NUMERICAL MODELLING OF JET-FORCED CIRCULATION IN RESERVOIRS USING BOUNDARY-FITTED COORDINATE SYSTEMS

by

Robert William Barber, B.Sc. (Hons).

August, 1990

This thesis is submitted in partial fulfilment
for the degree of Doctor of Philosophy of the
University of Salford.

Computational Hydraulics Area,
Water Resources Group,
Department of Civil Engineering,
University of Salford, U.K.

CONTENTS

ACKNOWLEDGEMENTS

The studies described in this thesis were carried out under the supervision of Dr. A.G.L. Borthwick, to whom I give my grateful thanks for his guidance, enthusiasm and encouragement throughout the research project.

I also give my thanks to : -

Professor E.M. Wilson, University of Salford, who encouraged my earlier interests in hydrodynamics.

Dr. B.H. Johnson and Dr. R.S. Bernard of the U.S. Army Engineer Waterways Experiment Station, Vicksburg, Mississippi, Dr. J.H.A. Wijbenga of Delft Hydraulics, The Netherlands and Dr. B.A. Younis, City University, London, for their valuable advice.

Mr. M. Baker and the staff of the Computer Services Section of the Department of Civil Engineering, for their help in familiarising me with the various computer systems.

Mrs. C.E. Czirok, for her assistance in the preparation of the diagrams.

The Science and Engineering Research Council, who funded the studentship.

Finally, I would like to express my gratitude to my parents for their support and assistance throughout my student career.

ABSTRACT

Throughout the past decade, interest has grown in the use of boundary-fitted coordinate systems in many areas of computational fluid dynamics. The boundary-fitted technique provides an exact method of implementing finite-difference numerical schemes in curved flow geometries and offers an alternative solution procedure to the finite-element method. The unavoidable large bandwidth of the global stiffness matrix, employed in finite-element algorithms, means that they are computationally less efficient than corresponding finite-difference schemes. As a consequence, the boundary-fitted method offers a more efficient process for solving partial differential flow equations in awkwardly shaped regions.

This thesis describes a versatile finite-difference numerical scheme for the solution of the shallow water equations on arbitrary boundary-fitted non-orthogonal curvilinear grids. The model is capable of simulating flows in irregular geometries typically encountered in river basin management. Validation tests have been conducted against the severe condition of jet-forced flow in a circular reservoir with vertical side walls, where initial reflections of free surface waves pose major problems in achieving a stable solution. Furthermore, the validation exercises have been designed to test the computer model for artificial diffusion which may be a consequence of the numerical scheme adopted to stabilise the shallow water equations.

The thesis also describes two subsidiary numerical studies of jet-forced recirculating flow in circular cylinders. The first of these implements a Biot-Savart discrete vortex method for simulating the vorticity in the shear layers of the inflow jet, whereas the second employs a stream function/vorticity-transport finite-difference procedure for solving the two-dimensional Navier-Stokes equations on a distorted orthogonal polar mesh. Although the predictions from the stream function/vorticity-transport model are confined to low Reynolds number flows, they provide a valuable set of benchmark velocity fields which are used to confirm the validity of the boundary-fitted shallow water equation solver.

CHAPTER 1
INTRODUCTION AND SYNOPSIS

An understanding of mixing processes in shallow bodies of water is important to the water resources engineer who is concerned with pollution transport problems which may arise as a result of secondary flows. In reservoirs, rivers, lakes and estuaries with poor mixing, regions of low velocities may stagnate and act as sinks for sediment. Furthermore, in rural areas, the extensive use of fertilisers may result in nitrates and phosphates accumulating in the stagnant regions leading to contamination with algae and zooplankton; this process is exacerbated if the climate is warm. The storage of water derived from nutrient-rich agricultural run-off is particularly susceptible to the occurrence of algal blooms. Whilst the presence of algae in water supply reservoirs is not entirely detrimental, since it may assist in the operation of sand filters (Ali, 1985), the presence of large quantities of micro-organisms is undesirable since additional filtration may be required, thereby increasing the costs of treatment. High concentrations of algae are a major problem in domestic water supplies since they increase the turbidity of the treated water, and may also give rise to unacceptable odours and tastes (Watson, 1989).

Although the toxic algal blooms which occurred in many reservoirs in the United Kingdom during the summer of 1989 drew attention to the intensive use of fertilisers, the problem of increased algal activity in hot summers is not new. Gray (1985) pointed out that severe eutrophication of Lough Neagh, in Northern Ireland, occurred as long ago as 1967, and led to a reduction in fish stocks as well as a temporary loss in amenities due to the formation of unsightly floating masses of blue-green algae which subsequently decayed along the shoreline. Young et al. (1988) reported a case study into the effects of ferric sulphate in controlling the phosphate levels in a rural supply reservoir. They chose to study Foxcote Reservoir in Buckinghamshire which had been regularly plagued by severe algal blooms since its construction in the mid 1950s. Prior to the implementation of a ferric sulphate dosing programme, the difficulties in treating the water supplied from the reservoir had sometimes necessitated its closure for periods of up to six months per year. Another typical example was provided by the closure of Hartley Service Reservoir in Plymouth, which became ridden with algae during the long dry summer of 1975. As a consequence, the reservoir had to be emptied and decontaminated; a very expensive procedure when the South West Water Authority (as it then was) could least afford to lose water.

The risk of losing supplies, in critical periods of drought, has compelled several

water authorities to construct treatment plants to remove excess phosphates before the water reaches the storage reservoirs (Young et al., 1988). In regions where the concentrations of algal nutrients are lower, engineers usually rely upon the natural circulation within a body of water to mix the flow and prevent stagnation (Ali et al., 1978). Wind-induced currents are often the most dominant flow feature, yet this form of agitation is uncontrollable and is often unreliable during the summer months of relatively calm weather. Therefore, the most favoured method of stirring a reservoir, to ensure adequate mixing, is accomplished through the use of suitably placed inflow jets formed from large pipes or culverts which discharge towards the centre of the storage area. The forced circulation is often characterised by the formation of two gyres, rotating in opposite directions and separated by a relatively thin high velocity throughflow zone.

The first studies into the use of high velocity jets to alleviate stagnation problems were undertaken by Cooley and Harris (1954) and White et al. (1955). These investigations concentrated on finding the optimum positions and orientations of the inlet jets for the Walton and Queen Mary Reservoirs of the Metropolitan Water Board (now Thames Water PLC). Further experimental studies into jet-forced reservoir circulation have been conducted by Sobey (1973), Sobey and Savage (1974), Ali et al. (1978), Robinson (1979), Ali and Pateman (1981) and Ali (1983, 1985). The assessment of flow velocities is of prime importance when a service reservoir is designed or upgraded. Physical models of a proposed scheme are still the main source of flow data (e.g., Falconer, 1984b), but physical models are expensive to construct and may suffer from scaling effects. Depth integrated numerical models, such as Falconer's shallow water equation solver (1976), have been increasingly used by water resource engineers to predict the nearly-horizontal flow regimes which are encountered in shallow service reservoirs. Computer models have the advantage that they are not subject to scaling effects and once they have been constructed, can readily predict the flows in a large variety of design options. Consequently, as part of the strategic management of storage reservoirs, accurate numerical models are required for predicting the flow hydrodynamics.

This thesis presents three separate computational methods for simulating the momentum transfer in nearly-horizontal flow regimes. The numerical procedures include a Biot-Savart discrete vortex model, a stream function/vorticity-transport finite-difference simulation of the Navier-Stokes equations and a boundary-fitted technique for solving the shallow water equations using non-orthogonal curvilinear coordinate meshes. Instead of attempting to investigate the properties of the various numerical schemes by simulating the flow in prototypical domains with complicated

geometries and bathymetries, it was decided that a more systematic review could be achieved by modelling the hydrodynamics in more readily defined geometries. Consequently, the numerical schemes are assessed by simulating jet-forced circulation patterns in flat-bottomed circular reservoirs with a single inlet and a single outlet. The geometries are chosen so that the hydrodynamic predictions may be compared against the benchmark analytical solutions provided by Dennis (1974) and Mills (1977), and the numerical and experimental results presented by Falconer (1976, 1980). As shown later in this thesis, the prediction of recirculating eddies formed by a rapid throughflow jet allows the computational schemes to be validated under highly non-linear flow conditions.

The introductory paragraphs have outlined the need for accurate numerical models for predicting the velocity fields in storage reservoirs. Although the thesis investigates three separate methods for computing nearly-horizontal flow regimes in a *circular* reservoir, the main emphasis of the study has been directed towards the development of a non-orthogonal curvilinear systems model of the shallow water equations applicable to irregular or curved geometries encountered in prototypical reservoirs. The curvilinear shallow water equation solver is extremely versatile and is able to predict the depth-averaged flow fields in many other types of civil engineering application, such as in river channels, tidal harbours and estuaries. It is believed that this is the first finite-difference model incorporating the shallow water equations in curvilinear coordinate form to have been developed and validated in the United Kingdom. The computational scheme's ability to simulate flows within awkwardly shaped boundaries is a considerable improvement over more conventional Cartesian finite-difference procedures. It is anticipated that boundary-fitted coordinate systems will replace Cartesian grid methods for the majority of flow simulations in the next decade. The following text provides a synopsis of the investigations detailed in this thesis.

Chapter 2 describes a 'direct summation' or Biot-Savart discrete vortex model for qualitatively assessing jet-forced circulation in circular tanks and reservoirs. The computational procedures utilised here are based upon the numerical model proposed by Borthwick et al. (1988). A two stage conformal mapping is employed to transform the circular flow domain onto a rectangle; this enables the irrotational flow within the reservoir to be determined via an exact analytical technique. Discrete vortices are then added at the inlet to simulate the inflow boundary layers. These vortices are allowed to flow into the reservoir and their positions are updated using a first order accurate finite-difference Lagrangian tracking procedure. The velocity field is computed by summing the contributions from the uniform potential

flow, the vortices and their images in the walls of the cylinder*. In an extension to Borthwick et al.'s work (1988), the computational scheme, described in Chapter 2, permits the user to employ either potential or Rankine vortices and also allows a stochastic random walk technique to be superimposed on the pure advection Lagrangian tracking scheme; the random walks are used to simulate the viscous diffusion effects of real fluids.

Chapter 3 details a two-dimensional finite-difference numerical model of the Navier-Stokes equations in a circular flow domain. The governing hydrodynamic expressions are rewritten in stream function/vorticity-transport (ψ,ω) form and are applied to a distorted polar coordinate collocated finite-difference mesh. Following Butler (1978b), Houston and Butler (1979) and Vemulakonda et al. (1985), the grid system is transformed using an algebraic mapping, in order to increase the mesh resolution near the boundaries of the flow domain. A central difference alternating-direction-implicit (A.D.I.) discretisation technique is employed to solve the transformed governing equations on the distorted polar mesh. The chapter concentrates on the development of low Reynolds number laminar flows; these predictions are useful, in that, they deal with exact flow behaviour without recourse to empiricism until the computer simulation breaks down.

The third (and major) numerical scheme, described from Chapter 4 onwards, is concerned with the solution of the shallow water equations on boundary-fitted non-orthogonal coordinate systems. Real-life watercourses, such as reservoirs, rivers, lakes and estuaries have irregular boundaries which may be curved and, in some cases, highly contorted. In the past, numerical modelling of nearly-horizontal flow regimes has usually been undertaken using Cartesian grid finite-difference discretisations of the depth-averaged Reynolds equations (otherwise known as the shallow water equations). In the United Kingdom, most models are based upon the work of Falconer (1976, 1980) who derived his approach from that of Leendertse (1967). However, Cartesian schemes have the disadvantage that it is difficult to model curved perimeters accurately. Leendertse (1967), Abbott et al. (1973) and Hodgins (1977) approximated the curvature of the boundary by using a 'staircase' of points to follow the perimeter. The alternative method, utilised by Kuipers and Vreugdenhil (1973) and Falconer (1976, 1980), also employs a stepped pattern to follow the curved boundary but unlike the Leendertse approach, where the 'staircase'

* The vortex images ensure that the cylinder surface remains a streamline (Milne-Thomson, 1968).

straddles the flow domain, the hydraulic region is located entirely *within* the numerical grid; interpolation has to be used to redistribute the fluid in the boundary regions so that overall fluid mass is conserved. In either case, unnecessary (or spurious) vorticity may be generated at each of the boundary nodes. In the second method, the problems may be aggravated by the creation of fictitious depth gradients in the vicinity of the perimeter. Weare (1979) has also shown that the maximum allowable time step for Cartesian grid implicit models applied to problems involving irregular boundaries is severely restricted compared with cases where the boundaries are aligned with the computational grid axes. Moreover, in Cartesian schemes, it is relatively difficult to concentrate extra grid refinement in zones of interest, such as in boundary layers. At first sight, it may appear that finite-element methods are ideally suited to solving flow problems on irregularly shaped domains. However, as discussed by Weare (1976a), the large bandwidth of the global stiffness matrix employed in the finite-element methodology means that the technique is computationally less efficient than corresponding finite-difference schemes.

Boundary-fitted coordinate systems provide an approach which combines the best aspects of finite-difference discretisation with the grid flexibility usually attributed to finite-element procedures. In essence, a coupled pair of elliptic equations are solved to generate the grid, the governing flow equations are rewritten in curvilinear form and then discretised using finite-differences ready for solution by computer. The idea behind boundary-fitted systems originated during the early 1970s in the U.S. aerospace industry as a response to NASA's need to predict high velocity flow patterns around space rockets. Since then, the boundary-fitted methodology has been utilised for the solution of a wide variety of fluid dynamic problems in aeronautical and mechanical engineering. Surprisingly, even though numerical grid generation techniques have been thoroughly documented, few publications are as yet available concerning the solution of the two-dimensional shallow water equations using curvilinear coordinate systems. Perhaps the earliest study on the complete transformed non-linear shallow water equations was conducted by Johnson (1980, 1982) and Johnson et al. (1982) in the United States. Other curvilinear models include Häuser et al.'s (1985, 1986b) and Raghunath et al.'s (1987) numerical schemes for the linearised shallow water equations. Wijbenga (1985a, 1985b) and Willemse et al. (1985), of Delft Hydraulics in The Netherlands, have developed an alternative computational approach requiring an orthogonal curvilinear mesh.

Chapters 4, 5 and 6 describe the non-orthogonal shallow water equation solver. The computational procedures consist of two modules; the first generates the grid within the prescribed flow domain, whereas the second solves the transformed shallow

water equations using a semi-implicit discretisation technique, based upon Leendertse's (1967) Cartesian numerical scheme. Chapter 4 details the boundary-fitted concept, proposed by Thompson et al. (1974), and describes the methods which are employed in the production of curvilinear grids whilst Chapter 5 presents the derivation of the depth-averaged Reynolds and two-equation turbulence transport equations and their conversion for use with non-orthogonal coordinate systems. The discretisation of the depth-averaged momentum and continuity expressions is described in Chapter 6. To overcome the destabilising influences of the non-linear advective acceleration terms, when modelling flow conditions where low values of eddy viscosity prevail, the *cross-advective* acceleration terms are approximated using a combination of second order upwind and weighted central differences, in a similar manner to that employed by Willemse et al. (1985) for an orthogonal shallow water equation scheme.

Chapter 7 describes a comprehensive series of flow tests which are utilised to validate the computational procedures of the curvilinear shallow water equation scheme. In keeping with the flow geometries employed in Chapters 2 and 3, the main validation exercises are based upon the prediction of jet-forced circulation patterns in constant depth, circular reservoirs. The tests are designed to examine the computer model for numerical (artificial) diffusion which may arise from the second-order upwind finite-difference scheme adopted to stabilise the shallow water equations. However, in contrast to the earlier numerical solution procedures of Chapters 2 and 3, the curvilinear shallow water equation discretisation technique is also able to model irregularly shaped flow domains.

The validation tests of Chapter 7 utilise *zero-equation* turbulence models, and therefore, it is desirable to extend the curvilinear computational scheme by providing a higher order turbulence transport simulation to predict the spatial variations of the eddy viscosity coefficient. Chapter 8 details two finite-difference approaches for approximating the depth-averaged $k-\epsilon$ transport equations on non-orthogonal boundary-fitted grids. The first computational scheme utilises an alternating-direction-implicit (A.D.I.) method in conjunction with central differences for the advective terms, whereas the second employs an explicit forward-time discretisation with second order upwind and weighted central differences for the non-linear advective components.

As well as providing detailed simulations of jet-forced flow regimes in circular reservoirs, the thesis demonstrates the advantages of boundary-fitted curvilinear coordinate systems modelling of shallow water situations. Provided the user is prepared to accept an increase in the complexity of the governing hydrodynamic

equations and is prepared to develop new computational algorithms for the solution of these equations, the boundary-fitted approach offers the water resources engineer a powerful and flexible tool for predicting the flow conditions in irregularly shaped geometries.

CHAPTER 2
DISCRETE VORTEX MODELLING

2.1 Introduction

The utilisation of discrete vortices as a means of simulating the viscous flow described by the Navier–Stokes equations provides numerical modellers with a powerful computational technique. The earliest research into the discrete vortex methodology was carried out by Rosenhead (1931) while studying the surface instability between two flow streams travelling in opposite directions. Rosenhead replaced the continuous sheet of vorticity between the two layers by a finite number of elemental vortices, and then used a time–stepping finite–difference procedure to calculate the deformation of the vortex sheet due to the mutually induced velocities at each vortex position. Although the calculation was only continued for four time steps (since the simulation was performed by hand), the results clearly showed the vortex sheet rolling up into clusters of vortices. Rosenhead demonstrated the feasibility of the discrete vortex method, but it was only with the advent of computers that the number of elemental vortices could be increased sufficiently to allow adequate simulation of the complex shear layers in wakes and jets. It is the intention of this introduction to discuss the developments that have been made since Rosenhead's pioneering work. For a detailed exposition on the discrete vortex method, the reader is referred to the review article by Sarpkaya (1989).

Abernathy and Kronauer (1962) modelled the interaction of two free shear layers with discrete vortices and simulated the development of a typical von Kármán vortex street; this was the precursor to the use of the technique for simulating flows past obstacles. Several authors, including Gerrard (1967), Sarpkaya (1968), Laird (1971), Chaplin (1973), Stansby (1977) and Sarpkaya and Schoaff (1979) have modelled the flow past a circular cylinder using discrete vortices superimposed on the potential flow solution. Varying degrees of complexity for introducing the nascent vortices have been devised, ranging from a fixed point of introduction (e.g. Gerrard; Laird; Chaplin) to a variable separation point (e.g. Sarpkaya; Stansby; Sarpkaya and Schoaff). Flows past square–ended bluff bodies were modelled by Clements and Maull (1975); in this case, fixed nascent vortex positions were utilised since separation would always occur at the sharp corners of the obstacle. Clements and Maull employed a Schwarz–Christoffel transformation to map the square ended body (extending to infinity in the upstream direction) onto the upper half of a 'transformed' coordinate frame. This mapping procedure allowed the analytical determination of the irrotational flow around the square ended obstacle.*

All early discrete vortex simulations employed pure Lagrangian numerical schemes to simulate the advection of vorticity. The individual velocities of the discrete vortices were obtained by summing the velocity contributions from all the vortices in the field and superimposing this on the potential flow solution. For this reason, the technique is often described as the 'direct summation method'. The methodology may also be referred to as the Biot-Savart approach on account of the similarity between the magnetic field intensity caused by an electric current and the velocity induced by a vortex element.

The advantage of the Biot-Savart technique is that it is entirely independent of grid based numerical methods, thereby eliminating the artificial diffusion usually associated with Eulerian finite-difference schemes. There are, however, two main difficulties with the direct summation method. First, the elemental vortices have singularities at their centres of rotation which produce extremely large velocities in the immediate vicinity of the vortices. This can cause instabilities in the simulation of the roll-up of a vortex sheet, as the vortices in the centre of a cluster move closer to each other. There is a tendency for the elemental vortices to orbit around themselves in an unrealistic chaotic manner, rather than roll up into the expected spiral. In order to overcome this particular difficulty, various 'cut-off' schemes have been devised to desingularise the velocity field at the centres of the vortices; examples include Chorin and Bernard's (1973) vortex 'blob' and Chaplin's (1973) use of Rankine vortices. The second disadvantage with the Biot-Savart approach is that the number of arithmetic operations required to update the vortex positions by one time step is approximately proportional to N^2 where N is the number of vortices. Thus the computational time required per time step increases dramatically as more vortices are added. One method, which partially overcomes this difficulty, is to use a computer algorithm to amalgamate vortices if they are closer than a certain distance apart. Not only will this help to alleviate the computational costs, but it can also remove the propensity of vortices to orbit one another.

The dependency of the Biot-Savart approach upon a work load proportional to N^2 implies that the technique is unable to cope with large numbers of vortices. Consequently, from the late 1970s a 'hybrid' Lagrangian-Eulerian discrete vortex method has been increasingly used instead of the direct summation approach (e.g. Baker, 1979; Stansby and Dixon, 1983; Smith and Stansby, 1987). The technique

*

A similar mapping principle is used in the present study, so that the potential flow inside a circular cylinder can be specified without recourse to numerical methods.

is now referred to as the vortex-in-cell (V.I.C.) method and employs a Lagrangian method to track the vortex positions and a Eulerian grid-based finite-difference scheme to solve the elliptic stream function equation that defines the velocity field. Implementation of the V.I.C. method is performed essentially in four separate stages:

(a) The vorticity of each discrete vortex in the field is allocated to the four mesh nodes surrounding the particular vortex position; the vorticity, ω, being divided according to an area-weighting technique.

(b) Poisson's equation ($\nabla^2\psi = -\omega$) is solved using a finite-difference scheme to obtain the stream function, ψ at all non-boundary mesh points.

(c) The velocity components at the computational nodes are calculated from the stream function distribution [i.e., $u=\partial\psi/\partial y$ and $v=-\partial\psi/\partial x$ for Cartesian grids, and $V_r=1/r.\partial\psi/\partial\theta$ and $V_\theta=-\partial\psi/\partial r$ for polar finite-difference meshes].

(d) An area-weighting bilinear interpolation technique is then employed to calculate the velocity components of the individual elemental vortices within the cells; these velocities are used to update the vortex positions to the new time level.

The main advantage with the vortex-in-cell approach is that the cost of the computational technique is linear in N. This allows the method to use a larger number of vortices. Furthermore, the apportionment of the vorticity to the four nodes of a cell effectively removes the singularity in the velocity field without having to resort to vortex 'blobs' or Rankine vortices. However, the drawback with the V.I.C. technique is that the grid independence of the original Biot-Savart approach has been abandoned. Savings from the removal of the N^2 cost constraint can be completely destroyed if a very fine grid is employed due to the computationally expensive Poisson solver. As pointed out by Sarpkaya (1989), classical Biot-Savart techniques therefore still have an important role to play in discrete vortex modelling.

This chapter implements a Biot-Savart discrete vortex model for calculating two-dimensional jet-forced flow within a flat-bottomed circular reservoir, and follows the methodology adopted by Borthwick et al. (1988). The first stage of the model consists of conformally mapping the circular perimeter of the reservoir onto a rectangle by means of a Schwarz-Christoffel transformation; the inlet and outlet openings are represented as opposite sides of the transformed rectangular domain. A potential flow solution is then obtained for the flow inside the rectangle and this is transformed to give the potential solution inside the circle. In the second stage of the simulation, discrete vortices are added at the inlet of the physical system in order to model the inflow shear layers. Velocity components resulting from the discrete vortices and their images in the walls of the reservoir are superimposed on

the uniform potential flow solution. The positions of the vortices are updated using a first order accurate Lagrangian tracking procedure. Although the numerical scheme fails to produce quantitative agreement with the finite-difference stream function/vorticity-transport discretisation (described in Chapter 3), the discrete vortex model, nevertheless, predicts qualitatively some of the important features of jet-forced reservoir flow.

Figure 2.1 illustrates a typical asymmetrical circular reservoir with a single inlet and single outlet. The radial inflow jet produces separation at the sharp inlet corner points and the shear layers which are formed cause two recirculating eddies (of opposite senses of rotation) to develop either side of the main throughflow. Results from the stream function/vorticity-transport finite-difference model indicate that the throughflow jet will orientate itself towards the outlet. The discrete vortex scheme reported herein qualitatively predicts these features in the early stages of the simulation. However, the numerical technique fails to reach steady-state and a chaotic motion eventually overwhelms the flow model.

