver Island by the Strait of Juan de Fuca and north of Vancouver Island by Queen Charlotte Sound. It is through these two passages that river run-off enters coastal waters.

For the purposes of a numerical model, the domain has been idealized to a square, with  $x$  designating the east-west axis and  $y$  the north-south axis, as illustrated by Figure 2. The origin for this domain is the southwest corner of the square. The land boundary is therefore taken to be parallel to the  $y$ -axis.

## 2.3 Boundary Conditions

The equation for the salinity defect, (12), requires that the gradient on all boundaries be zero; that is,

$$(16) \quad \partial S / \partial n = 0$$

where  $\partial / \partial n$  is the normal derivative. This is the usual form of the no-flow condition across a boundary. Where the land boundary is broken, representing fresh water flow into the domain,  $S$  is prescribed from historical data.

The boundary conditions for H are a little more difficult. It is assumed that near the open ocean boundaries, H does not change rapidly and can therefore be approximated from those values within the domain that are near the boundary. The actual method for finding H on these boundaries will be presented when the numerical scheme has been explained. On the land boundary, the x-direction transport must be zero, as there can be no flow across land, and therefore, from equation (12),

$$f(H^2S)_y + k(H^2S)_x = 0$$

Expanding the derivative terms yields

$$2fSHH_y + fH^2S_y + 2kSHH_x + kH^2S_x = 0$$

or

$$(17) H_y = -(fHS_y + 2kSH_x + kHS_x)/2fS$$

To find H on the land boundary, we note that  $H_x$  can be approximated numerically and  $S_y$  is known from the solution of (15). Thus, (17) is a first order initial value ordinary differential equation of H in y, with the initial value for the equation being the value of H given at the river discharge regions of the boundary.

#### 2.4 Initial Conditions

The solution of (14) and (15) requires initial conditions  $H(x,y,0)$  and  $S(x,y,0)$ . Since the model is intended for use with historical river run-off data, some historic average of thickness and salinity data will eventually be considered as initial values. For the purposes of our initial computer runs, constant field values will be used for both  $H(x,y,0)$  and

$S(x,y,0)$ .

## 2.5 How The Model Will Be Used

Equations (14) and (15), with boundary conditions as specified, describe the effect of river run-off on coastal circulation. The purpose of the model is to try to closely approximate actual currents and salinities by using collected data in boundary conditions. To that end, river discharge data has been collected for the coasts of Washington State, B.C. and Alaska. From these values, surface velocities (and hence transports) at the inflow regions were calculated and a weekly time series produced to be used as boundary values for the model equations. Similarly, daily surface salinity readings, recorded at lighthouses along the coast, have been smoothed to a weekly series for use as salinity boundary values. Upper layer thickness values have been approximated from transports and surface salinities.

A period of approximately twenty years of actual or interpolated weekly transport and surface salinity records are available to be used as boundary conditions for the model. The aim of the study is to approximate the coastal circulation for that length of time, and to compare the results with the known salmon migration data. From this it is hoped that some inferences can be drawn about the relationship between the two.



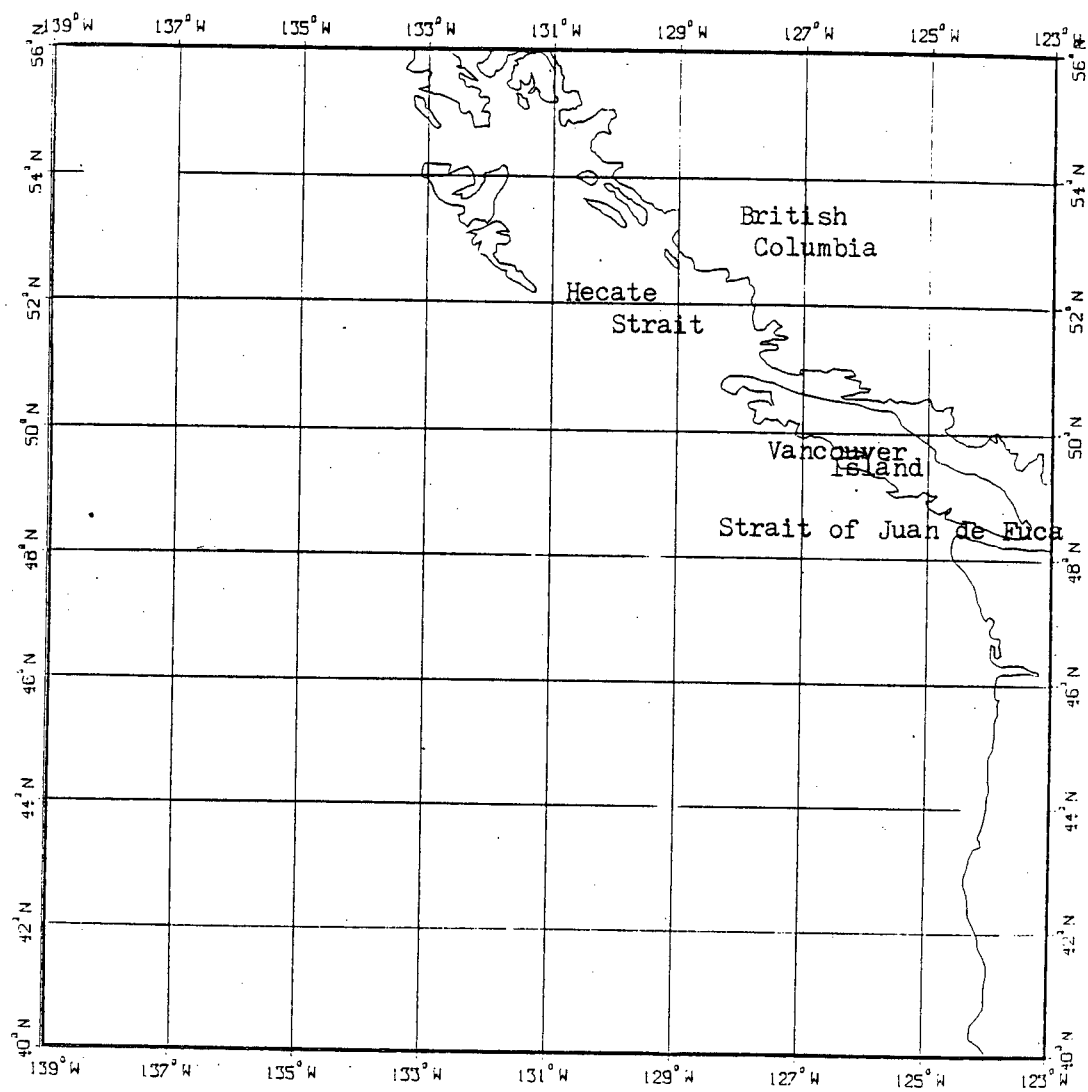


Figure 1 - Domain of the Coastal Circulation Model

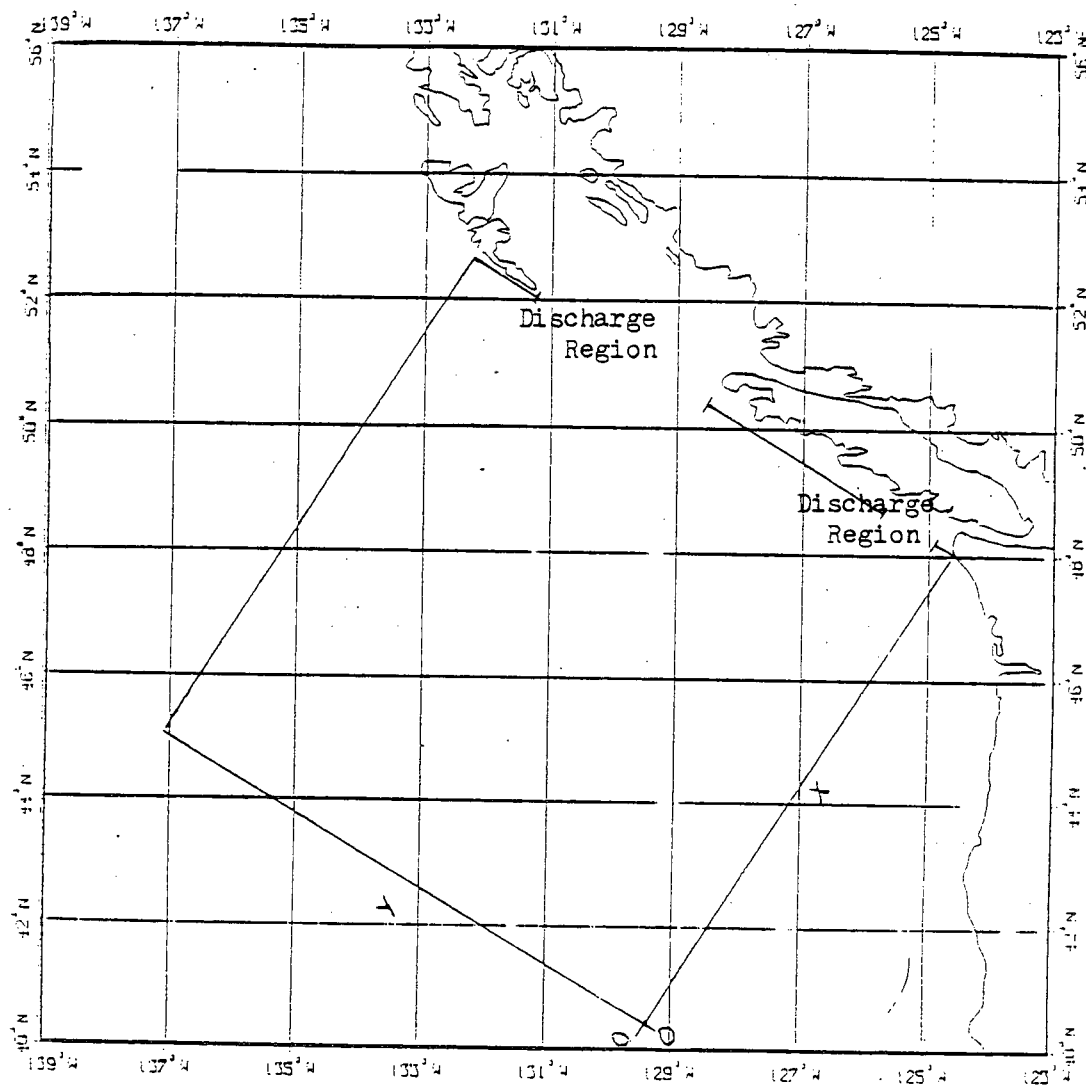


Figure 2 - Domain of the Numerical Model

III. CHAPTER 23.1 Notation

To solve the partial differential equations

$$(1) H_t = CV^2(H^2S) + W_e$$

$$(2) S_t = C/H (kSV^2(H^2S) + fJ(S, H^2S) + kVS \nabla(H^2S)) - (S/H)H_t$$

numerically, the domain is discretized by overlaying a rectangular grid. Initially, this domain will be a simplified version of that shown in Figure 2 by considering it to have just a single discharge point. The model equations are valid over the region represented by this grid, which has grid spacings of  $\Delta x$  and  $\Delta y$  in the  $x$  and  $y$  directions. Coordinates on the grid are specified by  $(i\Delta x, j\Delta y)$ , with  $i=1, \dots, L$  and  $j=1, \dots, M$ , and  $L\Delta x = M\Delta y = 600$  km. The time direction is also discretized by dividing the time domain in  $N$  equal time steps of size  $\Delta t$ . A specific time is represented by  $n\Delta t$ .

The solutions of (1) and (2) will be approximated at any grid point by the discrete solutions  $H(i\Delta x, j\Delta y, n\Delta t)$  and  $S(i\Delta x, j\Delta y, n\Delta t)$ , respectively. For ease of notation, the approximate solutions are simplified to  $H_{ij}^n$  and  $S_{ij}^n$ .

To specify each of (1) and (2) in discrete terms, the partial derivatives must be approximated in some way. Finite element or finite difference methods are used most often for finding numerical solutions to partial differential equations. There are a number of finite element methods for approximating (1) and (2), but they are for the most part difficult to program and in general are computationally expensive to solve. Finite

differences, in addition to being easier to program and less costly to solve, approximate derivatives using divided differences which are more suited to our rectangular domain.

### 3.2 Finite Differences

We will use centred differences for the first and second order space derivatives and the first order time derivative. They will be defined using linear operators  $D_{ox}$  and  $D_{ox}^2$  for the first and second order x-direction derivatives, and  $D_{oy}$  and  $D_{oy}^2$  for the y-direction derivatives.

For some function  $u$ ,  $u=u(x,y,t)$ , let

$$D_{ox} u_{ij} = (u_{i+1,j} - u_{i-1,j}) / 2\Delta x$$

and

$$D_{ox}^2 u_{ij} = (u_{i+1,j} - 2u_{ij} + u_{i-1,j}) / (\Delta x)^2$$

From Taylor series expansion of  $u((i+1)\Delta x, j\Delta y, t)$  and  $u((i-1)\Delta x, j\Delta y, t)$ , it is easily seen that

$$u_x = D_{ox} u + O((\Delta x)^2) \quad \text{and}$$

$$u_{xx} = D_{ox}^2 u + O((\Delta x)^2)$$

where  $O((\Delta x)^2)$  is the order of the remainder term. The  $D_{ox}$  and  $D_{ox}^2$  linear operators are said to be second order accurate approximations. Similarly,  $D_{oy}$  and  $D_{oy}^2$  are linear operators for the y-direction derivatives and are accurate to  $O((\Delta y)^2)$ .

Using the linear operators  $D_{ox}$ ,  $D_{ox}^2$ ,  $D_{oy}$  and  $D_{oy}^2$ , finite difference methods can be constructed to approximate the derivatives in (1) and (2). The solutions  $H$  and  $S$  are found by marching ahead in time in some fashion; the solutions at  $t=(n+1)\Delta t$  depend on values at lower time levels.

If  $H$  and  $S$  are found individually, using values from previous time levels, the method is said to be explicit. While computationally very simple, explicit methods are restricted to some maximum size of time step used to advance in time. Using a time step greater than this maximum, which depends on  $\Delta x$  and  $\Delta y$ , results in solutions that grow exponentially. Avoiding such instabilities by using a time step less than the maximum severely restricts the choice of  $\Delta x$  and  $\Delta y$ . As was mentioned in Chapter 1, various assumptions used in deriving the model equations put constraints on the size of  $\Delta x$  and  $\Delta y$ . An explicit method is therefore not appropriate for finding solutions to our equations.

A finite difference method is implicit if the values of the dependent variable at points at the next time level are interrelated. Implicit methods thus require solution of a system of linear equations at each time level. While naturally taking more computing time than explicit methods, implicit methods generally do not place restrictions on the time step, beyond that required for accuracy.

A well-known implicit method for solving parabolic equations is the Crank-Nicolson method. In terms of the linear operators defined above, the simple two-dimensional diffusion equation

$$u_t = u_{xx} + u_{yy}$$

is approximated by the Crank-Nicolson (or CN) method at  $(i\Delta x, j\Delta y, (n+1/2)\Delta t)$  by

$$u_{ij}^{n+1} - u_{ij}^n = \Delta t (D_{ox}^2 (u_{ij}^{n+1} + u_{ij}^n)/2 + D_{oy}^2 (u_{ij}^{n+1} + u_{ij}^n)/2)$$

The difference scheme is centred about  $t=n+1/2$  to preserve the second order accuracy in time, and is a two-level difference equation, solving for  $t=n+1$  from values at  $t=n$ .

The discussion in the remainder of the chapter which concerns finite difference equations and their application to (1) and (2) will be limited to the CN method.

Before applying the CN scheme to equations (1) and (2), we must decide how the spatial derivatives are to be expanded. The equations have derivatives of products of the dependent variables which must be expressed as difference equations. Consider as an example the second order x-direction derivative,  $(H^2S)_{xx}$ , in equation (1). Applying the product rule in the usual way yields

$$(3) \quad 2HS_{xx} + 2H_x^2 S_x + 4HH_x S_x + H^2 S_{xx}$$

while considering  $(H^2S)$  as a product of functions  $(HS)$  and  $H$  gives

$$(4) \quad (HS)H_{xx} + 2(HS)_x H_x + H(HS)_{xx}$$

Upon discretization, these expressions are clearly unequal, except for fortuitous values of  $H$  and  $S$ . Obviously, the solutions of both (1) and (2) will depend on which way the  $(H^2S)$  derivatives are expanded.

We are solving equation (1) for  $H$ , and it seems reasonable that those functions multiplying  $H$  should be considered as single, separate functions. That the derivatives should be expanded as (4) above was confirmed by experimental computer runs (as explained in Chapter 4).

Expanding equation (1) therefore yields

$$(5) H_t = C((HS)\nabla^2 H + 2(HS)_x H_x + 2(HS)_y H_y + H\nabla^2(HS)) + W_e.$$

The  $H\nabla^2(HS)$  term may be treated as the other terms in (5); that is,  $\nabla^2(HS)$  is considered the coefficient of  $H$ . It is usual ([12], p. 195) to assume that inhomogeneous terms involving the dependent variable are known. (5) becomes

$$(6) H_t = C((HS)\nabla^2 H + 2((HS)_x H_x + (HS)_y H_y)) + f_1$$

where  $f_1 = H\nabla^2(HS) + W_e$ .

The equation for  $S$  also contains derivatives of  $(H^2S)$ , but since we are solving for  $S$ , their expansion must be different than that outlined for  $H$ . Treating  $H^2$  as a single entity was found to work best, and thus, for  $(H^2S)_{xx}$ , the appropriate expansion is

$$H^2 S_{xx} + S(H^2)_{xx} + 2S_x(H^2)_x$$

Expanding equation (2) gives

$$(7) S_t = C/H (kS(H^2\nabla^2 S + 2\nabla H^2 \cdot \nabla S + S\nabla^2 H^2) + fSJ(S, H^2) + k(S\nabla S \cdot \nabla(H^2) + H^2(\nabla S)^2) - (S/H)H_t$$

Just as for  $H$ , terms not involving derivatives of  $S$  are treated as forcing functions, and (7) becomes

$$(8) S_t = C/H (kS(H^2\nabla^2 S + 2\nabla H^2 \cdot \nabla S) + fSJ(S, H^2) + k(S\nabla S \cdot \nabla(H^2) + H^2(\nabla S)^2) + f_2$$

where  $f_2 = (CkS/H)(S\nabla^2 H^2) - (S/H)H_t$ .

