SOME THREE-DIMENSIONAL COMPUTATIONAL FLUID DYNAMICS PROBLEMS FOR SHALLOW SEAS

by

Yuhe Song

M.Sc., Acadamia Sinica, Beijing, 1984
B.Sc., Zhengzhou University, 1982

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

in the Department
of
Mathematics and Statistics

SIMON FRASER UNIVERSITY

August, 1990

ISBN 0-315-78286-2

APPROVAL

Name: Yuhe Song

Degree: Ph.D. (Applied Mathematics)

Title of Thesis: Some Three-Dimensional Computational Fluid Dynamics Problems For Shallow Seas

Examining Committee:

Chairman: Dr. A. Lachlan

Dr. R.W. Lardner
Senior Supervisor

Dr. M. Trummer

Dr. C.Y. Shen

Dr. G.A.C. Graham

Dr. I.M. Navon
External Examiner
Professor, Department of Mathematics and
Supercomputer Computations Research Institute
Florida State University, Tallahassee, U.S.A.

Date Approved: August 3, 1990

PARTIAL COPYRIGHT LICENSE

Title of Thesis/Project/Extended Essay

SOME THREE-DIMENSIONAL COMPUTATIONAL FLUID DYNAMICS PROBLEMS FOR SHALLOW SEAS

Author:

(signature)

YUHE SONG

(name)

August 15, 90

(date)

ABSTRACT

In this thesis, some numerical modelling methods are developed to compute three-dimensional flows in shallow seas driven by tides,meteorological forcing or density gradient, and the convection-diffusion behaviour of dissolved or suspended substances.

A spectral method employing eddy-viscosity eigenfunctions is used to solve the full three-dimensional nonlinear hydrodynamic equations for the numerical computation of flows. An explicit finite elements method is developed to compute the nonlinear advective terms and an explicit treatment of bottom friction is used. This leads to a rapidly convergent expansion and relatively few eigenfunctions are required to obtain accurate solutions. An Arakawa B-grid is used in the horizontal directions and the eddy-viscosity eigenpaires are computed using the SLEIGN subroutine. Several model problems have been used to test the accuracy, stability and computational efficiency of the methods.

A vertical/horizontal splitting method is used to determine the numerical solutions of the three-dimensional convection-diffusion equation appropriate for the marine environment. This method splits the horizontal and vertical parts of the process, treating the horizontal convection and diffusion explicitly and the vertical convection and diffusion by an implicit finite elements method that is unconditionally stable. The overall stability conditions on this method are investigated and its accuracy is verified through a number of test problems whose exact solutions are known.

ACKNOWLEDGEMENTS

I would like to express my sincere admiration, appreciation and deep gratitude to my supervisor, Dr. R.W. Lardner for his kindly advice, patient guidance and constant support. I consider it an honour and privilege to have the opportunity to work with him these few years.

The greatest debt is to my family, my wife, Hong and my daughter, Shushan, to whom my gratitude can hardly be expressed. Without their support, this thesis would not be completed. I also would like to take this opportunity to thank everyone who contributed in one way or another to the completion of my thesis. Financial support from the Department of Mathematics and Statistics is much appreciated.

Table of Contents

List of Figures

INTRODUCTION

Computational fluid dynamics is the science of producing numerical solutions to a system of partial differential equations which describe fluid flow. It has become more and more important in recent years because computational fluid dynamics is more flexible and more cost-effective than experimental fluid dynamics and because it has become reliable for the simulation of a large variety of flow problems, and provides more detailed and comprehensive information than experimental fluid dynamics[1-2].

Although the era of computational fluid dynamics may be marked as beginning from 1922 [3], the real practice only began in the 1950s when the numerical theory has been developed and the main memory on computers expanded [4]. As computing power rapidly increased over the last decade larger and larger problems have become computationally practicable[1].

Within the area of oceanographic flows, early hydrodynamic numerical models were two-dimensional in that current structure was removed by integrating through the vertical from sea surface to sea bed, obtaining what are called the shallow water equations. These models were primarily used to study changes in sea surface elevation due to tides and meteorological events(storm surge models). A variety of computational methods have been used to obtain numerical solutions of these equations, using finite difference schemes[5-10], finite element schemes[11-16], harmonic analysis in time plus finite elements in space[17-19], the method of characteristics[20-23], and the vectorized computer implementation[2].

In recent years there has been a considerable interest in developing full three-dimensional flow models, since more detailed information about the currents is needed in practice. For example the surface velocity determines the motion of an oil slick and for a wind-driven flow, for instance, the surface velocity differs very much from the depth-averaged flow. Also information on the three-dimensional wind-induced circulation in a sea region is required in pollution problems and detailed information on bottom currents and sediment transport is required in many civil engineering projects[24-26]. For these purposes, some three-dimensional hydrodynamic problems and the associated marine pollutant transport problems have been considered in this thesis.

The first problem we are concerned with is the full three-dimensional nonlinear hydrodynamic equations with arbitrary variable eddy-viscosity for shallow seas. Numerical studies of the three-dimensional motion of the sea under the influence of wind and tide have in recent years been made by several authors using a number of different approaches[27-32]. Some of the most successful of these approaches make use of expansions of the two horizontal components of fluid velocity in terms of a set of basis functions of the vertical coordinate[33-41]. By this means the three-dimensional equations are reduced to a set of two-dimensional modal equations for the coefficients in these velocity expansions.

The use of a basis of "eddy-viscosity eigenfunctions" for this purpose was first proposed by Heaps [33-34] to solve the linearized three-dimensional tidal equations. The significant advantage of this particular basis is that the modal equations are uncoupled. Since Heaps employed analytic eigenfunctions, his use of the method was restricted to problems with simple eddy-viscosity profiles. Subsequently, the method has been extended to more general eddy-viscosities by Davies [35] and Furnes [36].

A more general Galerkin method was developed by Davies and Owen [37] for the linearized model and by Davies [38] for the fully nonlinear equations. In these papers basis sets consisting of cosine functions, Chebychev polynomials and Gram-Schmidt orthogonalized polynomials (equivalent to shifted Legendre polynomials) were used. Davies [39] later combined the Galerkin method with use of a basis set of eddy viscosity eigenfunctions.

The rate of convergence of the expansions in terms of eddy viscosity eigenfunctions was found by Davies [35,39] to be relatively slow, requiring typically 15-20 basis functions to obtain the desired accuracy. Much more rapid convergence is obtained using Chebychev or Legendre polynomials [37] but the disadvantage of these basis sets is that the modal equations are coupled even in the linear approximation. Using eddy viscosity eigenfunctions the equations are coupled only through the nonlinear terms. A later modification of the method that significantly accelerates the convergence was proposed for the linearized equations by Lardner [40]. With this modification it was found in one model problem that only 4-5 eddy-viscosity eigenfunctions were required to give the same accuracy as had previously [35] been obtained with 20-25 eigenfunctions.The rate of convergence is comparable to that obtained using B-splines [41] or Chebychev polynomials [38], and has the advantage over these approaches of uncoupling the linear modal equations.

In the first part of this thesis this modified eddy-viscosity eigenfunction method is extended to the nonlinear hydrodynamical equations. An explicit method is used to represent the nonlinear bottom friction and an explicit finite element method is used for the nonlinear advective terms (compared to the explicit time-splitting technique used by Davies [38]). This treatment of the bottom friction does not add significantly to the CPU requirements of the algorithm, but we have found, as did Davies, that computation of the advective terms is by far the most expensive part of the algorithm, in our case increasing the CPU requirements by a factor of more than 3. The finite element technique is the most effective method we have found for minimizing this cost.

Most of the finite difference schemes that have been developed for hydrodynamical modelling have been based on an Arakawa C-grid, starting with the two-dimensional algorithms of Hansen, Leendertse and others [5, 6] down to the more recent three-dimensional algorithms [26-32] and including the algorithms based on the spectral method [33-41]. While this choice of grid has the advantage of providing natural centred-difference approximations to most of the dominant terms and of minimizing numerical dispersion [42,75-78], it does lead to certain difficulties for some three-dimensional computations.

The first of these [43] is the occurrence of spurious numerical boundary layers unless the Coriolis terms are treated carefully. Since the two horizontal velocity components are computed at different spatial points with a C-grid, it is necessary to average the Coriolis terms in each momentum equation over the four neighbouring points at which the opposite velocity component is computed. Adjacent to a coast, one or more of these four points will actually lie on the coast, and the velocity components at such points are maintained at zero by the usual algorithms. The result is that the four-point average gives an incorrect value for the interior point, and this leads to spurious velocities near the coast.

This difficulty does not arise for two-dimensional models based on the depth-averaged equations, since it is correct within the model to set the depth-averaged velocity components equal to zero at coastal points. Within the approximations of the usual three-dimensional models, however, the velocity profile through the water column is not zero at a coast: the narrow coastal boundary layer in which the flow overturns, accommodating itself to the physical boundary condition of zero normal flow, is not contained within the usual model equations. It is worth noting that the problem does not arise for certain three-dimensional algorithms of splitting type [46-47] in which the C-grid is used only for the depth-averaged equations, and the vertical profiles are computed for both velocity components at the same horizontal grid-points.

It has been pointed out by Jamart and Ozer [42] that for algorithms using a spectral method this problem may be overcome by averaging the Coriolis terms only over the adjacent points that are interior to the water body (called the "wet points only" method). While this method is successful, it has the disadvantage of reducing the order of the local truncation error at near coastal points, and this may produce a serious loss of accuracy for a region with an intricate coastline, and consequently a large proportion of near-coastal points.

A second, and probably more serious, disadvantage of the C-grid that arises for spectral methods is that it is necessary to use the same basis functions at all grid points if the four-point average for the Coriolis terms is to give reasonably simple modal equations. The most efficient choice for the basis functions is to use eddy-viscosity eigenfunctions [33,34,36], since the modal equations are then de-coupled. These are independent of position only if the eddy viscosity function has the same vertical profile at all points, apart from an overall scaling factor, and this is a severe restriction for a water body with widely varying parameters such as depth or bottom roughness. If this condition on the eddy viscosity is not satisfied, the Coriolis terms couple the modal equations.

Because of these problems, it appears worthwhile to consider using an alternative spatial grid for which both horizontal velocity components are computed at the same grid points. Of the five grid types described by Arakawa and Lamb [42], those labelled A,B and E satisfy this requirement, so will certainly be simple to use when the eddy viscosity has arbitrary spatial variation. Since the two velocity components are computed at the same point, an additional benefit is that the two momentum equations can be solved simultaneously rather than sequentially as is necessary for the C-grid. This allows explicit treatment of the Coriolis terms to be easily avoided.

The A,B and E-grids lead to numerical schemes for the shallow water equations that have worse numerical dispersion than does the C-grid, particularly at wavelengths shorter than four grid-lengths. However, the hydrodynamical models with which we are concerned involve significant damping, through eddy viscosity and bottom friction, and to the extent we have tested these alternative grids, this does appear to be sufficient that the anomalous dispersion at short wavelengths does not cause difficulties. Arakawa's D-grid has both disadvantages of poor dispersion properties and velocity components at different points, so we have not considered it.

In **chapter 1** the full three-dimensional nonlinear hydrodynamic equations are derived from the physical equations. Some appropriate boundary conditions and initial values are specified. In **chapter 2** the modified spectral method is developed for the general equations. In **chapter 3** the appropriate numerical schemes for the A, B, C and E-grids are developed for the basic equations described in chapter 2. The different schemes are compared using two test problems. In **chapter 4** numerical results are given for a number of problems designed to test the accuracy of the algorithm and bounds on its stability are determined.

The second problem we are concerned with is the density gradient driven flows. The density gradients in near-coastal seas are established by such processes as evaporation, fresh-water run-off and precipitation. While the magnitude of these currents is usually quite small, compared to the currents driven by tides or meteorological forcing, they can be a dominant factor in phenomena such as pollutant transport where long-time drift is concerned.

An estimate of the density-driven currents was made for the Arabian Gulf by Lardner and co-workers[48] using a simple two-layer, sectionally integrated model, similar to that constructed by Pearson and Winter[49] for the flow in fjords. Later, Lardner and Das[50] developed an algorithm based on a splitting method to investigate density-driven currents and applied it to the Arabian Gulf. In this method the surface elevation and vertically integrated mass transports are computed from the depth-averaged equations, then the vertical structure of the currents is obtained from the horizontal momentum equation at each time-step.

In **chapter 5** the spectral method will be used to solve the three-dimensional hydrodynamic equations with a variable water density. The eddy-viscosity eigenfunctions are used again as a basis of the eigenexpansion, and the SLEIGN code is used to solve the eigenvalue problems at every horizontal points. The accuracy of the method is tested on some problems for which the exact steady-state solutions can be calculated. We found that the results are more accurate than those obtained by the splitting method used by Lardner and Das[50].

The third problem we are concerned with is the three-dimensional pollutant transport behaviour in shallow seas. Several numerical methods are available for the solution of the two-dimensional depth-integrated form of the convection-diffusion equation. The most popular finite-difference methods employed to overcome convection-dominated difficulties are the upwind and flux-corrected schemes [51,52]. Finite differences or finite elements combined with the characteristic Galerkin method have also been used [53-55]. Another widely used method for this kind of two-dimensional problem is the split-operator approach in which the convection and diffusion are solved separately by two different numerical methods [56-59]. But all of these methods have difficulty in solving three-dimensional problems, either because of expensive matrix inversions at each time-step or because of a time-step restricted by the vertical grid, which, for flows in near-coastal seas is two or three orders of magnitude smaller than the horizontal grid.

For many purposes, however, it is necessary to compute solutions of the full three-dimensional equations. For example, permits allowing discharge of effluent from industrial installations may place restrictions on the density of effluent in the surface layers of the water column but not at lower levels. The algorithm we shall discuss in this paper is intermediate between the two extremes of explicit and implicit schemes mentioned above. It is an extension of a vertical horizontal splitting algorithm [46,47] that has been developed for solution of the hydrodynamical equations for shallow seas. The basis of this approach is to treat explicitly the terms in the convection-diffusion equation that involve horizontal derivatives while the terms involving vertical derivatives are treated by an implicit finite element algorithm similar to that in [47]. This latter part of the algorithm is shown to be stable under not too restrictive conditions, so that the major stability restriction comes from the horizontal part.

In **chapter 6** the three-dimensional convection-diffusion equation will be recast in a form suitable for the numerical method. The numerical vertical / horizontal splitting algorithm is described and some stability properties are also discussed. In **chapter 7** details of some numerical experiments are given. A comparison is also given with the Monte-Carlo type of algorithm[60], which offers probably the most viable alternative for this type of problem.

Chapter 1. Governing Equations

1.1 *Physical equations*

We use xyz as Cartesian coordinates to describe a three-dimensional sea model with the z - axis pointing vertically upwards and the xy - plane occupying the undisturbed position of the water surface. The position of the bottom is taken to be $z = -h(x, y)$ while the surface at time t is $z = \zeta(x, y, t)$ (see Figure 1.1).

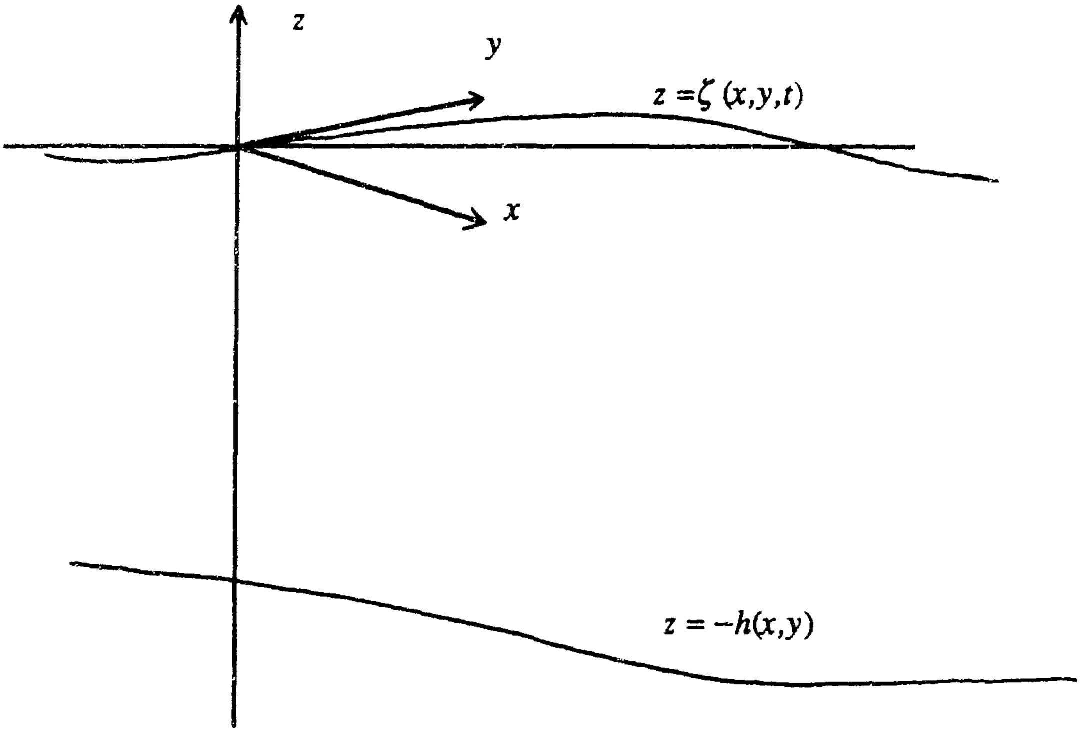


figure 1.1 shallow sea model

We first consider a viscous incompressible homogeneous fluid, the water density ρ is assumed constant with respect to both position and time (the variable density problem will be considered in chapter 5). The governing equations are the continuity equation and the three momentum equations, together with appropriate boundary and initial conditions (for the details, see [61-62]).

The continuity equation is

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0 \tag{1.1}$$

The horizontal momentum equations are

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}=fv+\frac{1}{\rho}\left\{-\frac{\partial p}{\partial x}+\frac{\partial \tau_{xx}}{\partial x}+\frac{\partial \tau_{xy}}{\partial y}+\frac{\partial \tau_{xz}}{\partial z}\right\} \tag{1.2}$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+w\frac{\partial v}{\partial z}=-fu+\frac{1}{\rho}\left\{-\frac{\partial p}{\partial y}+\frac{\partial \tau_{yx}}{\partial x}+\frac{\partial \tau_{yy}}{\partial y}+\frac{\partial \tau_{yz}}{\partial z}\right\} \tag{1.3}$$

The vertical momentum equation is usually approximated by the hydrostatic pressure

$$\frac{\partial p}{\partial t}+ g\rho =0 \tag{1.4}$$

The kinematical conditions at the sea surface and sea bed are

$$\frac{\partial \zeta}{\partial t}+u\frac{\partial \zeta}{\partial x}+v\frac{\partial \zeta}{\partial y}-w=0 \qquad on \;\; z=\zeta \tag{1.5}$$

$$u\frac{\partial h}{\partial x}+v\frac{\partial h}{\partial y}+w=0 \qquad on \;\; z=-h \tag{1.6}$$

The notation used in these equations is as follows:

$h\,(x, y\,)$	water depth
$\zeta\,(x, y, t\,)$	surface elevation at time t
$u\,(x, y, z, t\,), v\,(x, y, z, t\,)$	horizontal velocity components
$w\,(x, y, z, t\,)$	vertical velocity component

$p\ (x, y, z,t)$	pressure in water
g	acceleration due to gravity
$\tau_{xx}, \tau_{xy}, \tau_{xz}$	the stress tensor in x-direction
$\tau_{yx}, \tau_{yy}, \tau_{yz}$	the stress tensor in y-direction
$f = 2\Omega \sin\theta$	coriolis parameter

where Ω is the earth's angular rotation speed and θ is latitude.

In the horizontal momentum equations it is usual to neglect the terms involving $\tau_{xx}, \tau_{xy}, \tau_{yy}$ and to make an eddy-viscosity assumption about τ_{xz} and τ_{yz} :

$$\tau_{xz} = \rho\mu \frac{\partial u}{\partial z}, \qquad \tau_{yz} = \rho\mu \frac{\partial v}{\partial z} \tag{1.7}$$

where μ is the eddy viscosity. By integrating the vertical momentum equation from z to the water surface, we have

$$\begin{aligned} p(x,y,z,t) &= p_s(x,y,t) + \int_z^\zeta g\rho dz \\ &= p_s(x,y,t) + g\rho[\zeta(x,y,t) - z] \end{aligned} \tag{1.8}$$

Where p_S is the atmospheric pressure on the water surface. By differentiating (1.8) with respect to x and y, we have

$$\frac{\partial p}{\partial x} = \frac{\partial p_s}{\partial x} + g\rho \frac{\partial \zeta}{\partial x}, \qquad \frac{\partial p}{\partial y} = \frac{\partial p_s}{\partial y} + g\rho \frac{\partial \zeta}{\partial y} \tag{1.9}$$

Substituting equations (1.7) and (1.9) into the horizontal momentum equations, we combine equations (1.2), (1.3) and (1.4) into following two equations:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} - fv = -g\frac{\partial \zeta}{\partial x} - \frac{1}{\rho}\frac{\partial p_s}{\partial x} + \frac{\partial}{\partial z}\left(\mu\frac{\partial u}{\partial z}\right) \tag{1.10}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} + fu = -g\frac{\partial \zeta}{\partial y} - \frac{1}{\rho}\frac{\partial p_s}{\partial y} + \frac{\partial}{\partial z}\left(\mu\frac{\partial v}{\partial z}\right) \tag{1.11}$$

By integrating the continuity equation with respect to z from - h to ζ, we have

$$\int_{-h}^{\zeta}\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)dz+[w]_{-h}^{\zeta}=0$$

We use the Leibnitz formula:

$$\frac{\partial}{\partial x}\int_{a(x)}^{b(x)}c(x,z)dz=\int_a^b\frac{\partial c}{\partial x}dz+\frac{\partial b}{\partial x}c(x,b)-\frac{\partial a}{\partial x}c(x,a)$$

and get

$$\frac{\partial}{\partial x}\int_{-h}^{\zeta}udz+\frac{\partial}{\partial y}\int_{-h}^{\zeta}vdz+\left[w-\frac{\partial\zeta}{\partial x}u-\frac{\partial\zeta}{\partial y}v\right]_{z=\zeta}$$
$$-\left[w+\frac{\partial h}{\partial x}u+\frac{\partial h}{\partial y}v\right]_{z=-h}=0$$

By using the kinematical conditions (1.5) and (1.6) we obtain

$$\frac{\partial\zeta}{\partial t}+\frac{\partial p}{\partial x}+\frac{\partial q}{\partial y}=0 \tag{1.12}$$

where

$$p=\int_{-h}^{\zeta}u\,dz,\qquad q=\int_{-h}^{\zeta}v\,dz \tag{1.13}$$

are called the volume transports.

1.2 *Boundary-Initial Conditions*

In order to solve equations (1.10)–(1.13) for u, v, w and ζ, some boundary conditions and initial values have to be specified. We first give the boundary conditions at the sea surface and the sea bed.

The surface conditions, evaluated at $z = \zeta$, are

$$\rho\mu\frac{\partial u}{\partial z} = \tau_x^s, \qquad \rho\mu\frac{\partial v}{\partial z} = \tau_y^s \tag{1.14}$$

where ρ is the fluid density and τ_x^s τ_y^s the components of surface stress. For wind-induced flows, they are the components of wind stress acting on the free surface in the x and y directions, while for density gradient driven flows the surface stress is taken to be zero. Correspondingly, at the sea bed, $z = -h$, the conditions are taken to be

$$\rho\mu\frac{\partial u}{\partial z} = \tau_x{}^b, \qquad \rho\mu\frac{\partial v}{\partial z} = \tau_y{}^b \tag{1.15}$$

where τ_x^b, τ_y^b denote the x, y components of bottom stress.

In a nonlinear model, it is appropriate to use a quadratic formulation of bottom stress, thus

$$\tau_x^b = \kappa\rho\sqrt{u_b^2 + v_b^2}\, u_b\,, \qquad \tau_y^b = \kappa\rho\sqrt{u_b^2 + v_b^2}\, v_b \tag{1.16}$$

Where κ is a constant coefficient of bottom friction and u_b, v_b are the components of fluid velocity at sea bed.

In many problems, a linear model is suitable, in which case bottom friction can be linearized [63], giving the linear equivalent of (1.16), namely,

$$\tau_x^b = \kappa\rho u_b\,, \qquad \tau_y^b = \kappa\rho v_b \tag{1.17}$$

where κ is an appropriate coefficient of linear bottom friction.

An alternative to a slip bottom boundary condition is the application of a no-slip condition at sea bed, in which case

$$u_b = v_b = 0 \tag{1.18}$$

Physically, however, this condition can only be used if the bottom boundary layer is modelled through an appropriate choice of eddy-viscosity function.

In general, we write the bottom conditions in a form

$$\mu\frac{\partial u}{\partial z} = \left[\kappa_1 + \kappa_2\sqrt{u_b^{\,2} + v_b^{\,2}}\right]u_b, \qquad \mu\frac{\partial v}{\partial z} = \left[\kappa_1 + \kappa_2\sqrt{u_b^{\,2} + v_b^{\,2}}\right]v_b \tag{1.19}$$

where κ_1 and κ_2 are the coefficients of linear and quadratic bottom friction respectively. A no-slip condition on the bottom is obtained in the limiting case κ_1 and $\kappa_2 \to \infty$.

When a bounded flow field is considered, lateral boundary conditions also need to be specified. There are two kinds of such boundary conditions. A closed boundary condition is applied on coastal boundaries, while an open boundary condition is applied on parts of the boundary adjacent to another body of water.

On the closed part of the boundary, the usual condition is assumed to be

$$(p,q)\mathbf{n}=0 \tag{1.20}$$

where **n** is the outward normal vector to the closed boundary. Physically, condition (1.20) describes that there is no mass flow through the closed boundary.

On the open boundary, the most commonly used condition is to assume that the surface level ζ takes prescribed values. The data needed for this condition are usually obtained from measurements or from a larger model which encloses the model at hand. In practice, it appears to be more difficult to measure accurately the velocity than the elevation. As a consequence velocity data are mainly used for the boundary conditions if the model at hand is nested in a larger model[2].