2.2 Numerical scheme

2.2.1 Irrotational flow solution

A two stage conformal mapping is used to transform the circular flow domain onto a rectangle. This allows the potential flow to be determined analytically. Figure 2.2 shows the physical domain, R, located at the origin of the complex z-plane, $z = x + iy$ where $i = \sqrt{-1}$. The flow problem is non-dimensionalised and therefore the reservoir is represented by a unit circle. Points a, b, c and d on the perimeter of the cylinder define the inlet and outlet corners.

During the first stage of the mapping procedure (illustrated in Figure 2.2), the circular reservoir is transformed onto the upper half of the s-plane ($s = p + iq$) using the function:

$$s = - i \left[\frac{z - i}{z + i} \right] \quad , \tag{2.1}$$

which may be rewritten in terms of the x and y coordinates as

$$s = - i \left[\frac{x^2 + y^2 - 1 - 2ix}{x^2 + y^2 + 2y + 1} \right] \quad . \tag{2.2}$$

On the perimeter of the circle, $x^2 + y^2 = 1$ and therefore

$$s = p + iq = -i \left[\frac{-2ix}{2 + 2y} \right] \quad . \tag{2.3}$$

Thus, on the circumference of the reservoir

$$p = -\frac{x}{1 + y} \qquad \text{and} \qquad q = 0 \quad , \tag{2.4}$$

and consequently the perimeter wall is mapped onto the $q = 0$ axis. The points a', b', c' and d' in Figure 2.2 represent the images of the corner locations after the transformation.

The second stage of the conformal mapping employs a Schwarz–Christoffel transformation to convert the upper half of the s–plane into a rectangular region in the ζ–coordinate frame ($\zeta = \eta + i\xi$). Figure 2.3 illustrates the mapping, and shows the inlet and outlet openings, a''b'' and c''d'' forming opposite sides of the rectangular region, R''. The Schwarz–Christoffel transformation is defined by

$$\frac{d\zeta}{ds} = A\,(s-p_1)^{[\alpha_1/\pi-1]}(s-p_2)^{[\alpha_2/\pi-1]}\ldots.(s-p_n)^{[\alpha_n/\pi-1]} \tag{2.5}$$

where A is an arbitrary constant, α_1, α_2 ... α_n are the interior angles of the region in the ζ–plane, p_1, p_2 ... p_n are the p–values of the corner nodes and n is the total number of vertices. In this particular case, $n = 4$ and $\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \pi/2$. Assuming A to be unity allows the transformation to be simplified to

$$\frac{d\zeta}{ds} = \frac{1}{\sqrt{(s-a')(s-b')(s-c')(s-d')}} \quad . \tag{2.6}$$

The next stage of the numerical scheme involves generating a uniform potential flow within the rectangular region, R''. This is achieved by equating the η– and ξ–coordinates to the velocity potential, Φ, and stream function, ψ, respectively. Thus the complex potential, $w = \Phi + i\psi$, is equal to ζ.

The complex velocity, ν is defined as

$$\nu = \frac{\partial w}{\partial x} = \frac{\partial \Phi}{\partial x} + i\frac{\partial \psi}{\partial x} = u - iv \tag{2.7}$$

where u and v are the velocity components in the x– and y–directions. Using the Chain Rule, the complex velocity may be expressed as

$$\nu = \frac{\partial w}{\partial x} = \frac{dw}{dz}\frac{\partial z}{\partial x} \quad . \tag{2.8}$$

However, $\partial z/\partial x = 1$ and since the complex potential is equal to ζ, then

$$\nu = \frac{dw}{dz} = \frac{d\zeta}{dz} \quad . \tag{2.9}$$

The potential flow solution in the physical z-plane can therefore be determined by using the Chain Rule to expand $d\zeta/dz$, viz.

$$\nu = \frac{d\zeta}{dz} = \frac{d\zeta}{ds}\frac{ds}{dz} \quad . \tag{2.10}$$

Differentiating equation (2.1) with respect to z, yields

$$\frac{ds}{dz} = \frac{2}{(z+i)^2} \quad , \tag{2.11}$$

and hence, the complex velocity at any point in the circular reservoir may be found by substituting equations (2.11) and (2.6) into (2.10):

$$\nu = u - iv = \frac{1}{\sqrt{(s-a')(s-b')(s-c')(s-d')}} \cdot \frac{2}{(z+i)^2} \quad . \tag{2.12}$$

The complex velocity presented in the above expression must be scaled so that the average velocity across the inlet of the physical domain is unity. This is achieved by dividing equation (2.12) by the unscaled discharge, Q, and multiplying by the inlet width, L. The unscaled discharge through the reservoir is given by the difference in stream function values between the two inlet corner points at a'' and b''. Since a'' and b'' lie on the same equipotential line,

$$Q = \psi_{b''} - \psi_{a''} = \frac{1}{i}\left(\zeta_{b''} - \zeta_{a''}\right) = \frac{1}{i}\int_{a'}^{b'} \frac{d\zeta}{ds}\, ds \quad . \tag{2.13}$$

The limits of this integral lie on the real axis of the s-plane and so the integrand can be expressed entirely in terms of p. After rearrangement, the unscaled discharge becomes

$$Q = \int_{b'}^{a'} \frac{dp}{\sqrt{(a'-p)(p-b')(p-c')(p-d')}} \quad . \tag{2.14}$$

The above equation is an incomplete elliptic integral of the first kind and may be rewritten according to Byrd and Friedman (1954) as

$$Q = t\ F(\lambda, r) \tag{2.15}$$

where

$$t = \frac{2}{\sqrt{(a'-c')(b'-d')}}$$

$$\lambda = \sin^{-1}\left[\sqrt{\frac{(a'-c')(p-b')}{(a'-b')(p-c')}}\right] = \frac{\pi}{2} \quad \text{for } p = a$$

and

$$r = \sqrt{\frac{(a'-b')(c'-d')}{(a'-c')(b'-d')}} \quad .$$

F is a complete elliptic integral of the first kind and may be evaluated as a series approximation:

$$Q = t\ F(\pi/2, r) = t\ \frac{\pi}{2}\left\{1 + \left[\frac{1}{2}\right]^2 r^2 + \left[\frac{1.3}{2.4}\right]^2 r^4 + \left[\frac{1.3.5}{2.4.6}\right]^2 r^6 + \ldots\right\} . \tag{2.16}$$

Thus the scaled uniform complex velocity, ν_s, for the irrotational flow solution is given by

$$\nu_s = u_s - iv_s = \frac{L}{Q} \cdot \frac{1}{\sqrt{(s-a')(s-b')(s-c')(s-d')}} \cdot \frac{2}{(z+i)^2} \tag{2.17}$$

where u_s and v_s are the scaled velocity components in the x- and y-directions.

2.2.2 Discrete vortex simulation

The shear layers formed between the inflow jet and the inlet boundary are modelled by adding discrete vortices to the irrotational flow solution of equation (2.17). If the simulation employs potential vortices, the complex velocity at the point, z = x + iy due to the superposition of the scaled uniform irrotational flow and the vortices is given by

$$u - iv = \nu_s - \frac{i}{2\pi}\sum_{j=1}^{N}\frac{\Gamma_j}{z - z_j} + \frac{i}{2\pi}\sum_{j=1}^{N}\frac{\Gamma_j}{z - \frac{1}{\bar{z}_j}} \tag{2.18}$$

where Γ_j is the circulation of the j^{th} vortex (positive for anticlockwise rotation), z_j is the position of the j^{th} vortex, N is the total number of vortices in the flow field and the overbar denotes the complex conjugate. The velocity field due to the vortices is calculated as two separate summations; the first is the velocity contribution from the actual vortices, whilst the second arises from the fact that each vortex must have an image of opposite circulation at the 'inverse' position *outside* the reservoir. These vortex images ensure that the cylinder surface remains a streamline (see Milne-Thomson, 1968).

The singularity at the centre of a potential vortex may induce excessive velocities when two vortices approach each other; following Chaplin (1973), this difficulty is overcome through the use of Rankine vortices. Thus the complex velocity at the point, $z = x + iy$ becomes

$$u - iv = v_s - \frac{i}{2\pi} \sum \frac{\Gamma_j}{z - z_j} + \frac{i}{2\pi} \sum \frac{\Gamma_j}{z - \frac{1}{\bar{z}_j}}$$

$$- \frac{i}{2\pi r_0^2} \sum \Gamma_j \overline{(z - z_j)} + \frac{i}{2\pi r_0^2} \sum \Gamma_j \overline{(z - \frac{1}{\bar{z}_j})} \qquad (2.19)$$

where r_0 is the radius of the vortex core. The first summation is utilised for vortices whose cores do not cover the point, z, whilst the third summation is taken over the remaining vortices. Rankine velocity profiles are also used for the vortex images. Therefore, the second summation is employed whenever the core of an image vortex does not enclose the point, z, whereas the fourth summation is used if z lies within the image core.

2.2.3 Introduction of nascent vortices

The strengths of the vortices used to simulate the shear layers of the inflow jet are found from a consideration of the vorticity flux across the inlet boundary layers. Figure 2.4 illustrates the assumed velocity profile across the inflow, and defines a coordinate system with x' and y' measured parallel and perpendicular to the lower inlet wall. The vorticity flux across the boundary layer is given approximately by Raudkivi and Callander (1975) as

$$\frac{d\Gamma}{dt} = \int_0^{\delta} u'\omega \, dy' = -\frac{1}{2} U^2 \qquad (2.20)$$

where u' is the velocity component in the x'-direction, ω is the vorticity, δ is the nominal thickness of the boundary layer and U is the free stream velocity. In the discrete vortex model, the vorticity flux across the boundary layer is equivalent to the circulation of an individual discrete vortex, Γ_j, divided by the time increment, Δt_v, between the introduction of nascent vortices. Since the inflow velocity has been non-dimensionalised (U=1), the strength of vortices originating from the lower boundary layer can thus be expressed as

$$\Gamma_j = -\frac{\Delta t_v}{2} \qquad (2.21)$$

where the circulation, Γ_j, is positive for anticlockwise rotation. The vortices introduced at the upper boundary separation point are calculated in a similar manner:

$$\Gamma_j = +\frac{\Delta t_v}{2} \qquad (2.22)$$

2.2.4 Vortex tracking

As explained in Section 2.1, the computer time necessary to update the vortex positions in a Biot-Savart approach is approximately proportional to N^2 where N is the number of vortices. This is because the velocity of an individual vortex is found by summing the velocity contributions from each of the remaining N-1 vortices. In order to avoid using excessive computer resources, the time increment, Δt_v, between the addition of nascent vortices must be carefully chosen. Numerical experimentation has revealed, however, that although Δt_v may be suitable from the viewpoint of computer C.P.U. time, it is generally too large for the finite-difference vortex tracking discretisation; vortices near the circular perimeter of the reservoir are frequently transported across the boundary wall. Consequently, a smaller time step, Δt_m * is employed to update the vorticity field.

* For convenience, Δt_v is chosen to be a multiple of Δt_m. The addition of new vortices at the inlet will then coincide with the time steps employed in the Lagrangian tracking procedure.

The vortex positions are advanced over the 'vortex movement' time step, Δt_m, using a first order accurate finite-difference scheme:

$$\left.\begin{aligned} x_j(t+\Delta t_m) &= x_j(t) + u_j(t).\Delta t_m \\ \text{and} \\ y_j(t+\Delta t_m) &= y_j(t) + v_j(t).\Delta t_m \end{aligned}\right\} \quad (2.23)$$

where u_j and v_j are the velocity components of the j^{th} vortex [determined from either equation (2.18) or (2.19)]. In the event of a vortex being advected across a solid perimeter wall, it is 'reflected' back into the flow domain. Vortices which leave through the outlet of the reservoir are removed from subsequent flow calculations.

2.2.5 Random walk

The vortex tracking scheme presented in Section 2.2.4 only models the advective processes of the discretised vorticity field. Chorin (1973) proposed that the viscous effects could be simulated by treating the diffusive and advective phenomena in separate numerical procedures. This is equivalent to replacing the exact vorticity-transport equation (simultaneous advection and diffusion of vorticity):

$$\frac{D\omega}{Dt} = \frac{\partial\omega}{\partial t} + \mathbf{u}.\nabla\omega = \nu\,\nabla^2\omega \quad (2.24)$$

by two *sequential* equations:

$$\frac{\partial\omega}{\partial t} + \mathbf{u}.\nabla\omega = 0 \quad (2.25a)$$

and

$$\frac{\partial\omega}{\partial t} = \nu\,\nabla^2\omega \quad (2.25b)$$

where **u** represents the velocity *vector*.

A stochastic technique is used to model the pure diffusion of equation (2.25b); at the end of the advective tracking stage, each vortex undergoes a random walk in two orthogonal directions. The standard deviation of these random walks must be compatible with the analytical solution of the diffusion equation (2.25b) and therefore, the fluctuating random velocity components of a vortex are generated as (see Chorin, 1973):

$$u_r = r_1 \left[\frac{2\nu}{\Delta t_m} \right]^{\frac{1}{2}}$$

and

$$v_r = r_2 \left[\frac{2\nu}{\Delta t_m} \right]^{\frac{1}{2}} \qquad (2.26)$$

where ν is the fluid kinematic viscosity and r_1 and r_2 are *independent* normally distributed random numbers each with zero mean and standard deviation of unity. For convenience, however, the computer model is based upon the length of the random walks. Hence, equation (2.26) is recast as

$$\Delta x_r = u_r \Delta t_m = r_1 \, (2\nu \Delta t_m)^{\frac{1}{2}}$$

and

$$\Delta y_r = v_r \Delta t_m = r_2 \, (2\nu \Delta t_m)^{\frac{1}{2}} \qquad (2.27)$$

where Δx_r and Δy_r are the random walk *distances* in the x- and y-directions.

2.3 Optimisation of empirical flow parameters

Section 2.2 describes a Biot-Savart discrete vortex model for simulating two-dimensional jet-forced flows in flat-bottomed circular reservoirs. Although the governing equations originate from well defined physical principles, the computer model requires the user to select appropriate values for a number of important parameters. The success of the simulation depends crucially upon three variables; the exact position of introduction of the nascent vortices, the time increment, Δt_v, between successive additions of new vortices, and the time step, Δt_m, used in the Lagrangian tracking procedure. Furthermore, the 'smoothness' of the velocity vector diagrams depicting the reservoir flow are influenced by the use of either Rankine or potential vortices. Selection of the various empirical parameters is accomplished by conducting extensive numerical trials.

Borthwick et al. (1988) have previously outlined the method employed to select the positions of vortex introduction. Figure 2.5 depicts the velocity profile across the inlet of the reservoir as determined by the numerical model and shows that the most obvious defect in the potential flow solution is the complete absence of a boundary layer. Instead, potential flow theory predicts a velocity profile which tends

asymptotically to infinity either side of the inlet. Consequently, the transport of nascent vortices into the interior of the cylinder is significantly affected by the high velocities at the edges of the jet. The closer new vortices are introduced to the inlet corners, the more rapidly they enter the reservoir. This forces the numerical scheme to adopt a smaller time step in the Lagrangian tracking procedure and so increases the computational costs of the simulation. In the present study, the vortex introduction positions are located one-hundredth of the distance across the inlet.

The time interval, Δt_v, between successive additions of new vortices at the inlet separation points is the primary factor which governs the overall run time of the model. In order that the simulation can employ a large number of vortices, Δt_v is chosen to be as *small* as possible whilst maintaining a sensible overall run time. The numerical experiments to discover the optimum value for Δt_v are coupled with those used to find a suitable 'vortex movement' time increment, Δt_m. If large values of Δt_m are employed, vortices approaching the outlet often attempt to cross the solid perimeter wall; this demonstrates that vortices are not being tracked along the correct streamlines and indicates that a reduction in Δt_m is necessary. After many trial runs, the non-dimensional time increments, Δt_v and Δt_m were chosen to be 0.1 and 0.002 respectively. These values produced acceptable flow simulations and allowed the computer model to be advanced to the non-dimensional time level of t=10.0 in approximately 240 minutes of C.P.U. time on a PRIME 9955 computer (this corresponds to the length of a night-time background job).

The selection of the size of core for the Rankine vortices is a more arbitrary decision. Chaplin (1973) studied the flow past a circular cylinder of unit diameter and chose to use Rankine vortices having a core radius, r_0 equal to 0.1. In the present numerical study, however, the discrete vortex method is applied to the confined flow *within* a circular cylinder. Since the inlet width of the reservoir modelled in Section 2.4 is equal to $\pi/16$ (= 0.196), it was judged that the core radius should be smaller than that employed by Chaplin. Trial and error eventually led to the adoption of a core radius, r_0, equal to 0.05. This proved large enough to remove the excessive velocities close to the vortex centres (allowing the velocity vectors depicting the reservoir circulation to have a fair degree of smoothness) whilst still enabling the vortex positions at the early time stages of the calculation to resemble the potential vortex simulation.

2.4 Results

Borthwick et al. (1988) have already presented results from a Biot-Savart

discrete vortex model for simulating flows in circular tanks and reservoirs. They considered two separate reservoir geometries; the first case consisted of an inlet $\pi/2$ radians anticlockwise from the outlet, whereas the second geometry represented a reservoir having an inlet diametrically opposite the outflow opening. Their numerical model utilised potential vortices and only considered pure advection in the Lagrangian vortex tracking procedure. The computer model described herein extends Borthwick et al.'s investigation to include Rankine vortices and a random walk mechanism (to simulate diffusive processes). Test cases corresponding to the geometries studied by Borthwick et al. were considered initially, in order to validate the numerical algorithms used in the computer code. However, for presentational purposes, the results utilise the asymmetrical circular reservoir geometry investigated by Mills (1977). The inlet and outlet openings subtend an angle of $\pi/16$ radians and their centrelines are separated by $7\pi/8$. Details of the reservoir geometry are given in Table 2.1, together with a summary of the values of the semi-empirical flow parameters discussed in Section 2.3.

Figure 2.6 illustrates the irrotational flow solution determined from the analytical expression given by equation (2.17); the velocity vectors are scaled so that a length equal to the radius of the cylinder corresponds to a magnitude of 5.0. As a check, the irrotational flow solution was also determined numerically from the finite-difference Navier-Stokes discretisation presented in Chapter 3. The analytical and numerical solutions were identical, confirming the validity of the two-stage conformal mapping procedure.

Figures 2.7 and 2.8 depict the sequence of vortex positions and velocity distributions * obtained from a 'pure advection' potential vortex simulation (i.e., without the random walk mechanism). During the early stages of the calculation, vortices enter the reservoir in a symmetrical manner and roll up into two distinct eddies of opposite senses of rotation. By t=5.0, part of the eddy formed by the upper shear layer begins to exit the reservoir. The removal of vortices with positive circulation causes the net vorticity in the reservoir to become negative and accounts

* In order to clarify the velocity fields associated with the discrete vortex simulation, the lengths of the vectors in the velocity diagrams are reduced from that implemented in Figure 2.6. Thus the vectors in Figures 2.8, 2.10 and 2.12 are scaled so that an arrow length equal to the radius of the reservoir corresponds to a magnitude of 10.0; this allows the larger velocities in the throughflow jet to be clearly presented.

for the dominance of the clockwise rotating lower gyre structure (Figure 2.8). Between t=5.0 and t=10.0 the inflow jet realigns itself, so that most of the throughflow passes directly between inlet and outlet. In the later stages of the simulation, the jet begins to waver as the discrete vortices simulating the shear layers either side of the throughflow assemble into distinct clusters. The strength of the lower eddy increases throughout the calculation as vortices of clockwise rotation are recirculated in the lower gyre. By t=10.0 the main region of recirculation has moved back towards the inlet and starts to affect the direction of the inflow jet. Continuation of the simulation beyond t=10.0 results in a chaotic unrealistic motion overwhelming the numerical scheme; velocities in the lower eddy increase to unrealistically high values whilst the upper eddy practically disappears.

Figures 2.9 and 2.10 illustrate the results from a simulation employing Rankine vortices with a core radius, r_0, equal to 0.05. As for the previous case, the computer model tracks the vortices using a pure advective Lagrangian scheme. It can be seen that the vortex positions during the early stages of the flow are almost identical to the potential vortex positions shown in Figure 2.7. This indicates that the Rankine velocity profiles do not radically alter the discretisation of the vorticity field. Figure 2.10 illustrates the smoothing effect of the Rankine cores by demonstrating that the velocity vectors depicting the flow are much less susceptible to 'distortion' in the vicinity of the vortex centres. Moreover, the use of Rankine velocity profiles reduced the tendency for vortices to be advected across the solid perimeter walls of the flow domain. This can be attributed to the elimination of the large mutually induced velocities which are created when two potential vortices move close to each other. The Rankine vortex model predicted a slightly weaker recirculation zone in the lower portion of the reservoir. However, considering the large core size in relation to the dimensions of the reservoir, the potential and Rankine vortex simulations depicted in Figures 2.7 to 2.10 are remarkably consistent.

The final set of results utilise Rankine vortices (r_0=0.05) in conjunction with the random walk technique presented in Section 2.2.5. Figures 2.11 and 2.12 show the development of the discrete vortex simulation for an inlet Reynolds number * of 100. Figure 2.11 illustrates that the random walk algorithm has a dramatic effect on vortex positions; the symmetrical entry of vortices into the reservoir is completely destroyed and the characteristic 'roll-up' of vortices into two distinct eddies is less well defined. Diffusion of the shear layer vorticity manifests itself as a transverse spreading of the vortex positions. Although the individual vortices behave chaotically, the velocity vector plots (Figure 2.12) are notable for their smooth variation in time. In fact, the random walk mechanism improves the simulation by reducing the

circulation of the lower gyre whilst increasing the strength of the upper recirculation zone. Apart from the wavering motion of the main throughflow, the velocity vectors at t=10.0 are judged to be in qualitative agreement with the finite-difference solution of the Navier-Stokes equations, presented in Figure 3.8 of the next chapter. Beyond t=10.0, the inflow jet becomes increasingly distorted by the growth of the lower gyre. This leads to a totally unrealistic reservoir circulation pattern as the throughflow is forced towards the upper boundary of the reservoir.

2.5 Conclusions and recommendations

This chapter has described a discrete vortex model for qualitatively assessing jet-forced circulation patterns in circular tanks and reservoirs. The computational procedures utilised in the study are based upon the numerical model proposed by Borthwick et al. (1988). First, a two stage conformal mapping is employed to transform the circular flow domain onto a rectangle; this enables the irrotational flow within the circular reservoir to be determined via an analytical technique. In the second stage of the model, discrete vortices are superimposed on the irrotational flow solution in order to model the shear layers which are formed either side of the inflow jet. The velocities of the individual vortices are calculated using the 'direct summation' (Biot-Savart) method, and consequently the computer time required to update the vorticity field increases rapidly as new vortices are added at the inlet. By restricting the maximum number of vortices to approximately 200, it was found that just under four hours of C.P.U. time were required on a PRIME 9955 computer to run the flow simulation to the non-dimensionalised time, t=10.0. The use of only 200 vortices meant that the discretisation of the vorticity field was rather

* The inlet Reynolds number, Re_I is defined in an identical manner to Mills (1977) i.e.,

$$Re_I = \frac{U_I \epsilon R_0}{\nu} \qquad (2.28)$$

where U_I is the mean inlet velocity, ϵ is half the angle subtended by the inlet, R_0 is the radius of the reservoir and ν is the fluid kinematic viscosity. In the present non-dimensionalised problem, equation (2.28) may thus be simplified to

$$Re_I = \frac{\pi}{32\nu} \quad . \qquad (2.29)$$

coarse. Nevertheless, the results demonstrate that the model is capable of qualitatively predicting many of the important features of momentum driven reservoir flow.

The most satisfactory flow predictions are obtained when the Lagrangian vortex tracking algorithm implements the random walk technique, described in Section 2.2.5. The random walk mechanism effectively calms the velocity field and reduces the distortions of the throughflow jet. However, in a similar manner to other discrete vortex simulations (see Sarpkaya, 1989), the present numerical scheme fails to reach steady-state or limit cycle conditions, and instead produces an increasingly chaotic flow solution. For this particular model, the onset of the chaotic motion is characterised by the unrealistically high velocities in the lower recirculation zone. The growth in the circulation of the lower eddy may be attributed partly to the fact that the numerical scheme does not consider the generation of vorticity along the perimeter walls of the reservoir. Once the vortices simulating the lower inlet shear layer have been 'captured' in the recirculation zone, there is little opportunity for them to be transported out of the flow domain through the outlet. It is recommended that future investigations should study the possibility of adding nascent vortices along the perimeter walls of the reservoir in order to simulate the boundary shear layers. However, the large number of additional vortices which would be required effectively eliminates the Biot-Savart approach as a plausible solution technique and, in hindsight, the vortex-in-cell method may be more suitable because of the less severe computational costs in increasing the number of vortices. Moreover, the V.I.C. technique is less restricted by the shape of the domain since it does not rely upon an analytical determination of the irrotational flow solution. It is likely that the vortex-in-cell approach could be extended to utilise Thompson et al.'s (1974) boundary-fitted grid methodology; this would enable the technique to model flows in irregularly shaped confined areas. However, the empiricism in the discrete vortex method and the failure of the present scheme to produce quantitative flow solutions led to the decision to adopt Eulerian based finite-difference models for the remaining numerical studies.