While the method of expanding the derivatives has been determined, the coefficients of the derivatives that result are obviously functions of the dependent variables; both equations are nonlinear. Discretization of such equations produces nonlinear systems of equations which would require some kind of iterative scheme for solution, such as Newton's method. Since

iterative methods for nonlinear equations are in general computationally very expensive, we approximate further by linearizing the equations.

Although we leave the computational details until later in Chapter 2, the result of linearizing (6) and (8) are equations having variable coefficients. The equations can be simplified to

$$H_t = a_1 H_{xx} + a_2 H_{yy} + a_3 H_x + a_4 H_y + f_1$$

and

$$S_t = b_1 S_{xx} + b_2 S_{yy} + b_3 S_x + b_4 S_y + f_2$$

where  $a_i$ ,  $b_i$ ,  $i=1, \dots, 4$ , are functions of  $x$ ,  $y$  and  $t$ . In terms of the linear operators defined earlier, the CN discretization is

$$(9) \quad H_{ij}^{n+1} - H_{ij}^n = \Delta t (a_1 D_{ox}^2 H_{ij}^{n+1/2} + a_2 D_{oy}^2 H_{ij}^{n+1/2} + a_3 D_{ox} H_{ij}^{n+1/2} + a_4 D_{oy} H_{ij}^{n+1/2} + f_1)$$

and

$$(10) \quad S_{ij}^{n+1} - S_{ij}^n = \Delta t (b_1 D_{ox}^2 S_{ij}^{n+1/2} + b_2 D_{oy}^2 S_{ij}^{n+1/2} + b_3 D_{ox} S_{ij}^{n+1/2} + b_4 D_{oy} S_{ij}^{n+1/2} + f_2).$$

It should be noted here that evaluating the coefficients of each derivative of the equations shows that, for representative values of  $H$  and  $S$ , and using the constants defined in Chapter 1, both equations are strongly diffusive. This is important in determining the difference scheme because it is well known ([12], p. 26) that central differences are poor approximations to convective terms if their coefficients are large relative to those of the diffusion terms. While the convective terms are not insignificant, the primary moving force is produced by the



diffusion terms.

### 3.3 Consistency, Stability And Convergence

Finite difference equations are naturally only approximations to differential equations. It is necessary to show that the finite difference solution is a 'good' approximation to the exact solution. Briefly, this means that we require the difference equation to tend to the differential equation for  $\Delta x, \Delta t \rightarrow 0$ ; that the solution of the finite difference scheme will give a bounded solution (assuming the same for the exact solution); and that the finite difference solution will approach the exact solution as  $\Delta x, \Delta t \rightarrow 0$ . These properties are called consistency, stability and convergence.

The standard theory for stability and convergence of initial value problems generally assumes constant coefficients, and does not consider the effect of either boundary conditions or variable coefficients. We will assume for the moment that the model problems meet these criteria, and consider later in the chapter the effect on our analysis of variable coefficients and boundary conditions.

To simplify the presentation of these concepts for the CN scheme, we consider the one dimensional version of the model equations. What follows is easily extended to two dimensions.

For  $u = u(x, t)$ , let

$$(11) \quad u_t = au_{xx} + bu_x + f(x, t)$$

with  $a$  and  $b$  constant, and with initial condition

$$u(x, 0) = g(x).$$

Much of the analysis that follows is based on the notation and definitions of Kriess and Oliger ([5], p. 29). Applying the CN scheme to (11) yields

$$(1 - a\Delta t D_{ox}^2 - b\Delta t D_{ox})v(x, (n+1)\Delta t) = (1 + a\Delta t D_{ox}^2 + b\Delta t D_{ox})v(x, n\Delta t) + \Delta t f(x, n\Delta t)$$

with initial condition

$$v(x, 0) = h(x),$$

where  $v(x, t)$  is the solution to the finite difference equation. For any  $x$ , the discretization of (11) produces linear combinations of  $v(x, (n+1)\Delta t)$  and  $v(x, n\Delta t)$ . We can simplify these using matrix notation; (11) can be expressed as

$$(12) \quad Q_1 v(x, (n+1)\Delta t) = Q_0 v(x, n\Delta t) + \Delta t f(x, n\Delta t)$$

Definition The difference scheme is accurate of order  $(q_1, q_2)$  for the particular solution  $u(x, t)$ , if there is a constant  $c$  and a function  $C(t)$ , bounded on every finite interval  $[0, N\Delta t]$ , such that for all sufficiently small  $\Delta t, \Delta x$

$$||Q_1 u(x, t + \Delta t) - Q_0 u(x, t) - \Delta t f|| \leq C(t)(\Delta x^{q_1} + \Delta t^{q_2}) \quad \text{and}$$

$$||h(x) - g(x)|| \leq c(\Delta x^{q_1} + \Delta t^{q_2})$$

where  $||\cdot||$  is the  $l_2$  norm.

We have seen that the linear operators,  $D_{ox}$  and  $D_{ox}^2$ , approximate  $\partial/\partial x$  and  $\partial^2/\partial x^2$  with accuracy  $O((\Delta x)^2)$ , and that the CN approximation of  $\partial/\partial t$  is accurate to  $O((\Delta t)^2)$ . Assuming that  $h(x)=g(x)$ , that the initial condition is known exactly, the scheme is accurate of order  $(2, 2)$ .

The error in the scheme, caused by truncating the Taylor series in our approximations to the derivatives, is called the truncation error. For the CN scheme, the truncation error is  $\Delta t$

times the accuracy, or  $O(\Delta t^3 + \Delta t \Delta x^2)$ . Note that this is the local error, the error in the solution from one time step to the next.

If the difference equation converges to the differential equation being approximated, the method is said to be consistent. For the CN scheme, consistency follows directly from the definition of accuracy. Since  $C(t)$  is assumed to be bounded, and  $q_1, q_2 > 0$ ,

$$C(t)((\Delta x)^{q_1} + (\Delta t)^{q_2}) \rightarrow 0 \quad \text{as } \Delta t, \Delta x \rightarrow 0$$

and the scheme is consistent.

While the difference solution approximates the exact solution with the order of accuracy given above, the exact error depends on the size of the partial derivatives multiplying the  $(\Delta x)^2$  and  $(\Delta t)^2$  terms in the Taylor series expansion. In practice, these of course are unknown. To minimize the error, several steps are taken when actually finding a numerical solution. First, the size of the time step is chosen to be about the same size as the space step; thus  $(\Delta t)^3$  will be roughly equal to  $\Delta t(\Delta x)^2$ . Second, it is conventional to nondimensionalize the equations. This requires the equations to be scaled so that  $0 \leq x, y \leq 1$ . The time scale now depends on the spatial resolution used.

Stability requires that the solution to the difference equation,  $v(x, t)$ , at  $t = n\Delta t$ , is bounded as  $\Delta t, \Delta x \rightarrow 0$ .

Definition The difference approximation (11) is stable for a sequence  $\Delta t, \Delta x \rightarrow 0$  if there are constants  $a_s, K_s$  such that for  $t \geq 0$ , the estimate

$$||v(x,t)|| = ||S(t,0)v(x,0)|| \leq K_s \exp(a_s t) ||v(x,0)||$$

holds.

Here  $S(t,0)$  is a solution operator with

$$S(0,0)=I, S(t_2,0)=S(t_2,t_1)S(t_1,0); t_2 \geq t_1 \geq 0.$$

For  $t=n\Delta t$ , (12) is

$$(13) \quad Q_1 v(x, (n+1)\Delta t) = Q_0 v(x, n\Delta t) + \Delta f(x, n\Delta t)$$

and  $S(n\Delta t, 0)$  can easily be found. For  $Q = Q_1^{-1}Q_0$ ,

$$\begin{aligned} v(x, (n+1)\Delta t) &= Qv(x, n\Delta t) + Q_1^{-1}\Delta t f(x, n\Delta t) \\ &= Q(Qv(x, (n-1)\Delta t) + Q_1^{-1}\Delta t f(x, (n-1)\Delta t) + \\ &\quad Q_1^{-1}\Delta t f(x, n\Delta t)) \\ &\vdots \\ &= Q^{n+1}v(x, 0) + \Delta t \sum_{i=0}^{n+1} (Q^{n-i} Q_1^{-1} f(x, i\Delta t)) \\ &= S(n\Delta t, 0)v(x, 0) + \Delta t \sum_{i=0}^{n+1} S(n\Delta t, i\Delta t) f(x, i\Delta t) \end{aligned}$$

It is evident that the right side of the stability condition in the definition above should include the contribution from the forcing function,  $f$ . That is, the bound should include upper limits on  $\Delta t \sum_{i=0}^{n+1} S(n\Delta t, i\Delta t) f(x, i\Delta t)$ . However, this only increases the bound, and for simplicity, we let the definition stand as is. It is clear that we have to show that  $||Q||^{n+1}$  is bounded for stability to hold.

Assume first that a Fourier transform of  $v(x,t)$  exists. For every frequency,  $w$ , we must show that the amplification factor is not greater than one. That is, stability requires that from one time step to the next the amplitude of every frequency does not increase, except to allow truly increasing solutions. This is the von Neumann necessary condition for

stability ([12], p. 70).

Applying Fourier transforms to (13) requires only the transform of the linear operators  $D_{\theta x}$  and  $D_{\theta x}^2$ . As shown in ([9], p. 37), they are

$$(14) \quad \begin{aligned} D_{\theta x} &= i \sin(\beta/\Delta x) \\ D_{\theta x}^2 &= -4 \sin^2(\beta/2)/(\Delta x)^2, \end{aligned}$$

and  $G$ , the amplification factor, is (from [8], p. 196),

$$G = \frac{1 - 2a\sin^2\beta + i(b/2)(a\Delta t/a) \sin(2\beta) + \Delta tcf}{1 + 2a\sin^2\beta + i(b/2)(a\Delta t/a) \sin(2\beta)}$$

with  $a = a\Delta t/(\Delta x)^2$  and  $\beta = k\Delta x$ . Simplifying  $G$  leads to an equation with

$$(1 + 2a\sin^2\beta)^2 + (b^2/4)a\Delta t/a \sin^2(2\beta)$$

in the denominator; it is clearly greater than 1 for any  $\beta$ . So

$$G \leq \frac{(1 - 2a\sin^2\beta + i(b/2)(a\Delta t/a) \sin(2\beta) + \Delta tcf)}{(1 + 2a\sin^2\beta - ib/2(a\Delta t/a)\sin 2\beta)}$$

In modulus, the amplification factor can be given by

$$(15) \quad |G| \leq (1+h_1(\beta)\Delta t+h_2(\beta)(\Delta t)^2+h_3(\beta)(\Delta t)^3+h_4(\beta)(\Delta t)^4) \\ = 1 + O(\Delta t),$$

where  $h_1(\beta)$ ,  $h_2(\beta)$ ,  $h_3(\beta)$  and  $h_4(\beta)$  are bounded functions. The  $O(\Delta t)$  term allows exponential growth that may exist in the true solution because of the forcing function. The inequality above satisfies the von Neumann condition, which is necessary for stability; it is known ([12], p. 72) that for all two-level difference schemes for scalar equations, the von Neumann condition is sufficient as well.

This shows that the solution from one time level to the

next is bounded.  $||S(t,0)v(x,0)||$  must therefore also be bounded, and the CN scheme meets the requirements of the stability definition. It is clear that  $|G|$  is bounded for any values of  $\Delta t$  and  $\Delta x$ , and the scheme is said to be unconditionally stable.

If the difference between the solutions to the differential and difference equations tends to zero as  $\Delta t, \Delta x \rightarrow 0$ , a difference scheme is said to be convergent. From [5], p. 31, we have the following:

Theorem Assume that for fixed  $\Delta t$  and  $\Delta x$  the definitions for accuracy and stability are valid. Then for all  $t = n\Delta t$ ,  $n \geq 1$ ,

$$||u(x,t)-v(x,t)|| \leq K_s \exp(a_s t) (\Delta x^{q_1} + \Delta t^{q_2}) c.$$

Since it was shown, for the CN scheme, that  $(q_1, q_2) = (2, 2)$  and that stability is unconditional, there is no restriction on the way  $\Delta t, \Delta x \rightarrow 0$ . Therefore the right side tends to zero as  $\Delta t, \Delta x \rightarrow 0$ , and convergence is proven.

### 3.4 Variable Coefficients

As was mentioned, the analysis above assumes a differential equation with constant coefficients, a condition the model equations clearly do not meet. While the accuracy is unaffected by variable coefficients, assuming of course that there are no errors in their calculation, it is quite possible that the stability condition may change.

One method of ensuring stability is to look at the problem in terms of 'local stability'. That is, the coefficients are evaluated at each grid point in the domain and tested separately

for stability. While this does not guarantee convergence, it is usual that instabilities in the approximate solution start locally ([12], p. 91). However, to apply von Neumann's method to a large number of points presents obvious difficulties. One practical way to test for stability of variable coefficient problems is to perform computer runs of sample problems. This was carried out for several problems, and the results are given in Chapter 3.

### 3.5 Stability Of Boundary Conditions

To determine the effect of the boundary conditions on stability, we must examine the system of linear equations produced by the CN scheme for the initial value problem, and check that the introduction of boundary conditions do not add to the matrix  $Q$  any eigenvalues  $z$  with  $|z| > 1$ .

The strategy is to assume that such an eigenvalue  $z$  is introduced by the boundary condition, and show that this leads to a contradiction. In particular, we want to show that this occurs for the boundary conditions of the model equations.

As specified in Chapter 1, the boundary conditions for  $H$  are prescribed on all boundaries. The boundary conditions for  $S$  are prescribed at the inflow regions, and the no-flow condition elsewhere. Assuming smoothly changing functions of  $H$  and  $S$  (and thus no boundary layer problems), prescribed boundary conditions do not change the eigenvalues of the corresponding initial value problem, and so only the no-flow condition will be analysed.

We consider the sample equation (12), with no forcing

function, and  $0 \leq x \leq 1$ . It is known that for parabolic problems, stability of the left boundary condition implies stability for both boundaries [18], and so only the effect of the value at  $u(0,t)$  will be considered. In terms of  $u(x,t)$ , the second order accurate approximation to the no-flow condition at the left boundary is

$$u(0,n\Delta t) = (4/3)u(\Delta x,n\Delta t) - (1/3)u(2\Delta x,n\Delta t)$$

We have seen that the amplification matrix,  $Q = Q_1^{-1}Q_0$ , must have eigenvalues not greater than one for  $v(x,(n+1)\Delta t)$  to be a bounded solution. Let  $g = (g_0, \dots, g_n)$  be an eigenvector of  $Q$ , so that

$$Qg = zg \quad \text{or}$$

$$zQ_1g = Q_0g$$

with  $z$  an eigenvalue.  $Q_1$  and  $Q_0$  can be expressed as sums of their coefficients, so in terms of  $g$  and  $z$ ,

$$\begin{aligned} z \sum_{i=0}^n c_i g_i &= \sum_{i=0}^n a_i g_i \quad \text{or} \\ (16) \quad \sum_{i=0}^n (zc_i - a_i) g_i &= 0. \end{aligned}$$

From [18], (16) is a linear difference equation with constant coefficients, and the solution takes the form,

$$g = \sum_{j=1}^i a_j (T_j(z))$$

where the  $T_j(z)$  are the roots of (16). Substituting this into (16) gives

$$(17) \quad T^2 - 2\beta T + \tau/\eta = 0$$

$$\text{where } \beta = 1/(1+\Delta x) + ((z-1)/(z+1))/(\lambda(1+\Delta x))$$

$$\tau = \lambda/2 - \lambda\Delta x/2$$

$$\eta = \lambda/2 + \lambda\Delta x/2$$



and  $\lambda = \Delta t / (\Delta x)^2$

For  $|z| > 1$ , (17) can be shown to have roots  $T_1$  and  $T_2$  with  $|T_1(z)| < 1$ ,  $|T_2(z)| > 1$ . So

$$(18) \quad g_i = a_i T_1^i(z)$$

and, to be an eigenvector,  $g_i$  must also satisfy the no-flow boundary condition. Thus, we require that

$$g_0 = (4/3)g_1 - (1/3)g_2.$$

Substituting this expression into (18) gives

$$(19) \quad a_1 (1 - (4/3)T_1(z) + (1/3)T_1^2(z)) = 0$$

The boundary condition will produce an unstable solution if, for  $z$ , with  $|z| > 1$ , an eigenvalue, the corresponding root of (17) is also a root of (19). The roots of (19), however, are 1 and 3, contradicting our finding that one of the roots of (17) is less than one. Thus  $z$ , with  $|z| > 1$ , is not an eigenvalue; the no-flow boundary, approximated to second order accuracy, does not affect stability.

### 3.6 Nonlinear Terms

To avoid having to solve our nonlinear equations directly, it is necessary to linearize the coefficients of the derivative terms. Linear equations are, of course, much easier to solve, but the linearization procedure yields only an approximation to the true coefficients.

To examine this effect on the accuracy of the system, we note first the equations are quasilinear, since the coefficients of the second derivative terms involve only first derivatives or the function itself. Although the CN difference scheme can be

modified to produce predictor-corrector formulae [8], the truncation error estimate is  $O(\Delta t \Delta x^2 + \Delta t^{5/2})$  rather than  $O(\Delta t \Delta x^2 + \Delta t^3)$  which the CN scheme (for linear equations) yields. Lees [8] has shown that by estimating the quasilinear coefficients using extrapolated values from the two previous time steps, the truncation error remains  $O(\Delta t \Delta x^2 + \Delta t^3)$ . Since the CN scheme is centred about  $t=n+1/2$ , the coefficients of each term must be calculated at that time. In particular, if the HS terms in the coefficients of the second order partial derivatives of (9) are approximated by

$$HS^{n+1/2} = 1.5 HS^n - 0.5 HS^{n-1},$$

an error of  $O((\Delta t)^3)$  is added to the truncation error, which thus remains  $O(\Delta t \Delta x^2 + \Delta t^3)$ .