Finally, we specify the initial values by assuming the motion starts from some given values, $(u,v)=(u_o,v_o)$, $\zeta=\zeta_o$ at t=0. In practice, this is taken to be state of rest, that is

$$u=v=0, \qquad \zeta=0 \quad \text{at } t=0 \tag{1.21}$$

1.3 *Sigma Coordinate Equations*

The variable z in physical space has a range $-h(x,y) \le z \le \zeta(x,y,t)$ that varies with horizontal position and time. For numerical work it is better to change to a new variable that has a fixed range. The usual transformation is

$$\sigma = (a+b)\frac{z+h}{H} + a\,, \qquad H = z + \zeta \tag{1.22}$$

which transforms the variable vertical interval into a constant region a to b. The constants a and b are chosen to correspond to the interval over which the numerical methods are used. The transformation is similar to the one originally proposed by Phillips[64] and is often used when both the bottom topography and vertical resolution are important[45]. In our case, a=0 and b=1 have been chosen. That is

$$\sigma = (z+h)/H \tag{1.23}$$

From the chain rule, we have following transformations:

$$\frac{\partial}{\partial z} \to \frac{1}{H}\frac{\partial}{\partial\sigma}, \qquad \frac{\partial}{\partial t} \to \frac{\partial}{\partial t} - \frac{\sigma}{H}\frac{\partial\zeta}{\partial t}\frac{\partial}{\partial\sigma}$$

$$\frac{\partial}{\partial x} \to \frac{\partial}{\partial x} + \frac{1}{H}(\frac{\partial h}{\partial x} - \sigma\frac{\partial H}{\partial x})\frac{\partial}{\partial\sigma}$$

$$\frac{\partial}{\partial y} \to \frac{\partial}{\partial y} + \frac{1}{H}(\frac{\partial h}{\partial y} - \sigma\frac{\partial H}{\partial y})\frac{\partial}{\partial\sigma}$$

After using the above relations, the advective terms in the first momentum equation then become:

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} \to$$

$$u\{u_x + \frac{1}{H}(h_x - \sigma H_x)u_\sigma\} + v\{u_y + \frac{1}{H}(h_y - \sigma H_y)u_\sigma\} + \frac{w}{H}u_\sigma$$

$$= uu_x + vu_y + \frac{\tilde{w}}{H}u_\sigma \tag{1.24}$$

and similarly

$$u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}+w\frac{\partial v}{\partial z} \;\rightarrow\; uv_x+vv_y+\frac{\tilde{w}}{H}u_\sigma$$

where

$$\tilde{w}=w+u(h_x-\sigma H_x)+v(h_y-\sigma H_y) \tag{1.25}$$

and the x and y derivatives are now with σ held constant.

Now by integrating the continuity equation (1.1) from z=− h to z and using the kinematical boundary condition (1.6) on z=− h we get

$$w+\frac{\partial}{\partial x}\int_{-h}^{z}udz+\frac{\partial}{\partial y}\int_{-h}^{z}vdz=0$$

i.e.

$$w=-\frac{\partial}{\partial x}\left(H\int_0^\sigma ud\sigma\right)-\frac{\partial}{\partial y}\left(H\int_0^\sigma vd\sigma\right)$$

Using the above chain rule we transform this to σ instead of z:

$$\begin{aligned}w=&-\frac{\partial}{\partial x}\left(H\int_0^\sigma ud\sigma\right)-\frac{1}{H}(h_x-\sigma H_x)\frac{\partial}{\partial\sigma}\left(H\int_0^\sigma ud\sigma\right)\\&-\frac{\partial}{\partial y}\left(H\int_0^\sigma vd\sigma\right)-\frac{1}{H}(h_y-\sigma H_y)\frac{\partial}{\partial\sigma}\left(H\int_0^\sigma vd\sigma\right)\end{aligned}$$

and after simplification we have the convenient expression

$$\tilde{w}=-\frac{\partial}{\partial x}\left(H\int_0^\sigma ud\sigma\right)-\frac{\partial}{\partial y}\left(H\int_0^\sigma vd\sigma\right) \tag{1.26}$$

The momentum equations then have the form

$$\frac{\partial u}{\partial t}-\frac{1}{H^2}\frac{\partial}{\partial\sigma}\left(\mu\frac{\partial u}{\partial\sigma}\right)-fv=-g\frac{\partial\zeta}{\partial x}+F \tag{1.27}$$

$$\frac{\partial v}{\partial t}-\frac{1}{H^2}\frac{\partial}{\partial\sigma}\left(\mu\frac{\partial v}{\partial\sigma}\right)+fu=-g\frac{\partial\zeta}{\partial y}+G \tag{1.28}$$

where

$$F=\frac{\sigma\zeta_t}{H}\frac{\partial u}{\partial\sigma}-uu_x-vu_y-\frac{\tilde{w}}{H}u_\sigma-\frac{1}{\rho}\frac{\partial p_s}{\partial x}$$

$$G=\frac{\sigma\zeta_t}{H}\frac{\partial v}{\partial\sigma}-uv_x-vv_y-\frac{\tilde{w}}{H}v_\sigma-\frac{1}{\rho}\frac{\partial p_s}{\partial y}$$

It is convenient to write the momentum equation in complex form by setting U=u +i v. Then equation (1.27) and (1.28) can be combined into the form.

$$\frac{\partial U}{\partial t}+ifU-H^{-2}\frac{\partial}{\partial\sigma}\left(\mu\frac{\partial U}{\partial\sigma}\right)=-g\left(\frac{\partial\zeta}{\partial x}+i\frac{\partial\zeta}{\partial y}\right)+F_{NL} \tag{1.29}$$

where

$$F_{NL}=\frac{\sigma\zeta_t}{H}U_\sigma-uU_x-vU_y-\frac{\tilde{w}}{H}U_\sigma-\frac{1}{\rho}\left[\frac{\partial p_s}{\partial x}+i\frac{\partial p_s}{\partial y}\right] \tag{1.30}$$

Similarly, the boundary conditions (1.14) and (1.19)-(1.20) are transformed into σ-coordinate as

$$\begin{aligned}
&\mu\frac{\partial U}{\partial\sigma}=HS && on\ \ \sigma=1\\
&\mu\frac{\partial U}{\partial\sigma}=H(\kappa_1+\kappa_2|U|)U && on\ \ \sigma=0
\end{aligned} \tag{1.31}$$

where

$$S=\frac{\tau_x^s+i\,\tau_y^s}{\rho}$$

$$|U|=\sqrt{u^2+v^2}\,\Big|_{\sigma=0}$$

It is easy to transform the continuity equation (1.12) and (1.13) into sigma coordinates:

$$\frac{\partial\zeta}{\partial t}+\frac{\partial p}{\partial x}+\frac{\partial q}{\partial y}=0 \qquad (1.32)$$

and

$$p=H\int_0^1 u d\sigma \quad , \quad q=H\int_0^1 v d\sigma \qquad (1.33)$$

Chapter 2. Spectral Method

The fundamental idea of the spectral method is to expand the complex velocity U in terms of some set of basis functions.The use of a basis of "eddy-viscosity eigenfunctions" was first proposed by Heaps[33-34] to solve the linearized three-dimensional tidal equations. The significant advantage of this particular basis is that the modal equations are uncoupled. In this chapter, this method will be extended to a general nonlinear equations with an arbitrary variable eddy-viscosity function.

2.1 *Boundary Modifications*

In the spectral method, if the basis functions satisfy the boundary conditions of the problem, the expansion can converge uniformly. We first use a similar modification technique to that proposed by Lardner[40]. It has been shown that this modification significantly accelerates the convergence for the linearized equations. In our case, this modification also has another advantage in that it transforms the boundary conditions into type II conditions, therefore this will simplify the eddy-viscosity eigenvalue problem.

The modification consists of defining the function

$$V(\sigma) = SH\int_0^{\sigma} \frac{\sigma'}{\mu(\sigma')} d\sigma' + BH\int_1^{\sigma} \frac{1-\sigma'}{\mu(\sigma')} d\sigma'. \tag{2.1}$$

where $B(t) = (\kappa_1 + \kappa_2|U|)U|_{\sigma=0}$ (The dependence on x and y has been suppressed to avoid confusion), Integrating (2.1) from 0 to 1 gives that

$$\bar{V} = \int_0^1 V(\sigma)d\sigma = H(S-B)\int_0^1 \frac{\sigma(1-\sigma)}{\mu(\sigma)} d\sigma \tag{2.2}$$

This function V(σ) satisfies that

$$\frac{\partial}{\partial\sigma}\left(\mu\frac{\partial V}{\partial\sigma}\right) = H(S-B) \tag{2.3}$$

$$\mu \frac{\partial V}{\partial \sigma} = HS \quad \text{on} \quad \sigma = 1, \qquad \mu \frac{\partial V}{\partial \sigma} = HB \quad \text{on} \quad \sigma = 0. \tag{2.4}$$

Then, setting $U(\sigma) = V(\sigma) + W(\sigma)$, we obtain the following boundary value problem for W:

$$\frac{\partial W}{\partial t} + if\,W - H^{-2}\frac{\partial}{\partial \sigma}\left(\mu \frac{\partial W}{\partial \sigma}\right) = -g\left(\frac{\partial \zeta}{\partial x} + i\frac{\partial \zeta}{\partial y}\right) + F_{NL} + H^{-1}(S - B) - \frac{\partial V}{\partial t} - if\,V. \tag{2.5}$$

and $$\mu \frac{\partial W}{\partial \sigma} = 0 \quad \text{on} \quad \sigma = 0 \quad \text{and} \quad \sigma = 1. \tag{2.6}$$

Where F_{NL} is defined in (1.30). The essential point in this modification is that the boundary conditions (2.6) are homogeneous.

2.2 *Spectral Method for vertical direction*

In this section, we first consider the general Sturm-Liouville problem

$$\frac{d}{d\sigma}\left(p(\sigma)\frac{d\phi}{d\sigma}\right)+[q(\sigma)+\lambda r(\sigma)]\phi(\sigma)=0 \tag{2.7}$$

with boundary conditions

$$\alpha_1\phi(\sigma)+\alpha_2 p(\sigma)\frac{d\phi}{d\sigma}=0 \quad at\ \sigma=a \tag{2.8}$$

$$\beta_1\phi(\sigma)+\beta_2 p(\sigma)\frac{d\phi}{d\sigma}=0 \quad at\ \sigma=b \tag{2.9}$$

In the above equations ϕ is an eigenfunction of the Sturm-Liouville problem, and λ is the corresponding eigenvalue, α_1, α_2, β_1 and β_2 are arbitrary constants with $\alpha_1\alpha_2 \geq 0$ and $\beta_1\beta_2 \leq 0$. The coefficients p(σ), q(σ) and r(σ) are assumed to be real and continuous, with p(σ) and r(σ) also being positive in (a, b). When a and b are finite, and p(a) and p(b) are positive , and q(a) and q(b) are bounded, we have a regular Sturm-Liouville problem.

Several authors have developed algorithms to solve such Sturm-Liouville problems by using methods based on the Prufer transformation[65-66]. A modified Prufer transformation that offered computational advantages was later proposed by Bailey[67]. The transformation was built into the SLEIGN code[68] which was used in our problems. This code has also been used by Kuzmic[69] in a linearized model of wind-induced motions in shallow sea.

For our purposes, we consider the eddy-viscosity eigenvalue problem

$$\frac{d}{d\sigma}\left(\mu\frac{d\phi}{d\sigma}\right)+\lambda\phi(\sigma)=0 \tag{2.10}$$

and

$$\mu\frac{d\phi}{d\sigma}=0 \qquad \text{at } \sigma=0 \text{ and } 1 \tag{2.11}$$

We denote the eigenpairs by $\{\phi_j(\sigma), \lambda_j : j = 0,1,2,...\}$ where the lowest eigenpair is $\lambda_0 = 0$, $\phi_0(\sigma) = 1$. For a general eddy viscosity, it is necessary to compute the other eigenpairs numerically. For this purpose we have used the subroutine SLEIGN [68]*. We assume the eigenfunctions are normalized so that

$$\int_0^1 \phi_j(\sigma)^2 d\sigma = 1. \tag{2.12}$$

The eigenfunctions then form an orthonormal system, and in particular, orthogonality with ϕ_0 implies that

$$\int_0^1 \phi_j(\sigma) d\sigma = 0, \quad j \geq 1. \tag{2.13}$$

In general, these eigenpairs depend on x and y, and also on t if the eddy-viscosity is time dependent. For the general case they must be determined numerically, and if μ varies in a general way with t this usually makes the method impractical since the eigenfunctions must be re-determined at each time step for all horizontal points. Consequently, we shall from now on restrict to the case when μ is independent of t.** The eigenfunctions can then be determined at the beginning of the computation and although this can be quite expensive, it only has to be done once.

We now expand W in terms of the eigenfunctions:

$$W(\sigma) = c_0 + \sum_{j \geq 1} c_j \phi_j(\sigma). \tag{2.14}$$

* We are indebted to Dr. Paul Bailey for his help in supplying us with a recent version of this program.

** The method can readily be extended to the case when the eddy viscosity has the form

$$\mu(x, y, \sigma, t) = \mu_1(x, y, \sigma)\mu_2(t)$$

that is, at any point μ has a similar profile for all t.

In view of the orthonormality of the eigenfunctions, the coefficients in the expansion (2.14) are given by

$$c_0 = \int_0^1 W(\sigma)d\sigma \equiv \overline{W}, \qquad c_j = \int_0^1 W(\sigma)\phi_j(\sigma)d\sigma \quad j \geq 1. \tag{2.15}$$

Using the definition of W we have that $c_0 = \overline{U} - \overline{V}$ and from equation (2.2) it then follows that

$$c_0 = \overline{U} - H(S-B)\int_0^1 \frac{\sigma(1-\sigma)}{\mu(\sigma)}d\sigma. \tag{2.16}$$

Integrating the differential equation (2.5) from $\sigma = 0$ to $\sigma = 1$ and using the boundary conditions (2.6), we obtain an equation for c_0. This is equivalent to the usual depth averaged momentum equation, and it is in fact more convenient to use this equation in the latter form, which can be obtained more directly by integrating eqn (1.29):

$$\frac{\partial \overline{U}}{\partial t} + if\overline{U} + g\left(\frac{\partial \zeta}{\partial x} + i\frac{\partial \zeta}{\partial y}\right) = R_0 \tag{2.17}$$

where

$$R_0 = H^{-1}(S - B) + \int_0^1 F_{NL}\, d\sigma. \tag{2.18}$$

Multiplying the differential equation (2.5) by $\phi_j(\sigma)$, integrating from $\sigma = 0$ to $\sigma = 1$ and using the boundary conditions (2.6), we obtain a system of differential equations for the coefficients c_j:

$$\frac{\partial c_j}{\partial t} + \alpha_j c_j = R_j. \tag{2.19}$$

where $\alpha_j = h^{-2}\lambda_j^2 + if$ and the right side is given by

$$R_j = R_j^{NL} + R_j^{V} + (H^{-2} - h^{-2})\lambda_j^2 c_j \tag{2.20}$$

with

$$R_j^{NL} = \int_0^1 F_{NL}\phi_j\, d\sigma \tag{2.21}$$

and

$$R_j^V = \int_0^1 \left[-\frac{\partial V}{\partial t} - ifV \right] \phi_j \, d\sigma = -\frac{\partial I_j}{\partial t} - ifI_j \tag{2.22}$$

provided the eigenfunctions are time-independent, where

$$I_j \equiv \int_0^1 V(\sigma)\phi_j(\sigma) d\sigma = \frac{1}{\lambda_j^2}\{HS\,\phi_j(1) - HB\,\phi_j(0)\}. \tag{2.23}$$

In the numerical solution of equations (2.17) and (2.19) the terms on the right sides are treated explicitly. The final small nonlinear term on the right of eqn (2.20) can quite easily be accommodated on the left side of (2.19), but the coefficients on the left then become time-dependent and this adds significantly to the cost of the algorithm.

In order to obtain initial conditions for the system (2.17) and (2.19), we assume the motion starts from some given velocity, $U = U_0$, at $t = 0$. In practice, this is taken to be a state of rest, $U_0 = 0$. Then at $t = 0$, $W = -V$, so the initial values of the coefficients are given by

$$c_j(0) = -\int_0^1 V(\sigma)\phi_j(\sigma) d\sigma = -\frac{Sh}{\lambda_j^2}\phi_j(1). \tag{2.24}$$

where eqn (2.23) has been used together with the fact that $B = 0$ at $t = 0$.

To summarize the problem that we are now left with, equations (1.12) and (2.17) form a coupled system for ζ and $p + iq = H\overline{U}$ with these variables having zero initial values. Lateral boundary conditions on the coastal and open portions of the boundary are required to form a well-posed problem and these can be taken in the same form usually used for two-dimensional models. The coefficients c_j that determine the vertical structure of the fluid velocity are obtained by solving the system (2.19) with initial conditions (2.24). In terms of these quantities, the velocity field is given by

$$\begin{aligned} U(\sigma) = \overline{U} + HS\int_0^\sigma \frac{\sigma'}{\mu(\sigma')} d\sigma' + HB\int_1^\sigma \frac{1-\sigma'}{\mu(\sigma')} d\sigma' \\ - H(S-B)\int_0^1 \frac{\sigma'(1-\sigma')}{\mu(\sigma')} d\sigma' + \sum_{j\geq 1} c_j\phi_j(\sigma) \end{aligned} \tag{2.25}$$

The coupling between the system (1.12),(2.17) and the equations (2.19) occurs only through the nonlinear and bottom friction terms which are treated explicitly in this approach. Such treatment does raise the possibility of potential instability, and this is one of the questions we shall examine in the tests described in chapter 4

Some differences occur for the no-slip boundary condition (1.18). In the definition of the function V the final term in eqn (2.5) is dropped and the boundary conditions (2.11) on the eigenfunctions are changed to

$$\phi = 0 \ \ on \ \ \sigma = 0 \quad and \quad \mu \frac{\partial \phi}{\partial \sigma} = 0 \ \ on \ \ \sigma = 1 \tag{2.11'}$$

There is no zero eigenvalue in this case, so eqn (2.13) does not hold. Eqn (2.25) is replaced simply by

$$U(\sigma) = HS \int_0^{\sigma} \frac{\sigma'}{\mu(\sigma')} d\sigma' + \sum_{j \geq 1} c_j \phi_j(\sigma). \tag{2.25'}$$

Proceeding as before, we obtain the following system of differential equations for the coefficients in place of eqn (2.19):

$$\frac{\partial c_j}{\partial t} + \alpha_j c_j + \left\{ g \left(\frac{\partial \zeta}{\partial x} + i \frac{\partial \zeta}{\partial y} \right) + H^{-1} S \right\} \int_0^1 \phi_j d\sigma = R_j \tag{2.19'}$$

where R_j is again given by eqns (2.20)-(2.22). However, the right side of eqn (2.23) requires modification to

$$I_j \equiv \int_0^1 V(\sigma) \phi_j(\sigma) d\sigma = \frac{HS}{\lambda_j^2} \int_0^1 \sigma \, \phi_j'(\sigma) \, d\sigma. \tag{2.23'}$$

A corresponding change occurs in the initial condition (2.24).

2.3 *Finite Element Method for nonlinear terms*

The computation of the integrals involving the nonlinear terms in eqns (2.18) and (2.21) is the most expensive part of the algorithm: about two-thirds of the total CPU time is spent on it. A finite elements method has been adopted for this computation, which we have found significantly more efficient than to evaluate the integrals by direct use of the eigenfunction expansions.

Combining eqns (1.30) and (2.21), we have

$$R_j^{NL} = H^{-1}\int_0^1 [\sigma\zeta_t - \tilde{w}]U_\sigma\phi_j \, d\sigma - \int_0^1 [uU_x + vU_y|\phi_j \, d\sigma \qquad (2.26)$$

with a similar expression with $j = 0$ for the integral term in eqn (2.18).

We choose L equally spaced nodes $\{\sigma_j\}$ in the interval $[0,1]$ with $\sigma_1 = 0$ and $\sigma_L = 1$ and spacing $\Delta\sigma = 1/(L-1)$.. Each function of σ is approximated piecewise linearly in each sub-interval, for example, in $\sigma_l \le \sigma \le \sigma_{l+1}$,

$$U(\sigma) = U_l + \frac{U_{l+1} - U_l}{\Delta\sigma}(\sigma - \sigma_l), \qquad U_\sigma(\sigma) = \frac{U_{l+1} - U_l}{\Delta\sigma},$$

$$u(\sigma) = u_l + \frac{u_{l+1} - u_l}{\Delta\sigma}(\sigma - \sigma_l), \qquad \tilde{w}(\sigma) = \tilde{w}_l + \frac{\tilde{w}_{l+1} - \tilde{w}_l}{\Delta\sigma}(\sigma - \sigma_l)$$

where U_l is the value of $U(\sigma)$ at node l and so on. The various terms in eqn (2.30) are then approximated as follows.

$$\int_0^1 \sigma U_\sigma \phi_j \, d\sigma = \sum_{l=1}^{L-1}(U_{l+1} - U_l)S_{lj}^{(1)}$$

$$\int_0^1 \tilde{w} U_\sigma \phi_j \, d\sigma = \sum_{l=1}^{L-1}\frac{\tilde{w}_{l+1} - \tilde{w}_l}{\Delta\sigma}(U_{l+1} - U_l)S_{lj}^{(2)} + \sum_{l=1}^{L-1}\tilde{w}_l(U_{l+1} - U_l)S_{lj}^{(3)}$$

$$\int_0^1 [uU_x + vU_y|\phi_j \, d\sigma = \sum_{l=1}^{L-1}\Big\{(u_l U_{x,l} + v_l U_{y,l})S_{lj}^{(4)} + (u_{l+1}U_{x,l+1} + v_{l+1}U_{y,l+1})S_{lj}^{(6)}$$
$$+ (u_l U_{x,l+} u_{l+1}U_{x,l} + v_l U_{y,l+1} + v_{l+1}U_{y,l})S_{lj}^{(5)}\Big\}$$

where

$$S_{lj}^{(1)}=\frac{1}{\Delta\sigma}\int_{\sigma_l}^{\sigma_{l+1}} \sigma\,\phi_j\,d\sigma, \qquad S_{lj}^{(2)}=\frac{1}{\Delta\sigma}\int_{\sigma_l}^{\sigma_{l+1}} (\sigma-\sigma_l)\phi_j\,d\sigma,$$

$$S_{lj}^{(3)}=\frac{1}{\Delta\sigma}\int_{\sigma_l}^{\sigma_{l+1}} \phi_j\,d\sigma, \qquad S_{lj}^{(4)}=\int_{\sigma_l}^{\sigma_{l+1}} \left(\frac{\sigma-\sigma_{l+1}}{\Delta\sigma}\right)^2 \phi_j\,d\sigma,$$

$$S_{lj}^{(5)}=\int_{\sigma_l}^{\sigma_{l+1}} \left(\frac{\sigma_{l+1}-\sigma}{\Delta\sigma}\frac{\sigma-\sigma_l}{\Delta\sigma}\right)\phi_j\,d\sigma, \qquad S_{lj}^{(6)}=\int_{\sigma_l}^{\sigma_{l+1}} \left(\frac{\sigma-\sigma_l}{\Delta\sigma}\right)^2 \phi_j\,d\sigma.$$

It remains to compute $\tilde{w}$ from eqn (1.26). We first use a trapezoidal approximation:

$$\int_0^{\sigma_l} u\,d\sigma=\tfrac{1}{2}\Delta\sigma\sum_{k=1}^{l-1}[u_{k+1}+u_k], \qquad \int_0^{\sigma_l} v\,d\sigma=\tfrac{1}{2}\Delta\sigma\sum_{k=1}^{l-1}[v_{k+1}+v_k].$$

Then $\tilde{w}_l$ can be constructed recursively as

$$\tilde{w}_1=0, \qquad \tilde{w}_{l+1}=\tilde{w}_l-\tfrac{1}{2}\Delta\sigma\left\{\frac{\partial}{\partial x}[H(u_{l+1}+u_l)]+\frac{\partial}{\partial y}[H(v_{l+1}+v_l)]\right\}.$$

Chapter 3. Scheme Formulation

In the last chapter, by using the modified spectral method for the linear parts and a finite element method for the nonlinear terms, we transformed the three-dimensional equations (1.32) and (1.29) into a two-dimensional system (1.32), (2.17) and (2.19) or (2.19') with the connecting relations (2.25) or (2.25'). In order to discuss the horizontal discretization next, we write the system in a general form as follows:

$$\frac{\partial \zeta}{\partial t}+\frac{\partial p}{\partial x}+\frac{\partial q}{\partial y}=0 \tag{3.1}$$

$$\frac{\partial c_j}{\partial t}+\alpha_j c_j=R_{1j}\left[\zeta_x+i\zeta_y\right]+R_{2j} \tag{3.2}$$

3.1 *Interior points*

Figure 3.1 shows the distribution of grid points at which the variables ζ, u and v are computed for the five horizontal grids investigated by Arakawa and Lamb[42]. As discussed in the Introduction, only the grids A,B,C and E will be considered here. For the A,B and E-grids, all the variables u, v, p, q and c_j are computed at the same points. For the C-grid scheme it is necessary to split equation (3.2) into two real equations since the real and imaginary parts relate to different grid points, and we set

$$c_j=a_j+ib_j, \qquad R_{2j}=R_{2j}^x+iR_{2j}^y .$$

The spatial finite difference approximations to eqns (3.1) and (3.2) appropriate for these four grids are as follows.

Scheme A:

$$\frac{\partial \zeta}{\partial t}+\overline{(\delta_x p)}^x+\overline{(\delta_y q)}^y=0 \tag{3.3A}$$

$$\frac{\partial c_j}{\partial t}+\alpha_j c_j=R_{1j}\left\{\overline{(\delta_x \zeta)}^x+i\,\overline{(\delta_y \zeta)}^y\right\}+R_{2j} . \tag{3.4A}$$

Scheme B:

$$\frac{\partial\zeta}{\partial t}+\overline{(\delta_x p)}^y+\overline{(\delta_y q)}^x=0 \qquad (3.3B)$$

$$\frac{\partial c_j}{\partial t}+\alpha_j c_j=R_{1j}\left\{\overline{(\delta_x\zeta)}^y+i\,\overline{(\delta_y\zeta)}^x\right\}+R_{2j}. \qquad (3.4B)$$

Scheme C:

$$\frac{\partial\zeta}{\partial t}+\delta_x p+\delta_y q=0 \qquad (3.3C)$$

$$\left.\begin{aligned}\frac{\partial a_j}{\partial t}+k_j a_j-f\,\overline{b_j}&=R_{1j}\,\delta_x\zeta+R^x_{2j}\\ \frac{\partial b_j}{\partial t}+k_j b_j+f\,\overline{a_j}&=R_{1j}\,\delta_y\zeta+R^y_{2j}.\end{aligned}\right. \qquad (3.4C)$$

Scheme E:

$$\frac{\partial\zeta}{\partial t}+\delta_x p+\delta_y q=0 \qquad (3.3E)$$

$$\frac{\partial c_j}{\partial t}+\alpha_j c_j=R_{1j}\left\{\delta_x\zeta+i\,\delta_y\zeta\right\}+R_{2j}. \qquad (3.4E)$$

(It is worth noting that grid E can be regarded as two interlocking C-grids.)