Table 2.1 Reservoir geometry and empirical flow parameters: Discrete vortex simulation

Inlet coordinates	(-0.9952,-0.0980) and (-0.9952,+0.0980)
Outlet coordinates	(+0.8819,+0.4714) and (+0.9569,+0.2903)
Position of vortex introduction	(-0.9952,-0.0961) and (-0.9952,+0.0961)
Strength of vortices	± 0.05
Time increment between vortex movement, Δt_m	0.002
Time increment between addition of nascent vortices, Δt_v	0.10
Rankine vortex core size, r_0	0.05

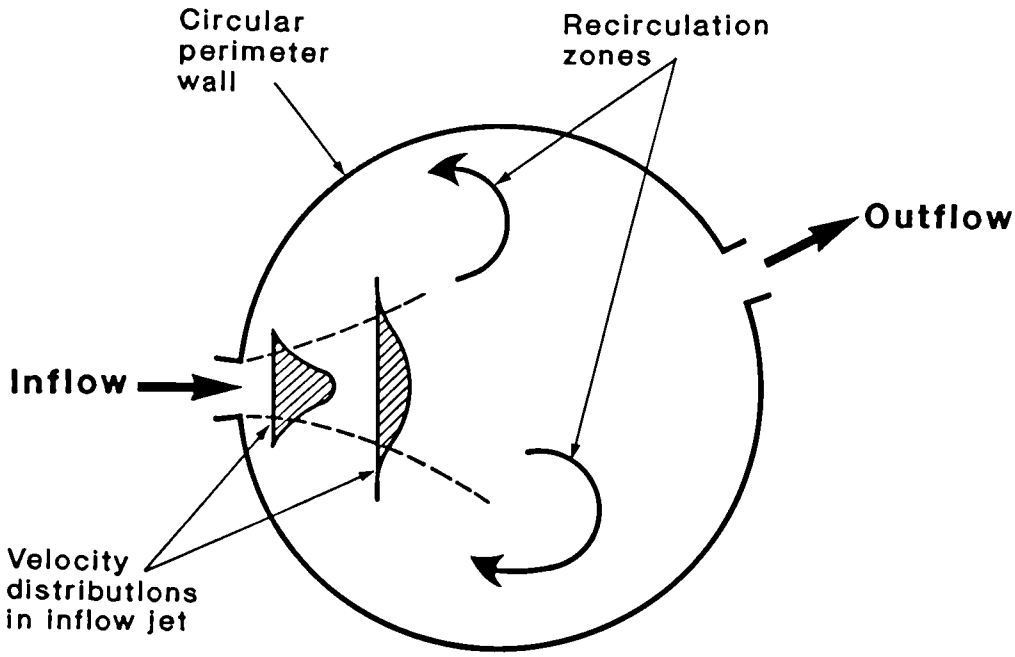


Figure 2.1 Problem formulation

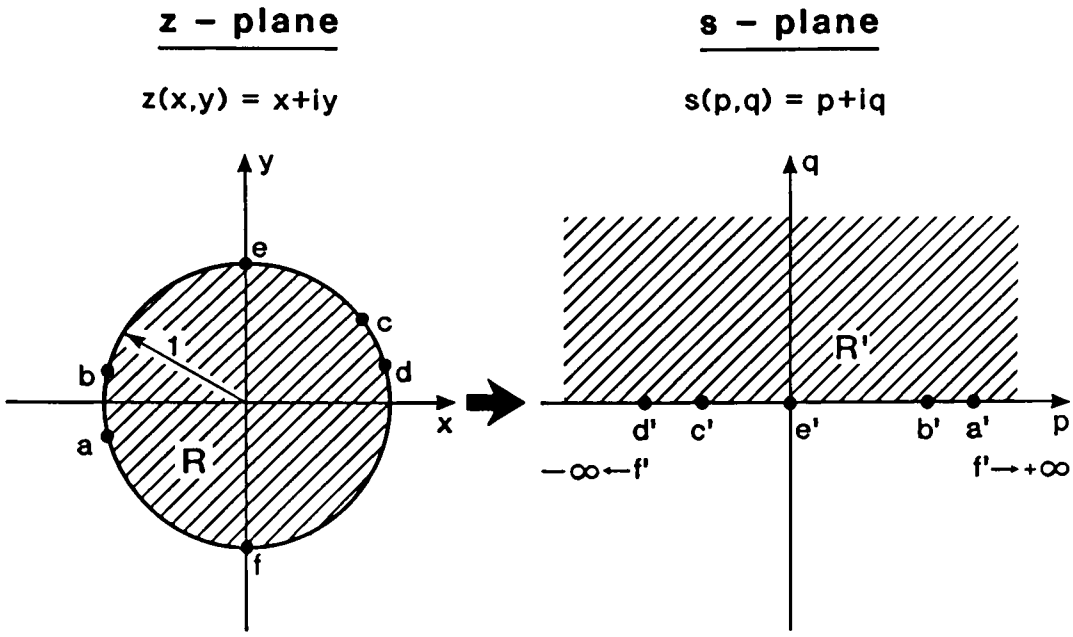


Figure 2.2 Conformal mapping: Step 1: z-plane to s-plane

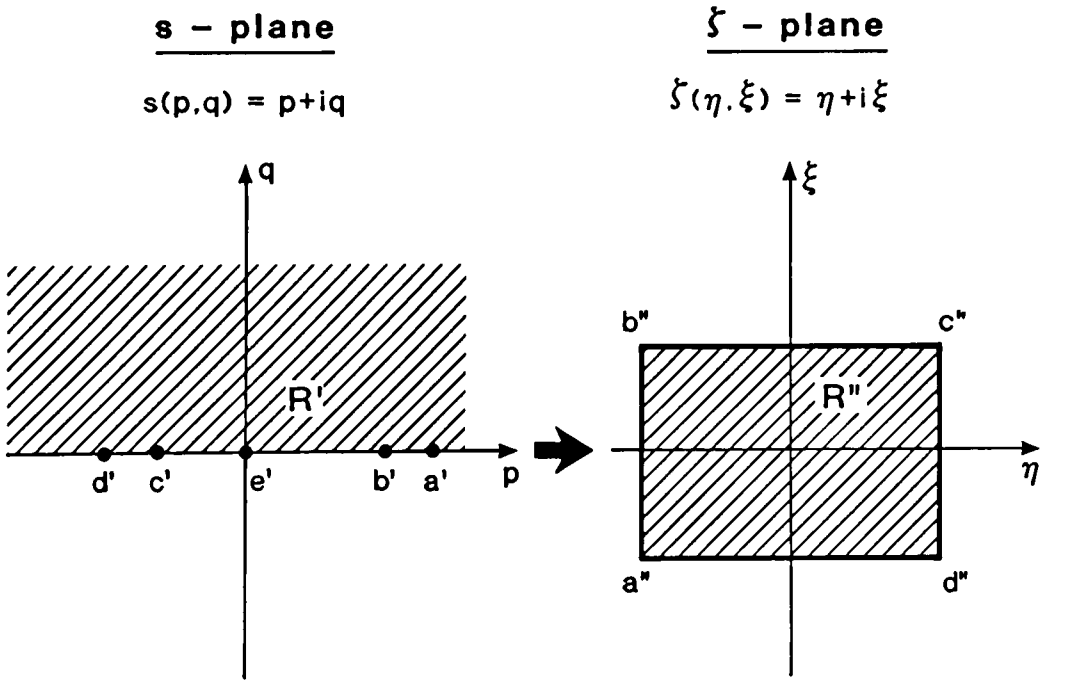


Figure 2.3 Conformal mapping: Step 2: s-plane to ζ-plane

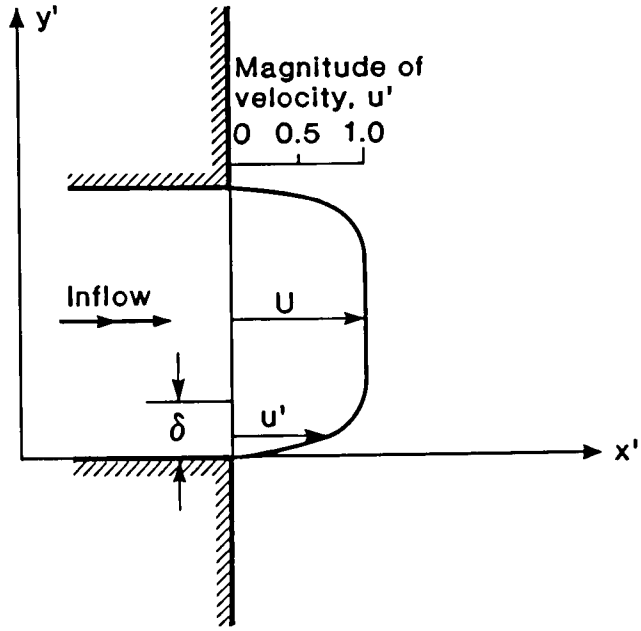


Figure 2.4 Assumed two-dimensional velocity profile at inlet (after Borthwick et al., 1988)

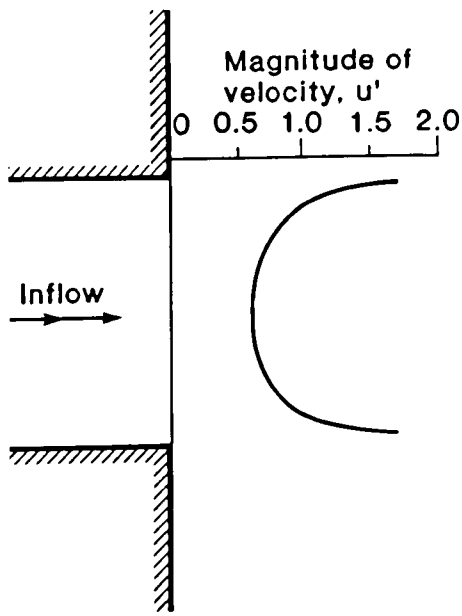


Figure 2.5 Computed velocity profile at inlet

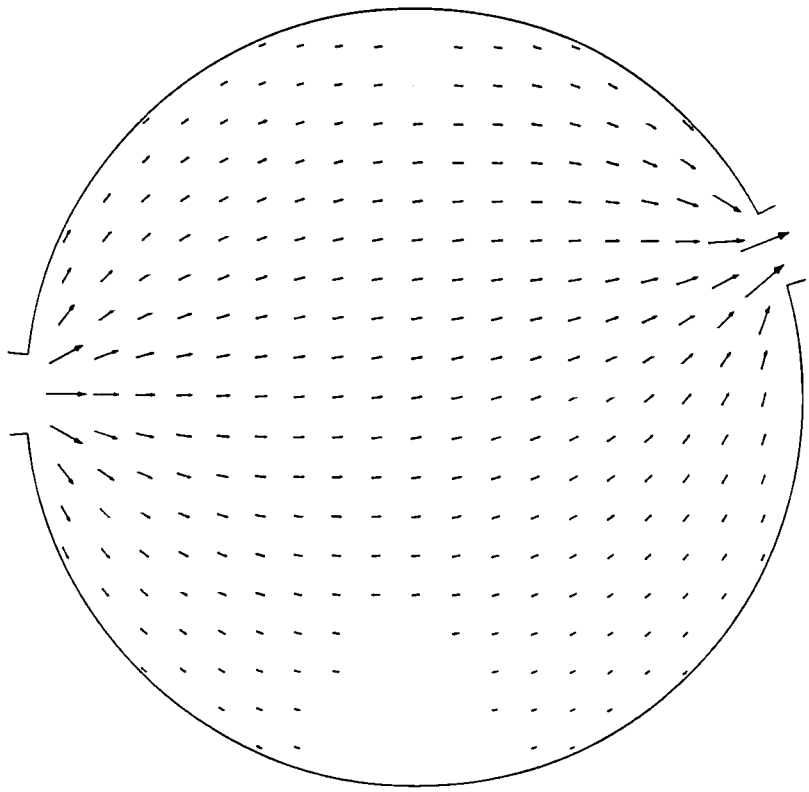


Figure 2.6 Velocity distribution for irrotational flow

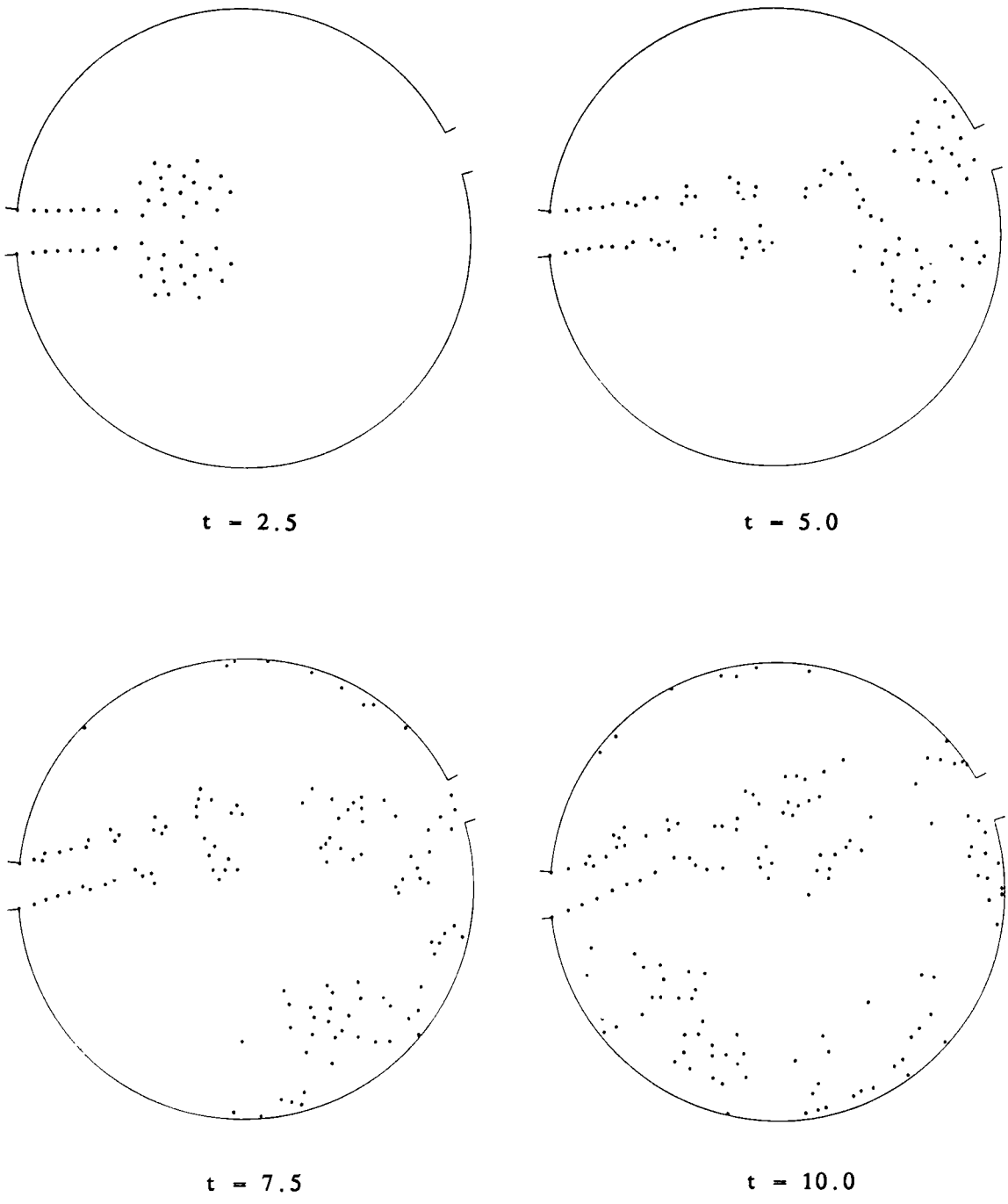


Figure 2.7 Vortex positions: potential vortex simulation

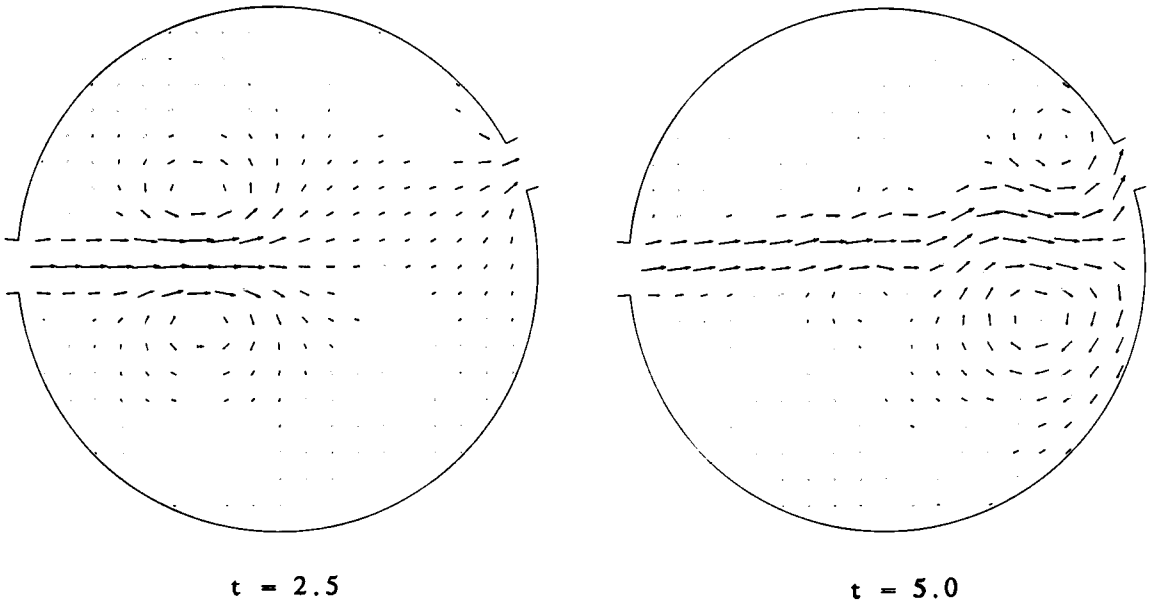


Figure 2.8 Velocity distributions: potential vortex simulation

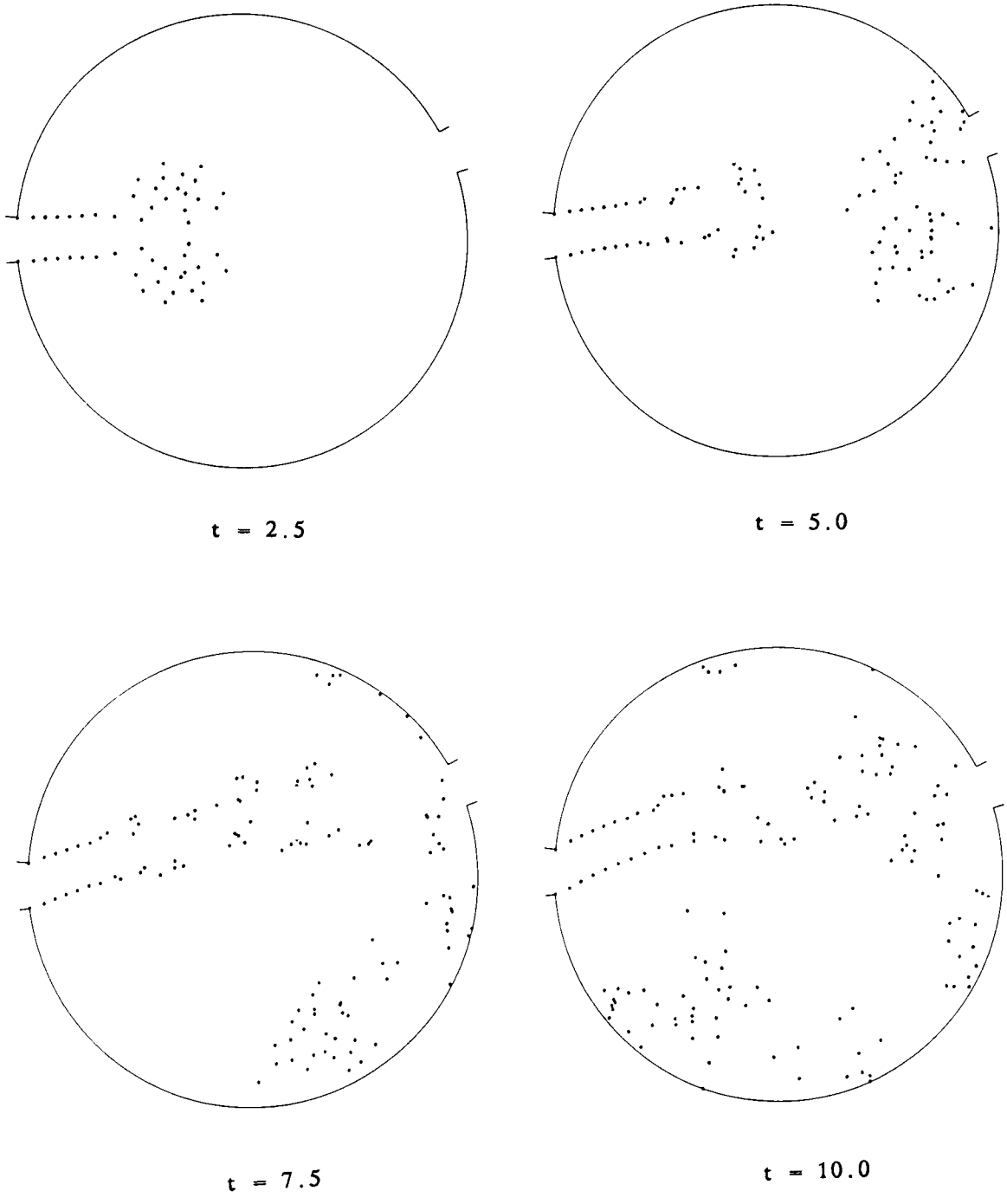


Figure 2.9 Vortex positions: Rankine vortex simulation (r_0=0.05)

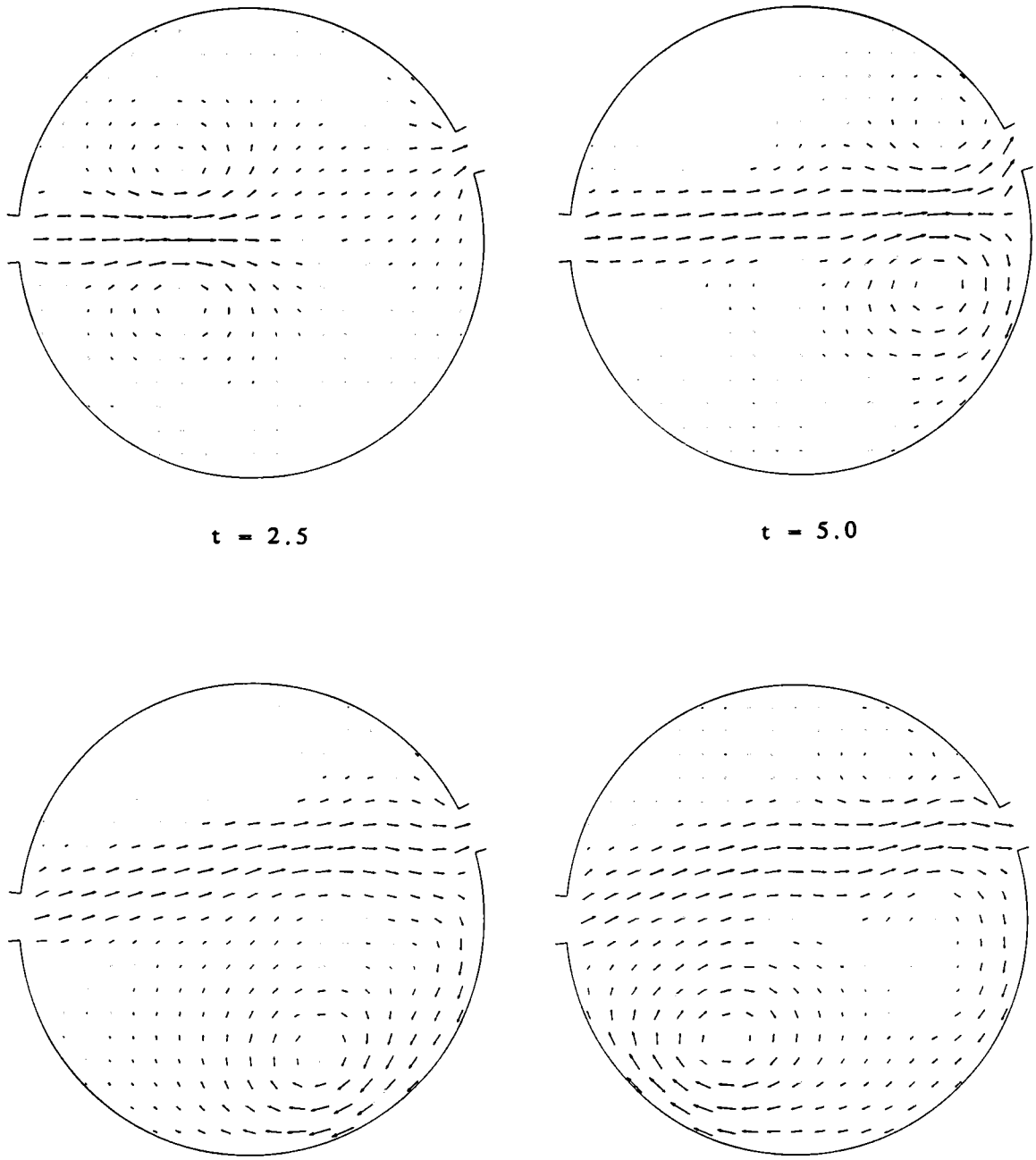


Figure 2.10 Velocity distributions: Rankine vortex simulation (r_0=0.05)