We see that the coefficients of  $\partial H / \partial x$  and  $\partial H / \partial y$ , which involve  $\partial(HS) / \partial x$  and  $\partial(HS) / \partial y$  terms, can be approximated in the same way. For (HS) at point  $(i, j)$ , the derivative at time  $n+1/2$  can be approximated by

$$\begin{aligned} (HS)_x &= ( (HS)_{i+1,j} - (HS)_{i-1,j} ) / (2\Delta x) & \text{and} \\ (HS)_y &= ( (HS)_{i,j+1} - (HS)_{i,j-1} ) / (2\Delta y) \end{aligned}$$

where all the (HS) values are calculated at  $t=n+1/2$  as specified above. This method can also be applied to the  $S_x^2$  and  $S_y^2$  terms that appear in the equation for S. One of each of the derivatives can be treated as the coefficient of the other, and approximated by extrapolating to time  $n+1/2$  ([2], p. 141). Equation (10) has quasilinear coefficients which are functions of  $SH^2$ ,  $H^2$ ,  $H_x$ ,  $H_y$ ,  $S_x$  and  $S_y$ . All of these are estimated in the same fashion as outlined above, without altering the order

of the truncation error.

The effect on stability of applying this extrapolation procedure to all the terms in each equation is not known. Varah [19], however, has shown that for an equation involving nonlinear terms in the first spatial derivative of the function, and the function itself, the stability of the CN scheme with extrapolation depends on the relative size of the coefficients of the diffusive and convective terms. For equations which are not highly convective, the CN scheme was found to be conditionally stable. Since both model equations are highly diffusive, it seems likely that stability will be maintained. While we can provide no stability analysis, the results of preliminary computer runs (described fully in Chapter 4) show that, for chosen space and time steps, this linearization produces results which appear to be conditionally stable.

### 3.7 Solving The Equations

To solve the model equations, we note first that they are coupled, in that the two independent variables are present in the two equations. While the Crank-Nicolson difference scheme could be applied to solve for H and S simultaneously, the only new terms to be solved directly rather than approximately are the forcing functions. Since they are nonlinear as well, one of either H or S would have to be approximated in any case. Little would be gained from this approach.

Instead, assume, as was mentioned in Chapter 1, that S varies slowly from one time level to the next. If equation (9)

is solved first, HS has to be estimated (by extrapolation). If S is in fact slowly varying, the error of approximation will be small (at least smaller than that caused by estimating  $H^2$ , as solving equation (10) would require). As we have seen, the addition to the error is  $O((\Delta t)^3)$ , which, will only cause a small increase in the overall error of approximating H.

Once H has been calculated, the average of  $H^{n+1}$  and  $H^n$ ,  $H^{n+1/2}$ , can be used in evaluating the coefficients of equation (10), rather than an extrapolated value. Solving for  $S^{n+1}$ , it is possible to repeat the process, using the new values of  $H^{n+1}$  and then  $S^{n+1}$  to calculate new approximations for  $H^{n+1/2}$  and  $S^{n+1/2}$ . Each step may be considered an iteration of a predictor-corrector method. We can lessen the effect of the interaction of H and S by alternating the order of solution from one iteration to the next. In practice, it is possible to continue iterating until the solutions from one iteration to the next change by only some arbitrary (small) amount. For smoothly changing values of H and S, the solutions will converge, it is hoped, in only a couple of iterations.

Applying the CN difference scheme to the model equations produces two systems of linear equations that must be solved at each time step. If some method of iteration is used, which will change the coefficients of the matrices, each iteration will require the systems to be solved again. It seems likely that the majority of computation time will be spent in solving linear equations, particularly as we increase the resolution to obtain better accuracy. The consequences are discussed in the next

chapter.

IV. CHAPTER 3

We now consider the structure of the matrices we must solve, the possible methods of their solution and the computational effort, in terms of operation counts, each would need. The structure of the matrices for equations (9) and (10) of Chapter 2 depend on the linear operators and the way the  $H_{ij}^{n+1}$  and  $S_{ij}^{n+1}$  are ordered in the vector of unknowns. The linear operators involve only grid points one grid spacing away in each spatial direction; that is, to solve for  $(i,j)$  involves  $(i+1,j)$ ,  $(i-1,j)$ ,  $(i,j+1)$ ,  $(i,j-1)$  as well as  $(i,j)$ . The conventional approach is to order the grid points by row with the resulting matrix being banded with half bandwidth  $L+1$ , where  $L$  is the number of grid points in the  $x$ -direction. For our equations, both matrices are nonsymmetric.

A banded, nonsymmetric matrix can be solved using either a variant of Gaussian elimination or any number of iterative methods. The former has the advantage of reliability, but the latter can be considerably faster. The rate at which iterative methods converge, and thus the number of iterations they require, depends on the condition number of the matrix. For an ill-conditioned matrix iterative methods will often converge slowly, if at all. Noting that the coefficients of our matrix are the result of approximations to nonlinear terms, and vary with  $x$ ,  $y$  and  $t$ , there is little we can say about how well-conditioned the matrices will be. (We assume of course that they remain nonsingular.) This being the case, Gaussian elimination would appear to be the best method.

To calculate the amount of computational effort required, we need only look at the order and the half-bandwidth of the matrices. If  $L$  is the number of grid points in each direction (so  $\Delta x = \Delta y$ ), the order is  $L^2$  and the half-bandwidth  $L+1$ . Although the matrices initially have zeros between the elements on the edge of the band and the diagonals, Gaussian elimination "fills in" these zeros. When calculating the amount of work, these elements must be considered as nonzeros. For such a matrix, having order  $L^2$ , the operation count (the number of multiplications) necessary to reduce it to triangular form is  $O(L^4)$ . However, difference schemes exist for equations with the same form of the model equations which produce matrices which can be solved in only  $O(L^2)$  operations. They are called alternating direction implicit methods, and are the subject of the next section.

#### 4.1 Introduction To ADI

Alternating direction implicit, or ADI, methods are perturbations of difference methods formulated directly from the difference approximations to derivatives. ADI methods most often arise by adding difference operators to a multi-dimensional difference scheme in a way that does not affect the accuracy of the scheme but allows an easier solution.

To illustrate the derivation of a particular ADI method, the CN scheme for  $u_t = u_{xx} + u_{yy}$ , is, using the linear operators introduced in Chapter 2,

$$(1) (1 - \lambda D_{ox}^2 - \lambda D_{oy}^2) u^{n+1} = (1 + \lambda D_{ox}^2 + \lambda D_{oy}^2) u^n$$

where  $\lambda = \Delta t/2$ . If the operator  $\lambda^2 D_{\sigma x}^2 D_{\sigma y}^2$  is added to the difference operator on both sides, (1) becomes

$$(2) \quad (1 - \lambda D_{\sigma x}^2 - \lambda D_{\sigma y}^2 + \lambda^2 D_{\sigma x}^2 D_{\sigma y}^2) u^{n+1} = (1 + \lambda D_{\sigma x}^2 + \lambda D_{\sigma y}^2 + \lambda^2 D_{\sigma x}^2 D_{\sigma y}^2) u^n .$$

Factoring (2) in terms of  $D_{\sigma x}^2$ ,  $D_{\sigma y}^2$  gives

$$(3) \quad (1 - \lambda D_{\sigma x}^2)(1 - \lambda D_{\sigma y}^2) u^{n+1} = (1 + \lambda D_{\sigma x}^2)(1 + \lambda D_{\sigma y}^2) u^n .$$

If we introduce an intermediate time level between  $n$  and  $n+1$ , say  $n^*$ , (3) can be 'split' into two equations, namely

$$(4) \quad \begin{aligned} (1 - \lambda D_{\sigma y}^2) u^{n^*} &= (1 + \lambda D_{\sigma x}^2) u^n \\ (1 - \lambda D_{\sigma x}^2) u^{n+1} &= (1 + \lambda D_{\sigma y}^2) u^{n^*} . \end{aligned}$$

The variable  $u^{n^*}$  is a dummy variable in the sense that the value of  $u^{n^*}$  is not an approximation to the solution at any particular time. The method of splitting used in (4) is called the Peaceman-Rachford formula ([9], p. 60). Although there are many methods for splitting a two-dimensional parabolic equation like (4), this formula will hereafter be referred to as the ADI scheme.

The advantage of (4) is that the solution at the next time level (either  $u^{n+1}$  or  $u^{n^*}$ ), is found along lines parallel to either the  $x$ - or  $y$ -axis. The matrix to be solved thus only involves points immediately adjacent to the point being found; that is, the matrix is tridiagonal. The computational effort required to solve a tridiagonal system of equations is small compared to that for the CN scheme. For an  $L^2$  by  $L^2$  system, the tridiagonal matrix requires only  $O(L^2)$  multiplications, compared to  $O(L^4)$  required by the CN scheme.

This considerable savings in computation time will not come



at the expense of accuracy or stability of the difference method. As we will show, the Peaceman-Rachford scheme, like the CN scheme, has a truncation error of  $O((\Delta t)^3 + \Delta t(\Delta x)^2)$  and is unconditionally stable, for a sample problem similar to that analyzed in Chapter 2. Note that in calculating the truncation error, we have assumed that  $\Delta x = \Delta y$ . For the rest of the thesis, this will be the case.

#### 4.2 Accuracy And Stability

To calculate the accuracy of the ADI scheme, we first consider an equation of the same form as our model equations, namely, for  $u=u(x,y,t)$ ,

$$(5) \quad u_t = au_{xx} + bu_{yy} + cu_x + du_y,$$

where  $a, b, c$  and  $d$  are constant coefficients. Factoring (5) using the ADI scheme in the same manner as above, gives

$$(6) \quad (1 - \lambda a D_{ox}^2 - \lambda c D_{ox})(1 - \lambda b D_{oy}^2 - \lambda d D_{oy})u^{n+1} = \\ (1 + \lambda a D_{ox}^2 + \lambda c D_{ox})(1 + \lambda b D_{oy}^2 + \lambda d D_{oy})u^n,$$

where  $\lambda$  is as above. Splitting (6) we get

$$(7) \quad \begin{aligned} (1 - \lambda b D_{oy}^2 - \lambda d D_{oy})u^{n+*} &= (1 + \lambda a D_{ox}^2 + \lambda c D_{ox})u^n \\ (1 - \lambda a D_{ox}^2 - \lambda c D_{ox})u^{n+1} &= (1 + \lambda b D_{oy}^2 + \lambda d D_{oy})u^{n+*} \end{aligned}$$

To calculate the error, we see first that the operators added to the CN scheme for (5) are  $(\lambda^2 cd)D_{ox} D_{oy}$ ,  $(\lambda^2 bc)D_{ox} D_{oy}^2$ ,  $(\lambda^2 ad)D_{oy} D_{ox}^2$  and  $(\lambda^2 ab)D_{ox}^2 D_{oy}^2$ . These terms do not result from Taylor series expansion of the derivatives, and the errors they produce can be added directly to the truncation error found for the CN scheme. Since each operator is applied to both  $u^n$  and  $u^{n+1}$ , the errors can easily be found by expanding  $u^{n+1}$  in a

Taylor series. Examining just one of these terms,  $(\lambda^2 cd) D D$ , we have

$$(\lambda^2 cd) D D (u + \Delta t u + O((\Delta t)^2)) = (\lambda^2 cd) D D u.$$

Cancelling  $u$  on both sides, we see that the addition to the truncation error of this term is  $O((\Delta t)^3)$ . The contribution to the error from the other operators is similarly found to be  $O((\Delta t)^3)$ , so the truncation error for ADI is of the same order as the CN scheme discussed earlier; the added terms leave the overall accuracy unchanged. Consistency follows in the same way as shown for CN in the last chapter.

Stability for the ADI scheme can be seen by again comparing with the CN scheme. Looking at the amplification factor,  $G$ , for CN applied to equation (5), we have

$$(8) \quad G = \frac{1 - 2(a_1 S_1 + a_2 S_2) + i((b/2)(a_1 \Delta t/a) S_3 + (d/2)(a_2 \Delta t/d) S_4)}{1 + 2(a_1 S_1 + a_2 S_2) + i((b/2)(a_1 \Delta t/a) S_3 + (d/2)(a_2 \Delta t/d) S_4)}$$

where  $S_1 = \sin^2 \beta$ ,  $S_2 = \sin^2 \eta$ ,  $S_3 = \sin 2\beta$ ,  $S_4 = \sin 2\eta$  and  $a_1 = a \Delta t / (\Delta x)^2$ ,  $a_2 = b \Delta t / (\Delta y)^2$ ,  $\beta = k_1 \Delta x$  and  $\eta = k_2 \Delta y$  ([12], p. 196).

It is easily shown that  $|G| \leq 1$ , for  $\Delta t, \Delta x \rightarrow 0$ . In calculating the amplification factor for ADI, we note that the additional terms of the ADI scheme are Fourier transformed to real valued terms involving products of  $\sin^2(\beta)$ ,  $\sin^2(\eta)$ ,  $\sin(2\beta)$  and  $\sin(2\eta)$ . But these terms are added to both the denominator and the numerator of (7); the inequality is unchanged. Thus, the amplification factor for ADI is also bounded by one and ADI is unconditionally stable for equation (5). Using the results from Chapter 2, convergence for ADI is guaranteed by the accuracy and unconditional stability of the

scheme.

#### 4.3 Computational Details And Sample Problems

We have assumed that the coefficients of (5) are constant, as required for the analysis to hold. To examine accuracy and stability for variable coefficients, and to test the effect of various extrapolation schemes for nonlinear terms, a series of equations with known solutions and of similar form to the model equations were solved using the ADI method. The Fortran code developed for this testing was also found to be useful when coding the program for the model equations.

Before discussing the sample problems, we must determine how to evaluate variable coefficients and forcing functions so the ADI equations remain consistent. In addition, the requirement of the ADI scheme for boundary values at  $t=n+*$  must be considered, since our equations do not have boundary conditions defined at intermediate time levels.

To see how to incorporate variable coefficients into the ADI scheme, consider equation (5), with  $a$ ,  $b$ ,  $c$  and  $d$  functions of  $x$ ,  $y$  and  $t$ . The Peaceman-Rachford formula is the same as (6) and it is conventional([9], p. 68) to evaluate the coefficients at  $t=n+1/2$ . This is consistent with the CN difference scheme, which is centred about  $t=n+1/2$ . The accuracy of the ADI scheme is unchanged from the constant coefficient case, assuming the coefficients are known exactly.

The addition of a forcing function to an equation poses a different problem. The forcing function itself must be split in

some way to maintain the consistency of the ADI formula. Considering as an example,

$$u_t = u_{xx} + u_{yy} + F(x, y, t),$$

we note first that  $F(x, y, t)$  must be evaluated at  $t = n + 1/2$ . Using

$$(9) \quad \begin{aligned} (1 - \lambda D_{oy}^2) u^{n+*} &= (1 + \lambda D_{ox}^2) u^n + F(x, y, (n + 1/2)\Delta t)/2 \\ (1 - \lambda D_{ox}^2) u^{n+1} &= (1 + \lambda D_{oy}^2) u^{n+*} + F(x, y, (n + 1/2)\Delta t)/2 \end{aligned}$$

and solving for  $u^{n+*}$ , we find that

$$(10) \quad u^{n+*} = (1 - \lambda D_{oy}^2)^{-1} ((1 + \lambda D_{ox}^2) u^n + F^{n+1/2}/2)$$

where  $F^{n+1/2} = F(x, y, (n + 1/2)\Delta t)$ . Substituting (10) into the second split equation gives

$$\begin{aligned} (1 - \lambda D_{oy}^2)(1 - \lambda D_{ox}^2) u^{n+1} &= (1 + \lambda D_{ox}^2)(1 + \lambda D_{oy}^2) u^n + \\ &\quad ((1 - \lambda D_{oy}^2) + (1 + \lambda D_{oy}^2)) F^{n+1/2}/2 \\ &= (1 + \lambda D_{ox}^2)(1 + \lambda D_{oy}^2) u^n + F^{n+1/2} \end{aligned}$$

which shows that (9) is consistent.

When solving either Peaceman-Rachford equation, time-dependent boundary values at the intermediate time level,  $t = n + 1/2$ , are needed. However, this time level does not correspond to any particular time. The values could be approximated by averaging values at  $t = n + 1$  and  $t = n$ , but the overall accuracy of the scheme would not be maintained ([15]). It is possible to find  $u^{n+*}$  explicitly in terms of the values at  $u^n$  and  $u^{n+1}$ ; the formula follows from equation (7). Adding the split equations gives

$$\begin{aligned} ((1 - \lambda D_{oy}^2 - \lambda D_{ox}^2) + (1 + \lambda D_{oy}^2 + \lambda D_{ox}^2)) u^{n+*} &= (1 - \lambda D_{ox}^2 - \lambda D_{oy}^2) u^{n+1} + \\ &\quad (1 + \lambda D_{ox}^2 + \lambda D_{oy}^2) u^n \end{aligned}$$

which is just

$$u^{n+*} = ((1 - \lambda D_{ox}^2 - \lambda D_{oy}^2) u^{n+1} + (1 + \lambda D_{ox}^2 + \lambda D_{oy}^2) u^n) / 2.$$

We require  $u^{n+1}$  on the x-direction boundaries, and it is easily seen that the difference operators involve values only on those boundaries. This formula ensures that calculation of the boundary values preserves the order of accuracy of the method.

We can now apply the ADI scheme to a series of sample problems. The problems we have chosen are intended to show that ADI has the accuracy predicted for the constant coefficient problem and that, although not as accurate for variable coefficient and nonlinear problems, the method appears to be at least conditionally stable.

The sample problems all have the general form

$$(11) \quad u_t = au_{xx} + bu_{yy} + cu_x + du_y$$

where the exact solution  $u$  and the coefficients are chosen to produce the various problems of interest. We use the exact solution on the boundaries except at the intermediate time levels, where the boundary conditions are calculated as explained above. The results, consisting of the maximum absolute error at the interior grid points at time = 1.0 for various values of  $\Delta t$  and  $\Delta x$ , are given in Table 1.

For the constant coefficient problem, we set  $a = b = 0.5$  and  $c = d = 1.0$ , which has exact solution  $u = \exp(-x-y-t)$ . The results are given in Table 1 under (I). The errors vary according to the truncation error of the method, which is  $O((\Delta t)^3 + \Delta t(\Delta x)^2)$ . We see that the errors are smaller than the truncation error would indicate, which is likely due to cancellation of error terms, whose coefficients are various higher derivatives of  $\exp(-x-y-t)$ .

To test a variable coefficient problem, (11) is solved with  $a=0.5/(y+.1)^2$ ,  $b=0.5/(x+.1)^2$ ,  $c=1/(y+.1)$  and  $d=1/(x+.1)$ , which has true solution  $u=\exp(-(x+.1)(y+.1)-t)$ . The results are given under (II) and are much the same as the constant coefficient case. For both these cases the error becomes constant after the first ten to twenty time steps (for some cases, errors were checked for 200 time steps), indicating stability of the method for the time and space steps given in the table.