In these equations the following notation is used for any net function β:

$$(\delta_x\beta)_{m,n}=\Delta x^{-1}\left(\beta_{m+\frac{1}{2},n}-\beta_{m-\frac{1}{2},n}\right) \qquad (\delta_y\beta)_{m,n}=\Delta y^{-1}\left(\beta_{m,n+\frac{1}{2}}-\beta_{m,n-\frac{1}{2}}\right)$$

$$\left(\overline{\beta}^x\right)_{m,n}=\tfrac{1}{2}\left(\beta_{m+\frac{1}{2},n}+\beta_{m-\frac{1}{2},n}\right), \qquad \left(\overline{\beta}^y\right)_{m,n}=\tfrac{1}{2}\left(\beta_{m,n+\frac{1}{2}}+\beta_{m,n-\frac{1}{2}}\right)$$

$$\left(\overline{\overline{\beta}}\right)_{m,n}=\left(\overline{\overline{\beta}^x}^y\right)_{m,n}=\left(\overline{\overline{\beta}^y}^x\right)_{m,n}.$$

Also, $k_j=h^{-2}\lambda_j^2$, $\quad \alpha_j=k_j+if$. It can be seen that the above schemes are all second order in the spatial grid dimensions at points in the interior of the region. The treatment of boundary points in the various schemes will be discussed below.

The time differencing scheme used for all the four grids has been a leap-frog scheme in which ζ and c_j are computed at alternating half-steps with eqns (3.1) and (3.2) being used alternately to update each of these variables in turn [40]. This scheme has the advantage of being explicit and also second order in the time step, though the size of the time-step is restricted by the CFL stability criterion.

Since for the A,B and E-grids the two velocity components are taken at the same grid points the differential equations for the c_j can be solved in complex form. Since this system is stiff, some care must be used in the choice of integration method. Writing eqn (3.4A), (3.4B) or (3.4E) in the symbolic form

$$\frac{\partial c_j}{\partial t} + \alpha_j c_j = G_j \tag{3.5}$$

we can update the solution over one time step from t to $t + \tau$ by the approximation [40]

$$\begin{aligned} c_j(t+\tau) &= c_j(t)e^{-\alpha_j\tau} + \int_0^\tau G(t+s)e^{-\alpha_j(\tau-s)}ds \\ &\approx c_j(t)e^{-\alpha_j\tau} + G(t+\tfrac{1}{2}\tau)r_j \end{aligned} \tag{3.6}$$

where

$$r_j = \frac{1}{\alpha_j}[1-e^{-\alpha_j\tau}].$$

The required value of G is found from ζ at the intermediate half step, which has already been determined from the continuity equation.

For the C-grid the two real equations (293.4C) must be integrated separately, but similar approximation formulas to (3.6) can be used, with α_j replaced by k_j and the Coriolis terms included with G. In order to have a stable treatment of the Coriolis terms, the two equations (3.4Ca) and (3.4Cb) are used in alternating order on successive time steps. This also has the advantage of providing a second order scheme. A similar situation would occur also for the D-grid.

3.2 *Treatment of boundary points*

At a coastal boundary, the boundary condition is taken to be that the normal component of volume flux, p or q, is equal to zero. As discussed in the Introduction, it is not assumed that the normal velocity is zero throughout the water column. At an open boundary, it is assumed for simplicity that the surface elevation, ζ, is prescribed. To illustrate how these boundaries are treated, examples of a left coastal boundary and a right open boundary are shown in Figures 3.2(a)-(d) for the four numerical schemes respectively. Scheme C is of course well-known but is included for completeness. It will be seen that in all the schemes the local truncation error is increased at the boundary points from second order to first.

Scheme A:

At point 1 in Figure 3.2(a), the boundary condition is $p = 0$. In order to compute c_j from (3.4A) at the adjacent interior point 2 it is necessary to know ζ at 1. To compute this, (3.3A) must be modified to have a one-sided difference in the x direction; furthermore, q is required at the adjacent boundary points, 3 and 4. These values can be found from (2.25) with c_j being computed from a modified version of (3.4A) that uses a one-sided difference in the x direction.

At the open boundary, ζ is given at point 5. In order to compute ζ from (3.3A) at the adjacent interior point 6 it is necessary to know p at 5. This can be found from (2.25) with c_j being computed from a modified version of (3.3A) that uses a one-sided difference in the x direction.

Scheme B:

At points 1 and 2 in Figure 3.2(b), the boundary condition is $p = 0$. In order to compute ζ at the adjacent interior point 3, (3.3B) must be modified to have a one-sided average of the q -term in the x direction. In this scheme, q is not computed at the boundary points 1 and 2 since use of (3.4B) at these points would necessitate an extrapolation of ζ values.

At the open boundary, ζ is given at points 5 and 6 and c_j can be computed from (3.4B) without modification at the adjacent interior point 7.

Scheme C:

At point 1 in Figure 3.2(c), the boundary condition is $p = 0$ and ζ may be computed from (3.3C) without modification at the adjacent interior point 2.The quantities a_j are not computed at the point 1. To compute b_j from (3.4Cb) at the interior point 3 all that is needed is to use Jamart and Ozer's wet points only averaging for the Coriolis term, that is to exclude points such as 1 that lie on the boundary.

At the open boundary, ζ is given at point 5. In order to compute a_j at the adjacent interior point, again (3.4Ca) is modified by including only interior points in the Coriolis term. The quantities b_j are not computed at points such as 7 that lie on the boundary.

Scheme E:

At point 1 in Figure 3.2(d), the boundary condition is $p = 0$. In order to compute ζ from (3.3E) at the adjacent interior point 2 it is necessary to know q at 3 and 4. These can be found from (2.25) with c_j being computed at 3 and 4 from a modified version of (3.4E) that uses an extrapolated one-sided difference of ζ in the x direction.

At the open boundary, ζ is given at point 5. In order to compute c_j from (3.4E) at the adjacent interior point 6 it is necessary to know ζ at points 7 and 8. This can be found from (3.3E) with an extrapolated one-sided difference used for p in the x direction. The use of extrapolated one-sided differences in this scheme is a potential weakness.

3.3 *Scheme comparisons*

(a) *Closed rectangular sea*

The four algorithms have been compared and tested using two model problems. The first of these is a simplified storm-surge model of the North Sea used by Owen and Davies [37] and subsequently by several investigators to test algorithms. The model region consists of a closed rectangular sea of dimensions 400 kms in the x -direction and 800 kms in the y -direction, with grid spacings $\Delta x = 400/9$ kms and $\Delta y = 800/17$ kms. The depth is taken uniformly as 65 m. The sea is initially in a state of equilibrium and starting at $t = 0$ is subjected to a constant surface shear stress in the negative y -direction, with values $\tau_x = 0$, $\tau_y = -1.5$ N/m^2. The values of the other parameters (all in MKS units) are $\rho = 1025$, $N = 0.065$, $\kappa = 0.002$, $g = 9.81$ and $f = 1.22\times10^{-4}$. A time step $\tau = 360$ s was used.

Some typical computed results are shown in Table 3.1, where the velocity profiles after 30 hours at three grid points are tabulated for each of the four numerical schemes. An analytical solution is not known for this problem, so as a means of testing the accuracy of the numerical results, the same model problem has been re-computed using the B-grid scheme with grid spacings and time step equal to one quarter of those stated above, and the results of this computation are listed in the first column of the table. Since the algorithms are all second order at interior points and first order at boundary points, it can be expected that the errors in the first column of the table are somewhere between one quarter and one sixteenth of the errors in the third column.

Comparing with the "exact" solution in the first column, we can see that the B,C and E schemes produce much more accurate results than the A scheme. This is not surprising in fact since the approximations to the spatial derivatives in equations (3.3A) and (3.4A) involve finite difference over intervals of twice the size of those in the other three schemes. There appears to be little difference in accuracy among the B,C and E schemes.

Secondly, none of the schemes generate spurious numerical boundary layers, presumably because for the A,B and E schemes the momentum equations are solved at the same point and averaging is not needed for the Coriolis terms. For the C-scheme such boundary layers do occur if the wet-points only averaging is not employed [43].

Thirdly, the CPU time required per time step of computation is about the same for the A,B and C schemes, but is approximately twice as much for the E scheme. Again this is to be expected since, for the same grid spacing, the E-grid has twice as many grid points as the other three grids, which all have about the same number of points.

(b) *Open rectangular sea*

The second model problem is one for which an analytical solution can be found. Only linear equations have been considered here.The region is again rectangular but with open boundaries on all sides on which the volume transports are specified as given below. The water depth is uniformly 65 m and the other parameters are given the same values as in the closed sea problem except that κ is zero.

When the bottom friction is zero, the depth-integrated equations form a closed system. Integrating the linear momentum equations from $z = -h$ to $z = 0$, and using conditions $\kappa = 0$, we get

$$\begin{aligned} \frac{\partial p}{\partial t} - fq &= -gh\frac{\partial \zeta}{\partial x} + s_x \\ \frac{\partial q}{\partial t} + fp &= -gh\frac{\partial \zeta}{\partial y} + s_y \end{aligned} \tag{3.7}$$

where $s_x = \tau_x / \rho$, $s_y = \tau_y / \rho$. Equations (3.7) and (3.1) can be solved for p, q and ζ.

Analytical solutions can be found that depend on only one of the coordinates x or y. If $s_y = 0$, a solution that starts from rest and is independent of y is given by

$$\zeta = s_x Z(x,t;l_x), \qquad p = s_x P(x,t,l_x), \qquad q = s_x Q(x,t,l_x)$$

where x runs from 0 to l_x and

$$Z(x,t,l)=-l\sum_{n\ \mathrm{odd}}\frac{4}{\omega_n^2}\left[1-\cos\left(\frac{\omega_n t}{l}\right)\right]\cos\left(\frac{n\pi x}{l}\right)$$
$$P(x,t,l)=l\sum_{n\ \mathrm{odd}}\frac{4}{n\pi\omega_n}\sin\left(\frac{\omega_n t}{l}\right)\sin\left(\frac{n\pi x}{l}\right)$$
$$Q(x,t,l)=-f\,l^2\sum_{n\ \mathrm{odd}}\frac{4}{n\pi\omega_n^2}\left[1-\cos\left(\frac{\omega_n t}{l}\right)\right]\sin\left(\frac{n\pi x}{l}\right)$$

where $\omega_n^2=(fl)^2+gh(n\pi)^2$. By superimposing two of these solutions, we get a solution of the form

$$\zeta(x,y,t)=s_x Z(x,t;l_x)+s_y Z(y,t;l_y)$$
$$p(x,y,t)=s_x P(x,t,l_x)-s_y Q(y,t,l_y)$$
$$q(x,y,t)=s_x Q(x,t,l_x)+s_y P(y,t,l_y)$$

This solution satisfies the boundary conditions

$$p=-s_y Q(y,t,l_y)\quad \text{on } x=0 \text{ or } l_x$$
$$q=s_x Q(x,t,l_x)\quad \text{on } y=0 \text{ or } l_y.$$

When the bottom friction is zero, the Sturm-Liouville problem (2.10) and (2.11) has a zero eigenvalue and the spectral amplitude of the corresponding (constant) eigenfunction is related to the depth-averaged solution discussed above. In the other modal equations (3.2), the coefficient $R_{1j}=0$, and the amplitudes c_j can be found independently of the depth-integrated equations. When S and N are independent of t the solution is given by [40]

$$c_j(t)=-\frac{hS\phi_j(1)}{\alpha_j\lambda_j^2}\left\{if+k_j e^{-\alpha_j t}\right\}$$

where $k_j=h^{-2}\lambda_j^2$, $\quad \alpha_j=k_j+if$. In the special case when N is constant,

$$\phi_j(\sigma)=\sqrt{2}\cos[j\pi(1-\sigma)]\qquad \lambda_j^2=N(j\pi)^2,\qquad k_j=N(j\pi/h)^2,$$

for $j\geq 1$, and the complete velocity field is given by

$$U(\sigma)=\frac{p+iq}{h}+\frac{hS}{2N}(\sigma^2-\tfrac{1}{3})-2hS\sum_{j\geq 1}\frac{\cos[j\pi(1-\sigma)]}{\alpha_j\lambda_j^2}\left\{if+k_j e^{-\alpha_j t}\right\}.$$

Some typical computed results are shown in Table 3.2, where the velocity profiles after 30 hours at three grid points are tabulated for the analytical solution and for each of the four numerical schemes. The conclusions that can be drawn are consistent with those from the first model problem. Scheme A is considerably less accurate than schemes B,C and E, and in fact is sufficiently inaccurate to be unusable with the chosen grid sizes. The computational errors from schemes B,C and E are of comparable magnitudes. The root mean square errors in u and v from all the grid points are almost equal for the B and C-grids, having the values 11 and 13 mm/s respectively, and are slightly smaller for the E-grid, being 10 and 11 mm/s.
These differences are probably not significant since the relative sizes of the errors from the three grids fluctuate from step to step. Figure 3.3 shows the rms error in u and v combined for the B and C-grids for the first 600 steps. Up to about 240 time steps (24 hours) the B-grid results are more accurate than those from the C-grid, between 240 and 450 steps the C-grid produces more accurate results and from 450 to 600 steps the B-grid again becomes the more accurate. The rms errors after 600 steps from the B-grid are about the same as after 240 steps, namely 11 and 14 mm/s in u and v respectively. The errors are almost uniform through the water column in every case, that is they are concentrated in the lowest mode.

	“Exact”		Scheme A		Scheme B		Scheme C		Scheme E	
Level	*u*	*v*	*u*	*v*	*u*	*v*	*u*	*v*	*u*	*v*
10	–97	–385	–129	–375	–98	–397	–97	–387	–94	–395
9	–83	–239	–116	–229	–84	–251	–83	–241	–80	–248
8	–48	–125	–81	–115	–49	–136	–48	–126	–45	–133
7	–5	–38	–38	–29	–6	–49	–5	–40	–2	–47
6	38	24	5	31	36	13	38	22	39	16
5	75	67	43	73	74	56	75	65	76	59
4	102	93	71	97	101	83	102	91	103	87
3	116	106	86	108	114	96	116	104	116	100
2	115	106	89	107	114	97	115	105	115	101
1	100	93	78	92	99	85	101	91	100	88

	“Exact”		Scheme A		Scheme B		Scheme C		Scheme E	
Level	*u*	*v*	*u*	*v*	*u*	*v*	*u*	*v*	*u*	*v*
10	–144	–338	–148	–380	–148	–349	–145	–353	–144	–347
9	–130	–193	–134	–235	–134	–204	–131	–207	–130	–202
8	–94	–79	–98	–120	–98	–89	–95	–93	–94	–88
7	–50	5	–52	–35	–53	–4	–50	–8	–49	–3
6	–5	66	–7	26	–8	56	–5	52	–4	57
5	34	106	33	68	31	97	34	93	35	98
4	63	129	64	94	61	121	64	117	65	121
3	80	137	82	105	78	131	82	127	82	130
2	84	133	86	105	82	127	86	124	86	127
1	74	114	76	92	72	110	75	107	75	109

	“Exact”		Scheme A		Scheme B		Scheme C		Scheme E	
Level	*u*	*v*	*u*	*v*	*u*	*v*	*u*	*v*	*u*	*v*
10	–102	–341	–111	–372	–105	–359	–100	–354	–102	–348
9	–88	–195	–97	–226	–92	–213	–87	–208	–89	–202
8	–53	–81	–62	–112	–57	–99	–52	–94	–54	–88
7	–10	4	–18	–26	–14	–13	–9	–8	–11	–2
6	32	65	24	35	29	48	33	53	31	58
5	69	106	62	76	66	90	71	95	69	100
4	96	131	90	101	93	115	93	120	96	124
3	110	140	105	113	107	126	112	130	110	134
2	110	136	106	111	108	123	111	128	110	131
1	95	117	92	97	94	107	97	110	95	113

Table 3.1. Exact and computed velocity profiles after 300 steps (30 hours) for Test Problem 1. Units are mm/s. The three tabulations refer to the C-grid points (2,2), (8,10) and (5,15) respectively. (The region is rectangular, running from (2,2) to (10,18).)

	Exact		Scheme A		Scheme B		Scheme C		Scheme E	
Level	*u*	*v*	*u*	*v*	*u*	*v*	*u*	*v*	*u*	*v*
10	346	–606	506	–809	339	–586	343	–595	346	–593
9	162	–440	321	–643	154	–420	158	–429	161	–427
8	45	–276	204	–479	37	–256	41	–264	44	–262
7	–21	–125	138	–328	–29	–105	–25	–114	–22	–112
6	–53	4	106	–198	–60	24	–56	15	–53	17
5	–61	110	97	–92	–69	130	–65	122	–62	124
4	–57	192	102	–10	–65	212	–61	204	–58	205
3	–49	250	110	47	–56	270	–52	261	–49	263
2	–41	284	117	81	–49	304	–45	295	–42	297
1	–39	294	120	92	–46	315	–42	306	–39	308

	Exact		Scheme A		Scheme B		Scheme C		Scheme E	
Level	*u*	*v*	*u*	*v*	*u*	*v*	*u*	*v*	*u*	*v*
10	339	–580	338	–542	316	–582	325	–587	324	–592
9	155	–414	154	–321	132	–416	141	–421	140	–426
8	38	–249	37	–212	15	–251	23	–256	23	–261
7	–28	–99	–29	–61	–51	–101	–43	–106	–43	–111
6	–60	30	–61	68	–83	28	–74	23	–74	18
5	–68	137	–69	174	–91	135	–83	130	–83	125
4	–64	218	–56	256	–87	217	–79	211	–79	206
3	–56	276	–51	313	–79	274	–70	269	–70	264
2	–48	310	–50	348	–71	308	–63	303	–63	298
1	–46	321	–47	359	–69	319	–60	314	–60	309

	Exact		Scheme A		Scheme B		Scheme C		Scheme E	
Level	*u*	*v*	*u*	*v*	*u*	*v*	*u*	*v*	*u*	*v*
10	356	–600	403	–546	383	–588	354	–583	360	–584
9	173	–433	218	–381	198	–423	170	–417	176	–418
8	55	–269	107	–216	81	–258	53	–252	59	–253
7	–11	–118	35	–65	14	–107	–13	–102	–7	–103
6	–42	11	3	64	–17	21	–44	27	–38	26
5	–51	117	–5	170	–25	128	–53	134	–47	132
4	–47	199	0	252	–21	210	–49	215	–43	214
3	–38	256	7	309	–13	267	–40	273	–34	272
2	–31	291	14	344	–5	301	–33	307	–27	306
1	–28	302	17	355	–2	313	–30	318	–24	317

Table 3.2. Exact and computed velocity profiles after 300 steps (30 hours) for Test Problem 2. Units are mm/s. The three tabulations refer to the C-grid points (2,2), (6,4) and (9,6) respectively. (The region is rectangular, running from (1,1) to (10,10).)

Chapter 4. Numerical Experiments

4.1 *One-dimensional channel flow*

It is not easy to find exact solutions of the nonlinear equations with which to test the accuracy of numerical algorithms. One such solution, however, occurs for the steady wind-driven flow in a channel of constant depth. If, for such a case of uni-directional flow, one makes the approximation $H \approx h$, the advective terms become identically zero and the only nonlinearity is the bottom friction. In this case, for constant eddy viscosity $\mu = N$, the final steady solution can be found analytically.

With these approximations, eqns (1.32) and (1.29) reduce, for steady flow, to

$$\frac{\partial p}{\partial x}=0, \qquad \frac{\partial}{\partial \sigma}\left(N\,\frac{\partial u}{\partial \sigma}\right)=gh^2\frac{\partial \zeta}{\partial x},$$

where $p = h\int_0^1 u\,d\sigma = h\bar{u}$, and the boundary conditions (11) are

$$N\,\frac{\partial u}{\partial \sigma}=hS \quad \text{on} \quad \sigma=1, \qquad N\,\frac{\partial u}{\partial \sigma}=hB \equiv h(\kappa_1+\kappa_2|u|)u \quad \text{on} \quad \sigma=0.$$

We take the channel to be closed at its ends, so that the boundary conditions are $p = 0$. Therefore $p = 0$ for all x. Then the solution of these equations is

$$u(\sigma)=\tfrac{1}{2}u(0)(3\sigma^2-6\sigma+2)+(Sh/4N)(3\sigma^2-2\sigma)$$

$$\partial\zeta/\partial x=(6Nu(0)+3Sh)/2gh^2,$$

where the bottom velocity is given by

$$u(0)=(\varepsilon/2\kappa_2 h)\left\{\kappa_1 h+3N-\sqrt{(\kappa_1 h+3N)^2+2\varepsilon\kappa_2 h^2 S}\right\}$$

with $\varepsilon = 1$ if $S \geq 0$ and $\varepsilon = -1$ if $S < 0$.

For the numerical solution, the dynamical equations (1.32),(2.17) and (2.19) are solved from an initial state of rest until the steady flow is reached. In this case, these equations simplify to

$$\frac{\partial\zeta}{\partial t}+\frac{\partial p}{\partial x}=0, \qquad \frac{\partial\bar{u}}{\partial t}+g\,\frac{\partial\zeta}{\partial x}=H^{-1}(S-B),$$

$$\frac{\partial c_j}{\partial t}+h^{-2}\lambda_j^2 c_j=\frac{h\phi_j(0)}{\lambda_j^2}\frac{\partial B}{\partial t}$$

while the velocity is obtained from eqn (2.25) as

$$u(\sigma)=\bar{u}+\frac{hS}{2N}(\sigma^2-\tfrac{1}{3})-\frac{hB}{2N}\{(1-\sigma)^2-\tfrac{1}{3}\}+\sum_{j\geq 1}c_j\phi_j(\sigma).$$

κ_1 and κ_2	Level	Computed solution at points			Exact solution
		(2)	(8)	(15)	
κ_1 =0.002 κ_2 = 0.005	S	−42.94	−42.89	−42.92	−42.96
	M	7.52	7.58	7.55	7.50
	B	12.93	12.97	12.95	12.92
κ_1 =0.002 κ_2 = 0.015	S	−41.99	−42.01	−42.00	−41.99
	M	7.74	7.73	7.74	7.75
	B	10.98	10.98	10.98	10.98
κ_1 =0.002 κ_2 = 0.05	S	−40.54	−40.56	−40.55	−40.53
	M	8.11	8.09	8.10	8.11
	B	8.07	8.07	8.07	8.07
κ_1 =0 κ_2 = 0.005	S	−45.66	−45.12	−45.44	−45.78
	M	6.91	7.43	7.12	6.80
	B	18.65	18.99	18.78	18.57
κ_1 =0 κ_2 = 0.015	S	−43.60	−43.60	−43.60	−43.61
	M	7.35	7.36	7.35	7.34
	B	14.22	14.23	14.22	14.22

TABLE 4.1. Computed and exact velocities in cm/s for steady wind-driven flow in a channel with different values of linear and quadratic friction coefficients. Velocities are given at the surface, mid-point and bottom of the water column.

The algorithm described in chapters 2 and 3 is easily adapted to these simplified equations by setting the Coriolis parameter equal to zero and by-passing computation of the y -component of velocity. All nonlinear terms are suppressed except for the bottom friction. Some typical results are given in Table 4.1. In these computations, the values of the various parameters were chosen as $h = 65$ m, $\Delta x = 47{,}059$ m, $\tau = 360$ s, $N = 0.065$ m^2/s and $g = 9.81$ m/s^2. Six eigenfunctions were used in the calculations and the length of the channel was 16 horizontal grids.

The table gives the exact solution in the last column and in the adjacent columns the computed solutions at three points spaced along the channel. The velocity is given at the surface, mid-depth and bottom in each case. Results are given for several combinations of values of the linear and quadratic friction coefficients κ_1 and κ_2. It can be seen that the numerical results are quite accurate, even for very small drag coefficients and that the algorithm remains stable for both small and large values of these coefficients.

4.2 Rectangular sea with constant eddy viscosity

In order to test the algorithm for the full three-dimensional nonlinear hydrodynamic equations, we have made use of the simplified storm-surge model of the North Sea used by Davies [37-38] and also several other investigators [e.g. 36,40]. The model region consists of a closed rectangular sea of dimensions 400 kms in the x -direction and 800 kms in the y -direction, with grid spacings $\Delta x = 400/9$ kms and $\Delta y = 800/17$ kms. Figure 4.1 shows the B-grid used for this region. The depth is taken uniformly as 65 m. The sea is initially in a state of equilibrium and starting at $t = 0$ is subjected to a constant surface shear stress in the negative y -direction, with values $\tau_x = 0$, $\tau_y = -1.5$ N/m^2. The values of the other parameters (all in MKS units) are $\rho = 1025$, $N = 0.065$, $g = 9.81$ and $f = 1.22\times10^{-4}$. A time step $\tau = 360$ s was used. These parameters have been chosen to be identical to those used in earlier work [37,39] so that comparisons can be made.

An initial series of computations was designed to test the accuracy and stability of the explicit treatment of bottom friction in a dynamical problem as opposed to the steady problem of the preceding subsection. Here we compared the solutions obtained for the linearized equations with those obtained using an earlier algorithm [44] in which the linear bottom friction is incorporated into the construction of the eigenfunctions, and so is treated implicitly.

Some typical computed results are shown in Table 4.2. The upper and lower parts of the table show the numerical solutions computed with the time-step $\tau = 360$ s using respectively the present algorithm and the old implicit algorithm of [44]. It is clear that no significant errors are introduced in this case by the explicit treatment of the bottom stress.