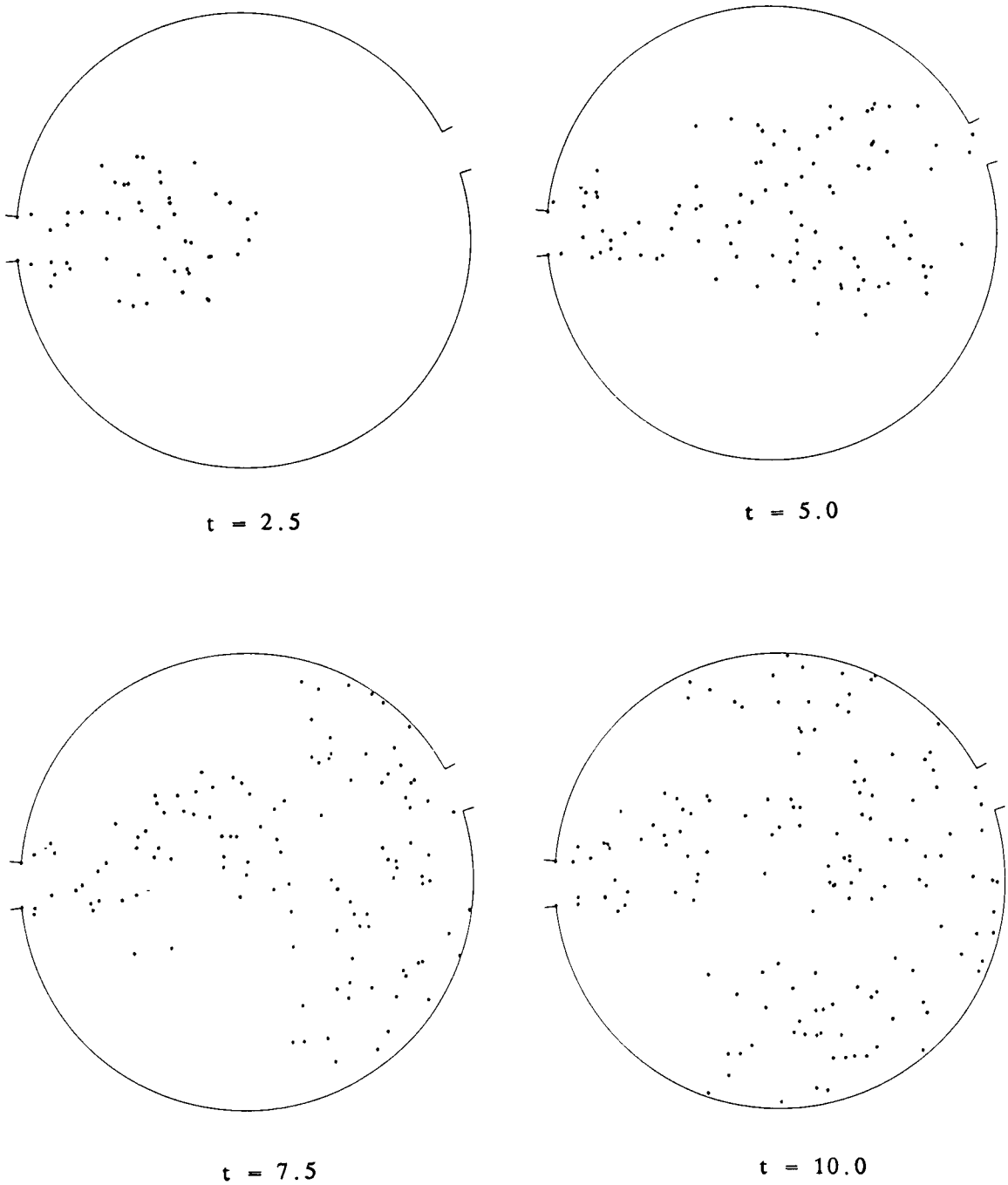


Figure 2.11 Vortex positions: Rankine vortex simulation with random walk (r_0=0.05; Re_I=100.0)

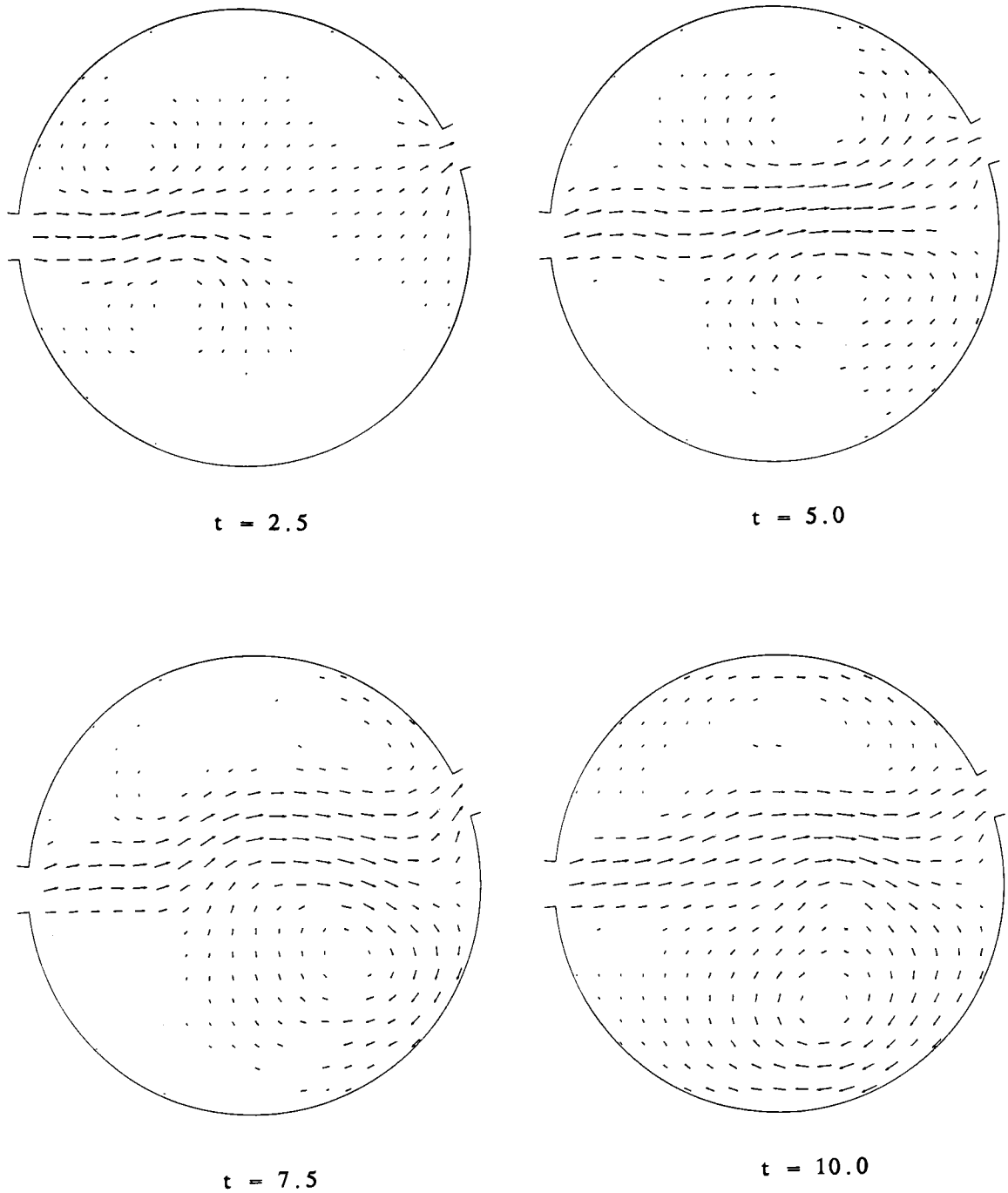


Figure 2.12 Velocity distributions: Rankine vortex simulation with random walk (r_0=0.05; Re_I=100.0)

CHAPTER 3
NAVIER-STOKES SIMULATIONS

3.1 Introduction

Chapter 2 has presented a semi-empirical reservoir flow model which uses discrete vortices to simulate the transport of vorticity originating from the boundary layers of the inflow jet. The model gave reasonable qualitative agreement with experimental and alternative numerical studies (e.g., Dennis, 1974; Falconer, 1976 and 1980; Mills, 1977; Ali and Pateman, 1981; Ali, 1983). However, the inability to reach equilibrium, the empiricism in the choice of position for the introduction of the vortices, the incorrect velocity distributions at the inlet/outlet openings and the difficulties in extending the technique to geometries other than a circle, indicate that the method (presented in Chapter 2) is not yet sufficiently refined to be of practical use in reservoir flow studies. As a consequence, Eulerian (grid based) numerical approaches were judged to be more appropriate.

This chapter describes a two-dimensional finite-difference numerical model of the Navier-Stokes equations for low inlet Reynolds number flows in a circular reservoir with a single inlet and a single outlet (representative of a typical flow problem involving a rapid throughflow stream bounded by adjacent slowly moving recirculation zones). For the purposes of the present study, the investigation ignores free surface and bed friction effects, and concentrates on planar flow in an infinitely deep reservoir. The fundamental flow equations are rewritten in stream function/vorticity-transport form and are applied to a distorted finite-difference mesh in a polar coordinate reference frame; the grid system is transformed using an algebraic mapping, in order to increase the resolution of the mesh near to the no-slip boundaries of the circular perimeter. Although the chapter concentrates on the development of very low Reynolds number laminar flows via solutions of the Navier-Stokes equations, the results are useful in that, until the computer simulation breaks down, they deal with exact flow behaviour without recourse to empiricism.

3.1.1 Stream function/vorticity-transport simulations of jet-forced circular reservoir flows.

Steady-state stream function/vorticity-transport reservoir circulation models were developed by Rayleigh (1893), Dennis (1974) and Mills (1977). These are necessarily restricted to very low speed impulsively-started flows where all quantities settle to

constant values independent of time. In the earliest study, Rayleigh (1893) obtained a Fourier solution of the biharmonic stream function equation at zero Reynolds number (Stokes flow) for a case where the inlet and outlet reduce to a point source and point sink at opposite sides of the circle. Although Rayleigh observed that no recirculation occurred for Stokes flow, he surmised that eddies should form at non-zero Reynolds numbers due to the non-linear terms in the Navier-Stokes equations.

Almost a Century later, Dennis (1974) considered the symmetrical case of a circular container with inlet and outlet directly opposite each other. Dennis expressed the stream function and steady-state vorticity-transport equations in non-dimensional polar form as a pair of Poisson equations coupled through a vorticity parameter. He substituted truncated Fourier series for the stream function and vorticity parameters into the Poisson equations; these were then solved in finite-difference form using Gauss-Seidel iteration to give the Fourier coefficients for constant values of stream function on the side walls and axis of symmetry. Dennis allowed the stream function to vary linearly across the inlet and outlet, thus assuming constant radial inflow and outflow velocities. Furthermore, he assumed that the tangential velocity components were zero at the openings. This is not strictly correct since the boundary layers which form along the inlet and outlet passages cause vena contracta effects on the flow into and out of the container. Dennis presented results for inlet Reynolds numbers, Re_I * of 0, 2, 5 and 10 with the inlet breadth b_I equal to $\pi Ro/15$. He discovered that zones of recirculation appeared as Re_I approached 5 with flow being more or less direct between inlet and outlet at Re_I equal to 10.

In an extension of Rayleigh's approach to Stokes flow in the circle, Mills (1977) combined the stream function and vorticity-transport equations into a single biharmonic equation $\nabla^4\psi = 0$ at zero Reynolds number, where ψ is the stream function. Mills obtained solutions for single and dual inflow-outflow problems by Fourier methods. At non-zero Reynolds numbers (Re_I = 2.5, 5 and 7.5), he computed iterative integral solutions based on the Green's function applied to a circular region. In each case, the inlet and outlet breadths were equal to $\pi Ro/16$ with their centres separated $7\pi Ro/8$ apart circumferentially. Mills observed that an

* The inlet Reynolds number, Re_I is defined as $Re_I = U_I \epsilon R_0/\nu$ where U_I is the mean inlet velocity, ϵ is half the angle subtended by the inlet, R_0 is the radius of the cylinder and ν is the fluid kinematic viscosity.

eddy first appeared on the side of the inflow–outflow stream containing the largest amount of fluid as the inlet Reynolds number approached 2.5. A second eddy formed at $Re_I \approx 2.6$ on the other side of the inflow–outflow stream. At higher Re_I the two eddies grew in size and strength. Mills noted that his solution technique became unstable for $Re_I > 10$ approximately, indicating that unsteadiness may have started to occur.

3.1.2 Directional–difference–explicit (D.D.E.) versus alternating–direction–implicit (A.D.I.) methods

Much of the impetus to develop stream function/vorticity–transport solutions of the Navier–Stokes equations originated from the study of flow past circular cylinders. Early finite–difference models had to be performed by hand (e.g., Thom, 1933; Kawaguti, 1953), but the introduction of computers in the early 1950's led to a rapid growth in the use of numerical models. Many of the early computational schemes employed explicit discretisations of the governing hydrodynamic equations. For example, Thoman (1966) and Thoman and Szewczyk (1969) devised a first order upwind directional difference algorithm for the approximation of the non–linear advective terms in the vorticity–transport equation. This provided their numerical scheme with extra stability and they claimed that flows at Reynolds numbers as high as 10^6 could be modelled.

Other researchers chose to develop numerical schemes based upon the alternating–direction–implicit method proposed by Peaceman and Rachford (1955) and Douglas (1955); the work of Paris and Whitaker (1965), Son and Hanratty (1969), Loc (1980) and Borthwick (1986a, 1986b) is representative of the A.D.I. approach. Unlike the explicit scheme of Thoman and Szewczck, most of the implicit computational methods were unable to model flows above a Reynolds number of about 1000. Lin et al. (1976) studied three different numerical approaches to the solution of the Navier–Stokes equations. First, they considered the D.D.E.–G.S.I. method proposed by Thoman (1966) in which the vorticity–transport equation was updated via a directional–differencing–explicit (D.D.E.) technique whilst the stream function equation was solved by a Gauss–Seidel iteration (G.S.I.) procedure. Secondly, they investigated an A.D.I.–S.O.R. scheme which used an alternating–direction–implicit algorithm to solve the vorticity–transport equation, and a successive–over–relaxation (S.O.R.) scheme for the determination of stream function. Finally, the strongly implicit procedure (proposed by Stone, 1968) was utilised for both the vorticity–transport and stream function equations (S.I.P.–S.I.P.). This latter

method employs a special matrix factorisation and elimination technique to prevent the destruction of the sparseness of the original fully implicit penta-diagonal flow matrix. Lin et al. observed that the A.D.I.-S.O.R. and S.I.P.-S.I.P. methods produced very similar flow solutions whilst the D.D.E.-G.S.I. technique yielded markedly different streamline patterns. A further comparison between explicit and implicit computational procedures for flows past a cylinder was conducted by Borthwick (1986a), who studied a D.D.E.-S.O.R. and an A.D.I.-S.O.R. approach. He showed that first order upwind directional differencing schemes created a significant amount of artificial (numerical) viscosity; this suppressed the vortex-shedding mechanisms entirely and led to incorrect flow predictions.

In view of the above, it was decided that the present numerical investigation should adopt an implicit discretisation technique. Since the finite-difference meshes were to include inlet and outlet stems, the strongly implicit procedure could not readily be utilised because Stone's factorisation (1968) requires a regular, NxM network of grid nodes. Consequently, the more flexible A.D.I.-S.O.R. technique was adopted.

3.2 Governing equations

For the purpose of simulating flow inside a circular reservoir it is convenient to express the governing equations in the polar coordinate frame of reference illustrated in Figure 3.1. The flow is assumed to be incompressible and two-dimensional, thus ignoring free surface and base boundary layer effects. Stratification is also not considered since the reservoir is assumed to have shallow depth relative to its lateral dimensions.

In an Eulerian system, the unsteady flow may be expressed by the conservation of mass (the continuity equation) and conservation of momentum in both the radial, r, and tangential, θ, directions (the Navier-Stokes equations):

Continuity

$$v_r + r\frac{\partial v_r}{\partial r} + \frac{\partial v_\theta}{\partial \theta} = 0 \qquad (3.1)$$

r-direction momentum

$$\rho \left[\frac{\partial V_r}{\partial t} + V_r \frac{\partial V_r}{\partial r} + \frac{V_\theta}{r} \frac{\partial V_r}{\partial \theta} - \frac{V_\theta^2}{r} \right] =$$

$$- \frac{\partial P}{\partial r} + \mu \left[\frac{\partial^2 V_r}{\partial r^2} + \frac{1}{r} \frac{\partial V_r}{\partial r} + \frac{1}{r^2} \frac{\partial^2 V_r}{\partial \theta^2} - \frac{V_r}{r^2} - \frac{2}{r^2} \frac{\partial V_\theta}{\partial \theta} \right] \qquad (3.2)$$

θ-direction momentum

$$\rho \left[\frac{\partial V_\theta}{\partial t} + V_r \frac{\partial V_\theta}{\partial r} + \frac{V_\theta}{r} \frac{\partial V_\theta}{\partial \theta} + \frac{V_r V_\theta}{r} \right] =$$

$$- \frac{1}{r} \frac{\partial P}{\partial \theta} + \mu \left[\frac{\partial^2 V_\theta}{\partial r^2} + \frac{1}{r} \frac{\partial V_\theta}{\partial r} + \frac{1}{r^2} \frac{\partial^2 V_\theta}{\partial \theta^2} + \frac{2}{r^2} \frac{\partial V_r}{\partial \theta} - \frac{V_\theta}{r^2} \right] \qquad (3.3)$$

where

r, θ = polar coordinates,

V_r, V_θ = radial and tangential velocity components,

t = time,

P = pressure,

ρ = fluid density,

and μ = coefficient of dynamic viscosity.

The stream function, ψ, and vorticity, ω, may be defined in terms of the radial and tangential velocity components as follows

Stream function : $$V_r = \frac{1}{r} \frac{\partial \psi}{\partial \theta} \qquad (3.4a)$$

and $$V_\theta = - \frac{\partial \psi}{\partial r} \quad , \qquad (3.4b)$$

Vorticity : $$\omega = \frac{1}{r} \left[\frac{\partial (r V_\theta)}{\partial r} - \frac{\partial V_r}{\partial \theta} \right] \quad . \qquad (3.5)$$

Substituting equations (3.4a) and (3.4b) into the definition of vorticity (equation (3.5)), yields the elliptic stream function equation:

$$\frac{\partial^2 \psi}{\partial r^2} + \frac{1}{r} \frac{\partial \psi}{\partial r} + \frac{1}{r^2} \frac{\partial^2 \psi}{\partial \theta^2} = - \omega \quad . \qquad (3.6)$$

The Navier-Stokes equations are manipulated to eliminate the pressure, P and are simplified using equation (3.5) to obtain the parabolic vorticity-transport expression given by

$$\frac{\partial \omega}{\partial t} + \frac{1}{r}\frac{\partial (V_\theta \omega)}{\partial \theta} + \frac{1}{r}\frac{\partial (rV_r \omega)}{\partial r} = \nu \left[\frac{1}{r}\frac{\partial}{\partial r}\left[r \frac{\partial \omega}{\partial r} \right] + \frac{1}{r^2}\frac{\partial^2 \omega}{\partial \theta^2} \right] \qquad (3.7)$$

where ν is the kinematic fluid viscosity. Equations (3.6) and (3.7) form a system of elliptic-parabolic equations which are solved respectively as boundary and initial value problems.

3.3 Coordinate transformation

The previous section presents the stream function/vorticity-transport equations governing incompressible, viscous, two-dimensional flow. However, it is the generation of vorticity by shearing action at the solid boundaries and the subsequent advection and diffusion of this vorticity to the inner regions of the hydraulic domain which characterise a particular flow. The discretisation of the partial differential hydrodynamic equations shown in Section 3.2 must be performed in conjunction with appropriate boundary conditions which model the effects of the stationary perimeter walls. Thus it is desirable to utilise a finite-difference mesh which is made finer in regions of greater importance, such as at the reservoir walls (where there are large vorticity gradients) and coarser in regions of less importance. Increased spatial resolution by coordinate transformation is preferred to merely altering the mesh spacing since the resulting finite-difference scheme is simpler and discretisation errors are generally smaller.

For these reasons, the radial grid components are transformed in order to concentrate lines near to the outer boundaries of the physical domain; the tangential direction is left unaltered. Figure 3.2 illustrates the mapping process whereby a uniform finite-difference mesh in the computational space is mapped onto a variable grid in the physical domain. The transformation from the computational radial coordinate, α_r to the physical radial coordinate, r is achieved using a polynomial function of the form

$$r = a\alpha_r^4 + b\alpha_r^3 + c\alpha_r^2 + d\alpha_r \qquad (3.8)$$

where a, b, c and d are constants.

The above algebraic mapping procedure is loosely based upon the technique employed by Butler (1978b), Houston and Butler (1979) and Vemulakonda et al. (1985) for solving the primitive variable shallow water equations in stretched rectangular coordinate systems.

Letting μ_r denote the expansion coefficient in the radial direction given by

$$\mu_r = \frac{\partial r}{\partial \alpha_r} = 4a\alpha_r^3 + 3b\alpha_r^2 + 2c\alpha_r + d \qquad (3.9)$$

the vorticity–transport, stream function and tangential velocity equations become

$$\frac{\partial \omega}{\partial t} + \frac{1}{r\mu_r}\frac{\partial (r\omega V_r)}{\partial \alpha_r} + \frac{1}{r}\frac{\partial (\omega V_\theta)}{\partial \theta}$$

$$= \nu \left[\frac{1}{\mu_r^2}\frac{\partial^2 \omega}{\partial \alpha_r^2} + \frac{1}{\mu_r}\left[\frac{1}{r} + \frac{\partial (1/\mu_r)}{\partial \alpha_r} \right]\frac{\partial \omega}{\partial \alpha_r} + \frac{1}{r^2}\frac{\partial^2 \omega}{\partial \theta^2} \right] , \qquad (3.10)$$

$$\frac{1}{\mu_r^2}\frac{\partial^2 \psi}{\partial \alpha_r^2} + \frac{1}{\mu_r}\left[\frac{1}{r} + \frac{\partial (1/\mu_r)}{\partial \alpha_r} \right]\frac{\partial \psi}{\partial \alpha_r} + \frac{1}{r^2}\frac{\partial^2 \psi}{\partial \theta^2} = -\omega , \qquad (3.11)$$

and $$V_\theta = -\frac{1}{\mu_r}\frac{\partial \psi}{\partial \alpha_r} . \qquad (3.12)$$

The expression for radial velocity given in equation (3.4a) remains unchanged.