One of the nonlinear terms both model equations contain is the square of the convective terms. As described in Chapter 2, these are approximated by discretizing one derivative and treating the other as its coefficient. The sample equation is solved with  $a$  and  $b$  as for the variable coefficient problem,  $c=-\bar{u}_x/(u_x(y+.1))$  and  $d=-\bar{u}_y/(u_y(x+.1))$ , where  $\bar{u}_x$ ,  $\bar{u}_y$  are approximated by extrapolation and  $u_x$ ,  $u_y$  are known exactly. The exact solution is the same as for the variable coefficient problem.

Case III in Table 1 contains the results found by approximating  $\bar{u}_x$  by  $D_{0x}u^n$  and  $\bar{u}_y$  by  $D_{0y}u^n$  and case IV the results with approximations  $D_{0x}(1.5u^n-0.5u^{n-1})$  and  $D_{0y}(1.5u^n-0.5u^{n-1})$ . The two-point extrapolation is slightly more accurate, although not as much better as expected. For several runs with this extrapolation, however, the solution is unstable at later times, oscillating and finally blowing up. This is apparently caused by a parasitic solution introduced numerically by linking three time levels together. (That the one-point extrapolation did not blow up would seem to confirm this.) Since

the solution with  $\Delta t = .05$ ,  $\Delta x = .10$  did not blow up, it appears this problem can be avoided by a careful choice of time and space steps.

Since the two-point extrapolation requires solutions from two previous time levels, a solution for time  $= \Delta t$  is necessary. We compared the effect of approximating this solution by a Taylor series expansion with using the exact solution and found no difference in the errors for the test values of  $\Delta x$  and  $\Delta t$ .

To test the effect of nonlinear coefficients of the diffusion terms, we solve (12) setting  $a = .5u/((y+.1)^2\bar{u})$ ,  $b = .5u/((x+.1)^2\bar{u})$  and  $c$  and  $d$  as for the variable coefficient problem. Here  $\bar{u} = 1.5u^n - 0.5u^{n-1}$  and the exact solution is again  $u = \exp(-(x+.1)(y+.1)-t)$ . The solution at time  $= \Delta t$  was found approximately by Taylor series expansion.

The results are given under (V). All the examples except  $\Delta t = .05$ ,  $\Delta x = .10$  blow up at later times; the scheme is obviously no longer unconditionally stable. As for the previous example, this may be due to the presence of parasitic solutions. To find the convergence rate for this problem, several other runs were made with  $\Delta t$  and  $\Delta x$  chosen to produce stable results. These solutions are not only less accurate than the corresponding variable coefficient problem, but do not converge at the rate we expect; the errors changes only as  $\Delta t$ . The reasons for this are not clear but seem to be due to the form of the nonlinear problem.

We conclude from these sample problems that the ADI scheme and the methods chosen for handling nonlinear terms produce

acceptable solutions. The larger errors and conditional stability for the nonlinear equations are due in part to the specific solution and the fact that the equations have coefficients which contain exact solutions or exact derivatives. The error that results from the approximations to these terms will always be in the same direction, which is certainly not necessarily the case for truly nonlinear equations. It should be noted as well that the error produced by extrapolating the coefficients of the convective terms is of little concern since the model equations are highly diffusive; the convective terms will have only a small contribution to the overall solution.

$\Delta t$	$\Delta x$	(I)	(II)	(III)	(IV)	(V)
.05	.05	1.5E-5	4.1E-6	5.5E-4	3.3E-4	1.6E-4
.05	.10	5.8E-5	1.5E-5	5.7E-4	3.4E-4	1.3E-4
.10	.05	1.9E-5	5.9E-6	1.1E-3	1.3E-3	2.1E-3
.10	.10	6.2E-5	1.6E-5	1.1E-3	6.4E-4	2.9E-4
.20	.20	2.5E-4	6.7E-5	2.3E-3	4.8E-3	3.9E-4

Table 1. Results of ADI solutions of sample problems.

#### 4.4 Applying ADI To The Model Equations

The final form of the ADI scheme for the model equations is found by applying the procedures discussed in this chapter to equations (9) and (10) of Chapter 2. The split equations for (9) are



$$\begin{aligned}
 (13) \quad & (1 - \lambda a_2 D_{oy}^2 - \lambda a_4 D_{oy}) H^{n+*} = (1 + \lambda a_1 D_{ox}^2 + \lambda a_3 D_{ox}) H^n + \Delta t f_1 / 2 \\
 & (1 - \lambda a_1 D_{ox}^2 - \lambda a_3 D_{ox}) H^{n+1} = (1 + \lambda a_2 D_{oy}^2 + \lambda a_4 D_{oy}) H^{n+*} \Delta t f_1 / 2
 \end{aligned}$$

and for (10) are

$$\begin{aligned}
 (14) \quad & (1 - \lambda b_2 D_{oy}^2 - \lambda b_4 D_{oy}) S^{n+*} = (1 + \lambda b_1 D_{ox}^2 + \lambda b_3 D_{ox}) S^n + \Delta t f_2 / 2 \\
 & (1 - \lambda b_1 D_{ox}^2 - \lambda b_3 D_{ox}) S^{n+1} = (1 + \lambda b_2 D_{oy}^2 + \lambda b_4 D_{oy}) S^{n+*} \Delta t f_2 / 2
 \end{aligned}$$

where the coefficients,  $f_1$  and  $f_2$  are defined as in Chapter 2 and evaluated at time =  $n+1/2$ .

## V. CHAPTER 4

In the first part of this chapter, several preliminary numerical tests of the model equations are discussed. They involve simplifying the boundary conditions to produce a closed system, allowing us to verify that the approximate solutions are conserving. We also discuss the computational details of both sets of initial and boundary conditions, neither of which, it was found, could be applied in a straightforward manner. The next section is devoted to testing the model equations with these initial and boundary conditions using various time series of input values. A physical interpretation of the plotted results follows. We conclude with recommendations for continued work on the model, both with steps which might be taken to more accurately represent the physical system being modelled and with improvements to the numerical methods used.

### 5.1 Preliminary Tests

The purpose of these tests is to ensure that applying the ADI scheme and the nonlinear approximations to the model equations conserve mass, as measured by  $H$ , and total salinity, as measured by the product of  $H$  and  $S$ . We consider the equations over the specified domain, but impose closed boundaries. By maintaining constant values of  $H$  and  $S$  on all boundaries, and setting the velocities  $u$  and  $v$  to zero on the  $y$ - and  $x$ -direction boundaries respectively, it is clear that the mass and total salinity in the system should remain constant.

For initial conditions we use constant values of  $H$  and  $S$

everywhere except at a single grid point. At this point, which is in the center of the domain, H or S or both is set to a value larger than the constant field. Three runs were made with constant values of 20 meters for H and 2 p.p.t. for S, and step sizes of  $\Delta x = \Delta y = \Delta t = 0.1$ . The results are given in Table 2 below. Mass and HS are calculated simply by summing over the grid points.

H(6,6) t=0	S(6,6) t=0	Mass t=0	HS t=0	Mass t=100	HS t=100
22.0	2.0	2422.0	4844.0	2420.2	4840.4
22.0	2.5	2422.0	4855.0	2422.9	4840.7
28.0	2.5	2428.0	4870.0	2421.4	4840.9

Table 2. Results of Conservation Tests

The method appears to conserve both quantities very well, with an error over 100 time steps of less than .1 percent for the first case, and slightly larger for the other two. For the crude 11 by 11 grid used, this is very good. The results of the second and third tests also indicate, not surprisingly, that the larger the point sources, the less accurate the spatial derivatives are approximated, and the larger the overall error in terms of conservation.

## 5.2 Initial Conditions At The Land Boundary

For purposes of testing, constant field values were assigned as initial conditions to both the height of the upper layer,  $H$ , and the salinity defect,  $S$ . The values chosen were 20 meters for  $H$  and 2 ppt for  $S$ , except at the river discharge point. Here the values we assigned to  $H$  and  $S$  depended on the time series we were testing.

To test the problem created by a value of  $S$  at the discharge point larger than those nearby, we used a weekly series with  $S$  equal to five. The nearly-discontinuous nature of the function at the initial time level resulted in extremely unlikely values of the solutions at time  $\Delta t$  for both  $H$  and  $S$  at grid points near the discharge. These results persisted despite several iterations, and there was no indication that the solutions for  $H$  and  $S$  would converge.

The problem stems from having to find the solution of equation (14), from Chapter 1, which is

$$(1) H_y = -(fHS_y + 2kSH_x + kHS_x)/2fS .$$

Due to the large gradients of  $S$  created by a large input value, (1) produces large values of  $H$ . To avoid the effect of this near-discontinuity, we assume [7] that  $S$  at the discharge point has an immediate effect on points in the region; that is, the impact of large changes in salinity defect is smoothed over the region. We arbitrarily chose five grid points in either direction along the land boundary, two grid points into the domain, and varied  $S$  linearly over these points.

We also want to find a value of  $H$  at the discharge point

such that  $H^2S$  is initially in equilibrium with nearby values of  $H^2S$ . To find such an  $H$ , we use the following procedure [7]. It is assumed that the discharge region consists of only one grid point, but includes the region up to but not including the grid points immediately north and south. The land boundary condition for  $U$  is

$$U/C = -(f(H^2S)_y + k(H^2S)_x) ,$$

with  $U$  given at the discharge point and  $U = 0$  elsewhere. The gradient of  $H^2S$  across the region is negative, due to the Coriolis force, and we approximate it using

$$(2) \quad \begin{aligned} (H^2S)_{K+1} &= (H^2S)_K - \delta(H^2S) \\ (H^2S)_{K-1} &= (H^2S)_K + \delta(H^2S) \end{aligned}$$

where the discharge point  $(L\Delta x, K\Delta x)$  is denoted by the subscript  $K$ , the first grid point to the north by  $K+1$ , and the first to the south by  $K-1$ .  $\delta(H^2S)$  is the increase in  $H^2S$  from point  $K$  to  $K-1$ . We see from (2) that a second order approximation to  $(H^2S)_y$  is

$$(H^2S)_y = \delta(H^2S)/\Delta x .$$

To find  $\delta(H^2S)$ , we note that  $U = 0$  at all grid points except point  $K$ , and so

$$-f(H^2S)_y = k(H^2S)_x$$

at these points. Denoting the initial constant field value by  $(H^2S)_\infty$ , we can approximate  $-f(H^2S)$  on the boundary north of the discharge point by

$$-f((H^2S)_\infty - (H^2S)_{K+1})/5\Delta x .$$

If we assume further that  $(H^2S)_x$  has the same value at the discharge point and at the five grid points north and south, we

have

$$k(H^2S)_x = -f((H^2S)_\infty - (H^2S)_{K+1})/5\Delta x$$

and so at the discharge point,

$$-U/C = f\delta(H^2S)/\Delta x - f((H^2S)_\infty - (H^2S)_{K+1})/5\Delta x$$

or

$$(3) -U\Delta x/(Cf) = 1.2\delta(H^2S) - 0.2((H^2S)_\infty - (H^2S)_K) .$$

We also require that the grid point inside the domain adjacent to the discharge point be equal to  $(H^2S)_\infty$ . Since this point is  $((L-1)\Delta x, K\Delta x)$ ,  $H^2S$  can be expanded in a Taylor series in the  $x$ -direction about  $L\Delta x$ , which is, to  $O(\Delta x)$ ,

$$H^2S((L-1)\Delta x, K\Delta x) = H^2S(L\Delta x, K\Delta x) - \Delta x(H^2S_x(L\Delta x, K\Delta x))$$

or

$$(4) (H^2S)_\infty = (H^2S)_K + (f/5k)((H^2S)_\infty - (H^2S)_K - \delta(H^2S)) .$$

We know  $S$  everywhere along the boundary,  $U$  at the discharge point, and  $(H^2S)_\infty$ , our initial condition, so the only unknowns in equations (3) and (4) are  $\delta(H^2S)$  and  $(H^2S)_K$ . Two equations with two unknowns are easily solved, and once  $\delta(H^2S)$  and  $(H^2S)_K$  are known,  $H(L\Delta x, (K+1)\Delta x)$  and  $H(L\Delta x, (K-1)\Delta x)$  can be determined from (1).  $H(L\Delta x, (K\pm i)\Delta x)$ ,  $i=1, \dots, 5$ , are found by assuming  $H^2S$  is linear from  $(H^2S)(L\Delta x, (K\pm 1)\Delta x)$  to  $(H^2S)_\infty$ . Finally, using a first order difference approximation for  $(H^2S)_x$ , the values for  $H((L-1)\Delta x, (K\pm i)\Delta x)$ ,  $i=1, \dots, 5$ , are easily determined from (2).

The result of all this is simply to smooth large initial values of  $S$ , and to provide a method of finding  $H$  at the discharge region so that initially an equilibrium in  $(H^2S)$  is maintained. When incorporated into the model, this procedure smoothed the results sufficiently to produce reasonable results.

### 5.3 Finding H On The Boundaries

In Chapter 1, we mentioned only that the boundary conditions for H were prescribed on both the ocean and land boundaries. The computational details are given in this section.

On the open ocean boundaries H is unknown. It is reasonable to assume, however, that H does not change rapidly, and that any changes that do occur are due to gradients of H (and hence S) within the domain. Except at the first time level, where H is set to the values used initially, we calculate H on the ocean boundaries by extrapolating from values at the two previous time levels using the extrapolation formula discussed in Chapter 2. After the model equation for H has been solved at the current time level, H on the boundary is further modified by extrapolating from the two nearest interior points on a grid line. That is, we use

$$H_{i0} = 2H_{i1} - H_{i2} ,$$

at point  $(i\Delta x, 0)$ , for example. The values on the ocean boundary thus reflect more closely what is occurring within the domain.

For H on the land boundary, we must solve equation (1) from the previous section with as an initial value the input value of H at the river discharge point. Since the discharge point is in the middle of the land boundary, (1) must be solved in each direction. When solving it for the southern portion of the boundary, the signs of the y-derivatives must be reversed.

In the process of finding a solution to equation (1), a number of difficulties were encountered. One was the effect of

the salinity defect values chosen as input at the river discharge point. For test purposes, we assigned input values to  $S$  that were large compared to adjacent boundary and interior points. Even with the smoothing procedure described in the last section, this led to a large gradient,  $S_y$ , in each direction and produced a stiff equation that was difficult to solve near the discharge point. Integration methods using  $\Delta x$  as a step size only produced physically impossible results.

It is necessary therefore to use an integration step size much smaller than  $\Delta x$ , which requires values for  $S$  and  $U$  at points between our grid points. For  $S$ , this is accomplished by fitting a cubic spline through  $S$  at the grid points and evaluating the curve at the required points. For  $U$ , which is zero at the grid points corresponding to the land boundary, but nonzero between the discharge point  $(J\Delta x, K\Delta x)$  and grid points  $(J\Delta x, (K+1)\Delta x)$  to the north and  $(J\Delta x, (K-1)\Delta x)$  to the south, we need some function which describes its behavior. We chose an exponential curve with a large negative exponent, such that  $U(J\Delta x, K\Delta x)$  is the input discharge value and  $U(J\Delta x, (K+1)\Delta x)$ ,  $U(J\Delta x, (K-1)\Delta x)$  are very nearly zero. This, it is hoped, is a reasonable approximation to  $U$  at the "river mouth". To include these nonzero values of  $U$  in equation (1), we must reintroduce the term  $-U/(2fSCH)$  on the right side of (1), and the ODE becomes

$$(5) \quad H_y = -(U/(CH) + fHS_y + 2kSH_x + kHS_x)/(2fS)$$

A further problem arises when evaluating the  $x$ -derivative terms. Approximating  $H$  and  $S$  with the second order difference



equation detailed earlier involves values at grid points inside the domain. Unfortunately, these interior points must be at the same time level (say  $\text{time} = n+1$ ) we are solving equation (5) for; values either at the interior points or on the land boundary must be estimated in some way. Rather than extrapolating from previous time levels to provide approximate values at the interior points, we first solve the model equations for  $H$  and  $S$ . The boundary values for  $H$  on all boundaries are extrapolated from the two previous levels in the usual way. Once solved, we have interior values for  $H$  and  $S$  at time  $n+1$ ,  $H$  and  $S$  can now be approximated, and equation (5) solved. This results in new boundary values for  $H$  which replace the extrapolated values.

A solution to equation (5) is found using the UBC GEARB subroutine, which is an iterative method using a backward differentiation formula as a predictor and the chord method as a corrector. The reader should consult [10] for a more detailed explanation.

We note here that since  $H$  must be extrapolated, and since  $H$  can change rapidly making the extrapolation less accurate, we wish to avoid having to find land boundary values at the intermediate time level, which does not correspond to an actual time. We do this by splitting the equation such that we solve for  $H$  first in the  $y$ -direction, then in the  $x$ -direction. Land boundary values are thus only required for times  $n$  and  $n+1$ .

#### 5.4 Boundary Condition For S

The boundary condition for the salinity defect is the no-flow condition, which requires setting the first derivative to zero. Using a first order accurate approximation to the derivative has one major advantage; when it is discretized, the matrix equations remain tridiagonal. As we have noted, the salinity defect changes rapidly on the land boundary, particularly near the river discharge region (depending on the input values used). For such cases, it is very important to approximate the boundary condition here with at least the same accuracy as the ADI scheme.

A second order accurate approximation to the no-flow condition is usually centred about the grid point on the boundary, requiring a value for  $S$  outside the domain. On the open ocean boundaries, this is a reasonable approach, but on the land boundary, such a value has no physical meaning. Instead we use a difference formula involving only interior points; for the western boundary, for example, this is

$$(6) \quad S_{0j} - 4S_{1j} / 3 + S_{2j} / 3 = 0, \quad j = 1, \dots, L.$$

Similar equations apply on the other boundaries.

Incorporating (6) into the ADI matrix equations for  $S$ , we see that each row corresponding to the boundary contains two elements to the right of the diagonal; the tridiagonality of the system is lost. Rather than try to solve a system with a half-bandwidth of three instead of two, a row reduction is performed on each of these rows before solving the whole system. For example, the first two rows of the matrix equation for  $S$  at

$t=n+*$  are

$$\begin{array}{ccccccc} 1 & -4/3 & 1/3 & 0 & \dots\dots\dots \\ a_{21} & a_{22} & a_{23} & 0 & \dots\dots\dots \end{array}$$

This can be reduced to

$$\begin{array}{ccccccc} (1-a_{21}/(3a_{23})) & (-4/3-a_{22}/(3a_{23})) & 0 & \dots\dots\dots \\ a_{21} & a_{22} & a_{23} & \dots\dots\dots \end{array}$$

Reducing each such row for both left and right boundaries returns the system to tridiagonality, and increases the operation count by only  $O(L)$ .