A second series of computations was designed to test the rate of convergence of the eigenfunction expansion in comparison with that obtained by Davies [38]. A quadratic law of bottom friction was used, with $\kappa_1 = 0$ and $\kappa_2 = 0.002$, and all nonlinear terms were

		(2,2)	(8,10)	(5,15)
Surface velocity	*u*	–9.86	–14.82	–10.56
	v	–39.68	–35.03	–35.95
Mid-level velocity	*u*	5.51	1.12	4.78
	v	3.47	7.67	6.91
Bottom velocity	*u*	9.96	7.28	9.38
	v	8.51	10.97	10.63

		(2,2)	(8,10)	(5,15)
Surface velocity	*u*	–9.85	–14.85	–10.58
	v	–39.72	–34.97	–35.98
Mid-level velocity	*u*	5.52	1.11	4.78
	v	3.47	7.72	6.92
Bottom velocity	*u*	9.95	7.27	9.41
	v	8.56	11.03	10.72

TABLE 4.2. Computed velocity profiles in cm/s after 30 hours for the linearized equations. Profiles are given at the three grid points (2,2), (8,10) and (5,15) (see Figure 4.1). The upper part of the table provides the results from the present algorithm and the lower part from the implicit algorithm of [44].

retained in the equations. The computation was repeated with 4,6 and 10 eigenfunctions included in the expansion (where the $j = 0$ eigenfunction is also counted).

Typical results are shown in Table 4.3 in which are given the components of fluid velocity at the centre of the rectangular sea and at the surface and bottom of the water column, as well as the corresponding surface elevation, 30 hours after the onset of the wind. It can be seen that the expansion converges quite fast: for the expansion truncated after six eigenfunctions, the maximum error in the computed current (using the solution with ten eigenfunctions as the standard) is only 0.04 cm/s. The maximum difference over the whole rectangle between the values of ζ computed using 4 or 10 eigenfunctions is 0.1 cm.

The final row of Table 4.3 lists the CPU times for each computation (using an IBM 3081). The interesting fact here is that the corresponding CPU times for the identical computations except for the omission of the nonlinear advective terms were in each case less than one-third of the figures listed. Thus the inclusion of the advective terms increases very significantly the cost of the spectral method.

It is interesting to compare the computational efficiency of the eigenfunction method with that of the direction-splitting algorithms such as the one described in [46-47]. For this rectangular sea problem, the eigenfunction method with four eigenfunctions requires slightly less CPU time than the splitting method if the linearized hydrodynamical equations are used. However for the nonlinear equations, the CPU requirements of the eigenfunction method are about fifty percent greater than those of the splitting algorithm, and of course increase if more eigenfunctions are employed.

	Component	Number of eigenfunctions		
		4	6	10
Surface velocity	u	−19.60	−19.73	−19.77
	v	−31.39	−31.37	−31.37
Bottom velocity	u	8.37	8.32	8.31
	v	17.08	17.09	17.09
Surface elevation	ζ	149.7	149.7	149.7
	CPU time	180 s	240 s	370 s

TABLE 4.3. Components of velocity in cm/s after 30 hours at the centre of the rectangle computed for the fully nonlinear equations with different numbers of eigenfunctions.

Table 4.4 shows the corresponding results obtained by Davies [38] using Gram-Schmidt polynomials (equivalent to shifted Legendre polynomials) as basis functions. The rate of convergence is somewhat slower than that in Table 4.3, the maximum difference between the velocities computed with 6 and 8 functions being 0.34 cm/s. Davies found much slower convergence if cosine functions are used, the reason being that these basis functions cannot satisfy the surface stress condition, giving a nonuniformity there. The Legendre (or Chebychev) basis come from a singular Sturm-Liouville problem, so the inhomogeneous boundary condition does not disturb the rate of convergence. It is therefore significant that comparable, or even slightly better, accuracy can be obtained using the eddy-viscosity eigenfunctions. The Legendre or Chebychev polynomials have of course the disadvantage of giving modal equations that are coupled via dominant terms.

The differences between Tables 4.3 and 4.4 in actual current values are probably due to the different grids used in the two computations. In [38] a C-grid was used which produces spurious numerical boundary layers [43] and consequent errors of several percent in flow variables throughout the rectangle while the B-grid used here automatically avoids such layers. Another possible source of difference is that the finite element method of computing the advective terms at every time-step may be expected to be more accurate than the time-splitting method used in [38], particularly if the higher frequency waves play a significant role in the solution.

	Component	Number of eigenfunctions		
		4	6	8
Surface velocity	u	–17.62	–17.58	–17.56
	v	–29.62	–29.55	–29.56
Bottom velocity	u	9.75	9.79	9.45
	v	18.41	18.47	18.45
Surface elevation	ζ	140.8	140.8	140.5

TABLE 4.4. Components of velocity after 30 hours at the centre of the rectangle computed by Davies [38] using different numbers of eigenfunctions.

	Component	$\kappa_1 = 0.002$ $\kappa_2 = 0$	$\kappa_1 = 0.002$ $\kappa_2 = 0.002$	$\kappa_1 = 0$ $\kappa_2 = 0.02$	$\kappa_1 = 0.02$ $\kappa_2 = 0.02$
Surface velocity	u	−15.51	−15.05	−14.51	−10.25
	v	−35.25	−35.64	−36.17	−37.98
Bottom velocity	u	6.91	6.48	6.61	1.39
	v	21.65	21.69	21.14	26.05
Surface elevation	ζ	140.0	138.2	133.5	136.7

TABLE 4.5. Components of velocity in cm/s after 30 hours at the centre of the rectangle computed for the fully nonlinear equations with different friction coefficients.

A final series of computations was designed to examine the effect of different friction coefficients and in particular to ensure that the algorithm remains stable for large friction. Typical results are shown in Table 4.5 for several values of κ_1 and κ_2

We have experimented with large values of κ_1 and κ_2 and small eddy viscosity to determine the stability limits of the algorithm. Using the time step of 360 s, the algorithm remains stable for all physically realistic values of these parameters. For example, with $N = 0.065$, stability is maintained up to and beyond $\kappa_2 = 0.2$ when $\kappa_1 = 0$ and up to $\kappa_2 = 0.02$ when $\kappa_1 = \kappa_2$ With $\kappa_1 = 0$ and $\kappa_2 = 0.002$ stability is maintained for $N \geq 0.001$ while for $\kappa_1 = 0.002$ and $\kappa_2 = 0$ it is maintained for $N \geq 0.004$. Figure 4.2 shows the regions of stability and instability in the N-κ_2 plane in the case when $\kappa_1 = 0$.

4.3 *Rectangular sea with variable eddy viscosity*

To further test the algorithm, a three-dimensional nonlinear problem with a more realistic eddy viscosity function was used. Three cases of variable viscosity were considered, again following reference [38] for the sake of comparison, as shown in Figure 4.3. The water depth was taken as 65 m, and the thicknesses d_1 and d_2 of the surface and bottom layers were taken as 11 m. Within these boundary layers N is assumed to vary linearly with the vertical coordinate, while N is constant through the rest of the water column. The values used for the parameters in the three cases were:

Case A: N_s = 0.013 m^2/s, N_m = 0.065 m^2/s, N_b = 0.013 m^2/s,
Case B: N_s = 0.117 m^2/s, N_m = 0.065 m^2/s, N_b = 0.013 m^2/s,
Case C: N_s = 0.117 m^2/s, N_m = 0.065 m^2/s, N_b = 0.065m^2/s.

In this case, again for the sake of comparison with [38], a no-slip boundary condition (2.19) was used The algorithm must then be modified as indicated in Section 2.2. Figure 4.4 shows the two components of velocity as a function of vertical coordinate 75 hours after the onset of the wind at the centre of the rectangle for the three eddy viscosity functions. These were computed using six eigenfunctions. The most striking feature of these figures is the sensitivity of the near-surface velocities to the value of N_s. The no-slip condition prevents the bottom velocity from being as strongly affected by N_b , although the effect on the velocity gradient is noticeably.

Also in Figure 4.5 we have reproduced the figure given in [38] for the corresponding Case A. This was computed using four Chebychev or shifted legendre polynomials. There is substantial but not complete agreement between this figure and that in Figure 4.4(a). The differences are again probably ascribable to the numerical boundary layers.

Chapter 5 Density-Driven Flows

In this chapter the density-gradient driven flow problem will be considered. The basic equations and the model solutions used in this chapter have been taken from Lardner and Das[50]. The spectral method described in chapter 2 has been used to solve this problems. By testing two model problems, we have found that this method gives more accurate solutions than those given by Lardner and Das[50].

5.1 *Basic Equations*

The equations that form the basis of the model are the usual momentum and mass conservation equation. Flows driven by density gradients are generally quite slow, so that the advective terms in the horizontal momentum equations are very small and can be ignored. These equations then take the form:

The mass conservation equation:

$$(\rho u)_x + (\rho v)_y + (\rho w)_z = 0 \tag{5.1}$$

The horizontal momentum equations, corresponding to (1.2)-(1.3):

$$\rho\frac{\partial u}{\partial t} - f\rho v = -\frac{\partial p}{\partial x} + \frac{\partial}{\partial z}\left(\mu\frac{\partial u}{\partial z}\right) \tag{5.2}$$

$$\rho\frac{\partial v}{\partial t} + f\rho u = -\frac{\partial p}{\partial y} + \frac{\partial}{\partial z}\left(\mu\frac{\partial v}{\partial z}\right) \tag{5.3}$$

As usual, the vertical momentum equation is approximated by the hydrostatic equation

$$p(x,y,z,t) = p_0 + \int_z^{\zeta} g\rho(x,y,z')dz' \tag{5.4}$$

where density ρ is depends on x, y and z; p_0 is atmospheric pressure, assumed constant.

Defining the average density and the components of mass transport by

$$\bar{\rho} = \frac{1}{H}\int_{-h}^{\zeta}\rho dz \tag{5.5}$$

$$p = \int_{-h}^{\zeta}\rho u dz \quad , \quad q = \int_{-h}^{\zeta}\rho v dz \tag{5.6}$$

and integrating equation (5.1) over the water column from $z = -h$ to $z = \zeta$, we get , after using (1.5)-(1.6),

$$\rho^{(s)}\zeta_t + p_x + q_y = 0 \tag{5.7}$$

where $\rho^{(s)}$ denotes the density on the surface.

Substituting (5.4) into equation (5.2)-(5.3), we obtain

$$\rho\frac{\partial u}{\partial t} - fgv - \frac{\partial}{\partial z}\left(\mu\frac{\partial u}{\partial z}\right) = -g\rho^{(s)}\zeta_x - g\int_{z}^{\zeta}\frac{\partial}{\partial x}\rho dz' \tag{5.8}$$

$$\rho\frac{\partial v}{\partial t} + fgu - \frac{\partial}{\partial z}\left(\mu\frac{\partial v}{\partial z}\right) = -g\rho^{(s)}\zeta_y - g\int_{z}^{\zeta}\frac{\partial}{\partial y}\rho dz' \tag{5.9}$$

It is convenient to write the equations in terms of the sigma coordinate in the vertical direction. By setting $\sigma=(z+h)/H$, where $H=h+\zeta$ is the total water depth, the free surface is then $\sigma=1$ while the bottom is $\sigma=0$. Using the chain rule shown in chapter 1, we have.

$$\rho\frac{\partial u}{\partial t} - \frac{\rho\sigma\zeta_t}{H}u_\sigma - fgv - H^{-2}\frac{\partial}{\partial\sigma}\left(\mu\frac{\partial u}{\partial\sigma}\right) = -g\rho\zeta_x + T_x \tag{5.10}$$

$$\rho\frac{\partial v}{\partial t} - \frac{\rho\sigma\zeta_t}{H}v_\sigma + fgu - H^{-2}\frac{\partial}{\partial\sigma}\left(\mu\frac{\partial v}{\partial\sigma}\right) = -g\rho\zeta_y + T \tag{5.11}$$

where

$$\begin{aligned} T_x &= H\int_{\sigma}^{1}\left[R_x - H(1-\sigma)\rho_x\right]d\sigma' \\ T_y &= H\int_{\sigma}^{1}\left[R_y - H(1-\sigma)\rho_y\right]d\sigma' \end{aligned} \tag{5.12}$$

and

$$R = H\int_{\sigma}^{1}[\rho(\sigma)-\rho(\sigma')]d\sigma' \tag{5.13}$$

In addition, we have boundary conditions on σ =0 and 1. We assume that the surface is free of shear traction, so we have the surface condition

$$u_{\sigma} = v_{\sigma} = 0 \tag{5.14}$$

On the bottom the boundary condition is

$$\mu u_{\sigma} = H\bar{\rho}(\kappa_1 + \kappa_2\sqrt{u_b^2 + v_b^2})u_b \quad \mu v_{\sigma} = H\bar{\rho}(\kappa_1 + \kappa_2\sqrt{u_b^2 + v_b^2})v_b \tag{5.15}$$

In general it is physically more realistic to use a quadratic dependence on bottom friction on velocity, but in the present case, where the density-driven flow is superimposed on other, possibly much larger, flows, such as tidal and wind-driven currents, it may be more appropriate to use the linear form for bottom friction[25,50], that is

$$\mu(u_{\sigma},v_{\sigma}) = H\bar{\rho}\kappa(u,v) \qquad on \quad \sigma = 0 \tag{5.16}$$

Writing the above problem in complex form, U=u + iv , we have

$$\rho^{(s)}\zeta_t + p_x + q_y = 0 \tag{5.17}$$

and

$$\rho U_t - \frac{\rho\sigma\zeta_t}{H}U_{\sigma} + if\rho U - H^{-2}\frac{\partial}{\partial\sigma}\left(\mu\frac{\partial U}{\partial\sigma}\right) = -g\rho(\zeta_x + i\zeta_y) + T_x + iT_y \tag{5.18}$$

with boundary conditions

$$U_{\sigma} = 0 \quad on \quad \sigma = 1 \tag{5.19}$$

and

$$\mu U_{\sigma} = H\bar{\rho}\kappa U \quad on \quad \sigma = 0 \tag{5.20}$$

5.2 *Eddy-viscosity eigenexpansion*

In order to solve problem (5.17)-(5.20), we consider the eddy-viscosity eigenvalue problem:

$$\frac{\partial}{\partial\sigma}\left(\mu\frac{\partial\phi}{\partial\sigma}\right)+\lambda\rho\phi(\sigma)=0 \tag{5.21}$$

$$\mu\frac{\partial\phi}{\partial\sigma}=0 \quad \text{on } \sigma=0 \ , \ \mu\frac{\partial\phi}{\partial\sigma}-\kappa h\bar{\rho}\phi=0 \quad \text{on } \sigma=1 \tag{5.22}$$

Using the SLEIGN subroutine to compute the eigenpairs $\{\lambda_j, \phi_j(\sigma)\}$ at every point (x,y), we have the orthogonal eigenfunctions with properties:

$$\int_0^1 \rho\phi_i\phi_j d\sigma=\begin{cases}1 \ , & \text{when } i=j\\ 0 \ , & \text{when } i\neq j\end{cases} \tag{5.23}$$

Expanding the current profile U in terms of the eigenfunctions, we obtain

$$U(\sigma)=\sum_{j\geq 1}c_j(x,y,t)\phi_j(\sigma) \tag{5.24}$$

where

$$c_j(x,y,t)=\int_0^1\rho U(\sigma)\phi_j(\sigma)d\sigma \tag{5.25}$$

Multiplying (5.18) by $\phi_j(\sigma)$ and integrating from $\sigma=0$ to 1, we get

$$\frac{\partial c_j}{\partial t}+(if+\frac{\lambda_j}{h^2})c_j=R_{1j}\left[\zeta_x+i\zeta_y\right]+R_{3j} \tag{5.26}$$

where

$$R_{1j}=-g\int_0^1\rho\phi_j d\sigma$$

$$R_{3j}=\int_0^1[T_x+iT_x]\phi_j d\sigma$$

The initial values for the set of equations (5.26) can be obtained by assuming the motion starts from a state of rest, then

$$U|_{t=0} = 0$$

and

$$c_j|_{t=0} = 0 \tag{2.27}$$

The mass transports can be obtained by combining (5.6) and (5.24) as

$$p + iq = \sum_{j \ge 1} c_j(x,y,t) H \int_0^1 \rho \phi_j d\sigma \tag{2.28}$$

We can compute the steady currents caused by a given density field by solving equations (5.16) and (5.26) with appropriate boundary conditions and initial values. The discretization scheme shown in chapter 3 has been used. The horizontal boundary conditions are specified as usual. That is, on coastal boundaries, the normal component of the mass flux vector (p , q) is taken to be zero; the appropriate condition on the open part of the boundary is the subject of considerable debate, but the simplest condition is to set ζ equal to zero at all open boundary points.

5.3 *Test Problems*

The accuracy of the algorithm was tested by using it solve two problem for which the exact steady solutions can be calculated. These solutions, taken from among those given by Lardner and Das[50], were designed to test the code's accuracy in handling two distinct physical features: vertical density variation and Coriolis forces.

Problem 1

For the first problem, we consider a channel occupying the region 0<*x* <L with the end *x*=0 being closed and *x*=L open. The eddy-viscosity μ is assumed constant and bottom drag is assumed linear (κ_2=0). Only longitudinal flow is considered, with *v* and *f* take as zero.

In this case we take the density to have the form

$$\rho = \rho_0[1 + x\beta(1 - 2\sigma)] \tag{5.29}$$

and the water depth *h* =*h* (*x*) was taken to increasing uniformly from 35m at the closed end of the channel to 95m at the open end.

For steady flow, equation (5.17) then reduces to p_x =0, and since *p* =0 at *x* =0, it follows that *p* =0 for all *x* . Therefore,

$$\int_0^1 \rho u d\sigma = 0 \tag{5.30}$$

Equation (5.18) then reduces to

$$(\mu u_\sigma)_\sigma = g\rho h^2\zeta_x - \beta\rho_0 g\left[h^3\sigma(1-\sigma) + h^2h'x(1-\sigma)^2\right] \tag{5.31}$$

Taking the boundary conditions (5.19)-(5.20) into account, we obtain the bottom velocity

$$u_b = -\frac{U}{15D}\left[1 + 5\beta x - 2(\beta x)^2 - S(2 - \beta x)^2\right] \tag{5.32}$$

Where

$$U = \frac{\beta g h^3 \rho_0}{24\mu}$$

$$D = 1 + 2K\left[2 - \beta x + (\beta x)^2 / 2\right]$$

$$K = \frac{\kappa_1 h \rho_0}{12\mu}$$

$$S = 4xh'(x) / h(x)$$

The rest of the solution is given by

$$u = u_b\left\{1 + 6k\left[2\sigma - \sigma^2 - \beta x \sigma^2(1 - 2\sigma/3)\right]\right\}$$

$$+ 2U\sigma^2\left[(1-\sigma)^2 + (1 + S/2)\beta x(1 - 2\sigma/3) - S(2-\sigma)^2/4\right] \qquad (5.33)$$

$$\zeta_x = \frac{1}{6}\beta h\left\{1 + S/2 - \frac{3Ku_b}{U}\right\} \qquad (5.34)$$

In the numerical results given below, the channel was taken with the closed end at grid point m=1 and the open end at m=18.5, the horizontal grid spacing being δ =40,000m.

The constants ρ_0=1.013, β =1/17 ,The other constants used were(in MKS units) g=9.81, κ_1=0.002 and μ =0.065. The time step was τ =360s, with the computation being run until a steady solution was obtained (typically, this took about 3000 steps, corresponding to 12 days of real time). The computed results are given in Table 5.1, which shows the exact and computed velocity profiles at three positions along the channel. The solutio s are given at six equally spaced levels form the top to the bottom of the water column.The two solutions are in very closed agreement.

Exact Solutions:

lev\ m	4	10	16
11	-0.173	-0.987	-2.696
9	-0.113	-0.714	-1.985
7	-0.005	-0.156	-0.480
5	0.072	0.372	1.013
3	0.093	0.661	1.868
1	0.074	0.601	1.635

Computed Solutions:

lev\ m	4	10	16
11	-0.177	-0.995	-2.710
9	-0.115	-0.714	-1.989
7	-0.005	-0.154	-0.473
5	0.075	0.378	1.025
3	0.095	0.665	1.870
1	0.076	0.601	1.623

Table 5.1 Exact and Computed velocity profiles at three points along the channel, in units of cm/s.

Problem 2

The second problem is designed to test the accuracy of the computer code's treatment of the Coriolis terms. We consider a rectangular body of water of constant depth, occupying the region $0<x<L$, $0<y<M$, with lateral boundary conditions $p=0$ on the sides x =0, L and q having certain prescribed values, to be given below, on the sides $y=0$, M. κ_2 is again zero and f, m and κ_1 constant. We suppose that the density $\rho(x)$ is a function of x only, and that all flow variables are independent of y , we define it as $\rho=\rho_0(1-\beta x)$. In this case, equations (5.10)-(5.11), for steady flow, reduce to

$$(\mu u_\sigma)_\sigma + \rho h^2 f v - g\rho h^3 \zeta_x - \rho h^3 \rho'(x)(1-\sigma) = 0 \tag{5.35}$$

$$(\mu v_\sigma)_\sigma - \rho h^2 f u = 0 \tag{5.36}$$

The continuity equation reduces to $p_x=0$, and in view of the lateral boundary conditions therefore, $p=0$. Equation (5.36) implies that $v_b=0$ on $\sigma=0$, so that the boundary conditions associated with equation (5.37) are

$$u_\sigma = v_\sigma = 0 \quad on \quad \sigma = 1$$
$$v = v_\sigma = \mu u_\sigma - \kappa_1 h\rho u = 0 \quad on \quad \sigma = 0$$

The general solution of equations (3.35)-(5.36) has the form

$$(u,v) = \frac{gh\rho'}{f\rho}(\alpha,\beta) \ , \quad \zeta_x = \frac{h\rho'}{\rho}\gamma$$

where

$$\alpha = e^{r\sigma}(a\cos r\sigma - b\sin r\sigma) + e^{-r\sigma}(c\cos r\sigma + d\sin r\sigma)$$
$$\beta = \gamma + 1 - \sigma + e^{r\sigma}(b\cos r\sigma + a\sin r\sigma) + e^{-r\sigma}(d\cos r\sigma - c\sin r\sigma)$$

The terms involving a,b,c and d are of course an Ekman spiral type of solution. These constants as well as γ are determined from the above boundary conditions, which take the form

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & -1 & -1 \\ 0 & \lambda-1 & 1 & \lambda+1 & -1 \\ 0 & -e^{r}\cos r & e^{r}\sin r & e^{-r}\cos r & e^{-r}\sin r \\ 0 & e^{r}\sin r & e^{r}\cos r & e^{-r}\sin r & -e^{-r}\cos r \end{bmatrix} \begin{bmatrix} \gamma+1 \\ a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 1/r \\ 0 \\ -1/2r \\ 1/2r \end{bmatrix}$$

where

$$r = \sqrt{\rho h^2 f / 2\mu} \ , \ \lambda = \kappa_1 h \rho / r\mu$$

This matrix equation is solved numerically to construct the exact solution.

For the computed solution, we require the boundary values of q on y=0, M, which are taken from the above exact solution. It is easily seem from (5.35)-(5.36) and the above solution that

$$q = \frac{gh^2\rho'}{f}\left[\gamma + \frac{1}{2} + \frac{\lambda}{2r}(a+c)\right]$$

The solution has been computed for a rectangle of dimensions 600 kms in the x-direction and 200 kms in the y-direction with a grid spacing of 20,000m. The parameters were taken as in problem 1 with the addition of f=1.22*10^{-4} s^{-1}. Table 5.2 gives the exact and computed velocities at three points of the rectangle. There is very little variation of the final steady solution with x and the accuracy of the computed solution is about the same throughout the rectangle.

Exact Solutions:

\ point	(3, 2)		(15,9)		(27,2)	
level \	u	v	u	v	u	v
11	-16.85	25.73	-16.91	25.71	-16.97	25.69
9	-12.81	23.08	-12.85	23.06	-12.90	23.05
7	-3.93	16.36	-3.95	16.35	-3.96	16.34
5	5.76	8.37	5.77	8.37	5.79	8.36
3	12.67	2.20	12.71	2.20	12.75	2.20
1	12.61	0.00	12.67	0.00	12.72	0.00

Computed Solutions:

\ point	(3, 2)		(15,9)		(27,2)	
level \	u	v	u	v	u	v
11	-16.84	25.73	-16.90	25.71	-16.95	25.69
9	-12.80	23.08	-12.84	23.06	-12.90	23.05
7	-3.94	16.36	-3.96	16.35	-3.97	16.34
5	5.77	8.37	5.79	8.37	5.79	8.36
3	12.67	2.20	12.72	2.20	12.76	2.20
1	12.60	0.00	12.62	0.00	12.78	0.00

Table 5.2 Exact and Computed velocity profiles at three points in the rectangle region, in units of cm/s.

Chapter 6 Convection-diffusion Problem

6.1. *Introduction and Basic equation*

We shall be concerned with modelling the behaviour of dissolved or suspended substances, such as pollutants, in the marine environment. For non-reactive substances, this is governed by the convection-diffusion equation,

$$S_t + uS_x + vS_y + wS_z = (D_h S_x)_x + (D_h S_y)_y + (D_v S_z)_z + F_s. \tag{6.1}$$

Here, x,y and z are Cartesian coordinates with the xy - plane horizontal and occupying the undisturbed position of the water surface, and the z -axis pointing vertically upwards. The density of the pollutant at the point (x,y,z) and time t is denoted by $S(x,y,z,t)$ and subscripts of x,y,z or t denote the corresponding partial derivative. The components of fluid velocity at (x,y,z,t) are denoted by u,v and w and D_h and D_v are respectively the horizontal and vertical diffusivities. F_s represents any source or sink.

The position of the free surface is denoted by $z = \zeta(x,y,t)$ and that of the bottom by $z = -h(x,y)$. It is assumed that ζ as well as u,v and w are known as functions of their respective arguments from some model of the hydrodynamics of the region in question.

In addition to (6.1), S satisfies initial conditions

$$S(x,y,z,0) = S_0(x,y,z) \tag{6.2}$$

and boundary conditions on the top and bottom surfaces

$$S_z = b_1(x,y,t) \text{ on } z = \zeta, \qquad S_z = b_2(x,y,t) \text{ on } z = -h, \tag{6.3}$$

and on the horizontal boundaries

$$\frac{\partial S}{\partial n} = 0 \text{ on } B_1, \qquad S = 0 \text{ on } B_2 \tag{6.4}$$

Here, B_1 and B_2 are the coastal and open boundaries respectively with $\boldsymbol{n}$ the outward normal in the xy -plane. In most applications, the fluxes b_1 and b_2 would be zero, but they are included for the purpose of certain test problems to be used later.