Although the above vorticity–transport equation was utilised successfully for the lower Reynolds number flow cases, it was found to give slightly distorted vorticity contours near the central point of the reservoir at higher Reynolds numbers. This was found to be attributable to inaccuracies in the finite–difference approximation of the second term in equation (3.10) near the singularity at r = 0. The diffusion term in the vorticity–transport equation becomes less dominant as Re_I is increased and so the effects of errors in the advective term grow more noticeable. Even though the solution could have been improved by refining the computational mesh, it was discovered that a significant increase in spatial accuracy can be achieved by recasting the advective terms as simple derivatives; this process is equivalent to the 'ZIP' differencing method, described by Hirt (1968). The vorticity–transport equation is thus reformulated as:

$$\frac{\partial\omega}{\partial t} + \frac{1}{r\mu_r}\left[\mu_r\omega V_r + rV_r\frac{\partial\omega}{\partial\alpha_r} + r\omega\frac{\partial V_r}{\partial\alpha_r} \right] + \frac{1}{r}\left[\omega\frac{\partial V_\theta}{\partial\theta} + V_\theta\frac{\partial\omega}{\partial\theta} \right]$$

$$= \nu \left[\frac{1}{\mu_r^2}\frac{\partial^2\omega}{\partial\alpha_r^2} + \frac{1}{\mu_r}\left[\frac{1}{r} + \frac{\partial(1/\mu_r)}{\partial\alpha_r} \right]\frac{\partial\omega}{\partial\alpha_r} + \frac{1}{r^2}\frac{\partial^2\omega}{\partial\theta^2} \right] \quad . \qquad (3.13)$$

In the numerical model, equations (3.4a), (3.11), (3.12) and (3.13) are approximated by finite-differences and solved in a time-stepping cyclic procedure taking account of the changing boundary conditions.

3.3.1 Selection of the mapping function

Letting the mapping function given in equation (3.8) be denoted by $r = f(\alpha_r)$ and setting $R_0=1$ for the purposes of the numerical formulation, then, in order for the computational mesh and real mesh to be coincident at the central point and at the circumference of the cylinder (r=1), the following conditions must be satisfied:

$$f(0) = 0 \qquad \text{and} \qquad f(1) = 1 \quad . \qquad (3.14)$$

Moreover, since the mapping function must be able to control the amount of 'stretching' in a particular region, the gradient of the function is also specified as:

$$f'(0) = k_1 \qquad \text{and} \qquad f'(1) = k_2 \quad . \qquad (3.15)$$

One further condition is stipulated; namely, $f''(1) = 0$, so that the r function becomes steeper once through $\alpha_r = 1$. This enables the mapping function to deal with long inlet and outlet passages using relatively few mesh points. The above requirements are fulfilled by equation (3.8) when

$$\left.\begin{aligned} a &= 3 - k_1 - 2k_2 \ , \\ b &= -8 + 3k_1 + 5k_2 \ , \\ c &= 6 - 3k_1 - 3k_2 \ , \\ \text{and} \quad d &= k_1 \quad . \end{aligned}\right\} \qquad (3.16)$$

The choice of the two parameters, k_1 and k_2 is somewhat arbitrary, bearing in

mind the following points:

(a) For the mesh to have increased spatial resolution near r = 1, k_2 should be smaller than 1.

(b) The value of k_1 should be larger than 1, in order to displace the circular mesh lines away from the central region.

(c) The $1/\mu_r$ terms in equations (3.11), (3.12) and (3.13) prohibit the use of $k_2 = 0$.

Selection of $k_1 = 2.0$ and $k_2 = 0.2$ results in a suitable mapping expression given by

$$r = 0.6\alpha_r^4 - 1.0\alpha_r^3 - 0.6\alpha_r^2 + 2.0\alpha_r \quad . \qquad (3.17)$$

Various other polynomial mapping functions with different values of the stretching parameters, k_1 and k_2 were also tested and found to give very similar flow solutions.

3.4 Finite–difference formulations

Before considering the finite–difference formulations which approximate the governing equations, it is necessary to define the notation that will be used to identify the mesh points. Figure 3.3 illustrates the general layout of the computational mesh and the arrangement of the spatial indices, i and j, whilst Figure 3.4 shows a more detailed view of a typical node and its four neighbouring mesh points. Since the flow solution is time–dependent, it is necessary to introduce a further index, n, to indicate the time level. Thus the value of any function, $F(\theta,\alpha_r,t)$ at $\theta = i\Delta\theta$, $\alpha_r = j\Delta\alpha_r$ and $t = n\Delta t$ is denoted as $F^n_{i,j}$.

3.4.1 Vorticity–transport equation

In view of the fact that explicit directional differencing techniques lead to the creation of significant artificial viscosity (Roache, 1972; Borthwick, 1986a), an alternating–direction–implicit (A.D.I.) technique is used to update the vorticity. This procedure advances the vorticity field from time level, n to the new time level, n+1 by the introduction of an intermediate stage at $n+\frac{1}{2}$. In the first half time step [$t=n\Delta t \rightarrow t=(n+\frac{1}{2})\Delta t$], equation (3.13) is 'released' in the θ–direction while holding

the α_r-components fixed. This leads to a finite-difference equation of the form

$$\frac{\omega_{i,j}^{n+\frac{1}{2}} - \omega_{i,j}^{n}}{\Delta t/2} = -\frac{1}{r_j \mu_{r_j}}\left[\mu_{r_j}\omega_{i,j}^{n} V_{r_{i,j}}^{n} + r_j V_{r_{i,j}}^{n}\left[\frac{\omega_{i,j+1}^{n} - \omega_{i,j-1}^{n}}{2\Delta\alpha_r} \right] \right.$$

$$\left. + r_j \omega_{i,j}^{n}\left[\frac{V_{r_{i,j+1}}^{n} - V_{r_{i,j-1}}^{n}}{2\Delta\alpha_r} \right] \right]$$

$$- \frac{1}{r_j}\left[\omega_{i,j}^{n+\frac{1}{2}}\left[\frac{V_{\theta_{i+1,j}}^{n+\frac{1}{2}} - V_{\theta_{i-1,j}}^{n+\frac{1}{2}}}{2\Delta\theta} \right] + V_{\theta_{i,j}}^{n+\frac{1}{2}}\left[\frac{\omega_{i+1,j}^{n+\frac{1}{2}} - \omega_{i-1,j}^{n+\frac{1}{2}}}{2\Delta\theta} \right] \right]$$

$$+ \nu \left[\frac{1}{\mu_{r_j}^2}\left[\frac{\omega_{i,j+1}^{n} - 2\omega_{i,j}^{n} + \omega_{i,j-1}^{n}}{(\Delta\alpha_r)^2} \right] \right.$$

$$+ \frac{1}{\mu_{r_j}}\left[\frac{1}{r_j} + \frac{\partial(1/\mu_r)_j}{\partial\alpha_r} \right]\left[\frac{\omega_{i,j+1}^{n} - \omega_{i,j-1}^{n}}{2\Delta\alpha_r} \right]$$

$$\left. + \frac{1}{r_j^2}\left[\frac{\omega_{i+1,j}^{n+\frac{1}{2}} - 2\omega_{i,j}^{n+\frac{1}{2}} + \omega_{i-1,j}^{n+\frac{1}{2}}}{(\Delta\theta)^2} \right] \right] . \qquad (3.18)$$

The second half time step $[t=(n+\frac{1}{2})\Delta t \rightarrow (n+1)\Delta t]$ is treated in a complementary manner by releasing equation (3.13) in the α_r-direction with the θ-components held at the $n+\frac{1}{2}$ time level:

$$\frac{\omega_{i,j}^{n+1} - \omega_{i,j}^{n+\frac{1}{2}}}{\Delta t/2} = -\frac{1}{r_j \mu_{r_j}}\left[\mu_{r_j}\omega_{i,j}^{n+1} V_{r_{i,j}}^{n+1} + r_j V_{r_{i,j}}^{n+1}\left[\frac{\omega_{i,j+1}^{n+1} - \omega_{i,j-1}^{n+1}}{2\Delta\alpha_r}\right]\right.$$

$$\left. + r_j \omega_{i,j}^{n+1}\left[\frac{V_{r_{i,j+1}}^{n+1} - V_{r_{i,j-1}}^{n+1}}{2\Delta\alpha_r}\right]\right]$$

$$-\frac{1}{r_j}\left[\omega_{i,j}^{n+\frac{1}{2}}\left[\frac{V_{\theta_{i+1,j}}^{n+\frac{1}{2}} - V_{\theta_{i-1,j}}^{n+\frac{1}{2}}}{2\Delta\theta}\right] + V_{\theta_{i,j}}^{n+\frac{1}{2}}\left[\frac{\omega_{i+1,j}^{n+\frac{1}{2}} - \omega_{i-1,j}^{n+\frac{1}{2}}}{2\Delta\theta}\right]\right]$$

$$+ \nu\left[\frac{1}{\mu_{r_j}^2}\left[\frac{\omega_{i,j+1}^{n+1} - 2\omega_{i,j}^{n+1} + \omega_{i,j-1}^{n+1}}{(\Delta\alpha_r)^2}\right]\right.$$

$$+ \frac{1}{\mu_{r_j}}\left[\frac{1}{r_j} + \frac{\partial(1/\mu_r)_j}{\partial\alpha_r}\right]\left[\frac{\omega_{i,j+1}^{n+1} - \omega_{i,j-1}^{n+1}}{2\Delta\alpha_r}\right]$$

$$\left. + \frac{1}{r_j^2}\left[\frac{\omega_{i+1,j}^{n+\frac{1}{2}} - 2\omega_{i,j}^{n+\frac{1}{2}} + \omega_{i-1,j}^{n+\frac{1}{2}}}{(\Delta\theta)^2}\right]\right] \quad . \qquad (3.19)$$

In each half time step the spatial derivatives are approximated using central differences, giving a formal accuracy of $(\Delta\alpha_r{}^2, \Delta\theta^2)$. Furthermore, the A.D.I. method may give the time dependency to (Δt^2) accuracy provided the non-linear advective terms involving $V_r{}^{n+1}$ and $V_\theta{}^{n+\frac{1}{2}}$ are correctly evaluated. In practice, second order time accuracy can be wasteful of computer resources since the only method of accurately assessing $V_r{}^{n+1}$ and $V_\theta{}^{n+\frac{1}{2}}$ is to use some form of iteration process which effectively couples the vorticity-transport and stream function equations back together (e.g., Aziz and Hellums, 1967; Briley, 1971). In the present study, the old values of $V_r{}^n$ and $V_\theta{}^n$ are used instead of the unknown advanced velocity components, leading to a theoretical deterioration in time accuracy. Provided the velocity components change relatively slowly with time, however, the effects of this reduction in accuracy will be insignificant. Indeed, for flows which tend to steady state, the current approach will lead to the same solution as the more exact, second-order time procedures proposed by Aziz and Hellums (1967) or Briley (1971).

Considering the first directional sweep in more detail, equation (3.18) may be applied at each grid point along a given j-line to give a set of N linear simultaneous equations, where N is the number of radial mesh lines. Using the fact that $\omega_{N,j} = \omega_{0,j}$ and writing these N equations in tri-diagonal matrix form gives

$$\begin{bmatrix} b_{0,j} & -c_{0,j} & & & & \\ -a_{1,j} & b_{1,j} & -c_{1,j} & & & \\ & \ddots & \ddots & \ddots & & \\ & & -a_{i,j} & b_{i,j} & -c_{i,j} & \\ & & & \ddots & \ddots & \ddots \\ & & & & -a_{N-1,j} & b_{N-1,j} \end{bmatrix} \begin{Bmatrix} \omega_{0,j}^{n+\frac{1}{2}} \\ \omega_{1,j}^{n+\frac{1}{2}} \\ \vdots \\ \omega_{i,j}^{n+\frac{1}{2}} \\ \vdots \\ \omega_{N-1,j}^{n+\frac{1}{2}} \end{Bmatrix} = \begin{Bmatrix} K_{0,j} \\ K_{1,j} \\ \vdots \\ K_{i,j} \\ \vdots \\ K_{N-1,j} \end{Bmatrix} \tag{3.20}$$

where $K_{0,j} = k_{0,j} + a_{0,j} \cdot \omega_{N-1,j}^{n+\frac{1}{2}}$,

$K_{i,j} = k_{i,j}$ for $1 \leq i \leq N-2$,

and $K_{N-1,j} = k_{N-1,j} + c_{N-1,j} \cdot \omega_{0,j}^{n+\frac{1}{2}}$.

The coefficients $a_{i,j}$, $b_{i,j}$, $c_{i,j}$ and $k_{i,j}$ in equation (3.20) are given by

$$a_{i,j} = \frac{V_{\theta_{i,j}}^{n+\frac{1}{2}}}{2r_j\Delta\theta} + \frac{\nu}{r_j^2\Delta\theta^2} ,$$

$$b_{i,j} = \frac{2}{\Delta t} + \frac{V_{\theta_{i+1,j}}^{n+\frac{1}{2}} - V_{\theta_{i-1,j}}^{n+\frac{1}{2}}}{2r_j\Delta\theta} + \frac{2\nu}{r_j^2\Delta\theta^2} ,$$

$$c_{i,j} = -\frac{V_{\theta_{i,j}}^{n+\frac{1}{2}}}{2r_j\Delta\theta} + \frac{\nu}{r_j^2\Delta\theta^2}$$

and

$$k_{i,j} = \frac{2\omega_{i,j}^n}{\Delta t} - \frac{1}{r_j\mu_{r_j}}\left[\mu_{r_j}\omega_{i,j}^n V_{r_{i,j}}^n + r_j V_{r_{i,j}}^n \left[\frac{\omega_{i,j+1}^n - \omega_{i,j-1}^n}{2\Delta\alpha_r} \right] \right.$$

$$\left. + r_j\omega_{i,j}^n \left[\frac{V_{r_{i,j+1}}^n - V_{r_{i,j-1}}^n}{2\Delta\alpha_r} \right] \right] + \nu \left[\frac{1}{\mu_{r_j}^2} \left[\frac{\omega_{i,j+1}^n - 2\omega_{i,j}^n + \omega_{i,j-1}^n}{(\Delta\alpha_r)^2} \right] \right.$$

$$\left. + \frac{1}{\mu_{r_j}} \left[\frac{1}{r_j} + \frac{\partial(1/\mu_r)_j}{\partial\alpha_r} \right] \left[\frac{\omega_{i,j+1}^n - \omega_{i,j-1}^n}{2\Delta\alpha_r} \right] \right] . \tag{3.21}$$

The intermediate vorticity values, $\omega_{i,j}^{n+\frac{1}{2}}$ are then obtained by solving equation (3.20) using Gaussian elimination for j = 1,...,M-1. In order to facilitate the solution, the advanced vorticity values, $\omega_{N-1,j}^{n+\frac{1}{2}}$ and $\omega_{0,j}^{n+\frac{1}{2}}$ are replaced with known lagged values $\omega_{N-1,j}^{n}$ and $\omega_{0,j}^{n}$. This does not affect the spatial accuracy of the low Reynolds number flow solutions since the vorticity field changes ever more slowly with time as steady state is approached.

The matrix equations presented above apply to values of j within the circle (i.e., $1 \leq j \leq M-1$). Two additional matrix equations are required for mesh points in the inlet and outlet passages ($j > M$). Considering the inlet stem, the resultant tri-diagonal matrix is

$$\begin{bmatrix} b_{NB+1,j} & -c_{NB+1,j} & & & \\ -a_{NB+2,j} & b_{NB+2,j} & -c_{NB+2,j} & & \\ & \vdots & & & \\ & -a_{i,j} & b_{i,j} & -c_{i,j} & \\ & & & \vdots & \\ & & & -a_{NA-1,j} & b_{NA-1,j} \end{bmatrix} \begin{bmatrix} \omega_{NB+1,j}^{n+\frac{1}{2}} \\ \omega_{NB+2,j}^{n+\frac{1}{2}} \\ \vdots \\ \omega_{i,j}^{n+\frac{1}{2}} \\ \vdots \\ \omega_{NA-1,j}^{n+\frac{1}{2}} \end{bmatrix} = \begin{bmatrix} K_{NB+1,j} \\ K_{NB+2,j} \\ \vdots \\ K_{i,j} \\ \vdots \\ K_{NA-1,j} \end{bmatrix} \tag{3.22}$$

where $K_{NB+1,j} = k_{NB+1,j} + a_{NB+1,j} \cdot \omega_{NB,j}^{n+\frac{1}{2}}$,

$K_{i,j} = k_{i,j}$ for $NB+2 \leq i \leq NA-2$,

and $K_{NA-1,j} = k_{NA-1,j} + c_{NA-1,j} \cdot \omega_{NA,j}^{n+\frac{1}{2}}$.

In order to solve equation (3.22), the advanced boundary vorticities, $\omega_{NB;j}^{n+\frac{1}{2}}$ and $\omega_{NA;j}^{n+\frac{1}{2}}$, in the first and last terms of the co-domain, are replaced with lagged values $\omega_{NB,j}^{n}$ and $\omega_{NA,j}^{n}$, in a similar manner to that already described for the θ = 0 axis. Thus, the first directional release is completed by solving equation (3.22) and an equivalent matrix expression for the outlet passage, for j = M,...,M'-1.

Returning to the α_r-direction release, a rearranged form of equation (3.19) is applied to each grid point along a given radial line, i, yielding

$$\begin{bmatrix} b_{i,1} & -c_{i,1} & & & & \\ -a_{i,2} & b_{i,2} & -c_{i,2} & & & \\ & \ddots & \ddots & \ddots & & \\ & & -a_{i,j} & b_{i,j} & -c_{i,j} & \\ & & & \ddots & \ddots & \ddots \\ & & & & -a_{i,M-1} & b_{i,M-1} \end{bmatrix} \begin{Bmatrix} \omega_{i,1}^{n+1} \\ \omega_{i,2}^{n+1} \\ \vdots \\ \omega_{i,j}^{n+1} \\ \vdots \\ \omega_{i,M-1}^{n+1} \end{Bmatrix} = \begin{Bmatrix} K_{i,1} \\ K_{i,2} \\ \vdots \\ K_{i,j} \\ \vdots \\ K_{i,M-1} \end{Bmatrix}$$

(3.23)

where $K_{i,1} = k_{i,1} + a_{i,1}.\omega_0^{n+1}$,

$K_{i,j} = k_{i,j}$ for $2 \leq j \leq M-2$,

$K_{i,M-1} = k_{i,M-1} + c_{i,M-1}.\omega_{i,M}^{n+1}$

and ω_0 is the vorticity at the central point (the double subscript notation being unnecessary).

In equation (3.23) the coefficients $a_{i,j}$, $b_{i,j}$, $c_{i,j}$ and $k_{i,j}$ are given by

$$a_{i,j} = \frac{V_{r_{i,j}}^{n+1}}{2\Delta\alpha_r\mu_{r_j}} - \nu\left[\frac{1}{2\Delta\alpha_r\mu_{r_j}}\left[\frac{1}{r_j} + \frac{\partial(1/\mu_r)_j}{\partial\alpha_r}\right] - \frac{1}{\Delta\alpha_r{}^2\mu_{r_j}{}^2}\right] ,$$

$$b_{i,j} = \frac{2}{\Delta t} + \frac{V_{r_{i,j}}^{n+1}}{r_j} + \frac{V_{r_{i,j+1}}^{n+1} - V_{r_{i,j-1}}^{n+1}}{2\Delta\alpha_r\mu_{r_j}} + \frac{2\nu}{\Delta\alpha_r{}^2\mu_{r_j}{}^2} ,$$

$$c_{i,j} = -\frac{V_{r_{i,j}}^{n+1}}{2\Delta\alpha_r\mu_{r_j}} + \nu\left[\frac{1}{2\Delta\alpha_r\mu_{r_j}}\left[\frac{1}{r_j} + \frac{\partial(1/\mu_r)_j}{\partial\alpha_r}\right] + \frac{1}{\Delta\alpha_r{}^2\mu_{r_j}{}^2}\right]$$

and

$$k_{i,j} = \frac{2\omega_{i,j}^{n+\frac{1}{2}}}{\Delta t} - \frac{1}{r_j}\left[\omega_{i,j}^{n+\frac{1}{2}}\left[\frac{V_{\theta_{i+1,j}}^{n+\frac{1}{2}} - V_{\theta_{i-1,j}}^{n+\frac{1}{2}}}{2\Delta\theta}\right] + V_{\theta_{i,j}}^{n+\frac{1}{2}}\left[\frac{\omega_{i+1,j}^{n+\frac{1}{2}} - \omega_{i-1,j}^{n+\frac{1}{2}}}{2\Delta\theta}\right]\right] + \frac{\nu}{r_j{}^2}\left[\frac{\omega_{i+1,j}^{n+\frac{1}{2}} - 2\omega_{i,j}^{n+\frac{1}{2}} + \omega_{i-1,j}^{n+\frac{1}{2}}}{(\Delta\theta)^2}\right] .$$

(3.24)

It should be noted that equation (3.23) assumes that the radial line in question does not extend into a passageway. For cases where the radial line does extend into an inlet or outlet stem, equation (3.23) is modified so that it corresponds to M'-1 simultaneous equations instead of M-1. Although it is again necessary to use a lagged boundary vorticity in place of $\omega_{1,M}^{n+1}$, the special treatment of the central point (described later in Section 3.5.3) allows the use of the advanced vorticity, ω_0^{n+1}, in the first term of the co-domain. The second directional release is completed by solving equation (3.23) to obtain the updated vorticity values, $\omega_{1,j}^{n+1}$ for each of the N radial lines in turn.

3.4.2 Stream function equation

Having updated the vorticity to the new time level, n+1, it is then necessary to solve the stream function equation in order to satisfy continuity requirements. After some rearrangement, the central difference form of equation (3.11) may be written

$$\psi_{i,j}^{n+1} = \frac{\omega_{i,j}^{n+1} + A_j\psi_{i,j+1}^{n+1} + B_j\psi_{i,j-1}^{n+1} + C_j[\psi_{i+1,j}^{n+1} + \psi_{i-1,j}^{n+1}]}{D_j} \qquad (3.25)$$

where

$$A_j = \frac{1}{\Delta\alpha_r{}^2\mu_{r_j}{}^2} + \frac{1}{2\Delta\alpha_r\mu_{r_j}}\left[\frac{1}{r_j} + \frac{\partial(1/\mu_r)_j}{\partial\alpha_r}\right] ,$$

$$B_j = \frac{1}{\Delta\alpha_r{}^2\mu_{r_j}{}^2} - \frac{1}{2\Delta\alpha_r\mu_{r_j}}\left[\frac{1}{r_j} + \frac{\partial(1/\mu_r)_j}{\partial\alpha_r}\right] ,$$

$$C_j = \frac{1}{r_j{}^2\Delta\theta^2} ,$$

and $$D_j = \frac{2}{\Delta\alpha_r{}^2\mu_{r_j}{}^2} + \frac{2}{r_j{}^2\Delta\theta^2} .$$

The stream function equation is solved by repeatedly applying equation (3.25) to each interior mesh point until the difference between successive stream function values at each node is less than a predetermined error parameter, ϵ. This iterative procedure is accelerated by successive-over-relaxation (S.O.R.):

$$\psi'_{i,j} = \psi_{i,j} + \Omega \left(\psi^{*}_{i,j} - \psi_{i,j} \right) \qquad (3.26)$$

where $\psi_{i,j}$ is the original stream function value at (i,j),

$\psi^{*}_{i,j}$ is the value obtained as a result of applying equation (3.25),

$\psi'_{i,j}$ is the updated stream function,

and Ω is the over-relaxation parameter.

Numerical experimentation indicates that the optimum value for the S.O.R. parameter is approximately 1.75.

3.4.3 Velocity components

The radial and tangential velocity components are determined from equations (3.4a) and (3.12) rewritten in central differences as

$$V_{r_{i,j}} = \frac{1}{r_j} \left[\frac{\psi_{i+1,j} - \psi_{i-1,j}}{2\Delta\theta} \right] \qquad \text{for} \quad 1 \leq i \leq N-2 \qquad (3.27a)$$

and

$$V_{\theta_{i,j}} = - \frac{1}{\mu_{r_j}} \left[\frac{\psi_{i,j+1} - \psi_{i,j-1}}{2\Delta\alpha_r} \right] . \qquad (3.27b)$$

Two special forms of equation (3.27a) are required to determine the radial velocity components near the $\theta = 0$ axis. Since $\psi_{N,j} = \psi_{0,j}$, then

$$V_{r_{0,j}} = \frac{1}{r_j} \left[\frac{\psi_{1,j} - \psi_{N-1,j}}{2\Delta\theta} \right] \qquad (3.27c)$$

and

$$V_{r_{N-1,j}} = \frac{1}{r_j} \left[\frac{\psi_{0,j} - \psi_{N-2,j}}{2\Delta\theta} \right] . \qquad (3.27d)$$

3.5 Boundary conditions

The previous section has dealt with the finite-difference approximations to the

vorticity–transport and stream function equations for nodes which are in the interior of the mesh. These finite–difference expressions model the advection and diffusion properties of the vorticity and continuity requirements of the flow. As yet, however, no mechanism has been discussed by which vorticity can be generated at walls nor have the boundary conditions of the elliptic stream function equation been described.