### 5.5 Tests Of The Model Equations

Incorporating these methods for initial and boundary conditions into the FORTRAN program tested in the first section of this chapter, we can proceed with tests of the full model equations. Input to the program consists of test data at the inflow, which was considered to be a single grid point located at the midpoint of the  $x=L\Delta x$  boundary. For each time step, the salinity defect,  $S$ , and  $x$ -direction velocity,  $u$ , corresponding to this point are input. The thickness of the upper layer,  $H$ , is calculated as explained in section 4.3 so as to be in equilibrium with these values of  $S$  and  $u$  at time  $t=0$ .

Input for the first experimental run consists of a pulse inflow, with  $S=5.0$  ppt,  $u=0.3$  m/sec, and the corresponding equilibrium value for  $H$ , which is  $H=17.66$ . Initial conditions are  $S=2.0$  ppt and  $H=20.0$  m throughout the domain. The inflow values are introduced into the model at time  $t=0$ , and remain the same until the final time step,  $t=50$ .

Based on the results of the first run, the second experimental run is identical with the first until  $t=15$ . At this point the boundary conditions at the inflow are changed to  $u=0$  and  $dS/dx=0$ , corresponding to shutting off the supply of fresh water into the model.  $H$  is found by extrapolating from the two previous time steps in the usual way. This run is continued until time step 40.

### 5.6 Interpretation Of Results

The results are presented in Figures 3 to 35 and consist of contour plots of  $H$  and  $S$  and a contour plot of  $H^2S$  (representing the pressure field) overlaid with a vector plot of velocity. The vector plot is normalized to the largest velocity at the time level, whose value in m/sec is printed on the figure. The point source of fresh water is at the midpoint of the right  $y$ -axis, at  $x=600$  km,  $y=300$  km.

Due to the algorithm used for contouring, a number of plots have regions of unusual looking contours. The areas where this occurs have values that are all approximately the same, and the contouring program cannot produce accurate contours. On all plots the results for the western half of the domain should be ignored; the accurate contours are always smooth curves.

The Figures for  $t=2$  reflect numerical difficulties created by the initial large gradients of  $S$ . Despite the smoothing procedure used, these gradients induce unlikely values of  $H$  around the inflow point. By using a large number of iterations over the next few time steps, values for  $H$  are smoothed, as can

be seen by  $t=5$ .

The region of low salinity moves slowly north, due to Coriolis force, growing larger as more fresh water flows in the system. This is accompanied by an increase in the thickness of the upper layer in this region; to the south,  $H$  decreases. The figures for the pressure term  $H^2S$  and velocity show the flow to be primarily northwards. That the velocities do not lie exactly along the  $H^2S$  contours reflect the effects of friction; the flow is pushed to the left of these contours.

By time step 15 this pattern is well established and Figures for later time steps show only an increase in the upper layer thickness and a spreading by advection of the fresh water along the surface. As noted in [6], increasing amounts of fresh water move west and south as well due to diffusive effects.

By time step 30, the limits of the domain and the boundary conditions in force there begin to have an affect on both upper layer thickness and flow. It appears that the flow is being "reflected" away from the northeast corner, causing flow back into the domain. A possible cause is the extrapolation scheme for  $H$ , which may give too large a value for  $H$  on the boundary in this area.

While the results are not shown, the same boundary conditions were used until time step 100. The reflecting flow increases along the northern boundary, but otherwise the results change very little from those at time step 40.

Being identical to the first test until  $t=15$ , the results for the second experimental run begin at time step 20 (Figures

27, 30 and 33). At  $t=16$ , the fresh water flow into the model is "turned off"; we would expect the values to slowly return to their initial conditions. As can be seen in Figures 27-35 this is in fact what happens, as the gradients of  $S$  and  $H$  decrease due to advection and entrainment of saltier lower layer water into the upper layer. These values decrease from one time step to the next until at  $t=40$  the maximum value of  $S$  is about 3.6 ppt while that of  $H$  has decreased from a high of about 28 m. to about 22 m.

The region of highest salinity defect moves south pushed by the flow, whose speeds (the maximum of which is given on the  $H^2S$  figures) diminish rapidly. The flow continues to the left of the  $H^2S$  contours. There appears to be an eddy (Figure 35) just to the south of the inflow point, perhaps due to the "reflection" in the northeast corner of the domain.

Both these experimental runs show that the numerical model produces reasonable results with no sign of any instabilities. The second run also shows that a given physical situation, the system returning to rest, is adequately modelled.

Two other tests were carried out which serve to illustrate further difficulties with solving the model equations. Both produced unstable solutions, so no figures are available.

The first used the boundary conditions of the first test discussed above until time step 40. Thereafter, the velocity  $u$  remained the same, but  $S$  was slowly decreased over time to  $S=2$ . The values of  $H$  were those found to be in equilibrium. This models the effect on the system of a flow of varying salinity.

The salinity defect was decreased at the rate of as little as 0.1 ppt per time step; even at this rate the model quickly developed instabilities in the values of  $S$  (soon followed by  $H$ ). Iterating a large number of times showed the solution for  $S$  to be actually diverging.

The other unsuccessful test involved the usual boundary conditions, but with a spatial mesh size one quarter of that indicated above. Instabilities resulted within two or three time steps. Increasing the number of iterations again led to a diverging solution.

### 5.7 Recommendations For Proceeding

There are several shortcomings in the model equations and the numerical methods we have chosen to solve them that should initially be considered when improving the model. The numerical difficulties must be dealt with before any changes can be made in the equations, so they will be discussed first.

The most obvious problem is illustrated by the two tests discussed at the end of the last section. In both cases, the instabilities seem to be caused by large gradients near the coastal boundary. In the first, it resulted from reducing the input salinity defect from 5.0 ppt after reaching a near-steady state solution. The input salinity defect dropped below  $S$  at surrounding grid points producing gradients that apparently created instabilities in equation (5), the land boundary ODE; the resulting values of  $S$  on the coast increase to values larger than any entering the model at the inflow.

Results from the second indicate that the same problem results from a decrease in mesh size. Since the right-hand side of (5) depends on x-direction derivatives of H and S which are approximated by difference equations, these values become larger (assuming adjacent values of H and S are approximately the same) and (5) gets even stiffer as the resolution increases. Again the solution blows up along the coastal boundary, even when increasing the accuracy of the derivative approximations to fourth order. A method of solving this problem is not readily apparent, although a variation of the averaging scheme used for the initial conditions might be considered.

Another numerical problem as yet unresolved is created when the true domain is considered; the land boundary now contains two segments of the coast where run-off enters the model. To the north of the northern segment and to the south of the southern segment, equation (5) can be solved as usual. Between these segments, however, the initial value problem (IVP) becomes a boundary value problem (BVP), as the values for H at both ends are fixed. Considering the difficulties encountered in solving (5) as an IVP, it seems likely that it will present further problems as a BVP, probably having boundary layers at both ends.

To more accurately reflect the physical processes of the model, a number of additions should be considered. First, a realistic set of initial conditions for H and S are needed, involving perhaps some sort of historical average. We should also overlay a geostrophic current field, both initially (the model is presently considered at rest), and at each time level.



As noted in the previous section, some method of more accurately reflecting behavior on the open-ocean boundaries is required, particularly at the intersection of the coastal boundary. Last, the model equations should include advection current terms and wind forcing, and wind-induced mixing should be added to  $W$ , the entrainment term. The effects of these changes on the numerical methods for the simplified model would have to be determined.