First, consider the feasibility of using an explicit finite difference method to compute numerical solutions to (6.1). Ignoring the convection part and the boundary conditions, the stability condition on such an algorithm would be [72-73]

$$\tau \le \frac{\Delta x^2}{4D_h} \quad \text{and} \quad \tau \le \frac{\Delta z^2}{2D_v} \tag{6.5}$$

where τ is the time step, Δx the horizontal grid spacing and Δz the vertical spacing. Later in the chapter, more complete stability criteria will be obtained, but these will suffice for the present preliminary estimates. A typical horizontal grid dimension would usually be in the range 1-20 km and horizontal diffusivities in the range 10-100 m^2 /s, depending mainly on the degree of turbulence. Within these ranges, the first of the above restrictions on the time-step is, in the worst case, $\tau \le 2500$s and generally it is much less restrictive than this. On the other hand, a typical vertical grid spacing is 5-20m and vertical diffusivity 0.1-1 m^2 /s, so the second of the above restrictions is, in the worst case, $\tau \le 12.5$s.

It is clear from these estimates that the stability limit on the time-step coming from the horizontal grid does not present a problem except perhaps for very fine-scale models of turbulent regions. On the other hand, the restriction from the vertical grid is often a serious difficulty, and effectively prevents the use of an explicit method. This is the problem that in the next two chapters we want to solve by a vertical / horizontal splitting method.

The variable z has a range $-h \le z \le \zeta$ that varies with time. For numerical work it is better to change to a new variable that has a fixed range. The usual choice is

$$\sigma = (z+h)/H, \qquad H = \zeta + h \tag{6.6}$$

so that $\sigma = 0$ at the bottom and $\sigma = 1$ at the top. Using the chain rule repeatedly, we can then rewrite (6.1) in the form

$$S_t - \frac{g}{H} S_\sigma - \frac{1}{H^2}(N S_\sigma)_\sigma = F \tag{6.7}$$

where

$$\begin{aligned} N &= D_h(X^2 + Y^2) + D_v, \\ g &= \sigma\zeta_t - \tilde{w} + D_h[X_x + Y_y], \\ F &= D_h(S_{xx} + S_{yy}) + 2H^{-1} D_h (X S_{x\sigma} + Y S_{y\sigma}) - uS_x - vS_y + F_s \end{aligned} \tag{6.8}$$

and

$$X = (1-\sigma)h_x - \sigma\zeta_x, \quad Y = (1-\sigma)h_y - \sigma\zeta_y \quad \text{and} \quad \tilde{w} = w + uX + vY . \tag{6.9}$$

It has been assumed here that D_h and D_v are independent of x and y, otherwise certain extra terms occur in g and F. Note that all derivatives with respect to x and y have been included on the left side of eqn (6.7), in accordance with the intention of treating these derivatives explicitly.

In terms of σ the boundary conditions (6.3) take the form

$$S_\sigma = Hb_1 \text{ on } \sigma = 1, \qquad S_\sigma = Hb_2 \text{ on } \sigma = 0. \tag{6.10}$$

6.2. *Discretization of the Problem*

First we discretize time in eqn (6.7). Let τ denote the size of the time-step, $S(\sigma)$ the approximation to the solution at the current step and $S^+(\sigma)$ the approximation at the next time step. We set

$$S^\lambda(\sigma) = \lambda S^+(\sigma) + (1-\lambda)S(\sigma), \tag{6.11}$$

where λ is an implicitness parameter, and approximate eqns (6.7) and (6.10) as

$$\frac{S^+ - S}{\tau} - \frac{g}{H} S_\sigma^\lambda - \frac{1}{H^2}(NS_\sigma^\lambda)_\sigma = F \tag{6.12}$$

$$S_\sigma^\lambda(1) = Hb_1^\lambda, \quad S_\sigma^\lambda(0) = Hb_2^\lambda. \tag{6.13}$$

The terms in the right side F that involve horizontal derivatives of S are evaluated explicitly, that is, at the current time level, while the other terms in F are evaluated midway between the current and new time-steps in order to minimize the errors.

Treating S^λ as the new unknown, we can rewrite eqn (6.12) as

$$\frac{1}{\lambda\tau} S^\lambda - \frac{g}{H} S_\sigma^\lambda - \frac{1}{H^2}(NS_\sigma^\lambda)_\sigma = F_c \tag{6.14}$$

where $F_c = F + S/\lambda\tau$. Once S^λ is determined, $S^+(\sigma)$ can be found from eqn (6.11) as

$$S^+(\sigma) = [S^\lambda(\sigma) - (1-\lambda)S(\sigma)]/\lambda.$$

This extrapolation is stable provided that $\frac{1}{2} \le \lambda \le 1$.

Next, we discretize the vertical coordinate in (6.4). In order to do this, as in [47] we replace (6.14) and the boundary conditions (6.13) by the integral identity

$$\begin{aligned} H\int_0^1 \left\{ V(\sigma)\left[\frac{1}{\lambda\tau} S^\lambda - \frac{g}{H} S_\sigma^\lambda\right] + V_\sigma(\sigma)\frac{N}{H^2} S_\sigma^\lambda \right\} d\sigma \\ = H\int_0^1 V(\sigma) F_c(\sigma)\, d\sigma + Nb_1^\lambda V(1) + Nb_2^\lambda V(0) \end{aligned} \tag{6.15}$$

where V is an arbitrary differentiable function. Within the class of C^2 functions, (6.15) is equivalent to the boundary value problem (6.13),(6.14).

This identity is now discretized using finite elements as follows. We choose J equally spaced nodes, $\{\sigma_j\}$ in the interval $[0,1]$ with $\sigma_1 = 0$, $\sigma_J = 1$, and spacing $\Delta\sigma = 1/(J - 1)$. We approximate each function piecewise linearly in each sub-interval, for example,

$$/ \, (\sigma) \sim V_j \frac{\sigma_{j+1} - \sigma}{\Delta\sigma} + V_{j+1} \frac{\sigma - \sigma_{j+1}}{\Delta\sigma}, \qquad \sigma \in [\sigma_j, \sigma_{j+1}],$$

where V_j is the value of V at node j. Then, after evaluating the integrals, we obtain identity (6.15) in the form

$$\begin{aligned}
&\frac{k}{\lambda\tau}\sum_{j=1}^{J-1} V_j \left\{\tfrac{1}{3}S_j^{\lambda} + \tfrac{1}{6}S_{j+1}^{\lambda}\right\} + \frac{k}{\lambda\tau}\sum_{j=2}^{J} V_j \left\{\tfrac{1}{3}S_j^{\lambda} + \tfrac{1}{6}S_{j-1}^{\lambda}\right\} \\
&- \sum_{j=1}^{J-1} V_j \left\{\tfrac{1}{3} g_j + \tfrac{1}{6} g_{j+1}\right\}\left(S_{j+1}^{\lambda} - S_j^{\lambda}\right) - \sum_{j=2}^{J} V_j \left\{\tfrac{1}{3} g_j + \tfrac{1}{6} g_{j-1}\right\}\left(S_j^{\lambda} - S_{j-1}^{\lambda}\right) \\
&- \frac{1}{2k}\sum_{j=1}^{J-1} V_j \left\{N_j + N_{j+1}\right\}\left(S_{j+1}^{\lambda} - S_j^{\lambda}\right) + \frac{1}{2k}\sum_{j=2}^{J} V_j \left\{N_j + N_{j-1}\right\}\left(S_j^{\lambda} - S_{j-1}^{\lambda}\right) \\
&= k\sum_{j=1}^{J-1} V_j \left\{\tfrac{1}{3}F_{cj} + \tfrac{1}{6}F_{c,j+1}\right\} + k\sum_{j=2}^{J} V_j \left\{\tfrac{1}{3}F_{cj} + \tfrac{1}{6}F_{cj-1}\right\} + N_J b_1^{\lambda} V_J - N_1 b_2^{\lambda} V_1 .
\end{aligned}$$

where $k = H\,\Delta\sigma$ is the vertical grid spacing in physical units. Equating the coefficients of each V_j we then obtain the system of equations

$$-P_j S_{j-1}^{\lambda} + Q_j S_j^{\lambda} - R_j S_{j+1}^{\lambda} = W_j, \qquad j = 1, \ldots, J, \tag{6.16}$$

where, for $2 \le j \le J - 1$,

$$\begin{aligned}
P_j &= -\frac{k}{6\lambda\tau} - \frac{g_j}{3} - \frac{g_{j-1}}{6} + \frac{1}{2k}\left(N_{j-1} + N_j\right) \\
R_j &= -\frac{k}{6\lambda\tau} + \frac{g_j}{3} + \frac{g_{j+1}}{6} + \frac{1}{2k}\left(N_{j+1} + N_j\right) \\
Q_j &= P_j + R_j + \frac{k}{\lambda\tau} \\
W_j &= k\left\{\tfrac{1}{6}F_{c,j-1} + \tfrac{2}{3}F_{cj} + \tfrac{1}{6}F_{cj+1}\right\};
\end{aligned} \tag{6.17}$$

for $j = 1$,

$$
\begin{aligned}
&P_1 = 0\\
&R_1 = -\frac{k}{6\lambda\tau} + \frac{g_1}{3} + \frac{g_2}{6} + \frac{1}{2k}(N_2 + N_1)\\
&Q_1 = R_1 + \frac{k}{2\lambda\tau}\\
&W_1 = k\left\{\tfrac{1}{3}F_{c,1} + \tfrac{1}{6}F_{c,2}\right\} - N_1 b_2^{\lambda};
\end{aligned}
\qquad (6.18)
$$

and for $j = J$,

$$
\begin{aligned}
&P_J = -\frac{k}{6\lambda\tau} - \frac{g_J}{3} - \frac{g_{J-1}}{6} + \frac{1}{2k}(N_{J-1} + N_J)\\
&R_j = 0\\
&Q_J = P_J + \frac{k}{2\lambda\tau}\\
&W_J = k\left\{\tfrac{1}{6}F_{c,J-1} + \tfrac{1}{3}F_{c,J}\right\} + N_J b_1^{\lambda}.
\end{aligned}
\qquad (6.19)
$$

The system (6.16) can be written in matrix form as $\mathbf{A}\mathbf{S}^{\lambda} = \mathbf{W}$ where $\mathbf{A}$ is tri-diagonal. It is solved by the usual decomposition $\mathbf{A} = \mathbf{L}\mathbf{U}$ where

$$
\mathbf{L} = \begin{bmatrix} 1 & & & & \\ L_2 & 1 & & & \\ & L_3 & 1 & & \\ & & \ddots & \ddots & \\ & & & L_J & 1 \end{bmatrix}, \quad
\mathbf{U} = \begin{bmatrix} U_1 & -R_1 & & & \\ & U_2 & -R_2 & & \\ & & U_3 & \ddots & \\ & & & \ddots & -R_{J-1} \\ & & & & U_J \end{bmatrix}.
$$

The elements of $\mathbf{L}$ and $\mathbf{U}$ are determined recursively from

$$
U_1 = Q_1, \quad U_j = Q_j - \frac{P_j R_{j-1}}{U_{j-1}}, \quad L_j = -\frac{P_j}{U_{j-1}}, \quad (j = 2, \ldots, J) \qquad (6.20)
$$

The solution of $\mathbf{A}\mathbf{S}^{\lambda} = \mathbf{W}$ is then obtained in two stages, by the forward elimination, $\mathbf{L}\mathbf{Y} = \mathbf{W}$, i.e.,

$$
Y_1 = W_1, \quad Y_j = W_j - L_j Y_{j-1}, \quad (j = 2, \ldots, J) \qquad (6.21)
$$

followed by the back substitution $\mathbf{U}\mathbf{S}^{\lambda} = \mathbf{Y}$, i.e.,

$$S_J^\lambda = \frac{Y_J}{U_J}, \quad S_j^\lambda = \frac{Y_j + R_j S_{j+1}^\lambda}{U_j}, \quad (j = J-1,\dots,1). \tag{6.22}$$

Finally, we discretize the horizontal derivatives, which have all been placed in the right side functions F in eqns (6.7) and (6.8), and are evaluated in terms of the current value S. The stability restriction arising from this explicit treatment will be discussed in the next section. The second derivatives are evaluated using a central difference,

$$S_{xx} = \frac{S_{m+1} - 2S_m + S_{m-1}}{\Delta x^2}, \qquad S_{yy} = \frac{S_{n+1} - 2S_n + S_{n-1}}{\Delta y^2},$$

where m and n are the grid indices in the x - and y - directions, while the first derivatives are evaluated using an up-wind difference,

$$S_x = \begin{cases} \dfrac{S_m - S_{m-1}}{\Delta x} & \text{if } u > 0 \\ \dfrac{S_{m+1} - S_m}{\Delta x} & \text{if } u < 0 \end{cases}, \qquad S_y = \begin{cases} \dfrac{S_n - S_{n-1}}{\Delta y} & \text{if } v > 0 \\ \dfrac{S_{n+1} - S_n}{\Delta y} & \text{if } v < 0. \end{cases} \tag{6.23}$$

6.3 *Stability restrictions*

First we discuss the stability of the factorization and back substitution (6.20)-(6.22). The theorem given by Smith [73, pp 27-28] cannot be applied to the present case because the elements P_j and R_j in A are not necessarily positive. However, a modification of Smith's argument can be made to show that the forward elimination (6.21) and back substitution (6.22) are stable provided we make the restriction

$$|g_j| < \min\left\{\frac{k}{2\lambda\tau}, \frac{k}{6\lambda\tau} + \frac{6N}{5k}\right\} \tag{6.24}$$

where $N = \min_j N_j$. The proof of this result is rather lengthy and will be given in the Appendix. It is worth noting that the theorem given by Golub and van Loan [79, Sec. 5.5] can be used to obtain a sufficient condition for the forward elimination (6.21) to be stable, but this condition does not guarantee the stability of the back substitution (6.22).

When $\zeta = 0$ and h is constant, we have from eqns (6.8) and (6.9) that $g = -w$ and $N = D_v$. More generally, on the basis of the order of magnitude estimates that are commonly applied to flows in coastal seas, it can be estimated that $N \approx D_v$ and g always has the same order of magnitude as w. Therefore, the above restriction is approximately equivalent to

$$|w| < \frac{k}{2\lambda\tau} \quad \textit{and} \quad |w| < \frac{k}{6\lambda\tau} + \frac{6D_v}{5k} \tag{6.25}$$

In most cases, the first of conditions (6.25) is more restrictive than the second. Assuming the value $\lambda = 0.5$, it requires that the convective displacement per time step should not exceed the vertical grid spacing k. For example, with $\lambda = 0.5$ and taking typical values of vertical grid spacing as 5m, D_V.= 0.1 m^2/s and time step as 1000s, these two conditions are $|w| < 0.005$ and 0.026 m/s respectively. These are not serious restrictions. When the time step is short enough that the second condition becomes the more restrictive, the bound on $|w|$ is even less serious. For example if k = 10m and τ = 500s, the two conditions are $|w| < 0.02$ and 0.019 m/s respectively.

The von Neumann stability of various explicit finite difference schemes for the multidimensional convection-diffusion equation was investigated by Hindmarsh, Gresho and Griffiths [80]. For the two-dimensional up-winding scheme we have used, their condition (92), which is necessary and sufficient for stability, is

$$\tau \leq \left[\frac{4D_H}{\Delta x^2} + \frac{|u| + |v|}{\Delta x} \right]^{-1} \tag{6.26}$$

When u and v are zero, this is the condition (6.5_1) quoted in the Section 6.1, and, as remarked there, for convection-diffusion in the marine environment, this restriction on the time step is usually not a severe limitation. However, when u and v are significantly non-zero, the stability condition is different from (6.5_1). In particular, when $|u|$ and $|v|$ are much greater than $4D_H / \Delta x$, the the stability becomes convection-limited rather than diffusion-limited and the condition reduces to

$$\tau \leq \frac{\Delta x}{|u| + |v|} \tag{6.27}$$

While again this is not usually a severe limitation, it is often more so than the condition (6.5_1). In the typical marine pollution problem the convection and diffusion contributions to the condition (6.26) are of comparable magnitudes.

The conditions (6.24) and (6.26) have been derived for the separate horizontal and vertical convection algorithms, and it is relevant to ask if they are valid for the joint three-dimensional algorithm. While we do not have a rigorous answer to this question, we have carried out numerous numerical experiments to obtain at least an empirical answer. These are reported in Section 7.3.

APPENDIX. PROOFS OF STABILITY THEOREMS

In this Appendix we shall give proofs of the stability conditions stated in Section 6.3. We first prove that the factorization used in the vertical part of the algorithm is stable provided

$$g < \frac{k}{2\lambda\tau}\text{, and } g < \frac{k}{6\lambda\tau} + \frac{6N}{5k} \qquad \text{(A.1)}$$

wher $g = \max_j |g_j|$ *and* $N = \min_j |N_j|$

The forward elimination (6.21) is stable if $|L_j| \le 1$ for all j. From (6.20), we have

$$L_{j+1} = -\frac{P_{j+1}}{Q_j + R_{j-1}L_j}, \qquad L_2 = -\frac{P_2}{Q_1}. \qquad \text{(A.2)}$$

First, we prove that $|L_2| \le 1$. From (6.18),

$$Q_1 = \frac{k}{3\lambda\tau} + \frac{g_1}{3} + \frac{g_2}{6} + \frac{1}{2k}(N_2 + N_1) > \frac{k}{3\lambda\tau}(1 - \tfrac{1}{6} - \tfrac{1}{12}) + \frac{1}{2k}(N_2 + N_1) > 0$$

where we have used (A.1). Then the required condition $|L_2| \le 1$ is equivalent to $-Q_1 \le P_2 \le Q_1$, or, $-Q_1 - P_2 \le 0 \le Q_1 - P_2$. From (6.17) and (6.18), this becomes

$$-\frac{k}{6\lambda\tau} - \frac{g_1}{6} + \frac{g_2}{6} - \frac{1}{k}(N_2 + N_1) \le 0 \le \frac{k}{2\lambda\tau} + \frac{g_1}{2} + \frac{g_2}{2}$$

and it is easily seen that these inequalities are satisfied if (A.1) holds.

Next, we proceed by induction, assuming that $|L_j| \le 1$ and using the recursion formula (A.2$_1$). The denominator here is again positive, since

$$\begin{aligned} Q_j + R_{j-1}L_j &= \frac{2k}{3\lambda\tau} + \frac{g_{j+1}}{6} - \frac{g_{j-1}}{6} + \frac{1}{2k}(N_{j+1} + 2N_j + N_{j-1}) \\ &\quad + L_j\left[-\frac{k}{6\lambda\tau} + \frac{g_{j-1}}{3} + \frac{g_j}{6} - \frac{1}{2k}(N_j + N_{j-1})\right] \\ &\ge \frac{k}{2\lambda\tau} - \frac{5g}{6} + \frac{1}{2k}(N_{j+1} + N_j) \\ &\ge \frac{k}{2\lambda\tau} - \frac{5g}{6} + \frac{N}{k} > 0 \end{aligned}$$

where again (A.1) has been used. Finally, we want $|L_{j+1}| \le 1$ and from (A.2$_1$) this is equivalent to $-(Q_j + R_{j-1}L_j) - P_{j+1} \le 0 \le Q_j + R_{j-1}L_j - P_{j+1}$. From the above and eqns (6.17), we have

$$\begin{aligned}(Q_j + R_{j-1}L_j) + P_{j+1} &\ge \frac{k}{2\lambda\tau} - \frac{5g}{6} + \frac{1}{2k}(N_{j+1} + N_j) \\ &+ \left[-\frac{k}{6\lambda\tau} - \frac{g_{j+1}}{3} - \frac{g_j}{6} + \frac{1}{2k}(N_j + N_{j+1})\right] \\ &\ge \frac{k}{3\lambda\tau} - \frac{4g}{3} + \frac{2N}{k} > 0\end{aligned}$$

and

$$\begin{aligned}(Q_j + R_{j-1}L_j) - P_{j+1} &\ge \frac{k}{2\lambda\tau} - \frac{5g}{6} + \frac{1}{2k}(N_{j+1} + N_j) \\ &+ \left[-\frac{k}{6\lambda\tau} - \frac{g_{j+1}}{3} - \frac{g_j}{6} + \frac{1}{2k}(N_j + N_{j+1})\right] \\ &\ge \frac{k}{3\lambda\tau} - \frac{4g}{3} > 0\end{aligned}$$

as required.

The back substitution (6.22) can be re-written as

$$S_j^{\lambda} = M_j S_{j+1}^{\lambda} + Y_j / U_j \qquad \text{where } M_j = R_j / U_j$$

and therefore is stable if $|M_j| \le 1$ for all j. From (6.20), we have

$$M_j = \frac{R_j}{Q_j - P_j M_{j-1}}, \qquad M_1 = \frac{R_1}{Q_1}. \tag{A.3}$$

First, we prove that $|M_1| \le 1$. We already showed that $Q_1 > 0$ if (A.1) holds. so we need to prove that $-Q_1 \le R_1 \le Q_1$. From (6.18$_3$), the right part of this inequality is always satisfied, while the left part is equivalent to

$$R_1 + \frac{k}{4\lambda\tau} \ge 0.$$

Then from (8.18$_2$),

$$R_1 + \frac{k}{4\lambda\tau} = \frac{k}{12\lambda\tau} + \frac{g_1}{3} + \frac{g_2}{6} + \frac{1}{2k}(N_2 + N_1)$$

$$\geq \frac{k}{12\lambda\tau} - \frac{|g_1|}{3} - \frac{|g_2|}{6} + \frac{1}{2k}(N_2 + N_1) > 0$$

if (A.1) holds.

Next, we show that if $|M_{j-1}| \leq 1$, the denominator in the recursion (A.3_1) is positive. We have

$$Q_j - P_j M_{j-1} = \frac{2k}{3\lambda\tau} + \frac{g_{j+1}}{6} - \frac{g_{j-1}}{6} + \frac{1}{2k}(N_{j+1} + 2N_j + N_{j-1})$$

$$- M_{j-1}\left[-\frac{k}{6\lambda\tau} + \frac{g_j}{3} + \frac{g_{j+1}}{6} + \frac{1}{2k}(N_{j+1} + N_j)\right]$$

$$\geq \frac{2k}{3\lambda\tau} - \frac{g}{3} + \frac{1}{2k}(N_{j+1} + 2N_j + N_{j-1})$$

$$- \frac{k}{6\lambda\tau} - \frac{g}{2} - \frac{1}{2k}(N_{j+1} + N_j)$$

$$\geq \frac{k}{2\lambda\tau} - \frac{5g}{6} + \frac{N}{k} > 0.$$

Finally, we show that from (A.3_1), it follows that $|M_1| \leq 1$. From (A.3_1) and (6.17_3), this is equivalent to

$$P_j(1 + M_{j-1}) - 2Q_j + \frac{k}{\lambda\tau} \leq 0 \leq P_j(1 - M_{j-1}) + \frac{k}{\lambda\tau}.$$

The right part of this is satisfied, since

$$P_j(1 - M_{j-1}) + \frac{k}{\lambda\tau} = (1 - M_{j-1})\left[-\frac{k}{6\lambda\tau} + \frac{g_j}{3} + \frac{g_{j+1}}{6} + \frac{1}{2k}(N_{j-1} + N_j)\right] + \frac{k}{\lambda\tau}$$

$$\geq 2\left[-\frac{k}{\lambda\tau} - \frac{g}{2}\right] + (0)(N_{j-1} + N_j) + \frac{k}{\lambda\tau}$$

$$= \frac{2k}{3\lambda\tau} - g > 0$$

and similarly, for the left part

$$2Q_j - P_j(1 + M_{j-1}) - \frac{k}{\lambda\tau} = 2\left[\frac{2k}{3\lambda\tau} + \frac{g_{j+1}}{6} - \frac{g_{j-1}}{6} + \frac{1}{2k}(N_{j+1} + 2N_j + N_{j-1})\right]$$

$$-(1 + M_{j-1})\left[-\frac{k}{6\lambda\tau} - \frac{g_j}{3} - \frac{g_{j-1}}{6} + \frac{1}{2k}(N_{j-1} + N_j)\right] - \frac{k}{\lambda\tau}$$

$$\begin{aligned}
&\geq \frac{4k}{3\lambda\tau} - \frac{2g}{3} + \frac{1}{k}\left(N_{j+1} + 2N_j + N_{j-1}\right) \\
&- \frac{2g}{3} - \frac{1}{k}\left(N_{j-1} + N_j\right) - \frac{k}{\lambda\tau} \\
&\geq \frac{k}{3\lambda\tau} + \frac{2N}{k} - \frac{4g}{3} > 0.
\end{aligned}$$

from (A.1). This completes the proof.

Chapter 7 Model Problems

Several numerical examples, chosen because they have analytical solutions, are presented here. Most of the examples involve the initial condition of a point source (delta function source). If instead, smooth initial conditions are used, the accuracy of the computed solution is very considerably improved in all the cases tested. We have chosen to present the point source results however since they provide a tougher test for the algorithm and in addition they correspond to a case of practical importance, that of a pollution accident resulting in a sudden influx of pollutant.

In all the test problems, we take $\zeta = 0$ and h, u, v, w, D_h and D_v constant. The value $h = 65$m has been used for definiteness and the number of vertical levels used is $J = 10$.

7.1 *Vertical convection-diffusion*

This example is designed to test the vertical finite element scheme. All quantities are assumed independent of x and y, and we set $D_v \equiv D$. Equation (6.1) then becomes

$$S_t + wS_z = DS_{zz}.$$

The initial condition (6.2) for a point source is $S(z,0) = K\delta(z - z_0)$ which is discretized as

$$S_j(0) = \begin{cases} K/\Delta z & \text{if } j = j_0 \\ 0 & \text{otherwise} \end{cases}$$

where $\Delta z = h\,\Delta\sigma$ is the physical grid spacing and j_0 is the grid point corresponding to z_0. For an infinite region, the exact solution for this problem is

$$S(z,t) = \frac{K}{\sqrt{4\pi Dt}} \exp\left[-\frac{(z - z_0 - wt)^2}{4Dt}\right].$$

This solution can be used for the bounded interval $-h \le z \le 0$ provided that in the boundary conditions (6.3) we take

$$b_1 = S(0,t)\frac{z_0 + wt}{2Dt}, \qquad b_2 = S(-h,t)\frac{z_0 + h + wt}{2Dt}.$$

It is discretized by setting $z - z_0 = \Delta z\,(j - j_0)$.