3.5.1 Stream function boundary conditions

(a) Walls

For comparison with Mills' computations (1977), the upper and lower walls of the reservoir are given constant ψ–values of 2 and 0 respectively.

(b) Inflow and Outflow Boundaries

In this case, the boundary conditions depend on the required form of the velocity distributions at the inlet and outlet. Dennis (1974) and Mills (1977) both assumed V_r to be constant, implying a linear variation in stream function across each opening. For several of the lower Reynolds number flows this condition is implemented in order to be compatible with the solutions presented by Mills and Dennis.

By extending the finite–difference mesh into stems beyond the circular perimeter of the reservoir, it is possible to implement more realistic boundary conditions which allow the development of boundary layers at the side walls of the inlet and outlet passageways. This is achieved by specifying only the tangential velocity component along the openings into the inflow/outflow passageways, leaving the radial component to develop freely as part of the numerical analysis. Using a boundary condition of the type first proposed by Paris and Whitaker (1965), the tangential velocity condition is

$$V_\theta = - \frac{1}{\mu_r} \frac{\partial \psi}{\partial \alpha_r} = 0 \qquad (3.28)$$

which, in terms of finite–differences, gives

$$\psi_{i,M'} = \psi_{i,M'-1} \qquad (3.29)$$

for $NB+1 \leq i \leq NA-1$ and $ND+1 \leq i \leq NC-1$.

Since the radial velocity component develops as part of the computation, equation (3.29) allows upstream/downstream effects beyond the actual finite-difference mesh to be 'felt' at the flow boundaries.

3.5.2 Vorticity boundary conditions

(a) No-Slip Walls

The accuracy of the flow simulations relies heavily on the vorticity values generated at solid walls. Two types of vorticity boundary condition are employed; a first order simple condition and a second order formulation of the type proposed by Woods (1954). Since the radial direction has undergone an algebraic mapping, the boundary vorticity equations are considered in the computational (α_r, θ) system.

In deriving the simple condition, ψ is expanded as a Taylor series about a boundary node in a direction normal to the wall and into the flow domain. Components related to V_r and V_θ at the wall are set to zero and terms above first order involving mesh spacings ignored. Thus, for the circular boundary, the wall vorticity can be shown to be:

$$\omega_{i,M} = \frac{2(\psi_{i,M} - \psi_{i,M-1})}{\mu_{r_M}^2 \, \Delta\alpha_r^2} + O(\Delta\alpha_r) \quad , \tag{3.30}$$

whereas for the radial wall at (NA,j) which forms one side of the inflow passageway:

$$\omega_{NA,j} = \frac{2(\psi_{NA,j} - \psi_{NA-1,j})}{r_j^2 \, \Delta\theta^2} + O(\Delta\theta) \quad . \tag{3.31}$$

Analogous expressions are obtained for the three other stem walls at (NB,j), (NC,j) and (ND,j).

It can be seen that the previously presented vorticity boundary conditions are only first order accurate. Therefore, to improve the spatial accuracy of the simulation, a second order method due to Woods (1954) is utilised. First the stream function equation is differentiated with respect to the ordinate normal to the wall. Simplification is achieved by eliminating terms directly dependent on V_r and V_θ which are both zero on the boundary. After some rearrangement, in which the boundary vorticity is also approximated by a Taylor series, Woods' vorticity

conditions are derived as

$$\omega_{i,M} = \frac{\left[\frac{6(\psi_{i,M} - \psi_{i,M-1})}{\Delta\alpha_r{}^2 \mu_{r_M}{}^2} - \omega_{i,M-1} \right]}{\left[2 + \Delta\alpha_r \left[3\mu_{r_M} \frac{\partial(1/\mu_r)_M}{\partial\alpha_r} + \frac{\mu_{r_M}}{r_M} \right] \right]} + O(\Delta\alpha_r{}^2) \qquad (3.32)$$

on the circular boundary, and

$$\omega_{NA,j} = \frac{3(\psi_{NA,j} - \psi_{NA-1,j})}{r_j{}^2 \Delta\theta^2} - \frac{\omega_{NA-1,j}}{2} + O(\Delta\theta^2) \qquad (3.33)$$

on the radial wall at (NA,j). Again, very similar expressions to equation (3.33) are obtained for the other radial walls at (NB,j), (NC,j) and (ND,j). Appendix A describes the derivation of the vorticity boundary conditions in more detail.

(b) Corner Points

At the junctions between the inflow/outflow passageway walls and the circular perimeter of the reservoir, special consideration of the nodal vorticity value is required. It is apparent there are two values of vorticity generated at the junction; one due to the circular wall, the other arising from the radial stem. Roache (1972) suggests that the best method of dealing with this difficulty is to store both values and use the most appropriate one depending on the direction of the computational sweep. By permitting a discontinuity in vorticity, the technique is more able to allow for the large velocity and vorticity gradients in the vicinity of the corner points than other methods (such as averaging).

(c) Inflow and Outflow Boundaries

The vorticity conditions at the flow boundaries depend on the prescribed velocity distributions across the inlet and outlet. For the velocity conditions stipulated by Dennis (1974) and Mills (1977), where V_r is a constant and V_θ is zero at the openings into the circle, the boundary vorticity is given by equations (3.30) or (3.32). It is interesting to note that, in this case, the vorticity conditions are the same at the solid walls and flow boundaries.

The second, less restrictive, condition was applied to cases involving inflow and outflow passageways where V_r was allowed to develop as part of the numerical

analysis. This results in the following vorticity condition due to Paris and Whitaker (1965):

$$\frac{\partial \omega}{\partial \alpha_r} = 0 \tag{3.34}$$

which is written in finite-differences as

$$\omega_{i,M'} = \omega_{i,M'-1} \tag{3.35}$$

for NB+1 $\leq$ i $\leq$ NA-1 and ND+1 $\leq$ i $\leq$ NC-1.

The implementation of equations (3.29) and (3.35) assumes that the flow remains confined by the radial walls for some distance beyond the outermost mesh points.

3.5.3 Treatment of central point

Both the vorticity-transport and stream function expressions, as formulated in equations (3.6) and (3.7) (or (3.10) and (3.11)), contain singularities at r = 0. This difficulty is overcome by replacing the polar coordinate forms of these expressions by their Cartesian equivalents at the central point. Making use of the right-angled symmetry of the radial lines in the polar mesh, it is possible to obtain an improved estimate at the origin by effectively rotating the Cartesian coordinate axes and averaging the resultant values. In an extension of the technique proposed by Smith (1985) for a heat conduction problem, the stream function at the central point can be written as (see Appendix B):

$$\psi_0 = \frac{1}{N} \sum_{i=0}^{N-1} \psi_{i,1} + \frac{\omega_0 \, \delta r^2}{4} \tag{3.36}$$

where ω_0 is the vorticity at r = 0,
and δr is the length of the first grid increment in the radial direction of the physical mesh.

The vorticity-transport equation is treated in a similar manner, and the updated central point vorticity obtained from the following explicit forward difference expression

$$\omega_0^{n+1} = \omega_0^n + \overline{\frac{\partial \omega_0}{\partial t}}\,\Delta t \qquad (3.37)$$

where ω_0^n is the previous vorticity at the central point and $\overline{\partial\omega_0/\partial t}$ is the mean value of the temporal vorticity gradient, $(\partial\omega/\partial t)_{i,0}$ obtained by averaging over a quadrant. Appendix B details the derivation of the formulae used to update the vorticity at the central node, and shows

$$\left.\frac{\partial \omega}{\partial t}\right|_{i,0} = -\frac{1}{4\delta r^2}(\psi_{i+N/4,1} - \psi_{i+3N/4,1})(\omega_{i,1} - \omega_{i+N/2,1})$$

$$-\frac{\omega_0}{2\delta r}(v_{r_{i,1}} + v_{r_{i+N/4,1}} + v_{r_{i+N/2,1}} + v_{r_{i+3N/4,1}})$$

$$-\frac{1}{4\delta r^2}(\psi_{i+N/2,1} - \psi_{i,1})(\omega_{i+N/4,1} - \omega_{i+3N/4,1})$$

$$+\frac{\nu}{\delta r^2}(\omega_{i,1} + \omega_{i+N/4,1} + \omega_{i+N/2,1} + \omega_{i+3N/4,1} - 4\omega_0)$$

for $0 \leq i \leq N/4 - 1$. (3.38)

By implementing equation (3.37) before the A.D.I. procedure for the remaining mesh points, the advanced vorticity at the origin can be utilised during the second directional release.

3.6 Initial conditions

The parabolic vorticity-transport and elliptic stream function equations are solved as a combination of initial and boundary value problems. In general, the potential flow solution is used as the starting condition; this is achieved by setting the vorticity to zero throughout the grid and then solving equation (3.25) to give initial values of ψ. It is also acceptable to use the final flow solution from a previous case as the starting condition. This approach was found to reduce the computation time significantly for the higher Reynolds number flows.

3.7 Numerical scheme

Borthwick (1986a, 1986b) has previously outlined the overall computational procedure by which the solution is built up in time steps from initial values by solving the stream function/vorticity-transport equations using a combination of S.O.R. and A.D.I. techniques. In the present study, grids of varying fineness were examined; the optimum chosen as being the one which gave almost identical results to a mesh of half the spacing. Even though the solution of the vorticity-transport equation by the A.D.I. technique is unconditionally stable according to a von Neumann analysis, a time step restriction is required in practice (see Roache, 1972). Here it may be attributed to the lagging of the advanced boundary vorticity values and the advanced velocity components, V_r^{n+1} and $V_\theta^{n+\frac{1}{2}}$. Hence, although the A.D.I. method allows larger time steps than explicit techniques, there is uncertainty over the appropriate value for Δt. Some authors (Roache, 1972; Borthwick, 1986a) recommend the use of a time step restriction applicable to an explicit scheme, in order to guarantee stability, yet this may be rather uneconomical. Therefore, the time steps for this particular model were determined by numerical experimentation and are given in Table 1, together with the sizes of the computational meshes to which the time steps apply.

3.8 Results

In order to validate the numerical model, test cases corresponding to those of Dennis (1974) and Mills (1977) were considered initially. Unlike Dennis and Mills, the solutions were extended to give computer plotted vorticity contours and velocity vector distributions, as well as stream function contours. In all cases, the boundary vorticity values were computed using the 2nd order Woods formulation, equations (3.32) and (3.33). Almost identical results were obtained, however, using the simple boundary vorticity expressions (3.30) and (3.31). In Figures 3.5 to 3.9 the velocity vectors are scaled so that a length equal to the radius of the reservoir corresponds to a magnitude of 50. It should also be noted that the stream function contours for the throughflow stream on Figures 3.7 to 3.9 have been left unlabelled, but have the same values as the corresponding contours in Figure 3.6.

Figure 3.5 contains the steady-state stream function contours, vorticity contours and velocity distribution at Re_I=10 for the symmetrical case of the inlet and outlet diametrically opposite each other. As with all the diagrams presented in this chapter the inflow enters at the left hand side and exits through the right hand outlet. Here,

the stream function contours are directly comparable with the solution obtained by Dennis (1974), using the series truncation method. Allowing for small drafting errors in Dennis' hand drawn diagrams there is almost exact agreement with the throughflow stream function contours. However, the present model predicts marginally lower strengths of recirculation zones in comparison to the series truncation method. Computer plots obtained at lower Re_I numbers indicate that, for $Re_I \leq 2$, the flow resembles the Stokes (zero Re_I) solution. By $Re_I=5$, symmetrical zones of recirculation develop either side of the inflow, close to the inlet. There is a divergent jet at the inlet corresponding to the inflow of vorticity. The situation is different at the outlet which acts as a sink. At $Re_I=10$ the counter rotating gyres occupy approximately two-thirds of the available space in the reservoir, and the throughflow streamlines begin to orientate themselves directly between inlet and outlet. There is, however, still some evidence of jet divergence at the inlet compared with radial flow at the outlet. Interestingly, the gyres in Figure 3.5 are bounded by separation streamlines (which were not included in Dennis' results). Large concentrations of vorticity occur close to the inlet with loops of vorticity extending parallel to the main stream beyond the middle of the reservoir. In effect, these regions of strong vorticity push back the outlet vorticity zones allowing contours of lower vorticity to spread out. This ties in with the velocity distribution where the inflow jet contains the largest velocities, there is some deceleration towards the outlet as the throughflow stream widens and, finally, there is a radial sink-like outflow. Either side of the throughflow, two slow moving eddies are noticeable with stagnation points close to the axis which bisects the circular reservoir normal to the throughflow direction.

For the remainder of this section, the case of a reservoir with asymmetrically placed inlet and outlet ($\alpha=7\pi/8$) will be examined. For $Re_I \leq 2$ the steady-state pattern again resembled Stokes flow. As Re_I increased a gyre first appeared in the lower part of the reservoir, and was shortly followed by a smaller gyre in the upper part. The steady-state situation at $Re_I=7.5$ is shown in Figure 3.6. The stream function contours differ from those presented by Mills (1977), with the present numerical model predicting stronger and larger recirculation zones. Furthermore, close scrutiny of the stream function plots presented by both Dennis and Mills, reveals differences between the strengths of the gyres in their cases which cannot be attributed entirely to the differing geometries; Mills' iterative integral solutions produce less recirculation than Dennis' series truncation method.

Examination of the stream function and vorticity contours in the vicinity of either the inlet or outlet in Figures 3.5 and 3.6, indicates the restrictive nature of

the boundary conditions used by both Dennis and Mills. (Their boundary conditions did not cater for the presence of boundary layer effects in the entrance or exit passageways). The streamlines are uniformly spaced across the inlet and outlet; this causes the contraction of the jet to occur within the reservoir as is discernible in both stream function contour plots. In particular, the vorticity contours become slightly distorted within a radial mesh interval of either the inlet or outlet. As a consequence, for the remaining higher Re_I cases, the computational mesh was extended to include the stems, and flexible flow boundary conditions were applied at the extreme ends.

Figure 3.7 shows the steady-state flow regime at Re_I=25. From the stream function contour plot, it is clear that the inflow stream tube narrows immediately after entering the reservoir forming a kind of vena contracta, then gradually widens as the flow traverses the central region before contracting sharply as it exits the reservoir. The bulbous shape of the streamtube at the exit is related to the formation of a second vena contracta in the outlet passageway. The centres of the two primary recirculation regions are closer to the outlet than at lower Reynolds numbers. This is due to long strands of high vorticity from the inlet walls coinciding with the shear layers which demarcate the rapid throughflow from the slowly rotating recirculation zones. These loops of high vorticity diffuse out towards the exit and so are responsible for the stagnation points in the gyres being located closer to the outlet passageway than at lower Re_I values. (It should be noted that, although the stems formed an integral part of the computational domain for all the results described from now on, they are not included in the diagrams.)

Similar steady-state patterns were obtained for Re_I=50; the inflow continued to narrow, the inflow vena contracta became less pronounced and the strength of the recirculation zones increased. By Re_I=100, an additional pair of extremely slow moving gyres developed either side of the throughflow close to the inlet as illustrated in Figure 3.8. Here the throughflow streamlines are almost parallel as they curve uniformly from inlet to outlet (with slight narrowing at the outlet). Although the positions of the primary eddies are still close to the outlet, their strengths have reduced due to the formation of the pair of counter-rotating secondary eddies. Two additional separation streamlines related to these eddies may be observed in the top and bottom left hand extremities of the reservoir.

Inspection of the vorticity contours reveals that there is a narrow zone of high vorticity gradient aligned between inlet and outlet. This corresponds to the centreline of the throughflow. Large values of vorticity are evident in the shear

layers at the edges of the main stream, with a portion of this vorticity exiting via the outlet while the main bulk is forced around the edges of the primary vortices. The secondary gyres are clearly affected by the 'tongues' of vorticity that have grown from fluid shearing at the borders of the primary gyres with the no-slip walls.

Figure 3.9 illustrates the flow characteristics at time t=33.2 for Re_I=200 immediately before the computational scheme became unstable. In this case, the previous steady-state solution at time t=30 was used for the initial conditions. In complete contrast to Re_I=100 however, the flow exhibited unsteadiness in the throughflow region, between the reservoir centre and the outlet, as the shear layers rolled up into vortices. At first, these vortices were advected towards the outlet in a symmetrical manner (relative to the axis of the throughflow stream) but very slight discrepancies in the positions of the vortex centres soon became accentuated, causing the main stream to waver as may be seen in Figure 3.9. Although these vortices disturbed the primary gyres, they did not, in the short span of the simulation, have any appreciable effect on the secondary gyres. At the instant before the computation process broke down, a vortex may be observed just below the main stream attempting to squeeze into the outlet passageway. As would be expected, no separation occurred in the inflow passageway due to the flow accelerating where the walls converged slightly. However, in the outflow stem the flow decelerates and separation was observed along the upper wall originating from the sharp corner between the circular perimeter and the radial outlet wall. As a consequence, the outflow was rather less uniform than the inflow. Furthermore, the proximity of the vortex approaching the other outlet corner (at i=ND, j=M) caused the lower boundary layer to become unstable. The computational grid simply did not have enough points to cater for the large vorticity gradients in the vicinity of the outlet and so further solutions could not be achieved.

3.9 Conclusions

Jet-forced flow in a circular reservoir with a single inlet and outlet has been examined for flows with low values of inlet Reynolds number. For $Re_I<2$, the solution is similar to Stokes flow in that zones of recirculation do not evolve. For $2<Re_I<50$, two gyres form either side of the throughflow at steady-state. By Re_I=100, steady-state is still achieved, but the number of gyres has increased to four. Unsteadiness in the shear layers causes vortices to form at the edges of the main stream at Re_I=200. At this point the solution broke down.

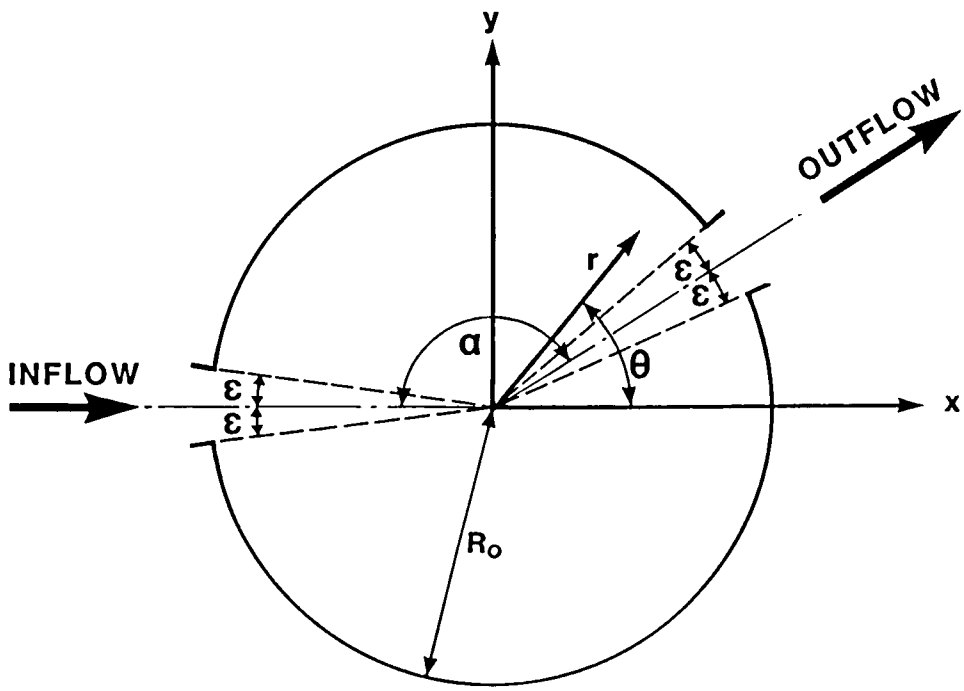
Although no entirely new numerical techniques are proposed in this chapter, the stream function/vorticity–transport model has simulated several interesting features of reservoir flows at low inlet Reynolds numbers. The model has indicated that the development of unsteadiness originates from vortex roll up at the sides of the throughflow. At higher Re_I, the inflow region of the main stream would be expected eventually to form a fully turbulent jet. It is reasonable to suppose that the onset of turbulence would finally result in a continuously wavering high velocity throughflow with single gyres either side composed of the aggregate of a large number of turbulent eddies.

The present numerical scheme is incapable of modelling small–scale turbulent conditions. However, the low Reynolds number flows considered in this chapter should not be viewed as being impractical since the model provides useful information for the development and validation of a more advanced boundary–fitted systems solution. Moreover, the modelling of turbulence generally utilises values of eddy viscosity which are several orders of magnitude greater than the coefficient of kinematic viscosity. The inlet Reynolds number for turbulent conditions (based upon the eddy viscosity coefficient) is therefore often in the same range as the flows studied in this chapter. Thus, although it may seem a paradox, the present low Reynolds number flow regimes will have many features which are similar to high Re_I turbulent reservoir conditions.

The next chapter describes a more advanced grid mapping technique proposed by Thompson et al. (1974 onwards), which enables finite–difference procedures to model complicated hydraulic domains with irregular perimeters. In order to simulate free surface effects (which are absent from the computations detailed in the present study), the boundary–fitted scheme will be used in conjunction with the depth–averaged Reynolds equations.