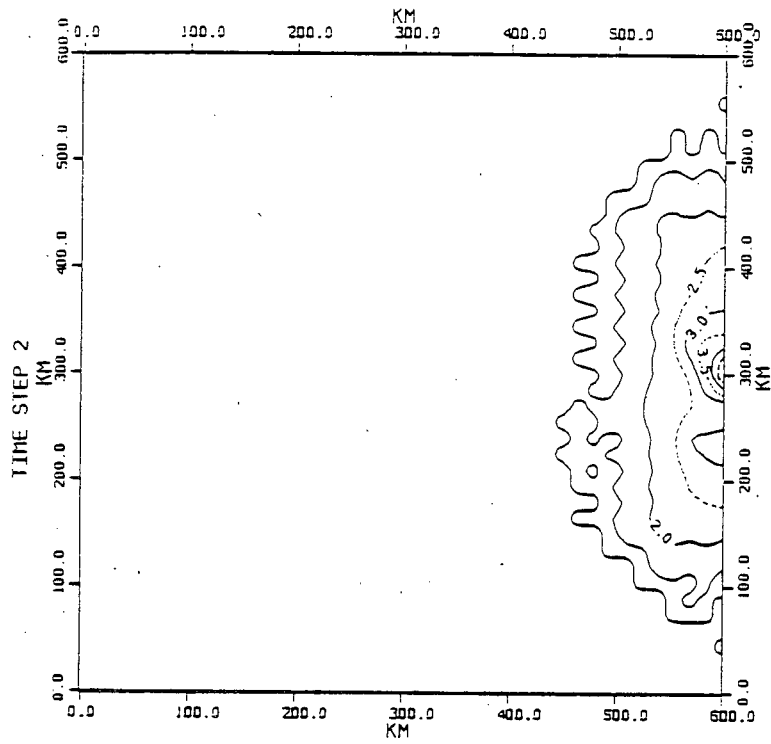


Figure 3 - Test 1: S at time step 2

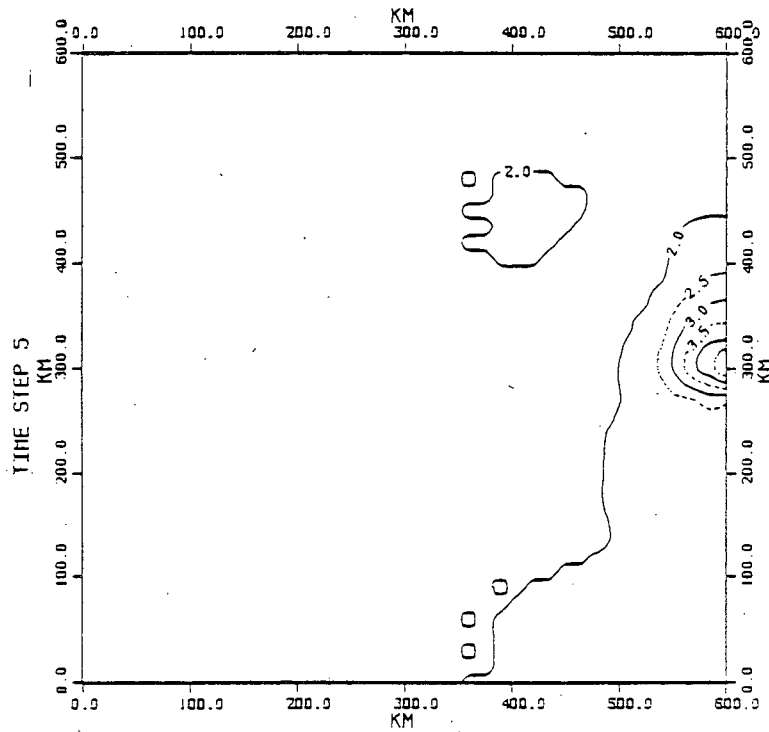


Figure 4 - Test 1: S at time step 5

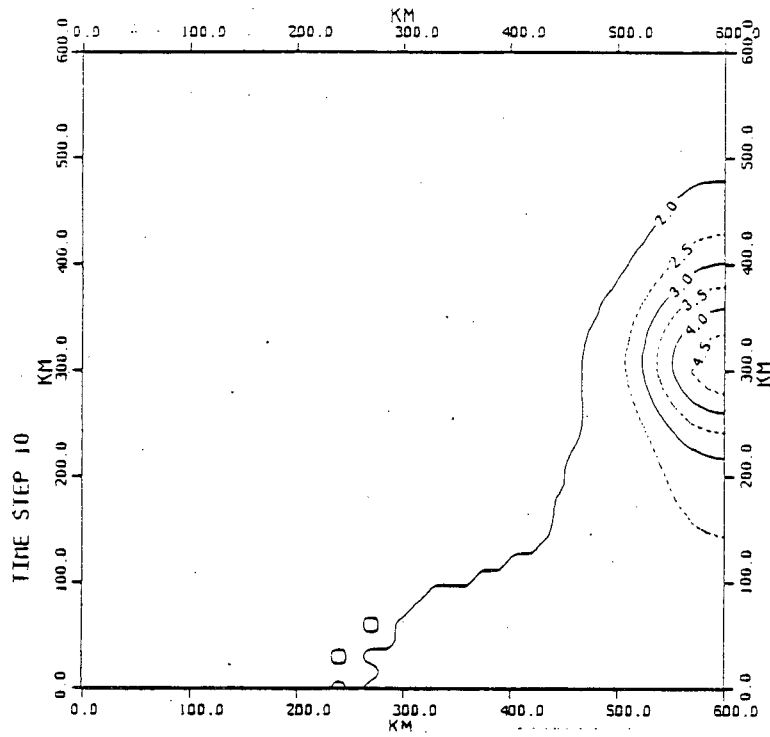


Figure 5 - Test 1: S at time step 10

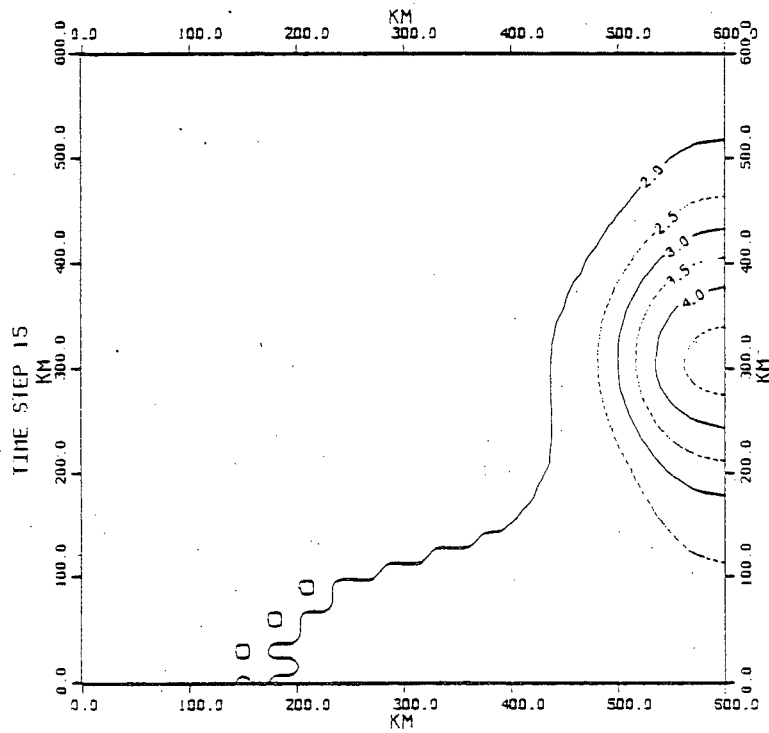


Figure 6 - Test 1: S at time step 15

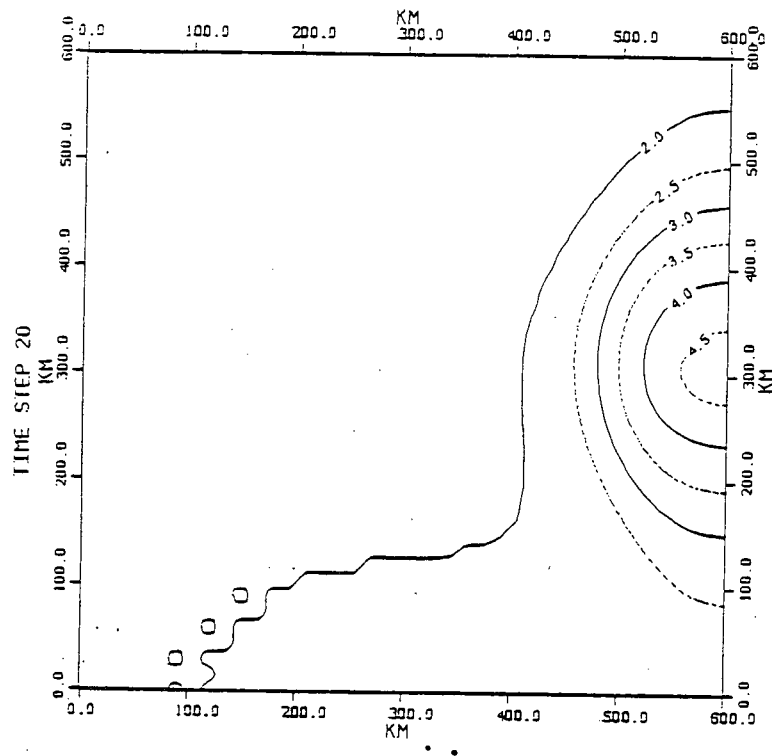


Figure 7 - Test 1: S at time step 20

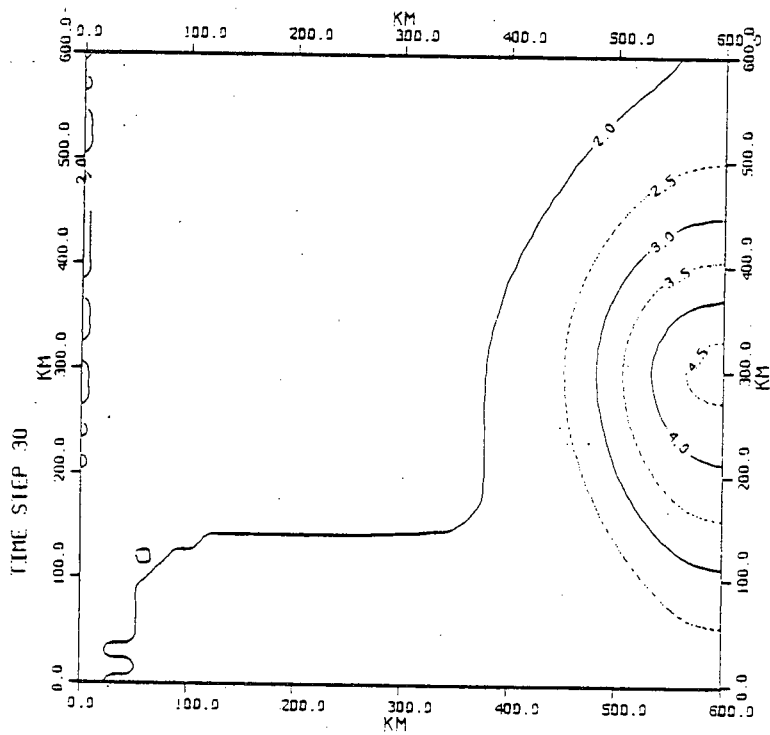


Figure 8 - Test 1: S at time step 30

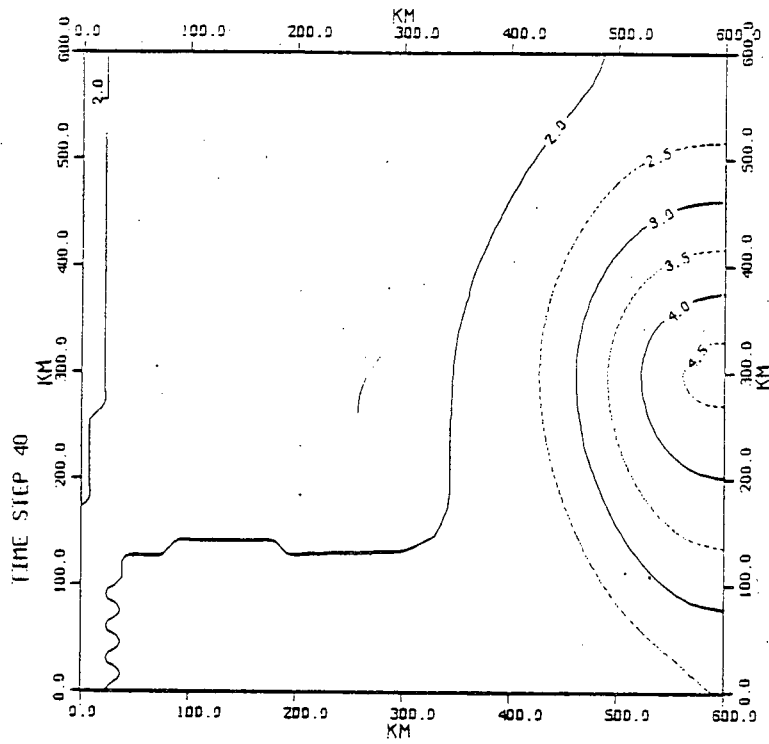


Figure 9 - Test 1: S at time step 40

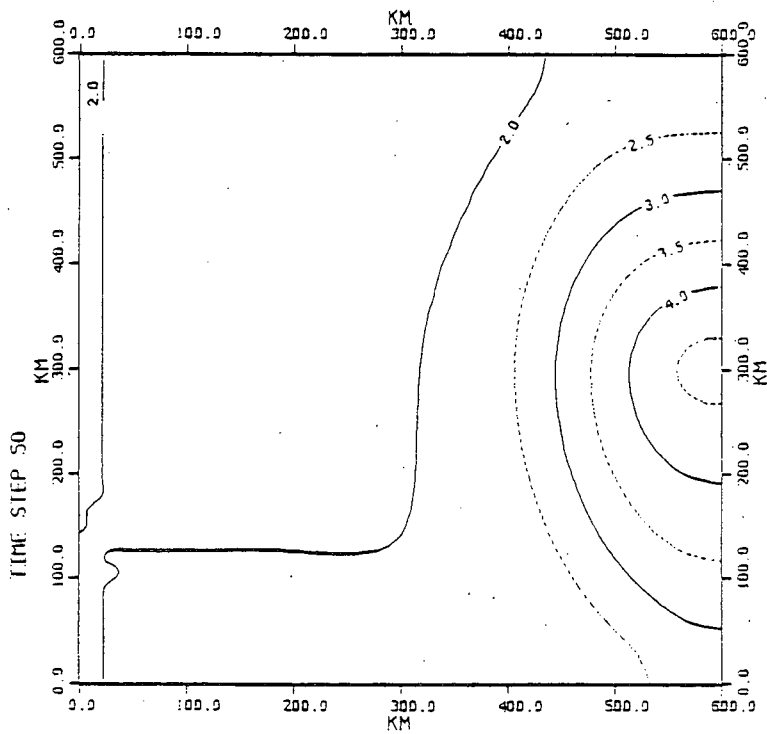


Figure 10 - Test 1: S at time step 50

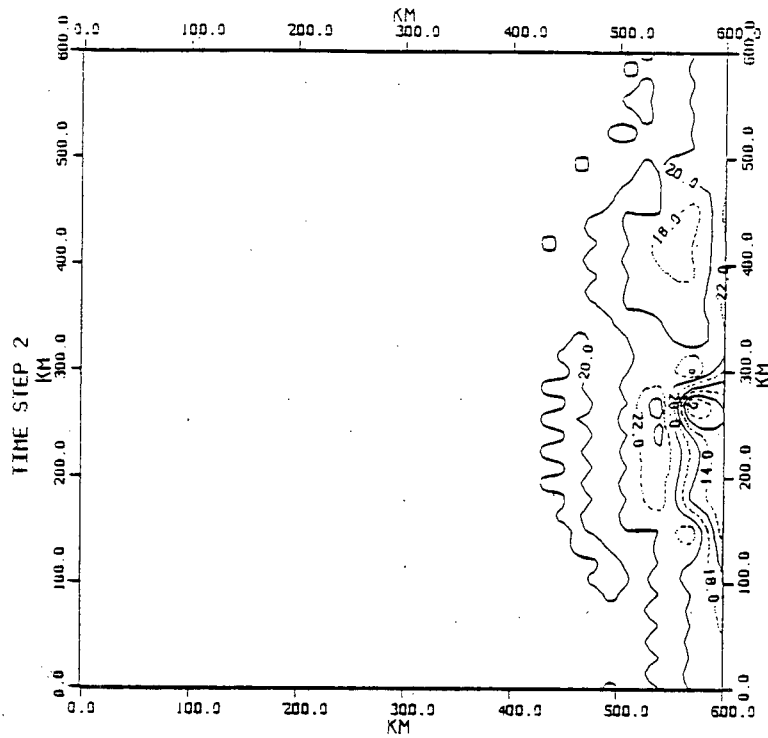


Figure 11 - Test 1: H at time step 2

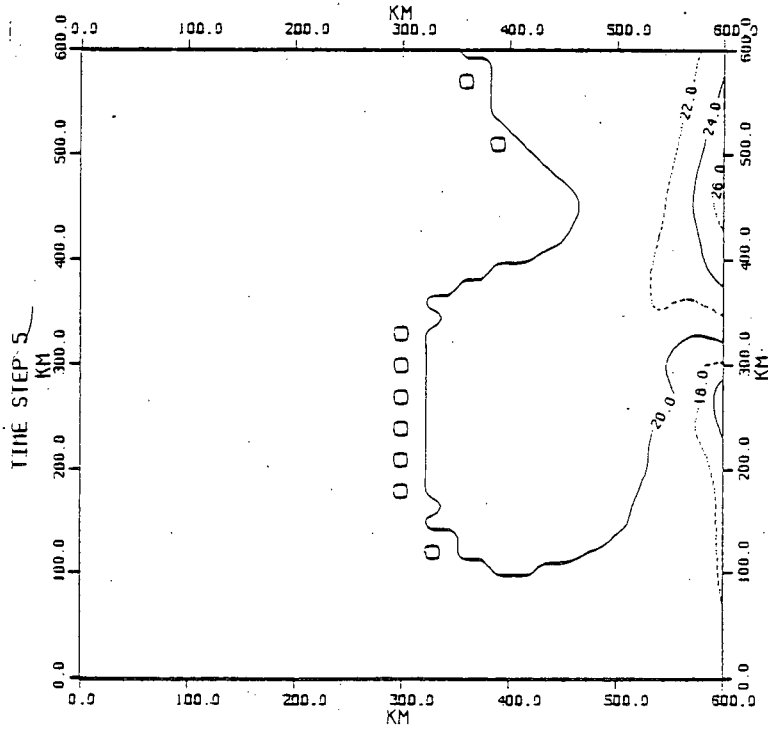


Figure 12 - Test 1: H at time step 5

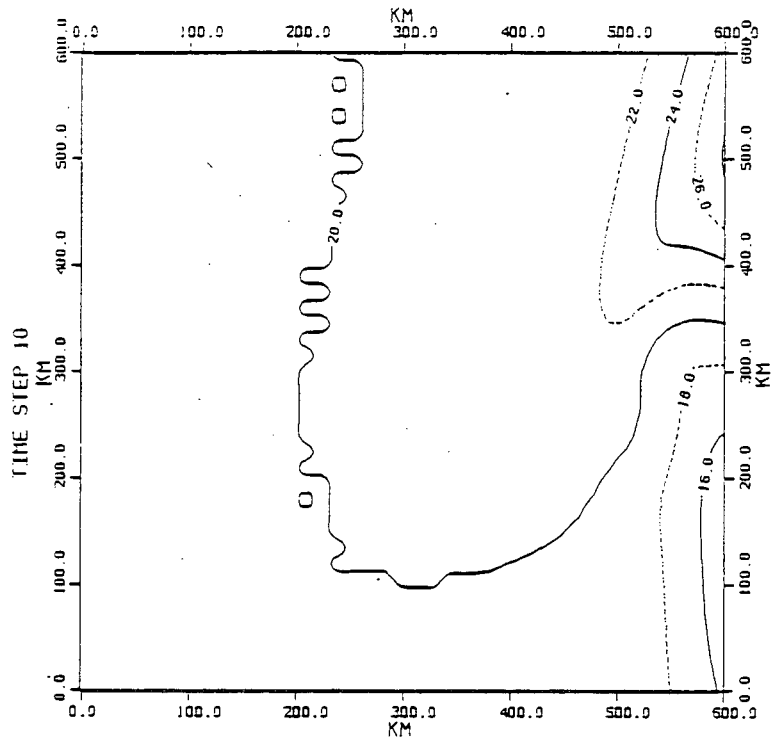


Figure 13 - Test 1: H at time step 10

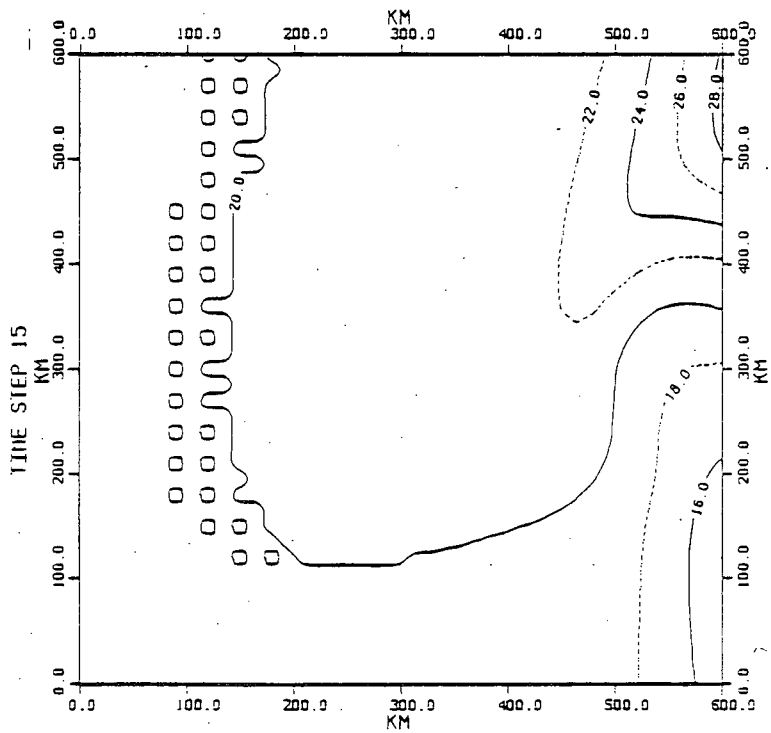


Figure 14 - Test 1: H at time step 15

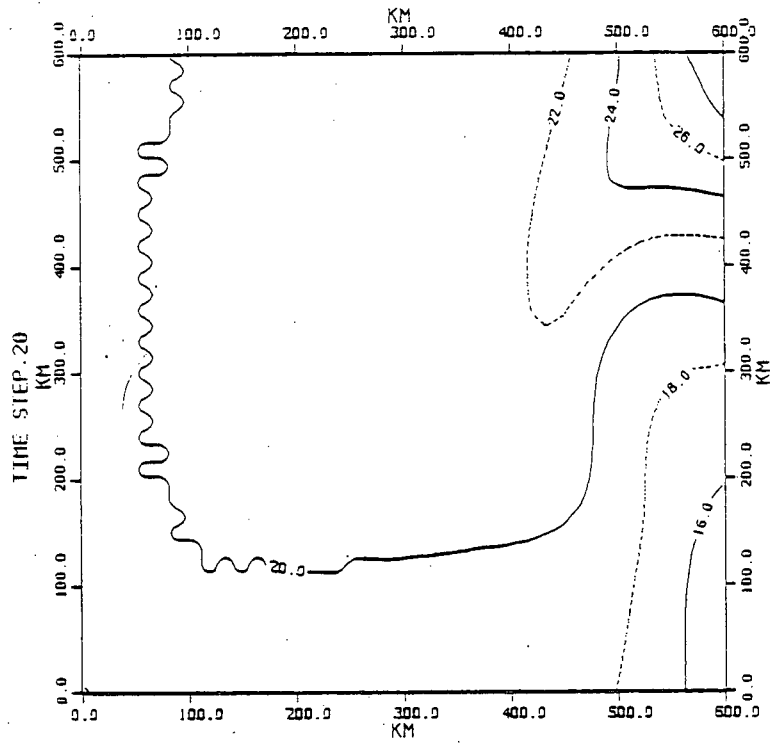


Figure 15 - Test 1: H at time step 20

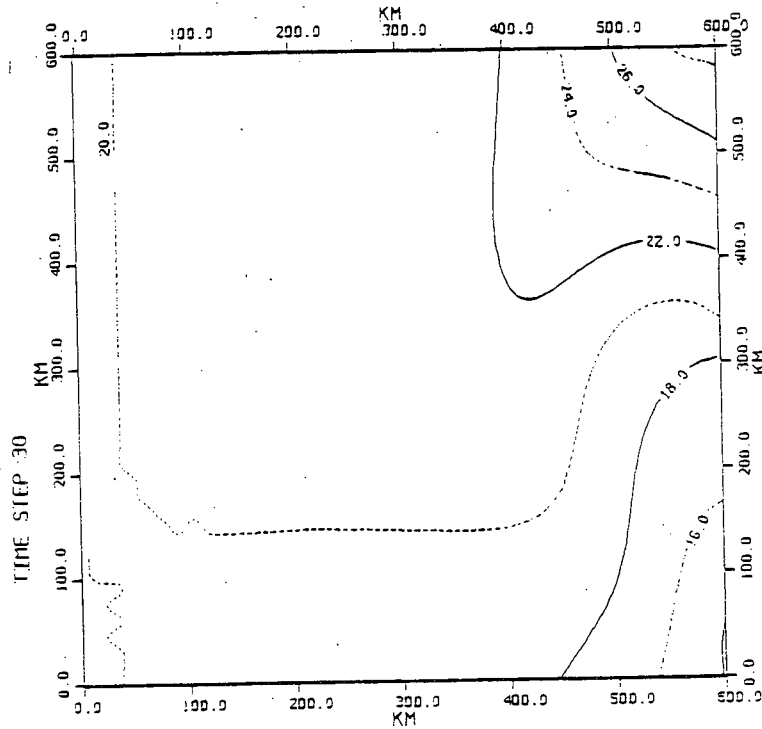


Figure 16 - Test 1: H at time step 30



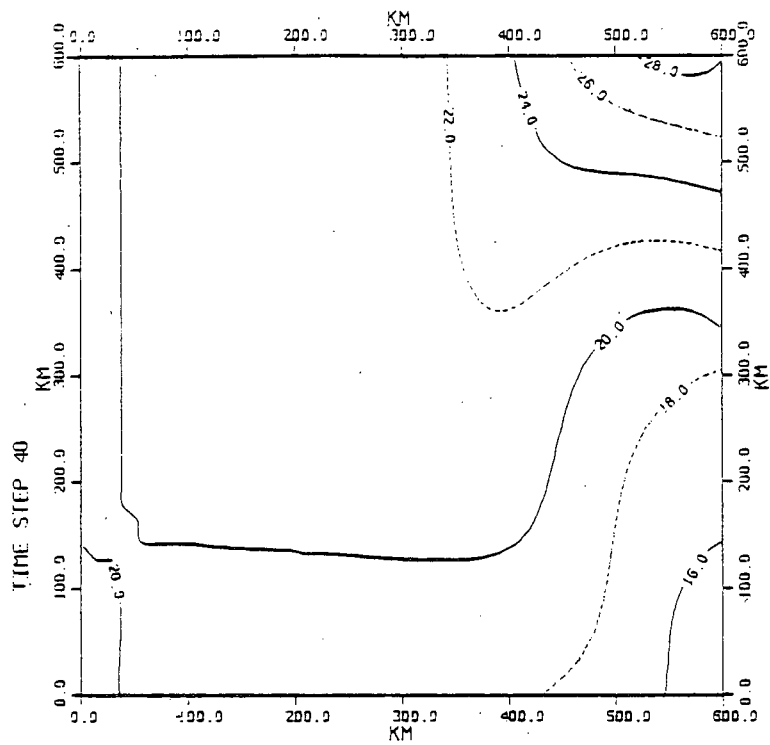


Figure 17 - Test 1: H at time step 40

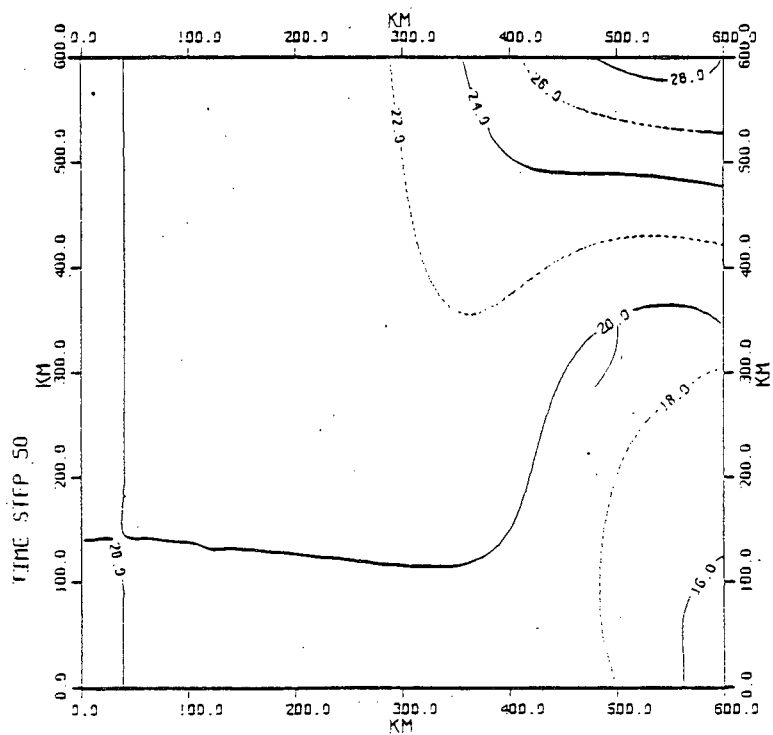


Figure 18 - Test 1: H at time step 50

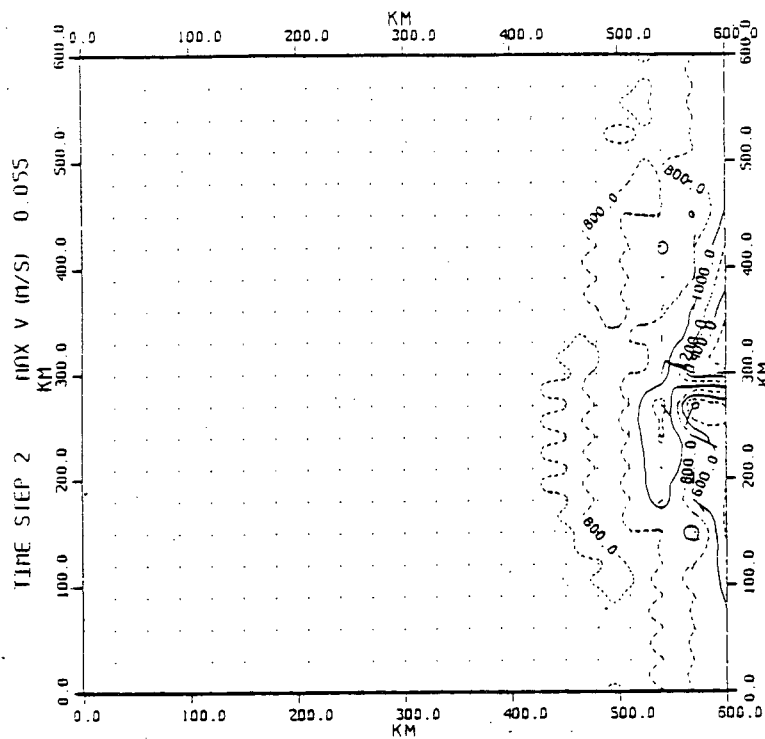


Figure 19 - Test 1:  $H^2S$  at time step 2

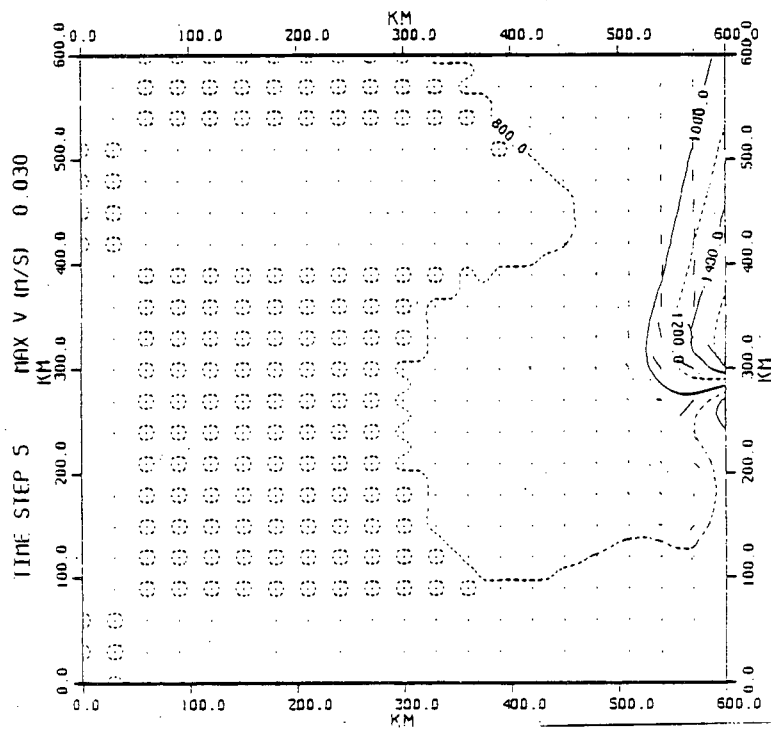


Figure 20 - Test 1:  $H^2S$  at time step 5

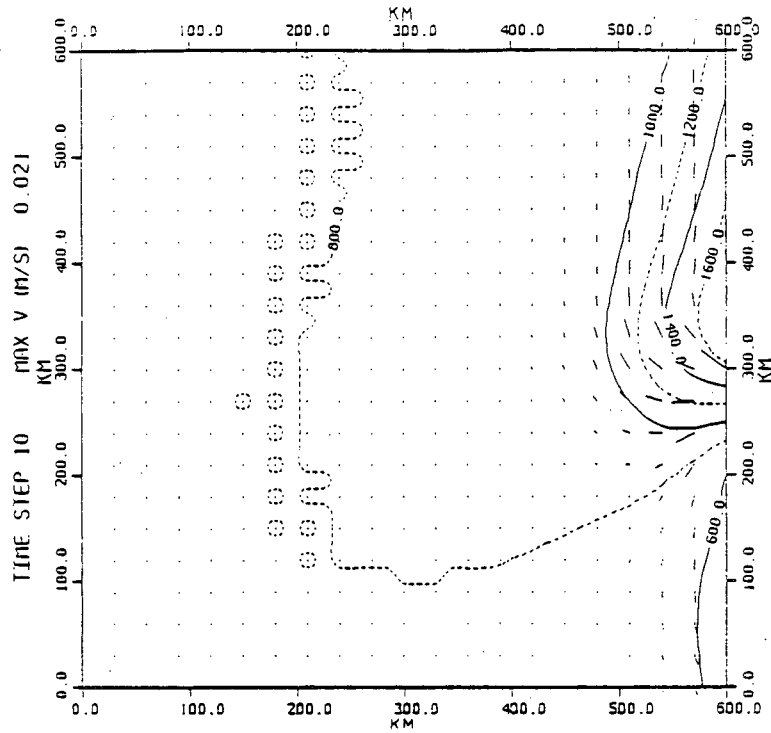


Figure 21 - Test 1:  $H^2S$  at time step 10

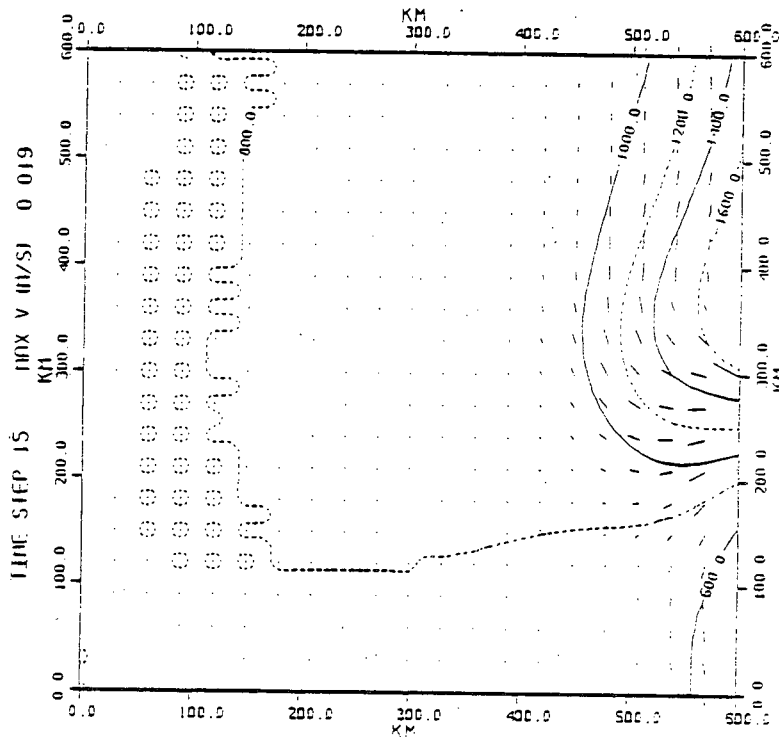


Figure 22 - Test 1:  $H^2S$  at time step 15

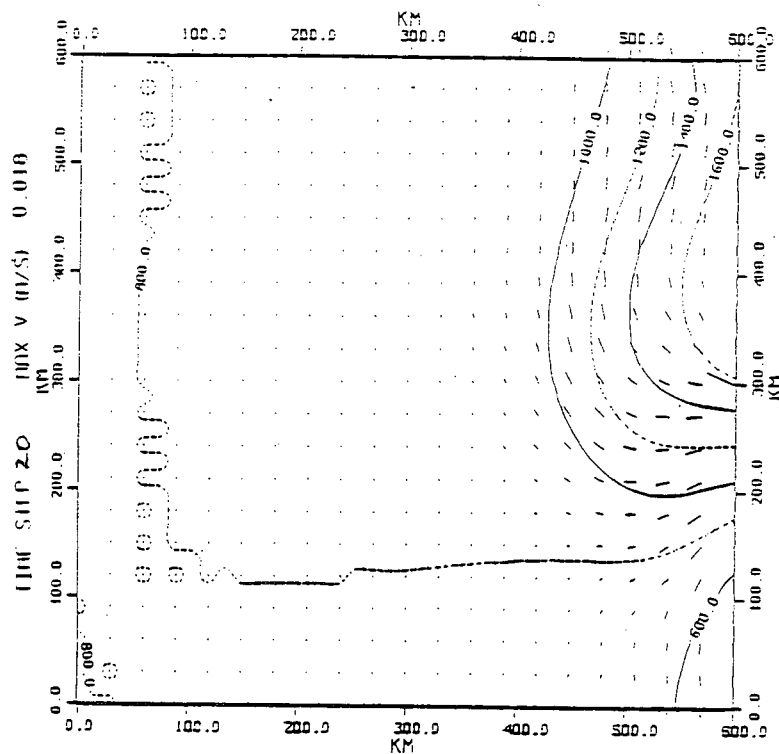


Figure 23 - Test 1:  $H^2S$  at time step 20

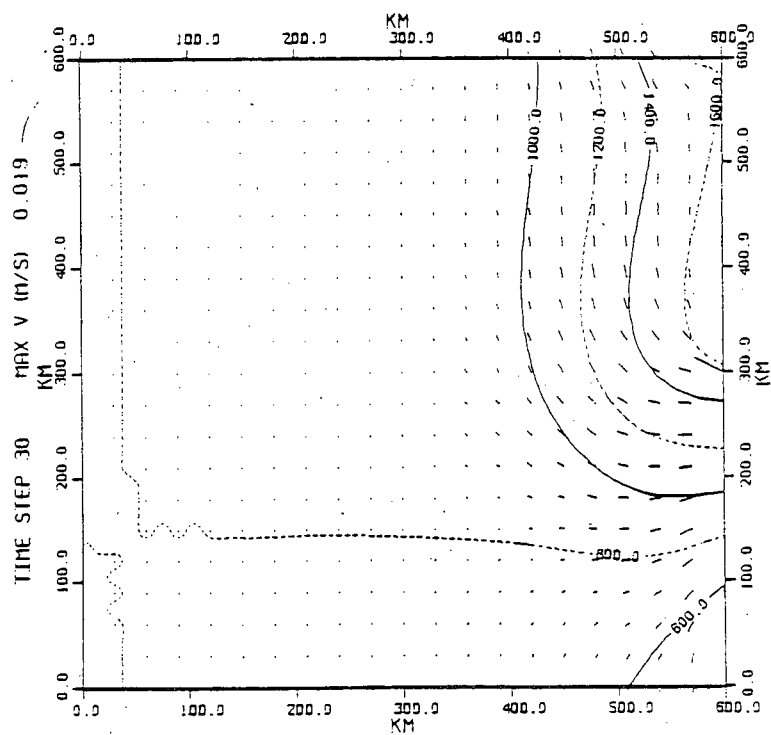


Figure 24 - Test 1:  $H^2S$  at time step 30

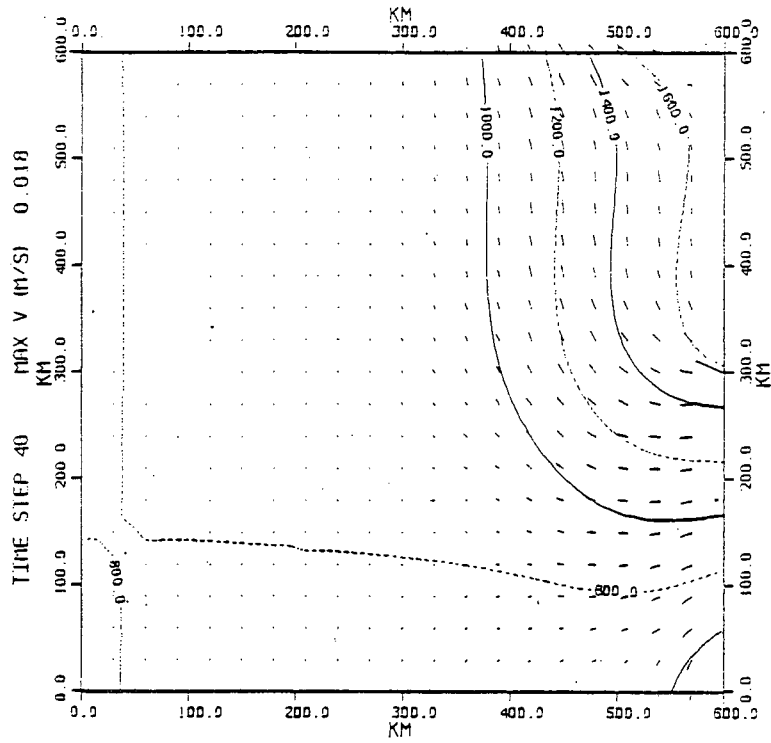


Figure 25 - Test 1:  $H^2S$  at time step 40