We have used a vertical velocity $w = 0.0005$ and diffusion coefficient $D = 0.005$, which values lead to roughly equal time-scales for the convection and diffusion. A time step of 360s was used. A comparison of the computed and exact solutions after 100 time steps is shown in Figure 7.1. The maximum relative error between the two solutions is approximately 1% and occurs at the maximum of S.

Some numerical experiments with large values of w and/or τ have been carried out to check the stability condition (6.24) or (6.25). In these tests the values $\lambda = 0.5$, $k = 7.2$ and $D_V = 0.005$ were used, with a sufficient number of vertical levels to keep the solution hump within the computation region. Figure 7.4 shows several pairs of values of w and τ that lead to stable and unstable solutions as well as the graph of the condition (6.25). It is interesting that this condition appears to be, at least approximately, a necessary as well as sufficient condition for stability.

7.2 *Horizontal convection-diffusion*

This example is designed to test the accuracy and stability of the horizontal explicit finite difference scheme. All quantities are assumed independent of z and we set $D_h \equiv D$. Equation (6.1) then becomes

$$S_t + uS_x + vS_y = D(S_{xx} + S_{yy}).$$

The initial condition (6.2) for a point source is $S(x, y, 0) = K\delta(x - x_0)\delta(y - y_0)$ which is discretized as

$$S_{m,n}(0) = \begin{cases} K/\Delta x\,\Delta y & \text{if } (m,n) = (m_0, n_0) \\ 0 & \text{otherwise} \end{cases}$$

where (m_0, n_0) is the grid point corresponding to $(x_0., y_0)$. For an infinite region, the exact solution for this problem is

$$S(x, y, t) = \frac{K}{4\pi Dt} \exp\left[-\frac{(x - x_0 - ut)^2 + (y - y_0 - vt)^2}{4Dt}\right]$$

This solution is used for the bounded region simply by taking the region large enough that the solution remains essentially zero at the boundaries for the time interval of the computation.

We have used a horizontal velocity $u = v = 0.5$, grid spacing $\Delta x = 5000$, time step $\tau = 360$ and diffusion coefficient $D = 10^4$. This value of D is rather large, but is chosen to give approximately equal time-scales for the convection and diffusion. A comparison of the computed and exact solutions after 100 time steps is shown in Figure 7.2. The relative error between the two solutions is not more than 5%.

As remarked above, a considerable improvement in accuracy is obtained if smooth initial values are used instead of the delta function. For example, if the above solution is used with the initial value of the numerical solution set equal to the exact solution after 40 time steps, the error after 100 steps is of the order of 2%. It is also the case that the accuracy is similarly improved even for the delta function source if the velocities u and v are zero. It therefore seems that the major source of the errors in such results as those shown in Figure 2 is the treatment of the convective terms via the discretizations (6.23) when the pollutant density S has large gradients.

Some numerical experiments have been carried out with very large time steps in order to verify the validity of the stability limit (6.26). We have found in numerous test cases that the condition (6.26) is precise. For example, with $u = v = 0, D_H = 10^4$ and $\Delta x = 20{,}000$, (6.26) requires $\tau \leq 10{,}000$. We found the computation with $\tau = 9800$ remains stable for at least 1000 time steps but with $\tau = 10{,}200$ oscillations grow and produce negative densities after about 400 steps. With $u = v = 0.5$ and the other parameters unchanged, (6.26) requires $\tau \leq 6667$. We found that the computation remains stable with $\tau = 6600$ but becomes unstable if a value $\tau = 6670$ is used. Of course in both these cases, even when stable, the results are not very accurate

7.3 *Three-dimensional convection-diffusion*

(a). *Comparison with Exact Solution*

This example is a combination of Examples (7.1) and (7.2). We define

$$f_h(x,t;u)=\frac{1}{\sqrt{4\pi D_h(t+t_0)}}\exp\left[-\frac{[x-x_0-u(t+t_0)]^2}{4D_h(t+t_0)}\right]$$

$$f_v(z,t;w)=\frac{1}{\sqrt{4\pi D_v(t+t_0)}}\exp\left[-\frac{[z-z_0-w(t+t_0)]^2}{4D_v(t+t_0)}\right].$$

Then for constant diffusivities and velocities, the homogeneous equation (6.1) has the following solution for an infinite region:

$$S(x,y,z,t)=Kf_h(x,t;u)f_h(y,t;v)f_v(z,t;w).$$

If $t_0 > 0$, this solution has the smooth initial condition

$$S(x,y,z,0)=Kf_h(x,0;u)f_h(y,0;v)f_v(z,0;w)$$

while if $t_0 = 0$, there is a point source: $S(x,y,z,0)=K\delta(x-x_0)\delta(y-y_0)\delta(z-z_0)$.

As in Examples (7.1) and (7.2), this solution is applied to the bounded region by suitably choosing the functions b_1 and b_2 in conditions (6.3) and by taking the extent of the region in the xy -plane large enough that the solution remains essentially zero at the boundaries.

Some typical results are shown in Figure 7.3. For these, we used velocity components $u = v = 0.2$, $w = 0$, grid spacing $\Delta x = 2000$, $J = 10$ vertical levels, time step $\tau = 180$ with $t_0 = 5000$ and diffusion coefficients $D_h = 2000$ and $D_v = 0.01$. Again, these values were chosen to give approximately equal time-scales for the horizontal convection and the two directions of diffusion. The figure shows the solution after 100 time steps on the central level (which is the source level) and on levels 2 and 8. The maximum relative error in the computed solution is about 4%.

As remarked in Section 6.3, it is important to know if the stability criteria derived separately for the horizontal and vertical convection-diffusion apply also for the joint three-dimensional algorithm. We have carried out some experiments to obtain at least an empirical answer to this question. The values $D_H = 2000$, $D_V = 0.01$, $u = v = 0.2$ have been used. Then provided $|w| < 0.00167$, condition (6.26) is more restrictive than (6.25). Figure 7.5 shows the graph of the condition (6.26) in the Δx-τ plane as well as several experimentally determined stable and unstable points. The evidence from these results indicates that only at extremely long time steps (for which the numerical solution is very inaccurate) does the vertical part of the algorithm interfere with the stability condition (6.26).

(b). *Comparison with Monte Carlo method*

A commonly used method for solving the convection-diffusion equation is the Monte-Carlo method, in which a cloud of pollutant particles are, on each time step, given random displacements to simulate the diffusion and in addition are given convective displacements according to the fluid velocities at their current locations. The resulting densities suffer from errors due to statistical fluctuations in the random displacements, and in order to obtain a sufficiently smooth function, a very large number of particles must be used. The resulting large number of cells on the random number generator increases the expense of this method.

In order to help resolve the question of the optimal method to use, we have repeated the computation in Example (7.2) using a Monte-Carlo algorithm. The algorithm employs the random number generator URAND [74]. In order to achieve errors comparable to the 5% found in Example (7.2), it was necessary in the Monte Carlo algorithm to use at least 4×10^4 particles. The resulting CPU time was 257 seconds on an IBM 3081 machine. (The program was run as a three-dimensional simulation, even though the vertical diffusion coefficient was zero.)

In Example (7.2), the horizontal grid was 30×30 with 10 vertical levels, the solution after 100 time steps being significantly non-zero over about half of this region. The computation in this example required 92 seconds of CPU time. (Again, this represents a CPU time for the complete three-dimensional algorithm, even though the solution is independent of the vertical coordinate.)

It is clear that at least in this type of situation, the present algorithm has an advantage, either in accuracy or speed. For a full three-dimensional simulation, the advantage would be even greater, since the particles in the Monte-Carlo simulation would need to be distributed among the various levels and consequently there would need to be about J times as many of them to achieve the same accuracy.

On the other hand, if the problem at hand involves a large sea region in which the pollutant is concentrated in one small sub-region, a method of the type discussed in this paper will be inefficient, in that a great many unnecessary calculations are performed. It may be possible to overcome this difficulty by artificially restricting the region in which the convection-diffusion equation is solved, but this may not be possible if the convective displacements are large. The efficiency of the Monte-Carlo algorithm can also be very significantly improved by using a fast "random" number generator such as RANDU in the IBM Scientific Subroutine Package, although the lack of complete randomness [74] may be a deterrent to this.

CONCLUSIONS & REMARKS

In this thesis, we have considered three kinds of problems: nonlinear wind-driven flow, density-driven flow and the associated marine convection-diffusion problem. In order to solve these problems, four basic algorithms have been developed.

The first algorithm described in chapter 3 has been developed for the numerical solution of the three-dimensional tidal equations using a spectral method in the vertical dimension and finite differences in the horizontal. Four difference schemes have been constructed based on the Arakawa A,B,C and E-grids. While the C-grid has traditionally been used for such hydrodynamical computations, the other three grids offer significant advantages when a spectral method is used in the vertical, especially in that they allow eddy viscosity functions to be used that vary quite arbitrarily with position without introducing coupling among the modal equations.

A second advantage of these three grids is that none of them produces the spurious numerical boundary layers that can occur for the C-grid unless the Coriolis terms are treated using "wet-points only" averaging at coastal points [43]. A third benefit is that the two modal momentum equations can be solved simultaneously in complex form, allowing explicit numerical treatment of the Coriolis terms to be easily avoided.

Two test models have been designed for comparison of these schemes.The conclusions reached are as follows.

The numerical errors arising for the A-grid were very significantly greater than those for the other three grids, the reason being, presumably, that the finite difference approximations to the various spatial derivatives must use intervals of twice the size.

The numerical errors for the E-grid were generally slightly lower than those for the C-grid for both test problems. The disadvantage of using the E-grid is that the computational cost is about twice that of the C-grid for the same grid dimensions. The computational costs of the B and C-grids are about the same.

The rms numerical errors in the velocity components for the B and C-grids fluctuated in both relative and absolute magnitudes as the computations progressed. For the second problem, in which an analytical solution is known, over the first 600 time steps

(60 hours of real time) the B-grid results were on average slightly more accurate than those of the C-grid.

It is significant that the errors are almost uniform through the water column in every case, that is they are concentrated in the lowest mode. When the bottom friction is zero, as in the second test problem, the lowest mode is governed by the shallow water equations in which there is no damping. It is therefore a pleasant surprise that the inferior numerical dispersion properties of the B and E-grids at short wavelengths [42] do not lead to substantially greater errors in this problem than those for the C-grid. Both grids lead to stable algorithms for these undamped equations.

For a two-dimensional (depth-averaged) model, the C-grid appears to have no disadvantages compared to the other grids. This would presumably also be true for a three-dimensional multi-level or splitting method. For spectral method algorithms, however, this grid imposes severe limitations on the physical model if the computation is to be easily feasible. Our results suggest that the B-grid can provide a viable alternative at the same computing cost and without imposing such limitations.

The second algorithm has been developed for the numerical solutions of the full nonlinear three-dimensional tidal equations.The principal features of the algorithm described in the first two chapters can be summarized as follows.

(a) It is directed towards solving the fully nonlinear hydrodynamic equations as usually approximated for flows in shallow seas of uniform density, with an eddy viscosity model of turbulence.

(b) The numerical approach to the dependence on the vertical coordinate is a spectral method of Galerkin type, using eddy viscosity eigenfunctions as the basis set. The advantage of this basis is that the modal equations are not coupled through the linear terms, which are the dominant ones in most cases.

(c) By an appropriate modification of the velocity before expansion the method provides a uniformly convergent series that converges rapidly and for which truncations exactly satisfy the surface and bottom boundary conditions.

(d) The cost of this modification is an explicit treatment of bottom friction. This certainly imposes a stability restriction, which, however has not turned out to be a serious limitation in the problems examined.

(e) The advective terms are computed explicitly using a finite element method. This is the most efficient means we have found for this part of the algorithm, which is by far the most expensive, requiring two-thirds of the total CPU time.

(f) The numerical approach to the horizontal variations is via a staggered B-grid, which has some advantages over the more usual C-grid when a spectral method is used.

(g) A leapfrog method is used for the time-stepping, in which the surface elevation and the velocity are computed on alternate half-steps. Account is taken of the stiffness of the system of modal equations.

The performance of the algorithm has been tested satisfactorily on a number of problems (see chapter 4). For steady wind-driven flow in a channel with nonlinear bottom friction, good agreement is obtained with the analytical solution. For a dynamical wind-driven flow in a rectangular sea, the algorithm gives good agreement for the linearized equations with an earlier algorithm that is specifically adapted for the linear case. For the nonlinear case with constant eddy-viscosity, the stability region has been determined by a series of computations. For the nonlinear case with variable eddy-viscosity, the expansions were found to converge at least as fast as those used earlier with bases consisting of Chebychev or shifted Legendre polynomials (and these bases lead to large coupling among the modal equations).

The third algorithm described in chapter 5 has been developed for the numerical solutions of the three-dimensional tidal equations driven by density gradients.In this case, linear equations have been considered. The spectral method for the vertical direction and B-grid scheme for the horizontal directions have been used to develop the algorithm. The accuracy of the computer code has been tested by solving two problems for which the exact steady solutions can be found. These two problems were designed to test the code's accuracy in handling two distinct physical features: vertical density variation and Coriolis forces. Both test problems give very accurate results.

The fourth algorithm in chapter 6 and chapter 7 has been developed for the numerical solution of the three-dimensional convection-diffusion equation in shallow seas. An implicit finite element discretization has been used for the vertical direction and an explicit finite difference discretization has been used for the horizontal directions. These kinds of discretizations lead to be a tridiagonal system which can be easily solved at every horizontal points in the considered region. The stability restrictions, both on the vertical direction and on the horizontal directions, have been determined theoretically(see chapter 6, appendix). Three problems have been used to test the accuracy and stability of the algorithm which gives good agreement with corresponding exact solutions. A comparison also has been made with the Monte-Carlo type of algorithm.

In the conclusion of this thesis, we give a summary of the basic techniques used in these four algorithms in the following table:

\Problems Techniques\		scheme comparisons	nonlinear problem	density problem	conv-diffus. problem
time	leapfrog	X	X	X	
	explicit				X
	implicit				X
horizontal	B-grid	X	X	X	
	finite-diff.				X
vertical	spectral	X	X	X	
	finite-elem.		X		X

REFERENCES

1. Rizzi, A. and Engquist, B. (1987) Selected topics in the theory and practice of computational fluid dynamics, J. comput. phys., **72**,1-69

2. Wubs, F. W. (1988) *Numerical Solution of the Shallow Water Equations*, Stichting Mathematisch Centrum, Amsterdam.

3. Richardson, L.F. (1922) *Weather Pridiction by Numerical Press* (Cambrige Univ. Press, London)

4. Charney, J.G., Fjφrtoft, R. and von Neumann, J (1950) Numerical integration of the barotropic verticity equation, Tellus, **2**, 237-254.

5. Hansen, W. (1962) Hydrodynamical methods applied to oceanographic problems. Proc. Symy. Math. Hydrodyn. Phys. Ocean., Institut fur meereskunde, Universat Hamburg, 24-34.

6. Leendertse, J.J. (1967) Aspects of a computational model for long period water-wave propagation, Rand. Corp. Rep. RM-5294-PR.

7. Mathews, J.B. and Mungall, J.C.H. (1972) A numerical tidal model and its application to Cook Inlet, Alaska, J. Mar. Res. **30**, 27-38.

8. Flather, R.A. and Heaps, N.S. (1975) Tidal computations for Morecambe Bay, Geophys. J. R. Astr. Soc. **42**, 489-517.

9. Lardner, R.W., Belen, M.S. and Cekirge, H.M. (1982) Finite difference model for tidal flows in the Aranbian Gulf, Comp. & Math. with Applic., **8**, 425-444.

10. Duff, G.F.D. (1983) A special ADI model for the Laplace tidal equations, Comp. & Math. with Applic. **9**, 507-516.

11. Connor, J.J. and Wang, J. (1973) Finite element modelling of hydrodynamic circulation, in *Numerical Methods in Fluid Mechanics* (ed. by C.A. Brebbia, and J.J. Connor, Pentech Press, London), 335-387.

12. Taylor, C. and Davis, J.M. (1975) Tidal propagation and dispersion in estuaries, in *Finite Elements in Fluids* (ed. by R.H. Gallagher, J.T. Oden, C. Taylor and O.C. Zienkiewics, Wiley), Vol.**1**, Chapter 5.

13. Brebbia, C.A. and Partridge, P.W. (1976) Finite element simulation of water circulation in the North Sea, Appl. Math. Modelling, **1**, 101-107.

14. Wang, J.D. (1978) Real time flow in unstratified shallow water, J. Waterway Port, Coast and Ocean Div., A.S.C.E., **104**, 53-68.

15. Connor, J.J. and Brebbia, C.A. (1976) *Finite Elements for Fluid Flow*, Butterworth, London.

16. Pinder, G.F. and Gray, W.G. (1977) *Finite Element Simulation in Surface and Subsurface Hydrology*, Academic Press.

17. Le Provost, C. and Poncet, A. (1978) Finite-element method for spectral modelling of tides, Int. J. Num. Methods Eng. **12**, 853-871.

18. Le Provost, C., Rougier, G. and Poncet, A. (1981) Numerical modelling of the harmonic constituents of the tides, with application to the English Channel, J. Phys. Ocean. **11**, 1123-1138.

19. Pearson, C.E. and Winter, D.F. (1977) On the calculation of tidal currents in homogeneous esturaries, J. Phys. Ocean. **7**, 520-531.

20. Townson, J.M. (1974) An application of the method of characteristics to tidal calculations in x-y-t space, J. Hydraul. Res. , **12**, 499-523.

21. Lai, C. (1976) Some computational aspects of one-and two-dimensional unsteady flow simulation by the method of characteristics, Proc. Int. Sym. Unsteady Flow Open Channels, Int. Assoc. Hydraul. Res., Newcastle-upon-Tyne, D1-12.

22. Lardner, R.W. Cekirge, H.M. and Gunay, N. (1986) Numerical solution of the two-dimensional tidal equations using the method of characteristics, Comp. and Math. with Applications, **12A**, 1065-1080.

23. Cekirge, H.M., Lardner, R.W. and Fraga, R.J. (1986) Adaption of the solution of the two-dimensional tidal equations using the method of characteristics to wind-induced currents and storm surges, Comp. anf Math. with Applications, **12A**, 1081-1090.

24. Gordon, R.L. (1982) Coastal ocean current response to storm winds, J. Geophys. Res., **98**, 1939-1951.

25. Daubert, O., Hervouet, J.M. and Jami, A. (1989) Description of some numerical tools for solving incompressible turbulent and free surface flows, Int. J. Numer. methods Engeg., **27**, 3-20.

26. Davies, A.M. (1987) Spectral models in continental shell sea oceanography, American Geophysical Union, 71-106.

27. Sundermann, J. (1974) A three dimensional model of a homogeneous estuary, Proc. Conf. Coastal Engrg. (A.S.C.E.) 2337-2390.

28. Simons, T.J. (1974) Verification of numerical models of Lake Ontatio, Part I, J. Phys. Oceanog., **4**, 504-519; Part II, J. Phys. Oceanog., **5**, (1975) 98-110.

29. Laevastu, T. (1975) Multilayer hydrodynamical-numerical models, Proc. Symp. Modelling Tech. (ASCE) 1010-1020.

30. Leendertse, J.J. and Liu, S.K. (1975a) A three-dimensional model for estuarine and coastal seas, Rand Corp. Tech. Rep. R-1764-OWRT;
Leendertse, J.J. and Liu, S.K. (1975b) Modelling of three-dimensional flows in estuarines, in *Modelling Techniques* (published by ASCE)

31. Sengupta, S., Lee, S.S. and Miller, H.P. (1978) Three-dimensional numerical investigations of tide and wind-induced transport processes in Biscayne Bay, Sea Drant Technical Bulletin No. 39, University of Miami.

32. Nihoul, J.C.J. and Jamart, B.M. (1987) *Three Dimensional Models of Marine and Estuarine Dynamics.*

33. Heaps, N.S. (1972) On the numerical solution of the three-dimensional hydrodynamical equations for tides and storm surges. Mem. Soc. Roy. Sci. Liege, Ser **6**, 143-180.

34 Heaps, N.S. (1981) Three-dimensional model for tides and surges with vertical eddy viscosity prescribed in two layers I. Mathematical formulation. Geophys. J. Roy. Astron. Soc., **64**, 291-302.

35. Davies, A.M. (1983) Formulation of a linear three-dimensional hydrodynamic sea model using a Galerkin eigenfunction method. Int J Num Methods Fluids 3, 33-60.

36. Furnes, G. (1983) A three-dimensional numerical sea model with eddy viscosity varying piecewise linearly in the vertical. Continental Shelf Res **2**, 231-241.

37. Davies, A.M. and Owen, A. (1979) Three-dimensional numerical sea model using the Galerkin method with a polynomial basis set. Appl. Math. Modelling **3**, 421-428.

38. Davies, A.M. (1980) Application of the Galerkin method to the formulation of a three-dimensional nonlinear hydrodynamical numerical sea model. Appl Math Modelling 4, 245-256.

39. Davies, A.M. (1988) On formulating two-dimensional vertically integrated hydrodynamic numerical models with an enhanced representation of bed stress. J Geophys Res **93**, 1241-1263.

40. Lardner,.R.W. (1990) On the numerical solution of the linearized three-dimensional tidal equations using eddy-viscosity eigenfunctions. To be published.

41. Davies, A.M. (1977) The numerical solution ot the three-dimensional hydrodynamic equations using a B-spline representation of the vertical current profile. *Bottom turbulence, Proceedings of 8th Liege Conference on Ocean Hydrodynamics* (ed. by J.C.J.Nihoul), Elsevier Oceanography Series, **19**, 1-25.

42. Arakawa, A. and Lamb, V.R. (1977) Computational designof the basic dynamical processes of the UCLA general circulation model, in *Methods in computational physics*, (Academic Press), **17**, 174-265

43. Jamart, B.M. and Ozer, J. (1986) Numerical boundary layers and spurious residual flows. J Geophys Res **91**, 10.621-10.631.

44. Lardner, R.W. and Song, Y. (1990) A comparison of spatial grids for numerical modelling of flows in near-coastal seas. Comp. Meth. Applied Mechanics & Eng., submttied.

45. Johns, B., Sinha, P.C., Dube, S.K., Mohanty, U.C. and Rao, A.D. (1983). Simulation of storm surges using a three-dimensional numerical model: an application to the 1977 Andhra cyclone. Quart. J. R. Met. Soc. **109**, 211-224.

46. Lardner, R.W. and Cekirge, H. M. (1988) A new algrithm for three-dimensional tidal and storm-surge computations, Appl. Math. Modelling, **12**, 471-481.

47. Lardner, R.W. and Smoczynski, P. (1990) A vertical/horizontal splitting algorithm for three-dimensional tidal and storm-surge computations, Proc. Roy. Soc. (London) to be published.

48. Lardner, R.W., Lehr, W.J., Fraga, R.J. and Sarhan, M. A. (1987) Residual currents in the Aranbian Gulf I: density-driven flow, Arabian J. Sci. Eng., **12**, 341-354.

49. Pearson, C.E. and Winter, D.F. (1978) Two-layer analysis of steady circulation in stratified fjords, in *Hydrodynamics of Estuaries aand Fjords* (ed. by J.C.J. Nihoul, Elsevier Publishing Co., Amsterdam).

50. Lardner, R.W. and Das, S. On the computational of flows driven by density gradient: residual currents in the Arabian Gulf. submitted.

51. Spalding, D.B. (1972). A novel finite difference formulation for differential expressions involving both first and second derivatives, Int. J. Num. Meth. Engng. **4**, 551-559.

52. Boris, J.P. and Book, D.L. (1973). Flux-corrected transport I shasta: a fluid transport algorithm that works, J. Comput. Phys. **11**, 38-69.

53. Dick, E. (1983). Accurate Petrov-Galerkin method for transient convection diffusion problem, Int. J. Num. Meth. Engng. **19**, 1425-1433.

54. Morton, K.W. and Sweby, P.K. (1987) A comparison of flux-limited difference methods and characteristic Galerkin method for shock modelling. J. Comput. Phys. **73**, 203-230

55. Noye, B. and Tan, H.H. (1988) A third order semi-implicit finite difference method for solving the one-dimensional convection-diffusion equation. Int. J. Num. Meth. Engng. **26**, 1615-1629.

56. Holly, F.M. and Usseglio-Polatera, J-M. (1984). Dispersion simulation in two-dimensional tidal flow, J. Hydraulic Engng. ASCE, **110(7)**, 905-926.

57. Sobey, R.J. (1983). Fractional step algorithm for estuarine mass transport, Int. J. Num. Meth. Fluids. **3**, 567-581.

58. Li, Y.S. and Chen, C.P. (1989). An efficient split-operator scheme for 2-D advection-diffusion using finite elements and characteristics, Appl. Math. Modelling **13**, 248-253.

59. Ding, D. and Liu, P.L-F. (1989). An operator splitting algorithm for two-dimensional convection-dispersion-reaction problems, Int. J. Num. Meth. Engng. **28**, 1023-1040.

60. Lardner, R.W., Lehr, W.J., Fraga, R.J. and Sarhan, M.A. (1988). Residual currents and pollutant transport in the Arabian Gulf. Appl. Math. Modelling **12**, 379-390.

61. Panton, R,L, (1984) *Incompressible Flow* (wiley, New York).

62. Fletcher, C.A.J. (1988) *Computational Techniques for Fluid Dynamics*, Springer ser. comput. phys. (Springer, Berlin, Heidelberg).

63. Bowden, K.F., Fairbain, L.A. and Hughes, P. (1959) The distribution of shearing stresses in a tidal current, Geophys. J. R. Astron. Soc., **2**, 288-305.

64. Phillips, N.A. (1957). A coordinate system having some spacial advantages for numerical forecasting. J. Meteorol. **14**, 184-186.

65. Godart, M. (1966) An iterative method for the solution of eigenvalue problem, Math. Comput., **20**, 399-406.

66. Banks, D.O. and Kurowski, G.J. (1968) Computation of eigenvalues of singular Sturm-Liouville systems, Math. Comput. **22**, 304-310.

67. Bailey, P.B. (1978) An slightly modified Prufer transformation useful for calculating Sturm-Liouville eigenvalues, J. Comput. Phys. **29**, 306-310.

68. Bailey, P.B. (1978) SLEIGN An eigenvalue-eigenfunction code for Sturm-Liouville problems, SAND77-2044, Sandia Laboratories, Albuquerque, N.M.