Table 3.1 Grid sizes and time steps utilised for Navier-Stokes reservoir circulation studies

Re_I	Geometry	N	M	M'	Δt	Figure
10	Dennis	180	40	40	0.00005	3.5
7.5	Mills	192	40	40	0.00005	3.6
25	Mills	192	40	60	0.00005	3.7
100	Mills	192	80	120	0.00025	3.8
200	Mills	192	80	120	0.00025	3.9



INLET BREADTH, $b_I = 2\varepsilon R_o$

INLET REYNOLDS NUMBER, $Re_I = \frac{U_I \varepsilon R_o}{\nu}$

Figure 3.1 Definition sketch

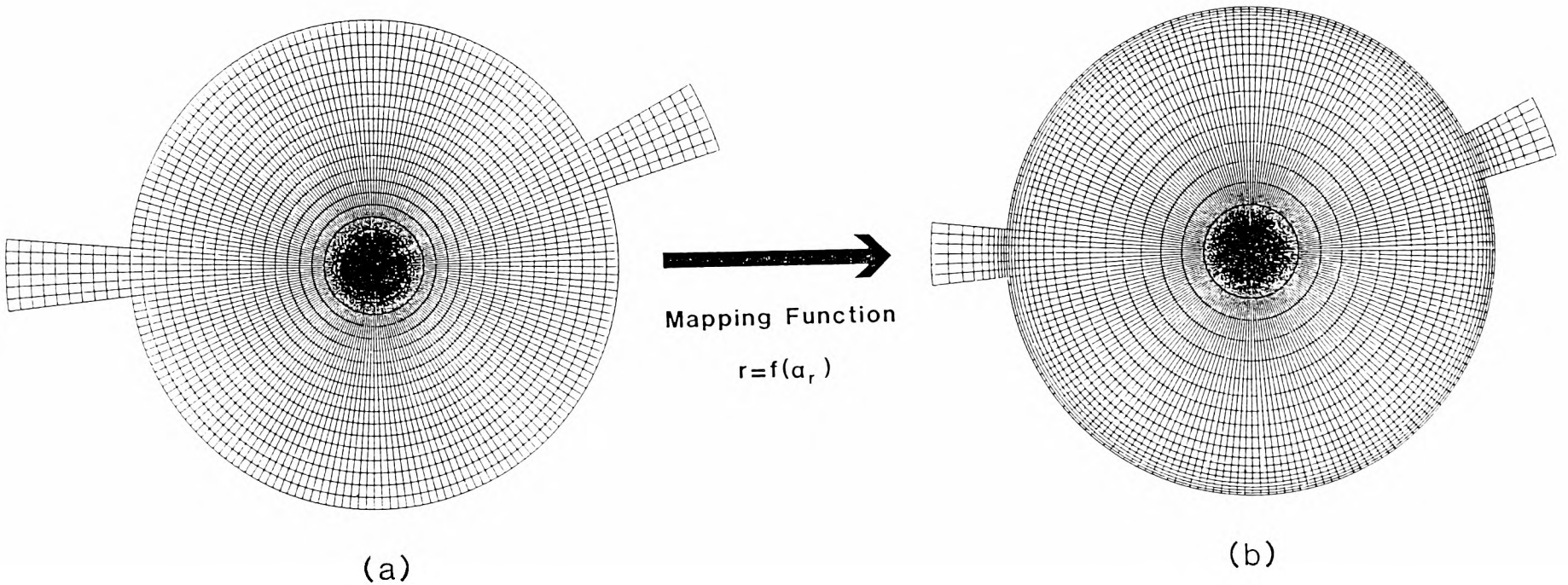


Figure 3.2 Mapping process

(a) Uniform finite-difference mesh in computational space

(b) Variable grid in the physical domain

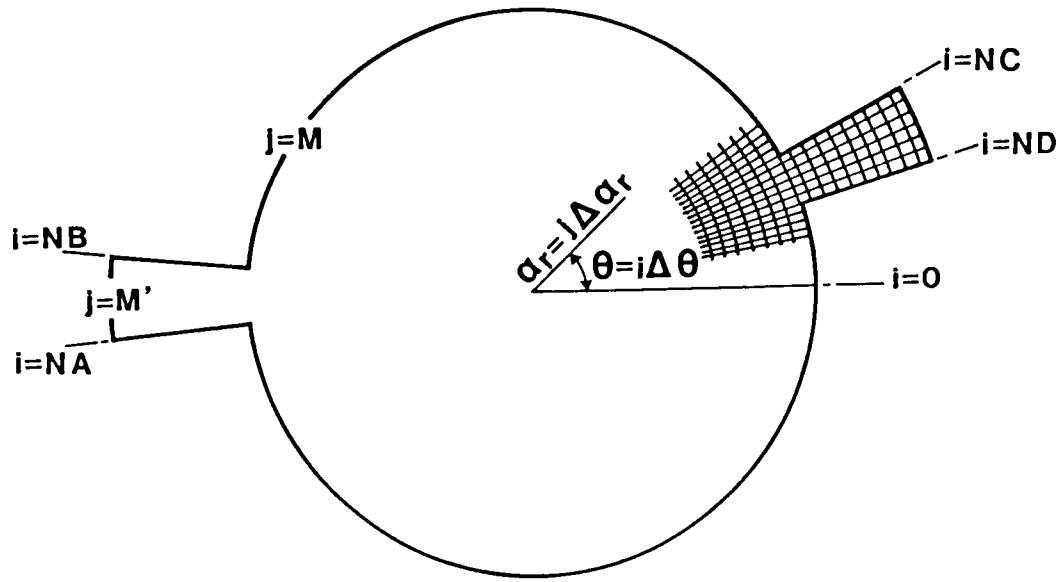


Figure 3.3 Computational mesh

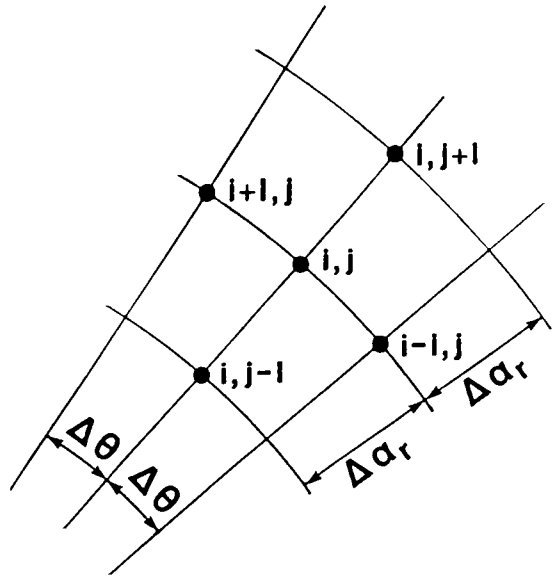


Figure 3.4 Detail of node and adjacent points (computational mesh)

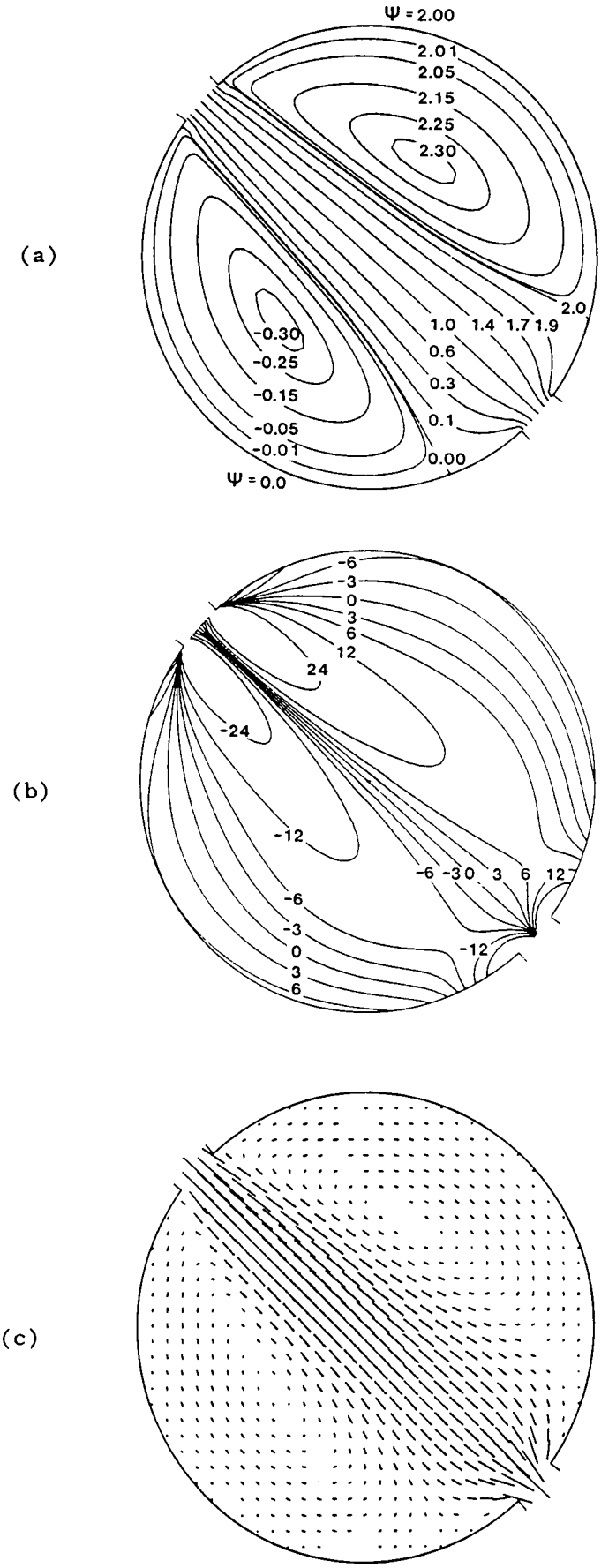


Figure 3.5 Stream function, vorticity and velocity distributions for Re_I=10, $\alpha=\pi$, $\epsilon=\pi/30$ and t=2

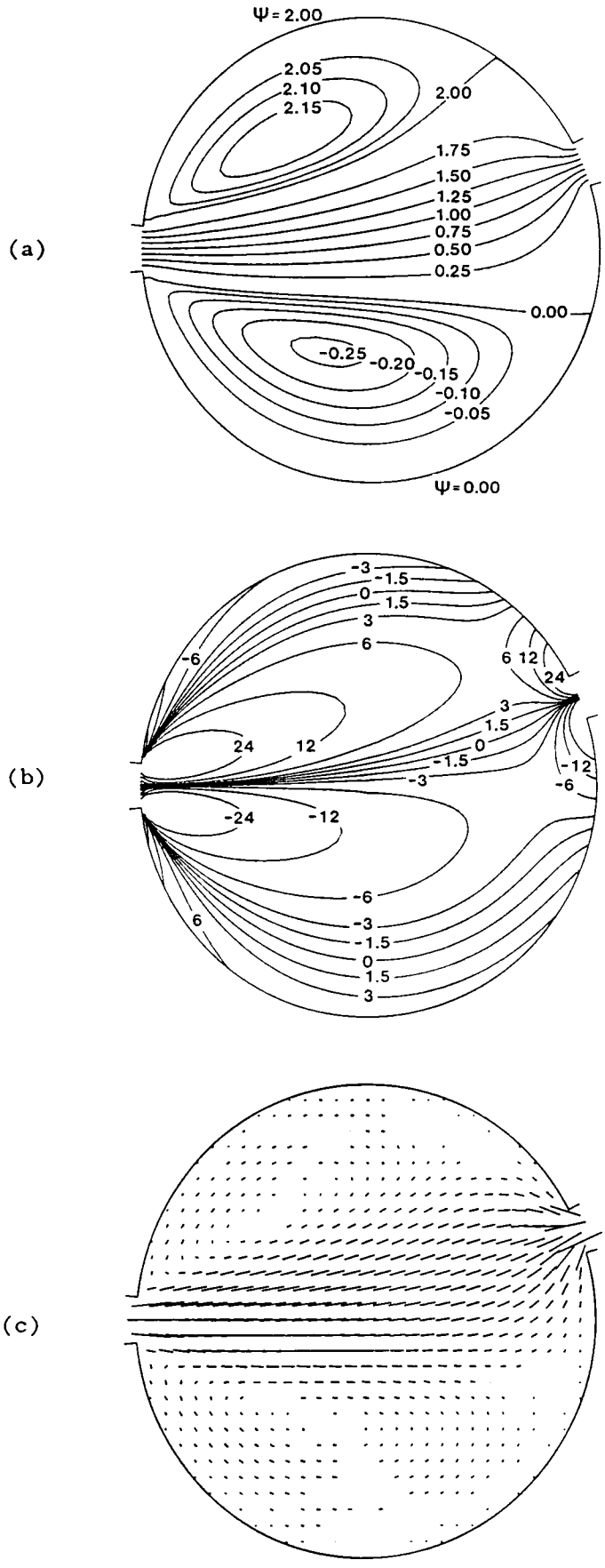


Figure 3.6 Stream function, vorticity and velocity distributions for Re_I=7.5, $\alpha=7\pi/8$, $\epsilon=\pi/32$ and t=2

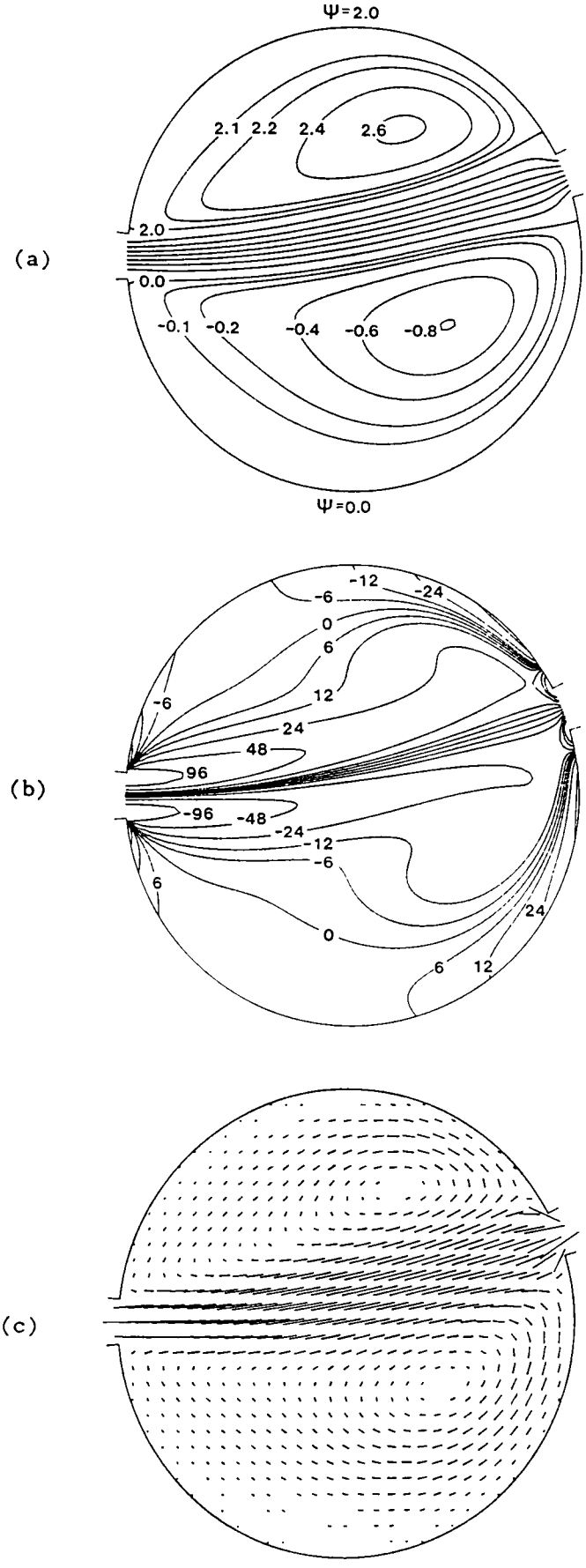


Figure 3.7 Stream function, vorticity and velocity distributions for Re_I=25, $\alpha=7\pi/8$, $\epsilon=\pi/32$ and t=7.5

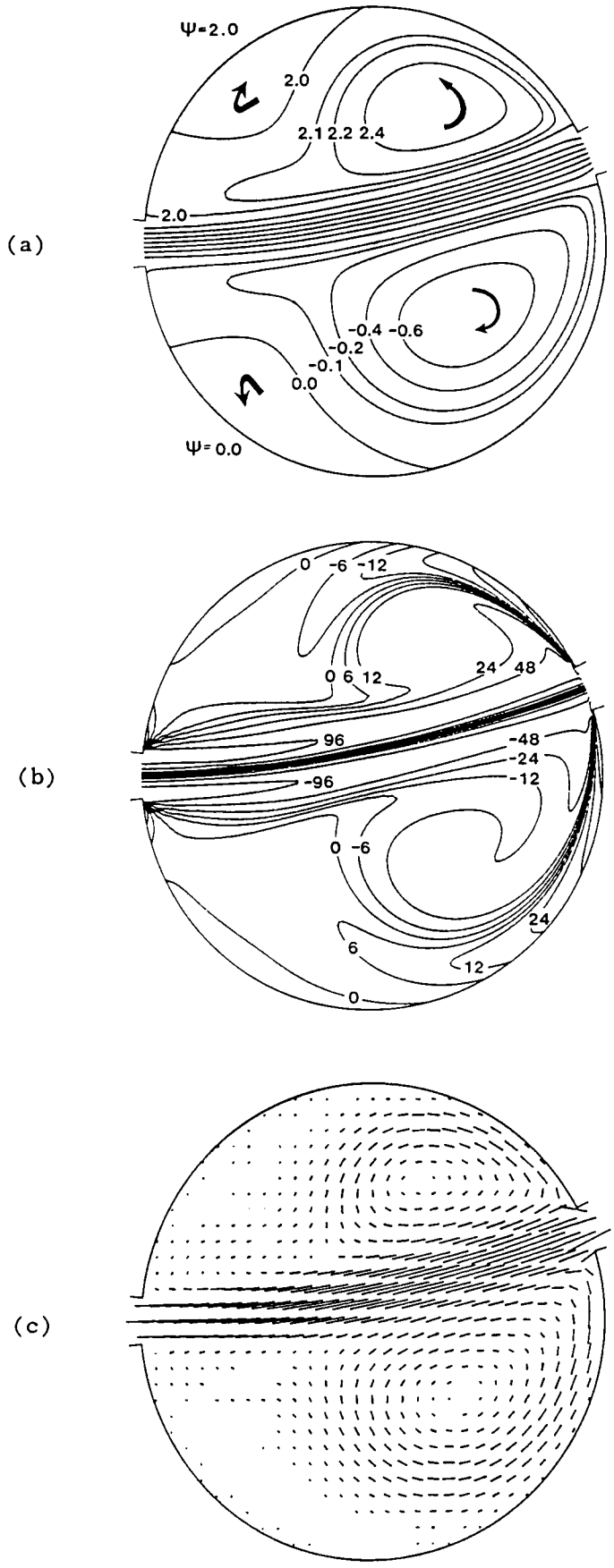


Figure 3.8 Stream function, vorticity and velocity distributions for Re_1=100, $\alpha=7\pi/8$, $\epsilon=\pi/32$ and t=30

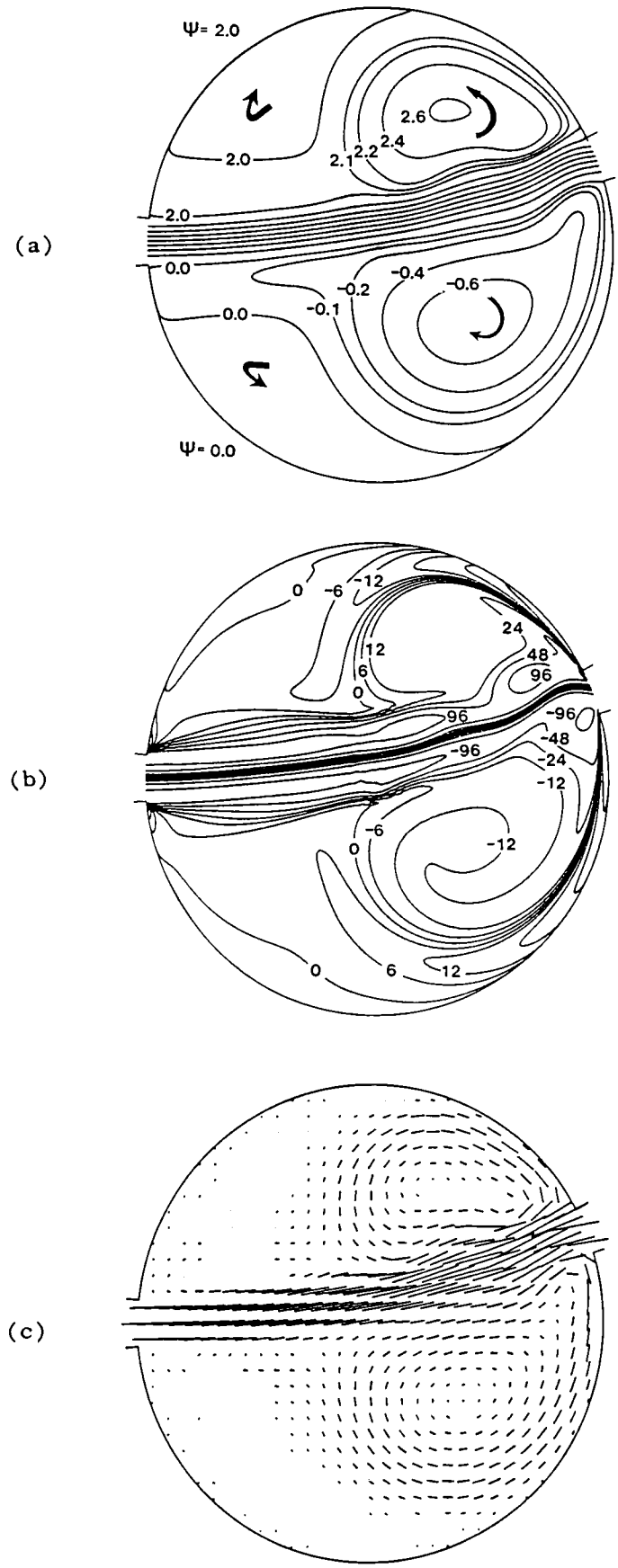


Figure 3.9 Stream function, vorticity and velocity distributions for Re_I=200, $\alpha=7\pi/8$, $\epsilon=\pi/32$ and t=33.2

CHAPTER 4
BOUNDARY-FITTED COORDINATE SYSTEMS

4.1 Introduction

The numerical solution of the Navier-Stokes equations, described in the previous chapter, implemented an algebraic coordinate transformation in the radial direction in order to concentrate the finite-difference mesh near to the outer boundaries of the circular reservoir. A polar mesh was the obvious choice for the grid system since a coordinate line of constant radius coincided with the circular boundary of the reservoir, thus considerably reducing both the complexity of the computer coding and the discretisation errors of the finite-difference scheme. Nevertheless, the choice of a polar mesh severely restricted the geometry of the reservoir because the walls defining the inlet and outlet passageways had to coincide with the radial lines of the grid system in order to avoid problems associated with interpolation schemes at the boundaries. In particular, the computer model could not handle readily the geometrically simple, and possibly more realistic, problem of a circular reservoir with parallel sided inlet and outlet stems. Thus it can be judged that the numerical simulation of jet-forced flows in circular reservoirs should benefit from the use of a more advanced grid mapping technique. This chapter is concerned with the construction of boundary-fitted grids (B.F.G.'s) and follows the widely used methodology suggested by Thompson et al. (1974 onwards) in which a curvilinear coordinate system is generated by the numerical solution of a pair of Poisson equations. The use of a boundary-fitted coordinate system increases the flexibility in defining the geometry of the reservoir's perimeter and could, if required, be used for more complicated flow domains with highly irregular shapes.

4.2 Literature review

4.2.1 Boundary representation in Cartesian coordinate systems

Most of the commonly used finite-difference schemes for predicting the flow behaviour in rivers, tidal harbours, estuaries and reservoirs are based upon regular, rectangular Cartesian coordinate meshes. Although Cartesian grid finite-difference schemes are very popular (as evidenced by the extensive literature on the subject), they have the disadvantage that it is difficult to model curved or irregular boundaries accurately. Leendertse (1967), Abbott et al. (1973) and Hodgins (1977) all approximated the curvature of a boundary by a 'staircase' of points in the

rectangular mesh. However, this leads to the spurious generation of vorticity at the sharp corners of the 'staircase' and hence such schemes may produce unnecessary recirculation. Weare (1979) has also shown that the maximum allowable time step for A.D.I. techniques with irregular boundaries is severely restricted compared to cases where the boundary is aligned with the computational grid axes.

Other popular schemes, including those of Kuipers and Vreugdenhil (1973) and Falconer (1976, 1980, 1984a), also used a 'stepped' pattern to approximate the curved boundary, but included a correction for the curvature by modifying the water depths at grid nodes on the edges of the finite-difference mesh. Unlike the Leendertse scheme where the 'staircase' falls partially inside and partially outside the flow domain, the shape of the flow region is located entirely within the boundaries of the grid; the water depths at grid points outside the boundary curve are then given appropriate values such that the cross-sectional area of each boundary cell is equal to that defined by the actual curved perimeter. However, this technique is questionable due to the imaginary depth gradients set up in the vicinity of the boundary, perhaps indicating why the majority of numerical models use the earlier method of boundary representation, as utilised by Leendertse.

In an attempt to model a large coastal region and yet maintain a sufficient grid resolution in desired areas of interest, Butler (1978b) and Houston and Butler (1979) developed a storm surge/tsunami inundation model which used a smoothly varying grid that allowed the cells to be small in certain areas and large in others. The x- and y-direction coordinate lines were stretched algebraically so that they were independent of each other. Although the resulting coordinate lines were still straight, the mapping gave much better resolution where needed. This procedure was also used by Vemulakonda et al. (1985) to study coastal and inlet processes; their grid stretching was enhanced by the use of piecewise algebraic transformation functions, determined by an interactive computer program, giving much more flexibility in the mapping process. However, both these models still required the use of the Leendertse form of 'staircase' boundaries.

All finite-difference models which require the use of either the simple 'staircase' approach or interpolation techniques for grid points not coincident with the boundary are clearly less accurate than models in which the boundaries of the flow domain coincide with the coordinate lines. This is especially the case for the solution of both the Navier-Stokes and shallow water equations, since the flow characteristics in the interior of the mesh are dominated by the boundary conditions. The large velocity gradients in the vicinity of boundaries are situated at precisely the place

where the numerical representation is least accurate.