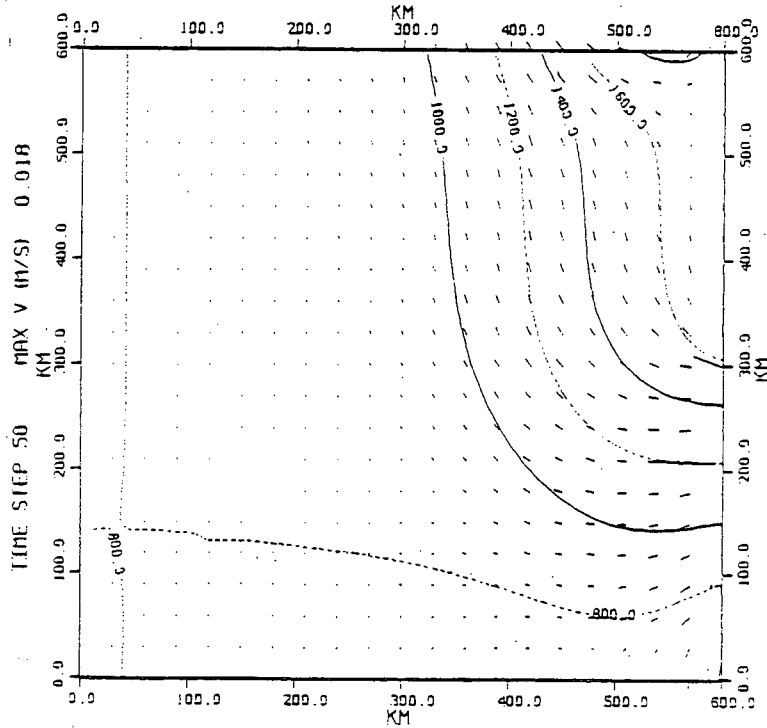


Figure 26 - Test 1:  $H^2S$  at time step 50

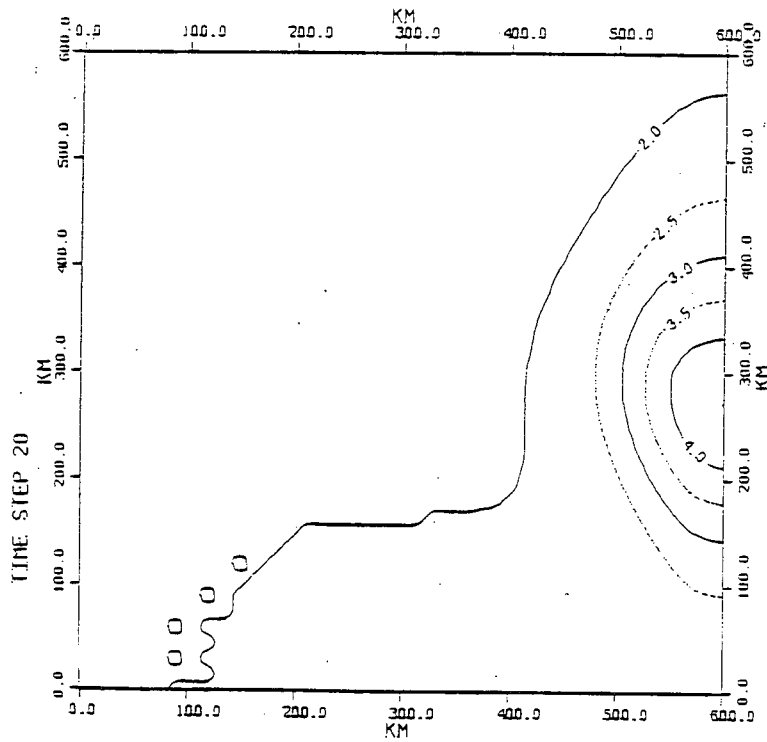


Figure 27 - Test 2: S at time step 20

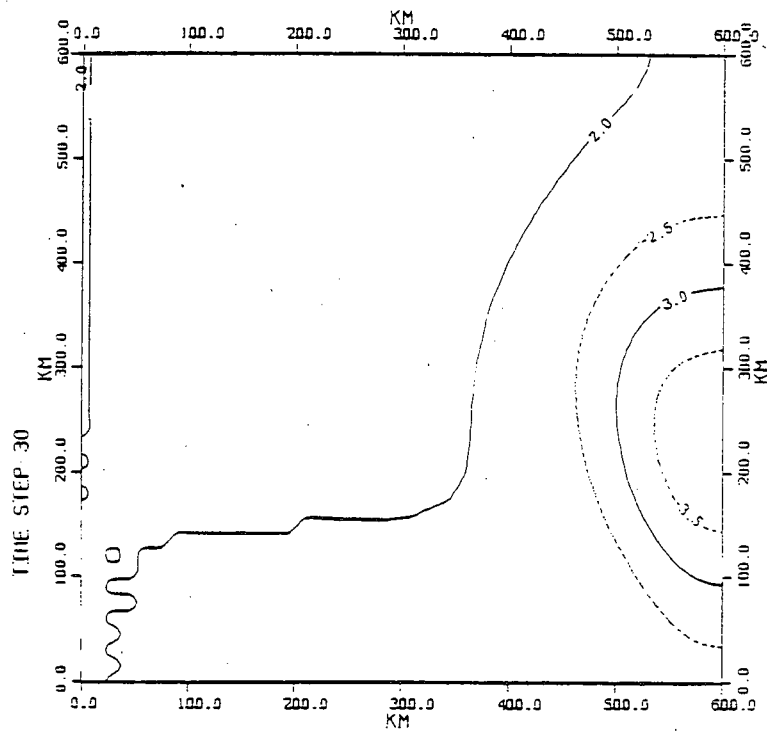


Figure 28 - Test 2: S at time step 30

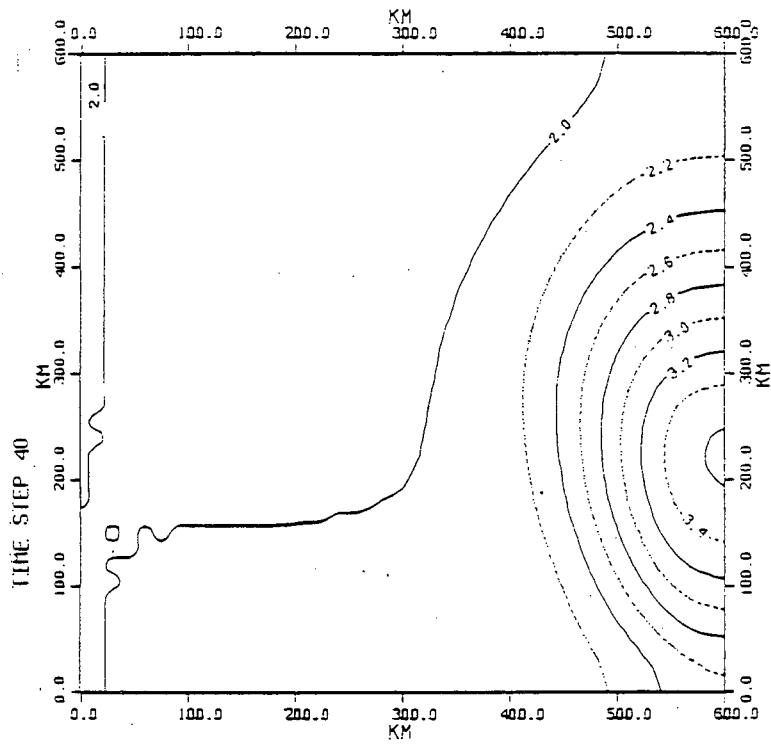


Figure 29 - Test 2: S at time step 40

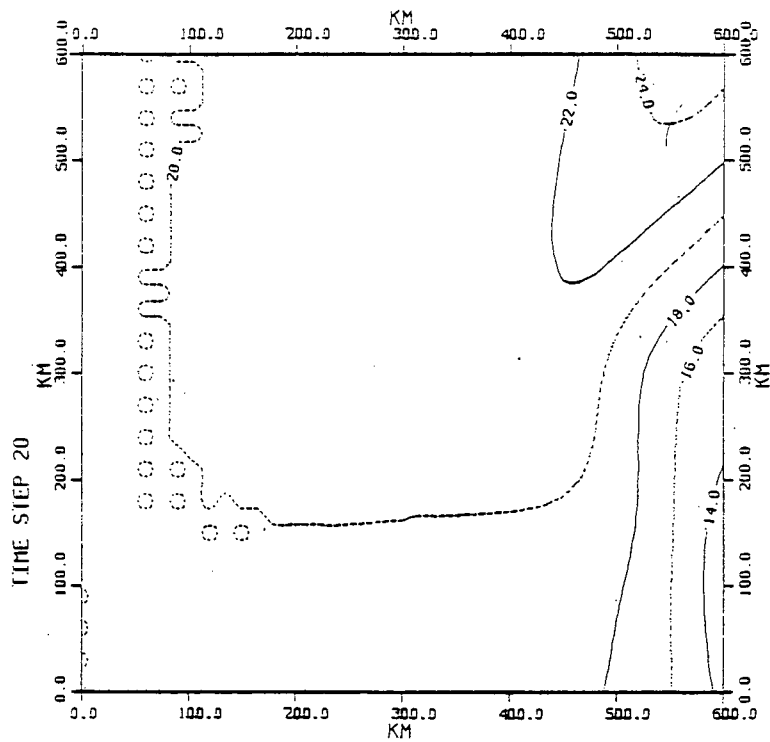


Figure 30 - Test 2: H at time step 20

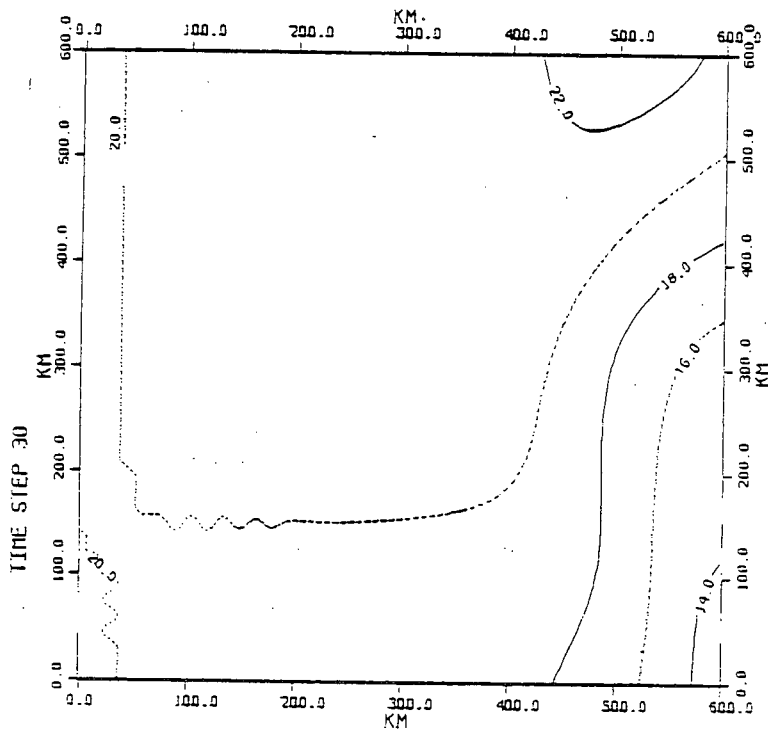


Figure 31 - Test 2: H at time step 30



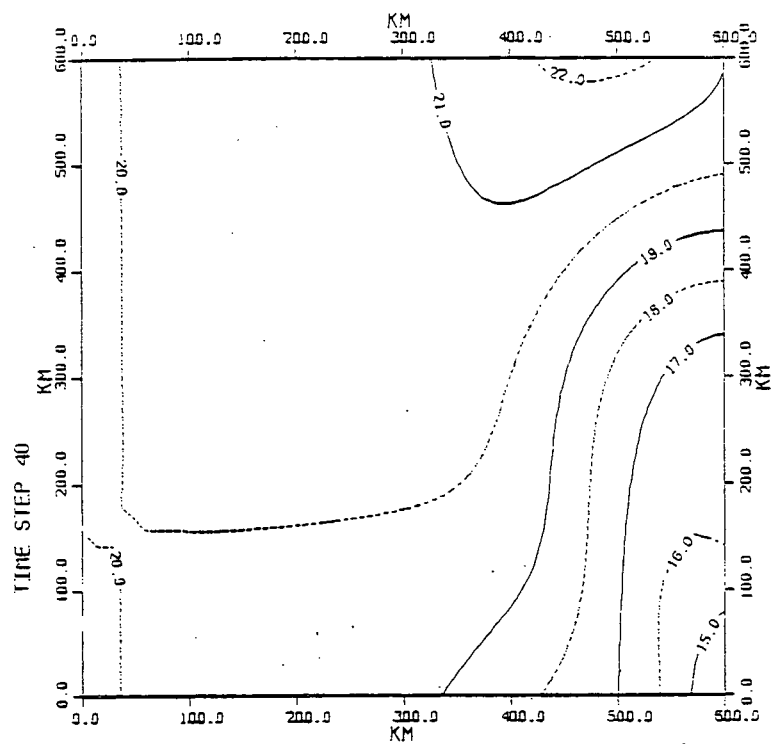


Figure 32 - Test 2: H at time step 40

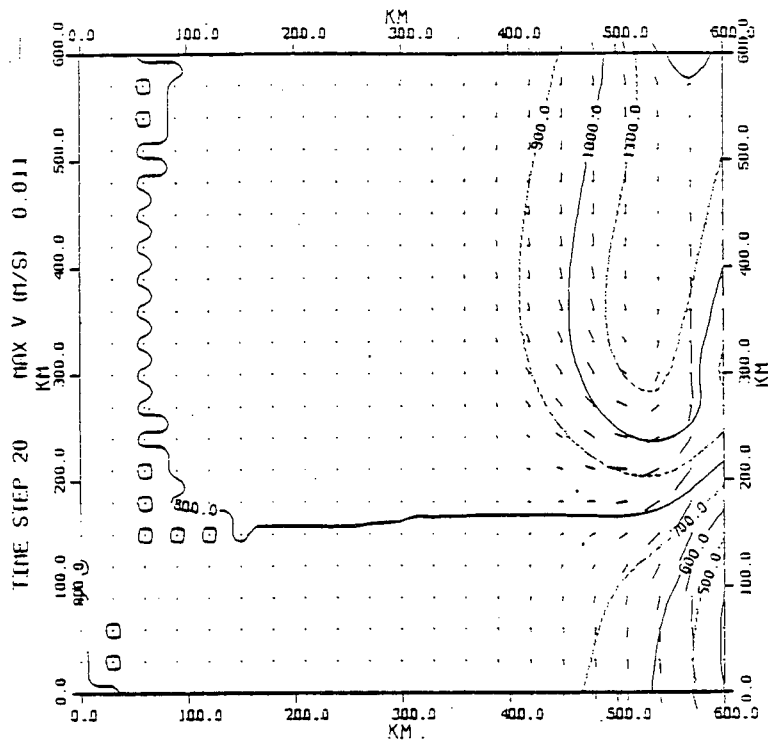


Figure 33 - Test 2:  $H^2S$  at time step 20

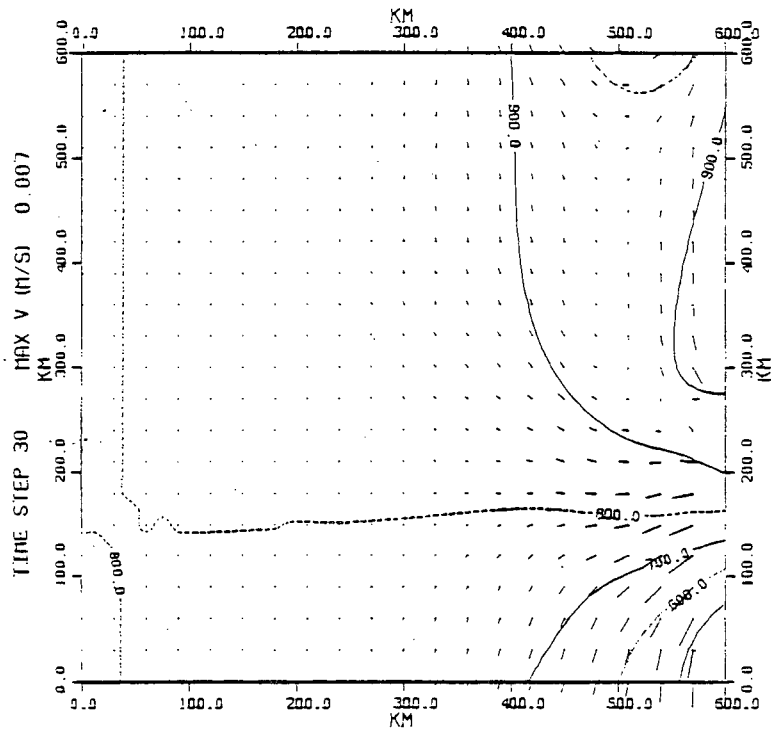


Figure 34 - Test 2:  $H^2S$  at time step 30

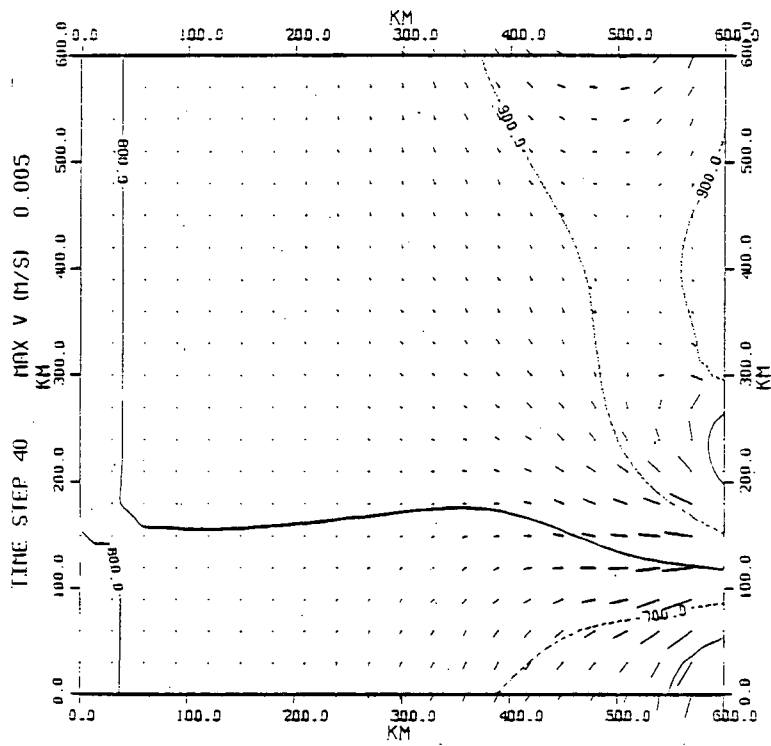


Figure 35 - Test 2:  $H^2S$  at time step 40

BIBLIOGRAPHY

1. Bo Pedersen, F., A Monograph on Turbulent Entrainment and Friction in Two Layer Stratified Flow, Inst. Hydrodynamics and Hydraulic Engineering, Tech. Univ. Denmark, Lyngby. Series Paper No. 25. (1980)
2. Forsythe, G.E. and Wasow, W.R., Finite Difference Methods for Partial Differential Equations, John Wiley & Sons, New York (1960).
3. Gear, C.W., Numerical Initial Value Problems in Ordinary Differential Equations, Prentice-Hall, Englewood Cliffs, N.J. (1971).
4. Gottlieb, D. and Gustafsson, B., Generalized Du Fort-Frankel Methods for Parabolic Initial-Boundary Value Problems, SIAM J. Numer. Anal., 13 (1976) 129-144.
5. Kriess, H. and Oliger, J., Methods of the Approximate Solution of Time Dependent Problems, GARP Publications Series No. 10 (1973).
6. LeBlond, P.H., Emery, W.J., Nicol, T.O., A Model of Runoff Driven Coastal Circulation, Submitted for publication to Coastal, Estuarine and Shelf Science, (1983)
7. LeBlond, P.H., Department of Oceanography, University of British Columbia, Vancouver, B.C., personal communication.