69. Kuzmic, M. (1989) A numerical study of wind-induced motions in shallow coastal seas: model and basic experiments, Appl. Math. Modelling, **13**, 178-191.

70. Gedney, R.T. and Lick, W. (1972) Wind-driven currents in Lake Erie. J. Geophys. Res. **77**, 2714-2723.

71. Backhaus, J.O. (1985) A three-dimensional model for simulation of shelf-sea dynamics. Dt. Hydrogr. Z. **38**, 165-187.

72. Richtmyer, R.D. and Morton, K.W. (1967) *Difference methods for initial value problems.*(Interscience, New York)

73. Smith, G.D. (1985) *Numerical solution of partial differential equations: finite difference methods*, 3rd ed. (Clarendon Press, Oxford)

74. Forsythe, G.E., Malcolm, M.A. and Moler, C.B. (1977). *Computer methods for mathematical computations* (Prentice-Hall). Chapter 10.

75. Neta, B. and Navon, I.M. (1989) Analysis of the Turkel-Zwas scheme for the shallow-water equations, J. of Computational Physics **81**, 277-299.

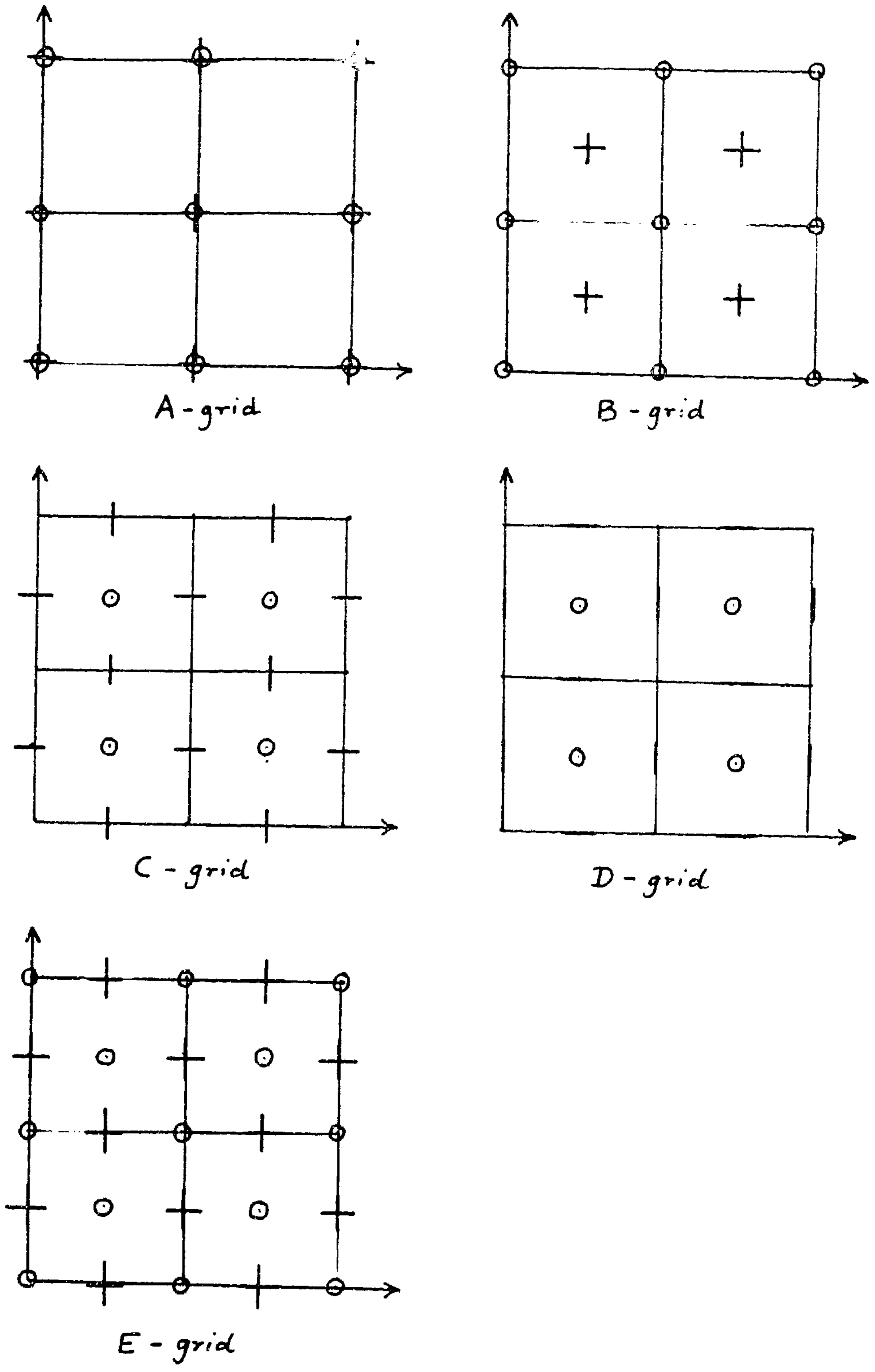
76. Haltiner, G.J. and Williams, R.T. (1980) *Numerical Prediction and Dynamic Meteorology*, John Wiley & Sons, Inc.

77. Schoenstadt, A.L. (1977) J. Comput. Phys. **23**, 364-

78. Schoenstadt, A.L. (1979) NPS Report NPS-53-79-001, Naval Post-Graduate School, (unpublished).

79. Golub, G.H. and van Loan, C.F. (1983). *Matrix Computations* (Baltimore: Johns Hopkins University Press).

80. Hindmarsh, A.C., Gresho, P.M. and Griffiths, D.F. (1984) The stability of explicit Euler time-integration for certain finite difference approximations of the multi-dimensional advection-diffusion equation, Int. J. Numer. Meth. Fluids, **4**, 853-897.



A-grid

B-grid

C-grid

D-grid

E-grid

Figure 3.1. The five Arakawa grids. ζ-points are indicated by O, u-points by — and v-points by |.

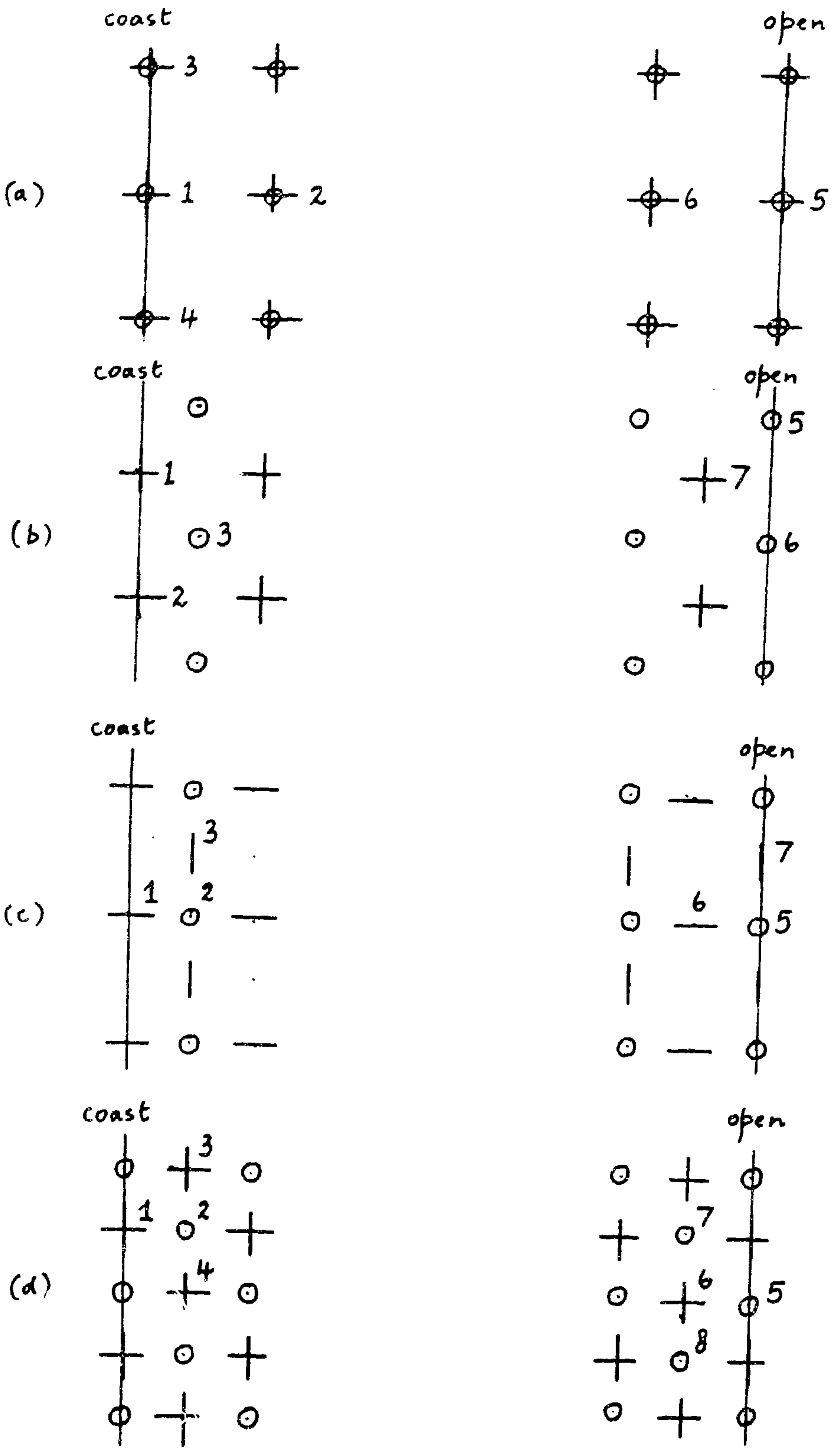


Figure 3.2. Treatment of coastal and open boundaries for the A, B, C and E-grids.

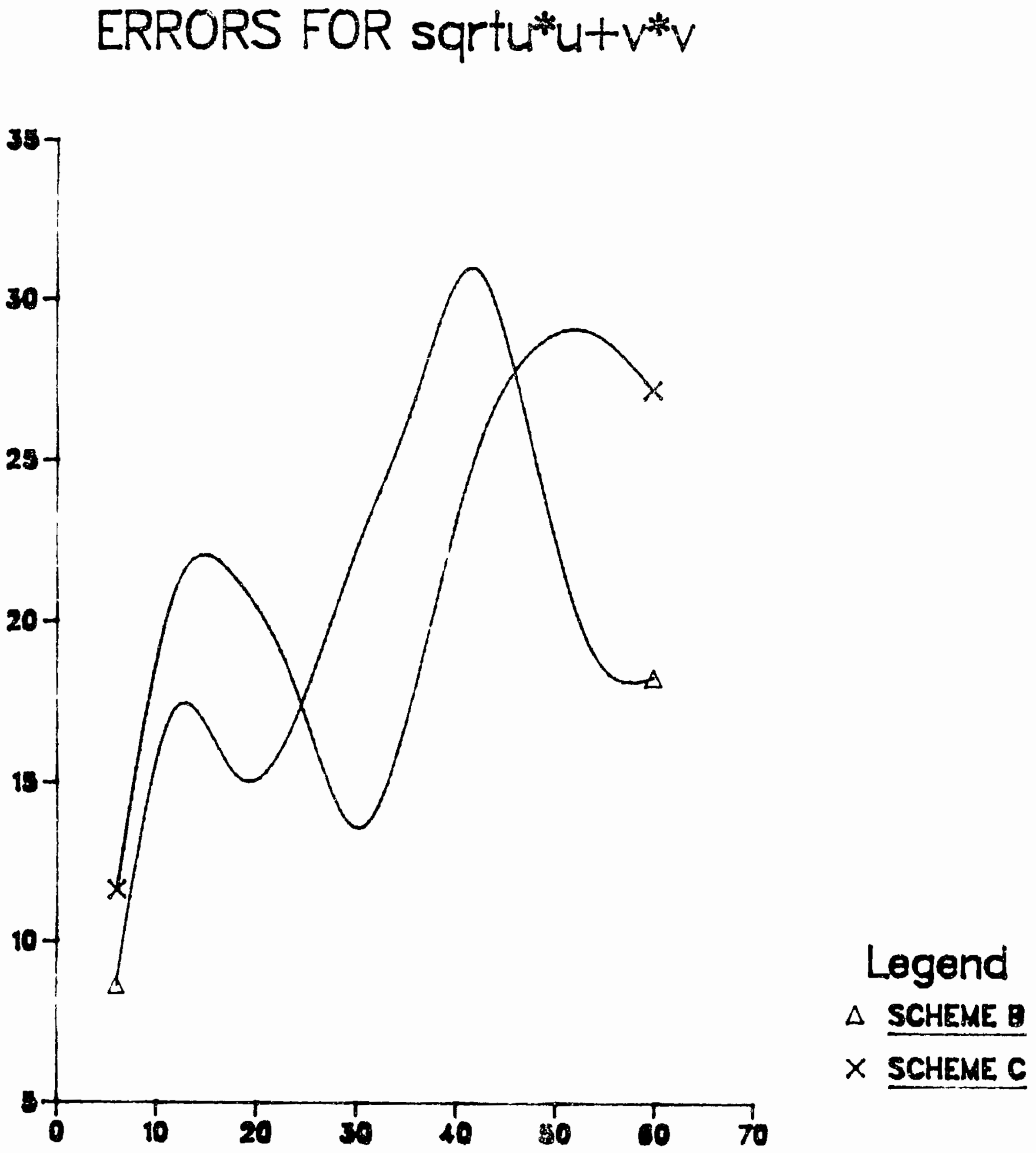


Figure33. Root mean square errors in velocity as a function of time step for schemes B and C.

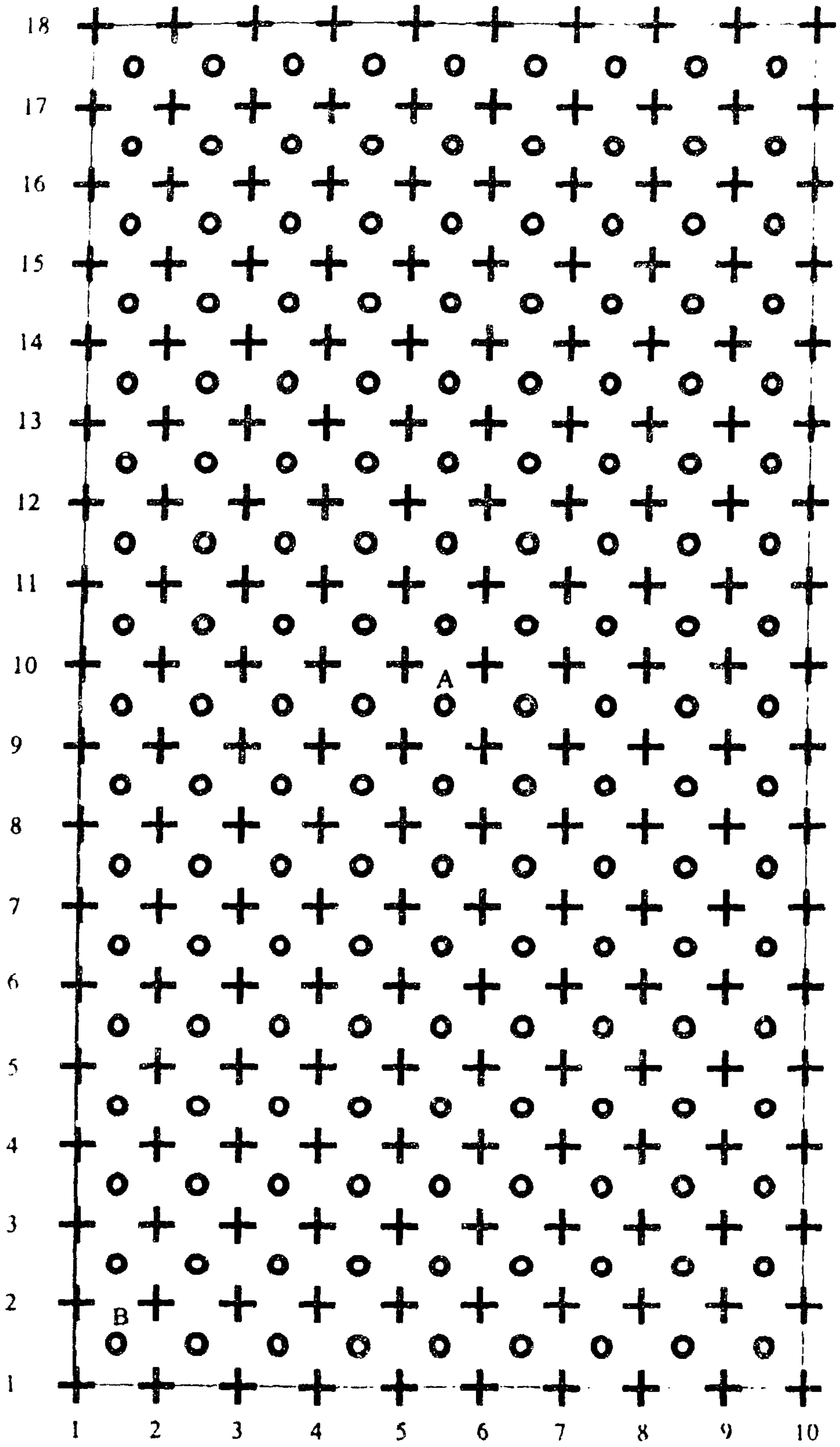


Figure 4.1

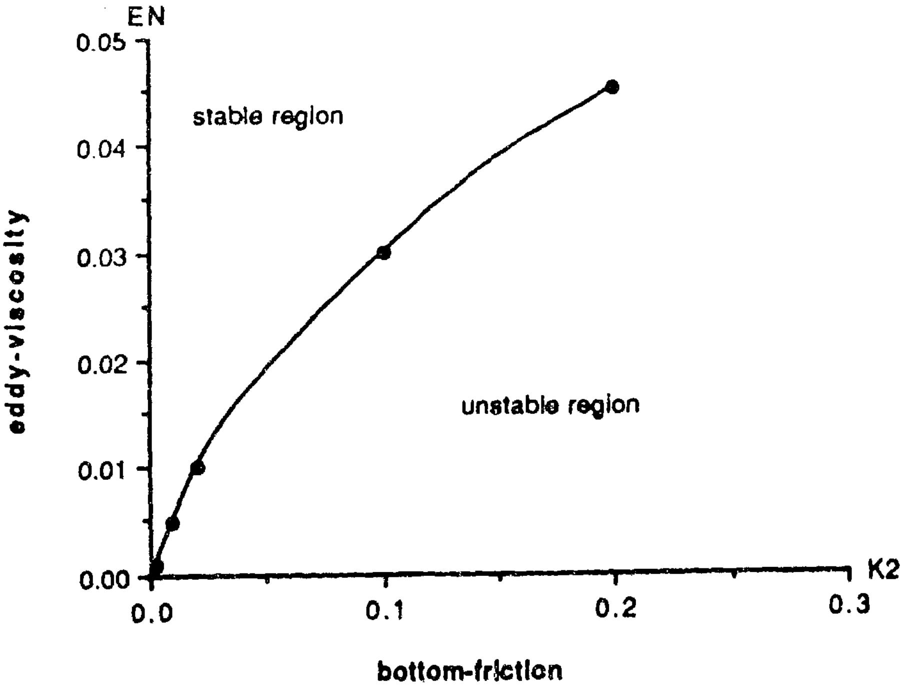

EN
0.05
0.04
0.03
0.02
0.01
0.00
eddy-viscosity
stable region
unstable region
0.0
0.1
0.2
0.3
K2
bottom-friction

Figure 4.2

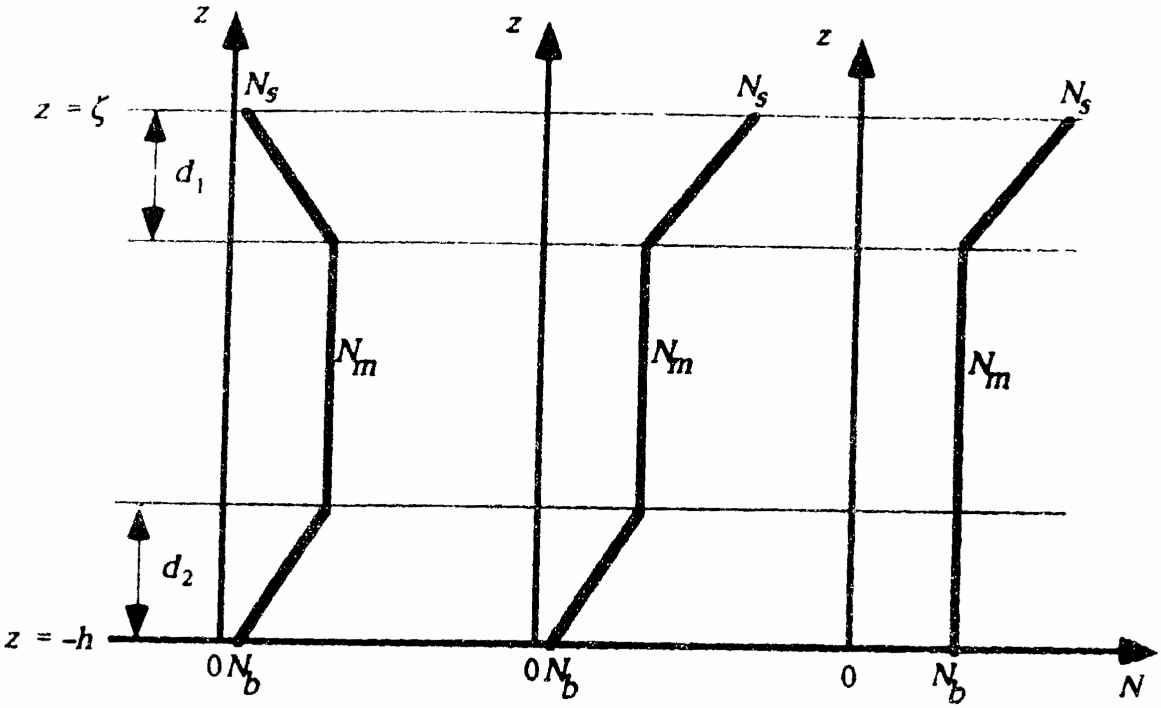
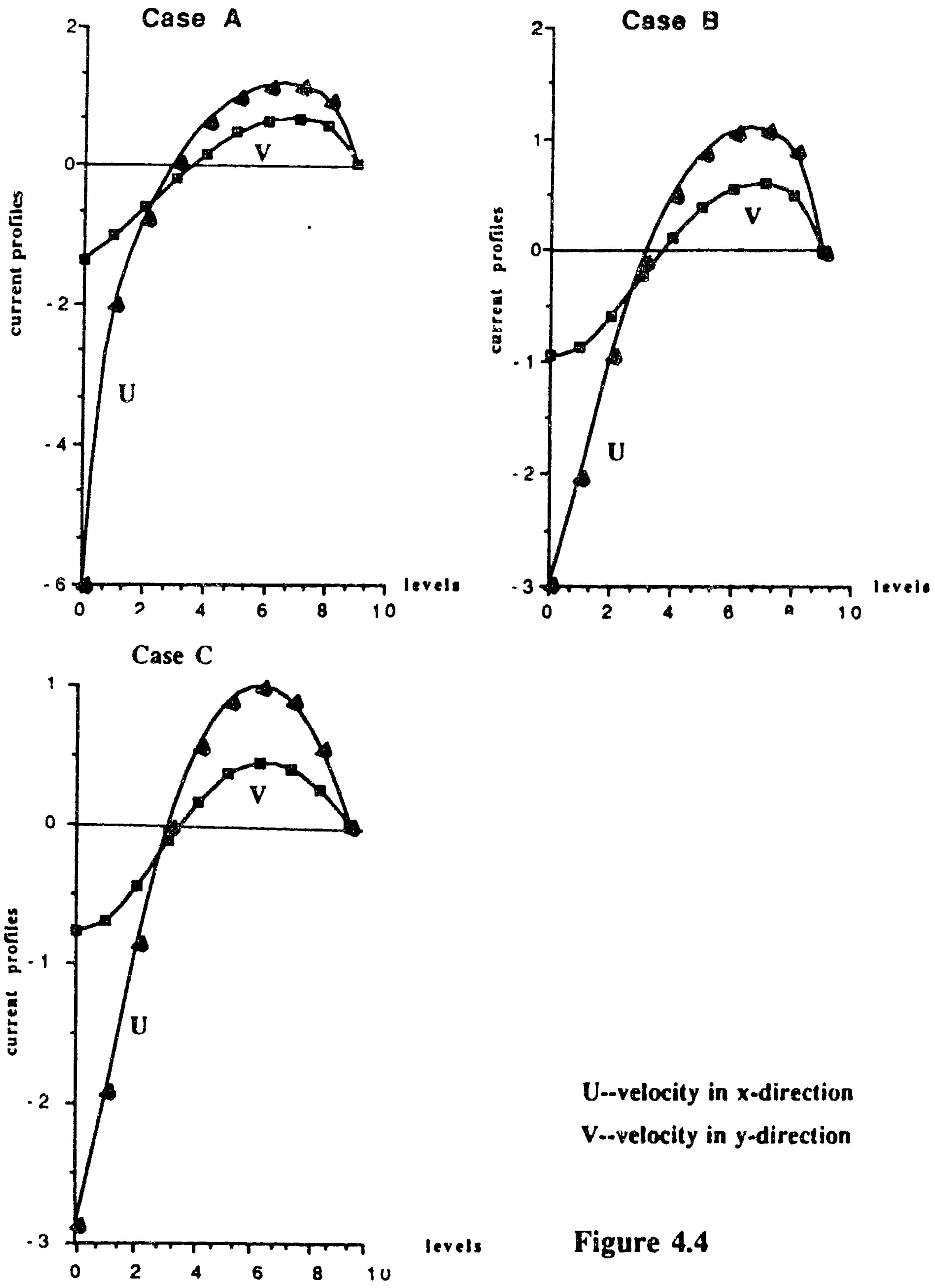


Figure 4.3



U--velocity in x-direction

V--velocity in y-direction

Figure 4.4

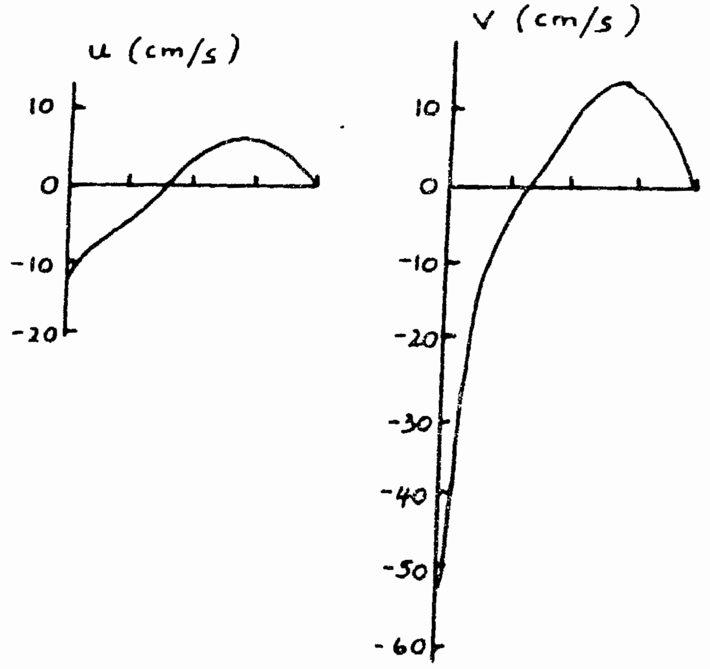


Figure 4.5

Figure 7.1

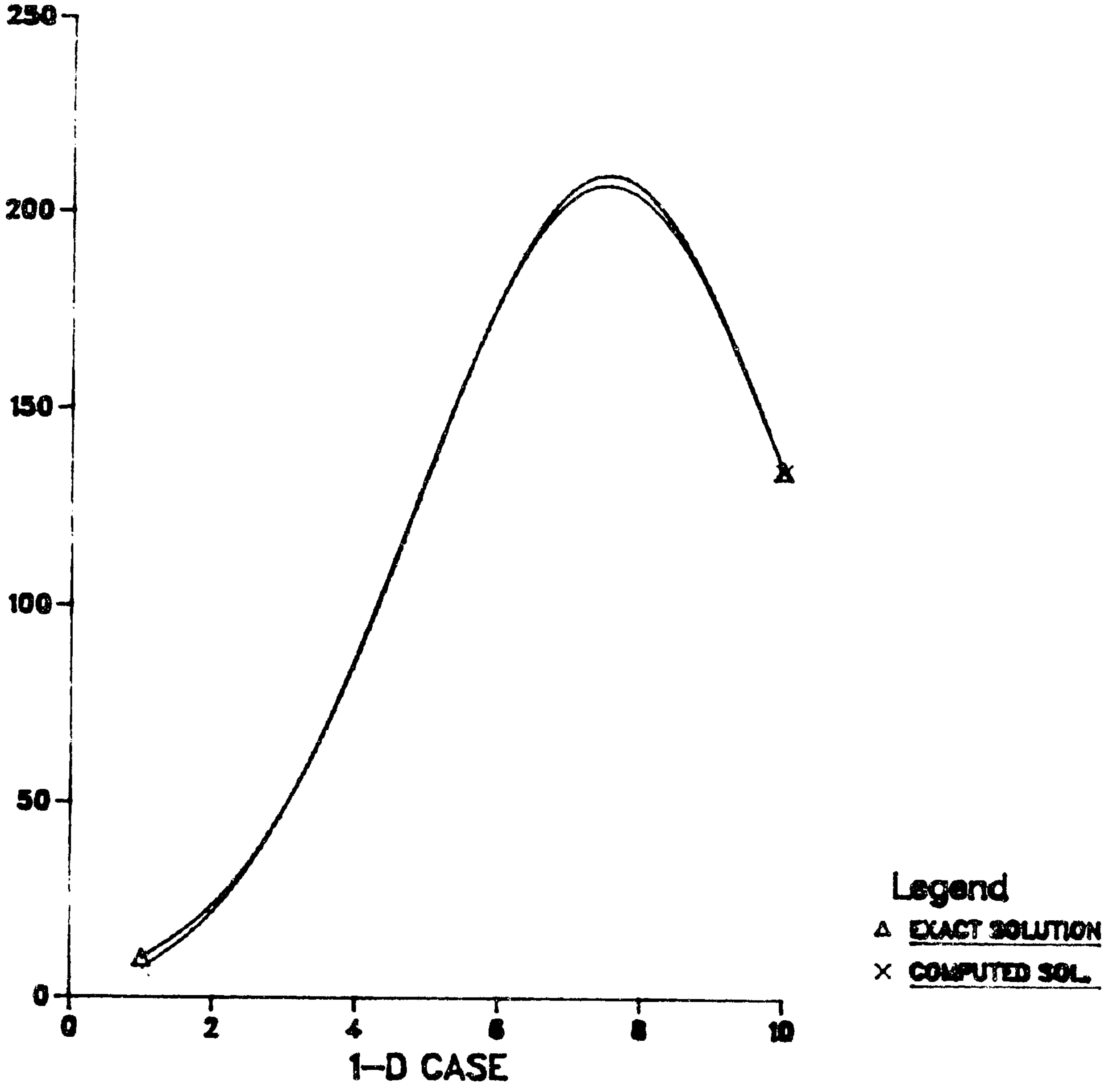
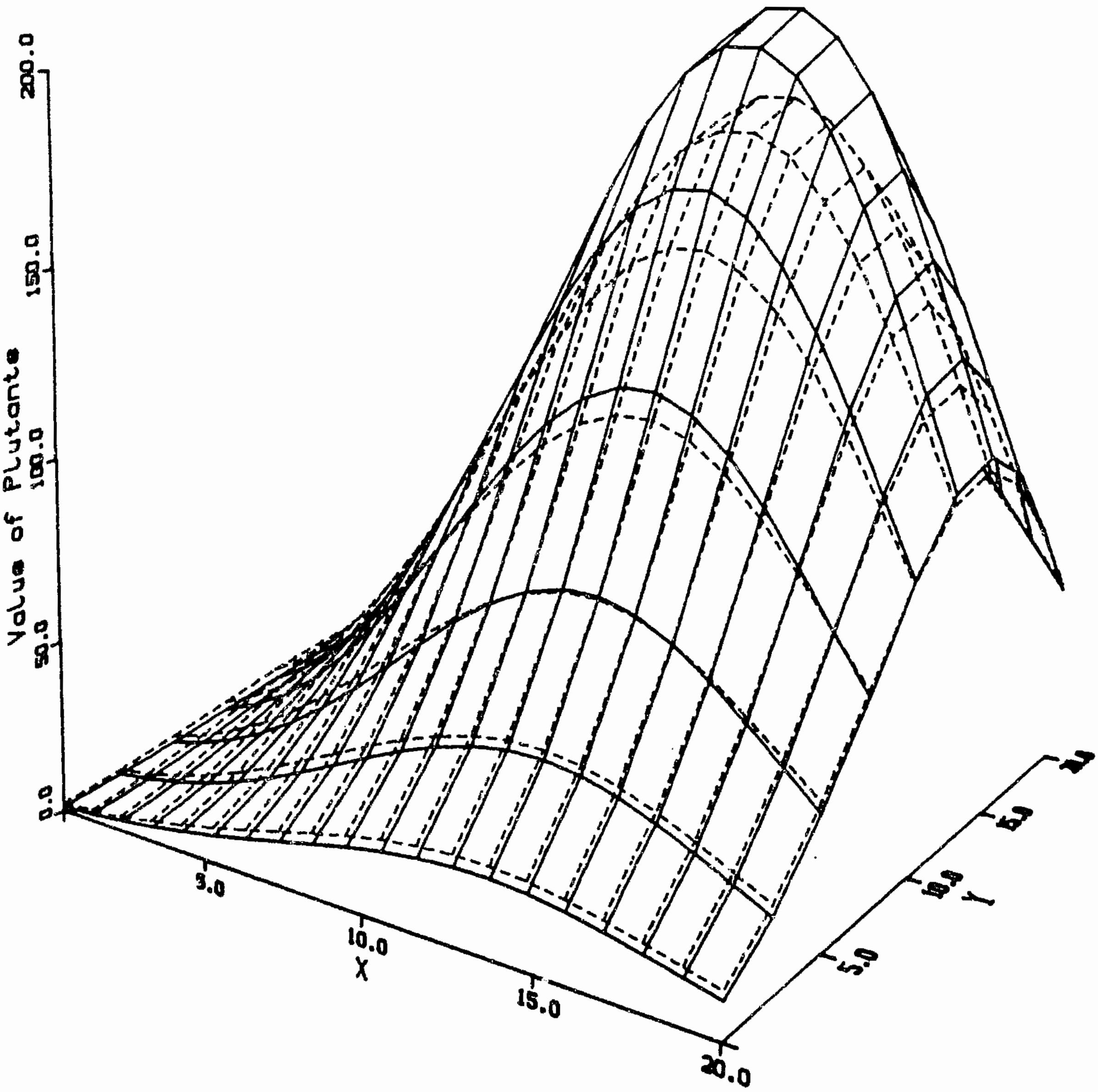

250
200
150
100
50
0
0
2
4
6
8
10
1–D CASE
Legend
Δ EXACT SOLUTION
× COMPUTED SOL.

Figure 7.2



Value of Plutants
0.0
50.0
100.0
150.0
200.0
5.0
10.0
X
15.0
20.0
5.0
10.0
Y
15.0
20.0

Figure 7.3(a)

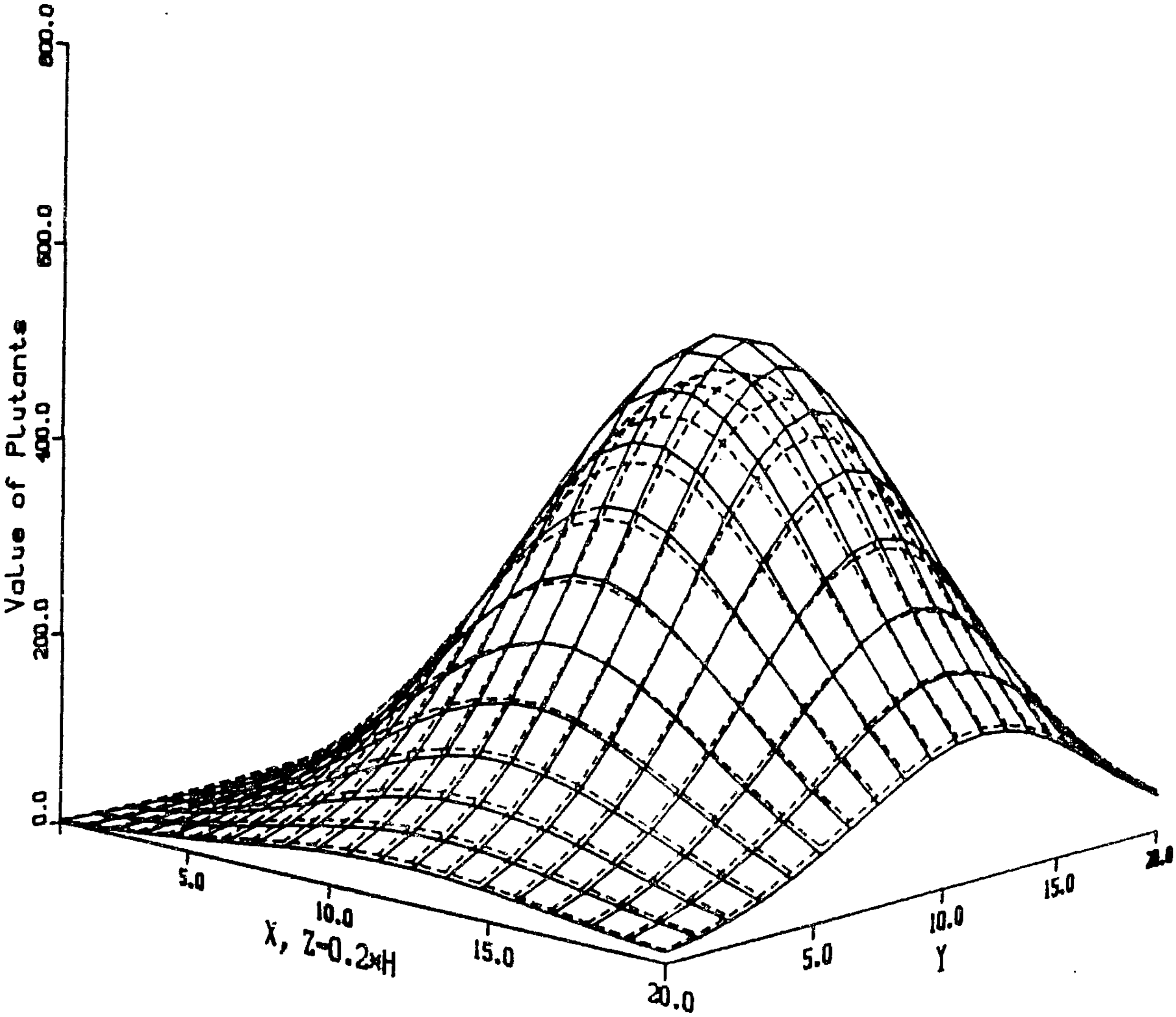

800.0
600.0
400.0
200.0
0.0
Value of PLutants
5.0
10.0
15.0
20.0
X, Z=0.2×H
5.0
10.0
15.0
20.0
Y

Figure 7.3(b)

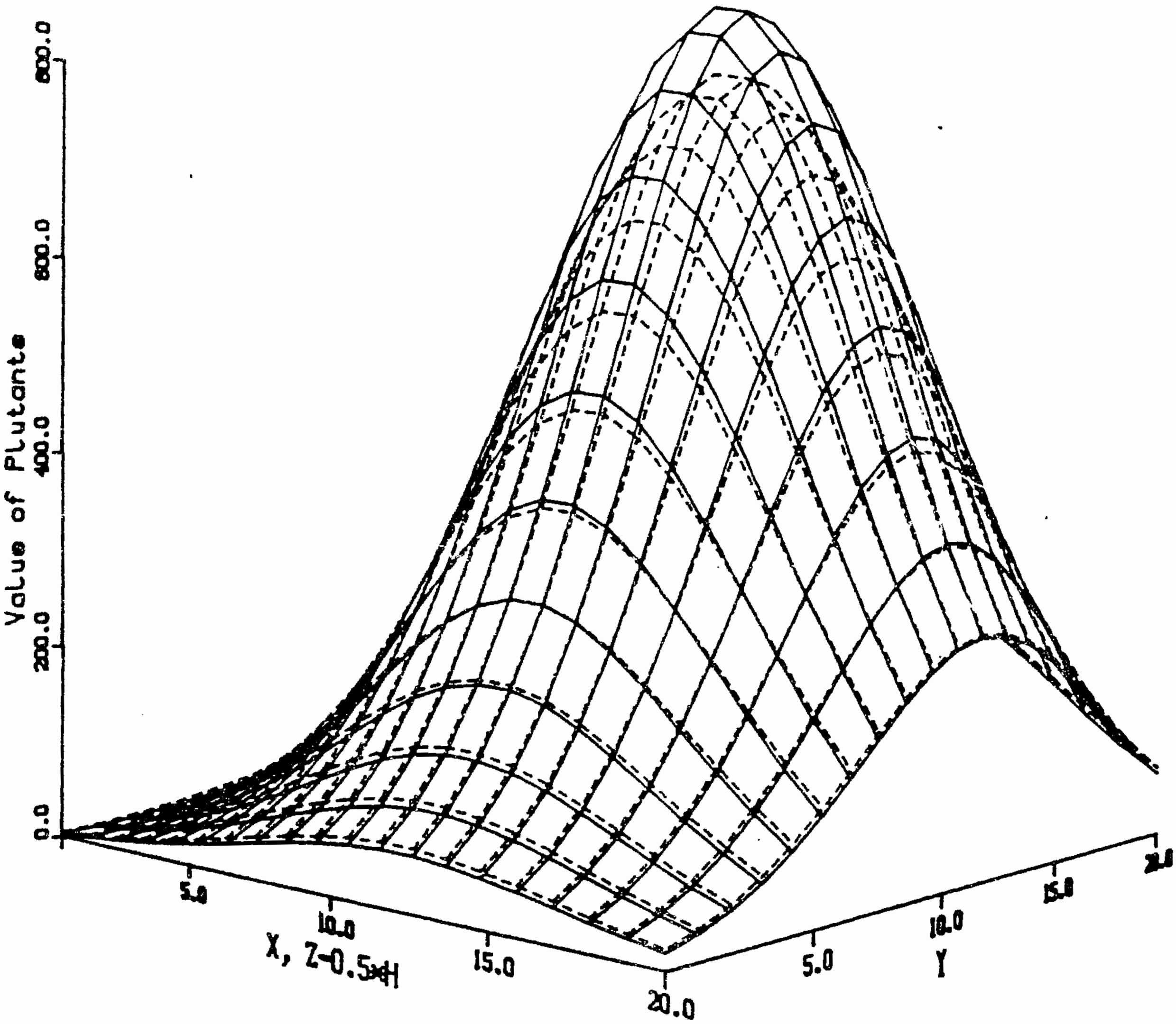
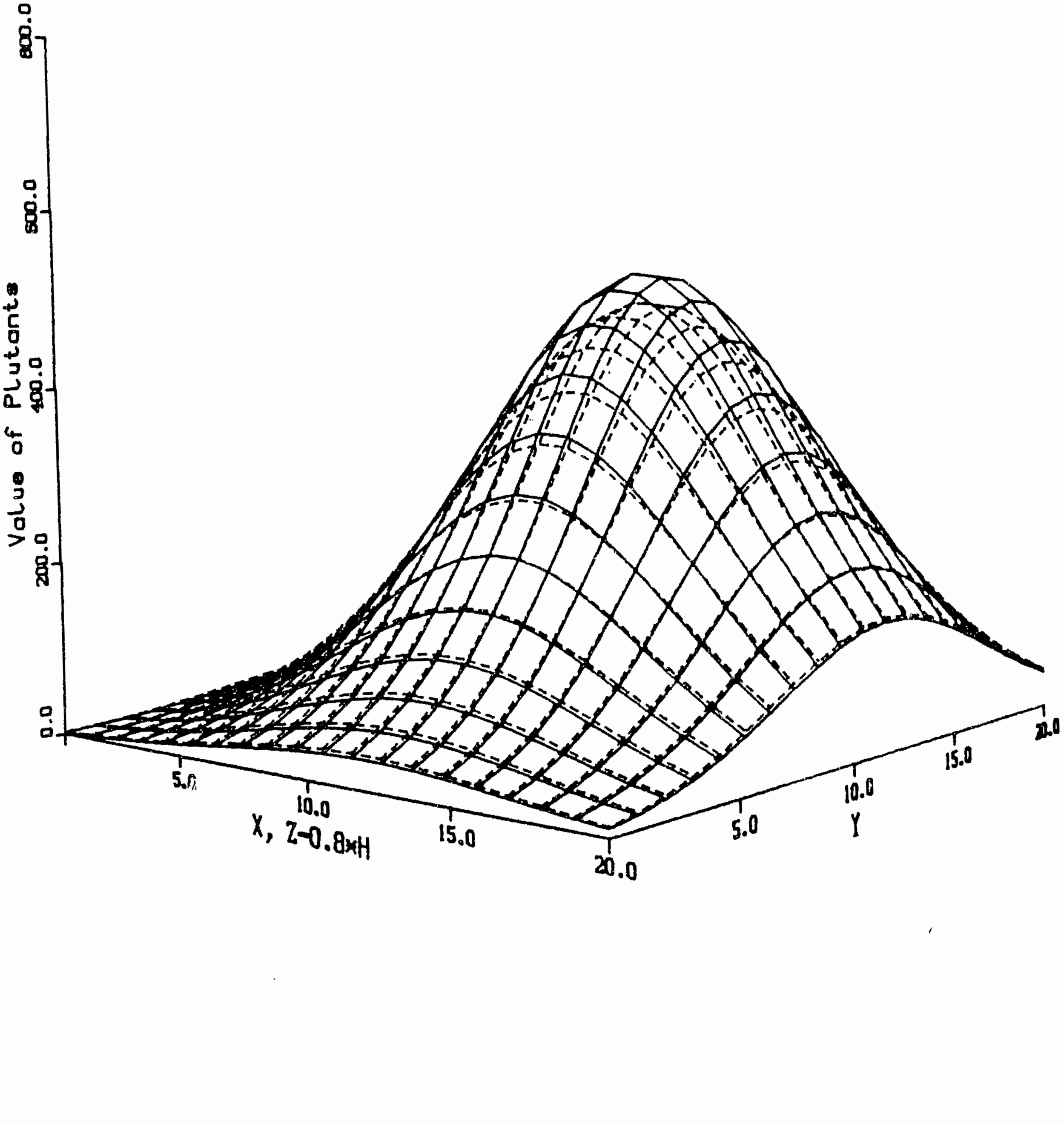

Value of Plutants
800.0
600.0
400.0
200.0
0.0
5.0
10.0
15.0
20.0
X, Z-0.5*H
5.0
10.0
15.0
20.0
Y

Figure 7.3(c)


800.0
600.0
400.0
200.0
0.0
Value of Plutants
5.0
10.0
15.0
20.0
X, Z=0.8×H
5.0
10.0
15.0
20.0
Y

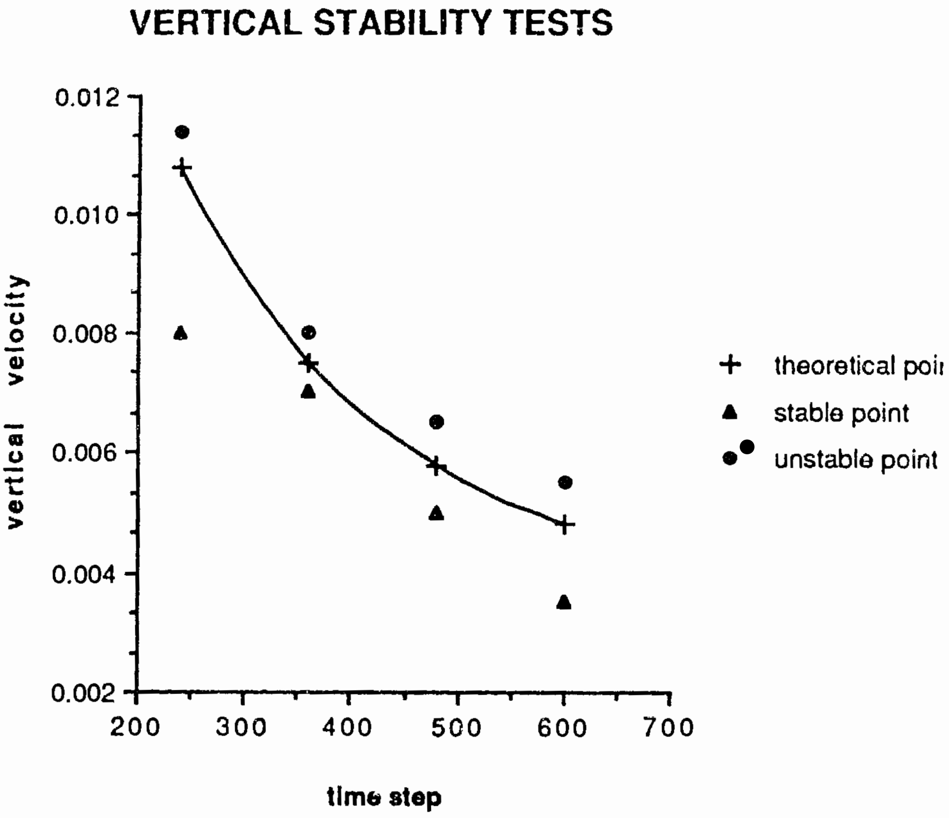

VERTICAL STABILITY TESTS
0.012
0.010
0.008
0.006
0.004
0.002
vertical velocity
200
300
400
500
600
700
time step
+ theoretical poi
▲ stable point
● unstable point

Figure 7.4

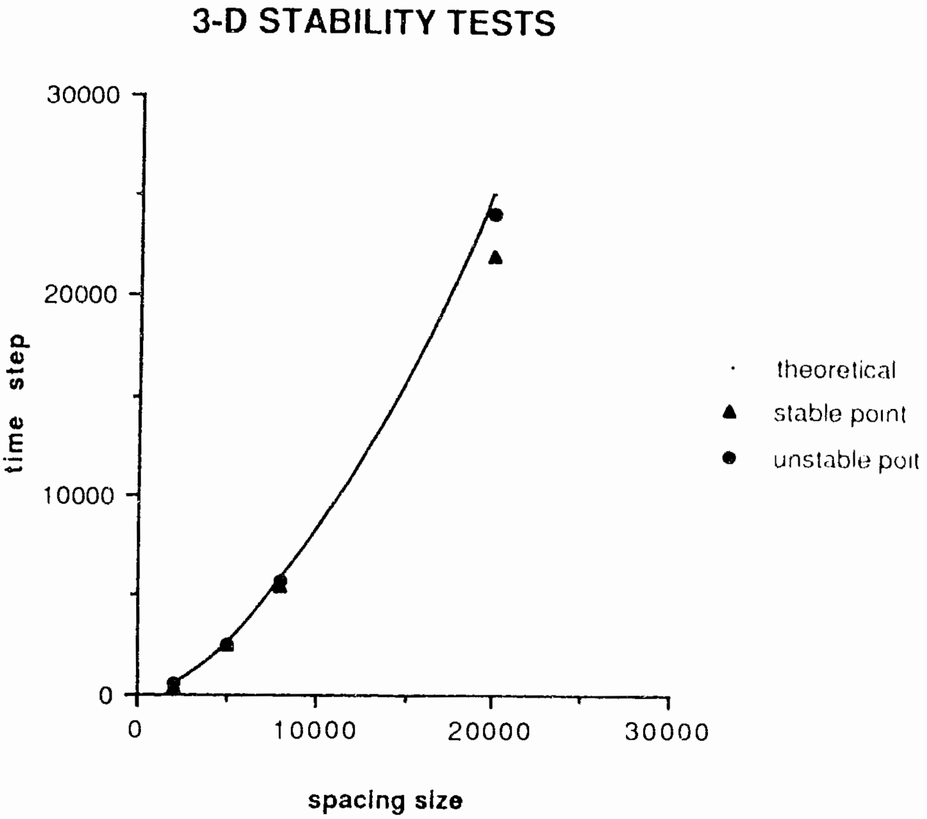

3-D STABILITY TESTS
30000
20000
10000
0
time step
0
10000
20000
30000
spacing size
· theoretical
▲ stable point
● unstable poit

Figure 7.5