8. Lees, M., An Extrapolated Crank-Nicolson Difference Scheme for Quasilinear Parabolic Equations, Numerical Solution of Partial Differential Equations, Ames, W.F. (Ed.), Barnes & Noble, New York (1969), 193-201.
9. Mitchell, A.R. and Griffiths, D.F., The Finite Difference Method in Partial Differential Equations, John Wiley & Sons, Chichester (1980)
10. Moore, C.F., Integration of Stiff Systems of Ordinary Differential Equations Having a Banded Jacobian, Computing Centre, University of British Columbia (1979).
11. Morris, J.L., On the Numerical Solution of a Heat Equation Associated with a Thermal Print-Head, J. Comp. Phys., 5 (1970), 208-228.
12. Richtmyer, R.D. and Morton, K.W., Difference Methods for Initial-Value Problems, Interscience Publishers, New York (1967).

13. Royer, T.C., On the Effect of Precipitation and Runoff on Coastal Circulation in the Gulf of Alaska, J. Phys. Oceanogr., 9 (1979), 555-563.
14. Royer, T.C., Hansen, D.V. and Paslinski, D.J., Coastal Flow in the Northern Gulf of Alaska as Observed by Dynamic Topography and Satellite-Tracked Drogued Drift Buoys, J. Phys. Oceanogr., 9 (1979), 785-801.
15. Royer, T.C., Coastal Fresh Water Discharge in the Northeast Pacific, J. Geophys. Res., 83, C3 (1982) 2017-2021.
16. Sommeijer, B.P., Van Der Houwen, P.J. and Verwer, J.G., On the Treatment of Time-Dependent Boundary Conditions in Splitting Methods for Parabolic Differential Equations, Int. J. Numer. Meth. Eng. 17 (1981) 335-346.
17. Varah, J.M., Stability of Difference Approximations to the Mixed Initial Boundary Value Problems for Parabolic Systems, SIAM J. Numer. Anal., 8 (1971) 598-615.
18. Varah, J.M., On the Stability of Boundary Conditions for Separable Difference Approximations to Parabolic Equations, SIAM J. Numer. Anal., 14 (1977) 1114-1125.
19. Varah, J.M., Stability Restrictions on Second Order, Three Level Finite Difference Schemes for Parabolic Systems, SIAM J. Numer. Anal., 17 (1980) 300-309.