The apparent inability of finite-difference schemes to model curved boundaries adequately may imply that finite-element methods are more suitable for computing hydrodynamic flow phenomena in complex geometries. However, Weare (1976a) has shown that the solution of the two-dimensional shallow water equations using finite-element techniques is not economically attractive compared with finite-difference methods. This is because the work required per grid point per time step increases with the band-width of the global stiffness matrix for finite-element methods, whereas it is independent of the size of the computational mesh for finite-difference schemes. Ramanathan and Kumar (1988) have investigated transient heat conduction using a boundary-fitted finite-difference scheme and a finite-element method. For identically sized meshes, they conclude that the boundary-fitted coordinate approach is more accurate and economical (in terms of computational time and storage requirements) than the finite-element technique. Another important consideration is the fact that finite-element methods are more suited to partial differential systems that can be expressed in terms of a variational principle. The non-linear Navier-Stokes and shallow water equations, however, cannot be expressed as a minimisation of a true physical quantity, and hence, their solution has to be obtained through the use of a weighted residual technique such as Galerkins' method (Zienkiewicz, 1977; Smith and Griffiths, 1988). Using this procedure, the partial differential equations themselves are not actually satisfied, but instead weighted residual expressions are solved. It is generally accepted that, for a given accuracy, the elements have to be finer for residual methods than for variational approaches, in order to compensate for the solution of the approximated residual equations (Johnson and Thompson, 1978). Moreover, there still exists doubt as to the assessment of the accuracy of finite-element methods when solving the non-linear shallow water equations (Basco, 1983). As stated by Thompson and Johnson (1985), a further disadvantage is that finite-element methods are generally more unwieldy to code and this causes problems not only during program development but also in subsequent modifications of the model.

4.2.2 Alternative finite-difference schemes on irregular meshes

Possibly one of the earliest successful attempts at overcoming the shortcomings of regular, rectangular finite-difference schemes was made by Boericke and Hall (1974), for the investigation of thermal dispersion in an estuary. The unique feature of their model was the use of a coordinate transformation to map the irregular

shorelines of the estuary onto a rectangular computational space, thereby avoiding the problems associated with 'staircase' boundaries or interpolation schemes. This mapping process was only performed in the transverse direction of the estuary, with the longitudinal mesh spacing remaining unaltered. Even so, the principle of mapping an irregular physical mesh onto a regular transformed grid underpins the whole methodology of boundary-fitted coordinate systems.

At approximately the same time as Boericke and Hall were developing their estuary model, Thompson et al. (1974) proposed a much more flexible transformation procedure which mapped two coordinate directions at the same time. This system allowed the generation of a completely arbitrary non-orthogonal coordinate grid with coordinate lines coincident with all boundaries of the flow domain. Thompson et al.'s work on coordinate transformation revolutionised computational fluid dynamics in that it provided the means by which finite-difference schemes could treat irregular boundaries accurately. Much of the emphasis of research into the finite-element method (since it was previously the only method to deal adequately with complex boundaries) has now switched to the implementation of boundary-fitted coordinate systems with finite-difference and, more recently, finite-volume techniques.

Before reviewing Thompson's work more fully, it is worthwhile considering some of the other finite-difference schemes that have been implemented on non-standard coordinate meshes. One of the most interesting and novel approaches is that of Thacker (1977) who described a finite-difference method with completely irregular grids for solving the linearised shallow water equations in a circular basin. Thacker considered the irregular two-dimensional grid to be a patchwork of triangular elements whose vertices were the grid nodes. In the vicinity of each such triangle, the partial derivatives of a function were approximated by the slopes of the plane determined by the values of the function at the vertices of the triangle. Since each grid point was the vertex of several different triangular elements, Thacker used a weighted average of all the slopes from adjacent triangles surrounding the grid point in order to determine the partial derivatives. He was thus able to calculate finite-differences on completely irregular meshes, thereby effectively combining the best aspects of finite-element and finite-difference schemes. Later on, Thacker (1979) used the method for the calculation of storm surges in a bay with a curved coastline and the same principles were also used by Bauer and Schmidt (1983) to simulate the effects of storm surges on Lake Geneva.

Another important technique for the solution of hydrodynamic flow problems in domains with irregular boundaries involves the use of orthogonal curvilinear

coordinate systems. The method can be subdivided further into those techniques which determine the orthogonal curvilinear grid algebraically and those which use numerical means. Wanstrath (1977) studied nearshore storm surge and tidal processes using a conformal mapping process, based on Fourier series, to transform the irregular coastline and off-shore boundary curves on to a non-uniformly spaced rectangular mesh. A second algebraic mapping was then used so that the non-uniformly spaced rectangular mesh could be transformed into a regular grid for the purposes of the finite-difference scheme. Although the bi-curve fitting conformal mapping coefficients have to be determined iteratively, the process is essentially algebraic. Hamilton (1979) also used a conformal mapping procedure for the investigation of storm surges in narrow straits; the emphasis, in this case, being directed towards the numerical scheme for the solution of the shallow water equations on orthogonal grids rather than the practicalities of grid generation.

Numerically generated orthogonal curvilinear coordinate systems have received much attention in The Netherlands, with Wijbenga (1985a, 1985b) and Willemse et al. (1985) using such systems to simulate flows in rivers and flood plains. Much of the impetus to use orthogonal curvilinear coordinate systems has stemmed from the fact that the 'transformed' governing hydrodynamic equations are simpler and contain less terms than the equivalent flow equations for non-orthogonal grids. Verboom (1983) showed that the transformed shallow water equations are remarkably similar in form to the standard Cartesian shallow water expressions, provided the velocities in the governing flow equations are written in terms of the velocity components tangential to the coordinate lines of the orthogonal mesh. This has meant that many of the older existing computer models, based upon rectangular Cartesian finite-difference meshes have easily been modified and updated for use with orthogonal curvilinear grid systems.

The numerical generation of orthogonal curvilinear transformations is performed in a very similar manner to Thompson's general technique for non-orthogonal mappings (discussed in detail in Section 4.3), except for the addition of extra constraints to ensure that the two sets of curvilinear coordinate lines are indeed orthogonal. Wijbenga and Willemse et al. achieved orthogonality by rewriting Thompson's generalised Poisson equations in a form that did not need the explicit specification of the so-called attraction-operators. Physically, the modified transformation expressions could be interpreted as the solution of a friction dominated flow in a confined aquifer, with the piezometric pressure head against the rigid lid and the stream function effectively defining the values of the two new coordinates. Since the lines of constant stream function and constant pressure head would be

normal to each other, the equivalent transformation equations automatically produced an orthogonal mesh. One of the major uses of these curvilinear coordinate meshes was in the numerical study of rivers with wide flood plains, where it was desirable to have a coordinate system that allowed the grid lines to congregate near the main river channel in order to achieve better resolution in regions with high flow velocities. This was easily accomplished because the actual depth of the flow in the river could also be specified in the modified transformation equations, using the analogy that the confined aquifer could have a flat top surface but a variable depth. The technique proved to be very suitable for the generation of coordinate meshes for practical applications with complex geometries and bathymetries.

The main disadvantage with the use of orthogonal transformations for boundary-fitted coordinate systems arises from the fact that, by necessity, the coordinate lines must be orthogonal at the boundaries of the grid. This greatly reduces the flexibility in determining the distribution of the grid points at the boundary (Thompson et al., 1977b). Furthermore, those schemes that rely on the orthogonality of the mesh for their computational procedures, have additional problems associated with the fact that numerically generated coordinate systems can never achieve perfect orthogonality. This means considerable attention must be paid to the effects of the orthogonality deviations on the accuracy of the flow calculations.

Provided the user is prepared to accept the increased complexity of the transformed hydrodynamic equations and is prepared to develop new techniques for the numerical solution of such equations, Thompson's method for generating non-orthogonal coordinate systems offers much more flexibility in the specification of the boundary points. Moreover, the numerical generation of non-orthogonal meshes is simpler than that for orthogonal curvilinear systems, because of the absence of orthogonality constraints. Thompson's procedure for creating non-orthogonal meshes is the approach adopted in this thesis; the mathematical generation of such coordinate systems being described in detail later in this chapter.

4.2.3 Thompson's non-orthogonal boundary-fitted coordinate system

A number of Thompson's papers concerning boundary-fitted coordinates have already been cited in the previous sections of this chapter. There is now an extensive amount of literature on the subject and the interested reader is referred to the comprehensive review articles by Thompson et al. (1982) and Thompson (1984). Although Thompson modestly states that the principles behind numerical grid

generation already existed before his earliest papers on the subject, his approach is now used almost universally for the numerical generation of curvilinear meshes. This is because he was able to demonstrate the usefulness of the technique, by presenting finite-difference solutions of potential flow around irregularly shaped aerofoils, including multiple aerofoil systems (Thompson et al., 1974). The later work of Thames (1975), Thompson et al. (1975) and Thames et al. (1977) extended the solutions to include viscous flow over aerofoils using the Navier-Stokes equations. More importantly, the papers and reports on TOMCAT (Thompson et al., 1977a and 1977b), a computer code for the numerical generation of boundary-fitted curvilinear coordinate systems, made Thompson the leading exponent of such techniques. The free availability of the TOMCAT program in his report for NASA, allowed many other workers to set up curvilinear coordinate codes. Indeed, the computer program developed to generate the curvilinear coordinates for the circular reservoir problems discussed in this thesis, owes much to the TOMCAT code.

The use of boundary-fitted grids, with coordinate lines coincident with the boundaries of the flow domain, no matter how irregular the shape of the region, avoids the problems associated with the previously described 'staircase' or interpolation boundary schemes of Cartesian finite-difference methods. Boundary-fitted coordinate systems, therefore, offer the possibility of accurately simulating flows in complex geometries using finite-difference techniques. However, even if the primary method of solution is by means of finite-elements, boundary-fitted systems can still play an important role in the production of the elemental grid (Baker, 1985). Thompson's boundary-fitted technique uses elliptic Poisson equations to generate the mesh and therefore produces very smooth curvilinear grids, ideal for finite-element methods.

Examination of the proceedings of a symposium on numerical grid generation techniques (Thompson, 1982a) indicates the rapid increase in the popularity and diversity of the basic method, just under eight years after Thompson published his first paper on the subject. Thompson's more recent work (1987) is concerned with the implementation of a fully three-dimensional boundary-fitted technique to compute the flow past very complex geometries such as the complete fuselage and wing assemblies of aircraft. Throughout the history of boundary-fitted coordinates, much of the impetus has come from aerodynamics and the requirement to compute flows around aerofoils of unusual geometry. For example, the TOMCAT code enabled the flow region to contain any number of arbitrary shaped bodies and was thus applicable to the flows past a wing/aileron section and the flows in a turbine cascade.

The practical flow regions which require to be modelled in civil engineering applications (such as rivers, lakes, estuaries and harbours) appear to be more complicated than the multiple sectioned aerofoils and turbine blades that can be created by TOMCAT. In actual fact, this is probably an optical illusion caused by the more irregularly curved boundaries. Nevertheless, the TOMCAT code is generally more suited to the geometries of flows around obstacles rather than for the flow through them. For example, the code has difficulty dealing with the generation of the relatively simple geometry of a bifurcation. In view of this, Thompson developed WESC0R, a curvilinear coordinate generation program more applicable to the type of geometries found in environmental hydrodynamic modelling and water quality studies. The extensive use of the topological operations of branch cuts and re-entrant boundaries, implemented in TOMCAT to produce the multiple obstacles, was discarded in WESC0R in favour of the simpler and more direct approach of using slits and slabs in the transformed plane. The shape of the outer perimeter of the transformed coordinate mesh, restricted in TOMCAT to a simple rectangle, was given more flexibility in the WESC0R code. Furthermore, WESC0R facilitated the use of a more elaborate form of coordinate control, with the attraction of coordinate lines determined partly by the boundary grid point distribution. Thompson (1983) impressively demonstrated the capabilities of the new grid generation code by presenting diagrams of intricate curvilinear coordinate systems representative of Charleston Harbour, the Delaware River and Lake Erie.

An important distinction between the TOMCAT and WESC0R codes was the redefinition of the attraction-operators in the elliptic Poisson equations used to generate the transformation. Although the modifications to the mathematical definition of the mapping procedure resulted in values of the control functions changing by several orders of magnitude (Thompson, 1982b and Thompson et al., 1985), the principles of the transformation remain unaltered. The changes in the attraction-operators enabled them to be determined from the boundary point distribution and also allowed the system of elliptic Poisson equations to take on a particularly simple form for the theoretical one-dimensional case. For the purposes of the grid generation in the present reservoir study, the older (pre 1982) definition of the attraction-operators was found to be perfectly adequate. Moreover, many of the curvilinear grids, employed in Chapter 7, do not use any coordinate line control and hence the Poisson equations often reduce to a Laplacian system. In these particular cases, without coordinate attraction, the mapping expressions used by the TOMCAT and WESC0R codes are therefore equivalent.

Although simpler than WESCOR, the boundary-fitted coordinate generation code developed by the author has a number of additional features which are described later in this Chapter. These include the option of using curve-wise parameterisation of cubic-splines to define the boundary of the flow region and the development of a technique to simplify the computer 'book-keeping' arrangements for the solution of the governing hydrodynamic equations on irregularly shaped transformed meshes. The latter development allowed the transformed mesh to have an irregular shape, without having to resort to the usual procedure of splitting the transformed mesh into a number of simple rectangular regions (the so-called composite block structure approach - Coleman (1982); Thompson et al. (1985)). Composite block structures have particular difficulty in solving the flow equations using implicit techniques because of the problems associated with the continuation of implicit flow calculations from one block to another.

4.2.4 Alternative methods for generating non-orthogonal boundary-fitted coordinates

Thompson's grid generation technique, based on the numerical solution of a pair of Poisson equations, is by far the most popular method of generating non-orthogonal curvilinear meshes. However, the computational inefficiency of having to solve the elliptic equations iteratively and the difficulty of controlling the coordinate lines within the interior of the mesh have encouraged developments in alternative boundary-fitted techniques. One such grid generation scheme is that presented by Gilding (1988), which used algebraic mappings based on the blending of a number of shearing transformations to create the non-orthogonal boundary-fitted mesh. Gilding quotes the main advantage of the method as being its computational cost effectiveness over conventional grid mapping techniques.

The major disadvantage with Gilding's method is that the curvilinear meshes are considerably less smooth than those generated by more traditional means. This is particularly noticeable at the interface between the zones of influence of different shearing operators, where there are large discontinuities in the direction of the coordinate lines. Nevertheless, the grid generation method was used successfully in conjunction with ODYSSEE, the Navier-Stokes equation solver for boundary-fitted coordinate systems developed at the Delft Hydraulics Laboratory (Officier and Wiersma, 1986 and Officier et al., 1986). Tests were conducted by Officier et al. to establish the influence of grid smoothness on the solution of the hydrodynamic equations. After comparing results from simulations using both Gilding's and Thompson's grid generation techniques, Officier et al. concluded that the non-smooth

grids based upon shearing functions were acceptable, and that minor differences in the flow solutions produced by ODYSSEE could be attributable to the boundary treatment rather than the grids themselves. However, the sudden spatial variations in the direction of the curvilinear coordinates for this type of shearing transformation imply that elliptic equation approaches are generally more suitable for the generation of boundary-fitted meshes.

4.2.5 Adaptive grid techniques

A review of boundary-fitted coordinate systems would be incomplete without a brief description of the relatively new technique of adaptive grids. The fundamental principle of the method is to have an automatic adjustment of the coordinate mesh whilst the physical flow solution develops, so as to concentrate grid points in regions of large solution variation. As Thompson et al. (1985) pointed out, although the mesh nodes must concentrate in regions of large solution variation, no region can be allowed to become depleted of grid points. Moreover, the coordinate system must retain sufficient smoothness and must not become too skewed. The mechanism for converting the mutual influence of neighbouring grid points into a grid movement can either be based upon a form of attraction/repulsion between mesh nodes (a discretised approach) or on a variational principle (a continuous field approach).

Rai and Anderson (1981) and Anderson (1983) used the analogy that the grid points behaved like electrostatic charges. The numerical truncation error in the flow solution was estimated at each mesh node and the average discretisation error over the whole computation space calculated. If the error at a particular grid point was larger than the average value, a mutually attractive force was created that caused the surrounding mesh nodes to move towards the grid point, thereby reducing the excess truncation error. Conversely, repulsive forces were generated at grid points where the discretisation error was smaller than the mean value. The forces between the nodes were assumed to obey an inverse power law of distance, and were converted into a resultant set of grid velocities in order to update the mesh coordinates. Rai and Anderson demonstrated the procedure by studying the time-asymptotic, one-dimensional and linearised two-dimensional viscous Burgers' equations. Another very similar method of grid adaptation was suggested by Nakahashi and Deiwert (1984), who proposed a tension and torsion spring analogy for the purposes of simulating compressible fluid flow problems involving shocks. The tension spring controlled the mesh clustering whilst the torsion spring prevented excessive grid skewness.

The alternative, variational technique for grid adaptation involves the adjustment of the grid to minimise certain integral expressions representing the mesh concentration, grid smoothness and grid orthogonality. By weighting the importance of each of the three integrals separately, the resultant coordinate mesh can be made to emphasize one or other of the above properties. Johnson et al. (1984) discussed the possibilities of using such a technique for the static, depth-adaptation of grids employed in estuarine/coastal models. This resulted in the modification of the WESCOR code to allow the coordinate lines to follow deep navigation channels (Thompson and Johnson, 1985; Johnson and Thompson, 1986).

Thompson and Johnson (1985) also tried a depth-adaptation process based upon the control functions used in the Poisson equations of the standard elliptic generating technique, but concluded that the variational formulation was more reliable. In addition, they found that natural channels in real estuaries contained insufficient depth gradients to allow satisfactory grid adaptation, and thus they had to resort to the use of false depth values to force the coordinate lines to congregate in the regions of deeper water.

In the future, boundary-fitted coordinate systems may offer the interesting possibility of accurately modelling flows in estuarial areas, where there are large changes in the cross sectional profiles of the flow during each tidal cycle. The conventional techniques for modelling the changes in the wetted plan-form area (e.g., Leendertse and Gritton, 1971; Falconer and Owens, 1987), using flooding/drying cells, can often produce troublesome oscillations in the water surface elevations at mesh points near the boundary. Time-dependent boundary-fitted coordinate systems, in which the grid constantly adapts to the changing shape of the wetted perimeter, may overcome this difficulty, by removing the need for a flooding/drying technique entirely. However, as yet, no published research has been presented along these lines.

4.3 Mathematical development of boundary-fitted coordinate systems

As stated earlier, boundary-fitted grids have coordinate lines coincident with the boundaries of the flow domain, no matter how irregular the shape of the region. However, before describing the equations which are used to generate the non-orthogonal curvilinear coordinate systems, it is necessary to demonstrate certain derivative relationships for the reversible mapping between the transformed and

physical planes.

4.3.1 Derivative relationships

Figure 4.1 shows a diagrammatic representation of the mapping of an irregularly shaped domain, R, in the physical plane onto the rectangle, S, in the transformed plane. As will be evident later on, the transformed (or computational) domain is composed of a square-celled finite-difference grid whilst the physical domain contains the associated non-orthogonal curvilinear coordinate mesh. It is assumed that the mapping procedure is one-to-one, and consequently reversible.

Considering the mapping from the transformed plane to the physical plane, then if $x=x(\xi,\eta)$ and $y=y(\xi,\eta)$, the Jacobian of x and y with respect to ξ and η is given by

$$J = \frac{\partial(x,y)}{\partial(\xi,\eta)} = \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{vmatrix} = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi} \quad . \qquad (4.1)$$

Furthermore, considering the mapping in the opposite direction (ie. from the physical plane to the transformed plane), then if $\xi=\xi(x,y)$ and $\eta=\eta(x,y)$, the Jacobian of ξ and η with respect to x and y is given by

$$J' = \frac{\partial(\xi,\eta)}{\partial(x,y)} = \begin{vmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} \end{vmatrix} = \frac{\partial \xi}{\partial x}\frac{\partial \eta}{\partial y} - \frac{\partial \xi}{\partial y}\frac{\partial \eta}{\partial x} \quad . \qquad (4.2)$$

Since the mapping is reversible, each Jacobian must be the inverse of the other ie., $J^{-1} = J'$.

Therefore,

$$J^{-1} = \frac{1}{J} \begin{vmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial x}{\partial \eta} \\ -\frac{\partial y}{\partial \xi} & \frac{\partial x}{\partial \xi} \end{vmatrix} = \begin{vmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} \end{vmatrix} \quad . \qquad (4.3)$$

By inspection:

$$\left.\begin{aligned} \frac{\partial \xi}{\partial x} &= \frac{1}{J}\frac{\partial y}{\partial \eta} \quad ; \quad \frac{\partial \xi}{\partial y} = -\frac{1}{J}\frac{\partial x}{\partial \eta} \\ \frac{\partial \eta}{\partial x} &= -\frac{1}{J}\frac{\partial y}{\partial \xi} \quad ; \quad \frac{\partial \eta}{\partial y} = \frac{1}{J}\frac{\partial x}{\partial \xi} \end{aligned}\right\} \qquad (4.4)$$

where

$$J = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi} \quad .$$

In order to reduce the size of the presented equations, it is advantageous to adopt the usual subscript shorthand form for partial derivatives. For example,

$\frac{\partial y}{\partial \eta}$ is referred to as y_η, whilst a second derivative such as

$\frac{\partial^2 x}{\partial \xi^2}$ is written as $x_{\xi\xi}$.

By differentiating the expressions in (4.4), the second derivatives of the transformed coordinates can be obtained:

$$\left.\begin{aligned} \xi_{xx} &= (\xi_x y_{\xi\eta} + \eta_x y_{\eta\eta})/J - ([\xi_x]^2 J_\xi + \xi_x \eta_x J_\eta)/J \\ \xi_{yy} &= -(\eta_y x_{\eta\eta} + \xi_y x_{\xi\eta})/J - (\xi_y \eta_y J_\eta + [\xi_y]^2 J_\xi)/J \\ \eta_{xx} &= -(\xi_x y_{\xi\xi} + \eta_x y_{\xi\eta})/J - (\xi_x \eta_x J_\xi + [\eta_x]^2 J_\eta)/J \\ \eta_{yy} &= (\eta_y x_{\xi\eta} + \xi_y x_{\xi\xi})/J - (\xi_y \eta_y J_\xi + [\eta_y]^2 J_\eta)/J \end{aligned}\right\} \qquad (4.5)$$

These expressions are needed to interchange the dependent and independent variables of the grid generation equations (see Section 4.3.2).

For the purposes of curvilinear mesh generation, the derivatives of a scalar quantity, f in the physical plane do not require to be transformed. Later in Chapter 5, however, during the transformation of the Cartesian governing hydrodynamic equations, extensive use is made of the formulae expressing f_x and f_y in terms of the variables ξ and η. Therefore, it is worthwhile considering the transformation of these derivatives at this stage. Using the Chain Rule for partial differential equations gives,

$$\frac{\partial f}{\partial x} = \frac{\partial(f,y)}{\partial(\xi,\eta)} \div \frac{\partial(x,y)}{\partial(\xi,\eta)} = \frac{1}{J} \left(y_\eta f_\xi - f_\eta y_\xi \right) \qquad (4.6a)$$

and

$$\frac{\partial f}{\partial y} = \frac{\partial(x,f)}{\partial(\xi,\eta)} \div \frac{\partial(x,y)}{\partial(\xi,\eta)} = \frac{1}{J} \left(x_\xi f_\eta - f_\xi x_\eta \right) \quad . \qquad (4.6b)$$

These equations are in the so-called non-conservative form, but they can easily be converted into the conservative expressions by the addition of the terms $f(y_{\eta\xi}-y_{\xi\eta})/J=0$ to (4.6a) and $f(x_{\xi\eta}-x_{\eta\xi})/J=0$ to (4.6b). For example, the x-direction derivative then becomes

$$\frac{\partial f}{\partial x} = \frac{1}{J} (y_\eta f_\xi + f y_{\eta\xi} - f y_{\xi\eta} - f_\eta y_\xi) = \frac{1}{J} \left[(f y_\eta)_\xi - (f y_\xi)_\eta \right] \qquad (4.7a)$$

and the y-derivative becomes

$$\frac{\partial f}{\partial y} = \frac{1}{J} (x_\xi f_\eta + f x_{\xi\eta} - f x_{\eta\xi} - f_\xi x_\eta) = \frac{1}{J} \left[(f x_\xi)_\eta - (f x_\eta)_\xi \right] \quad . \qquad (4.7b)$$

Although equations (4.6) and (4.7) are analytically equivalent, the expressions will be different if evaluated numerically. This is because the second derivatives, such as $y_{\eta\xi}$ and $y_{\xi\eta}$ are not interchangeable in a numerical scheme due to the discretisation process. The difference between the conservative and non-conservative forms can be easily illustrated by considering the x-direction derivative:

$$J f_x = \underbrace{(f y_\eta)_\xi - (f y_\xi)_\eta}_{\text{'conservative'}} = \underbrace{y_\eta f_\xi - f_\eta y_\xi}_{\text{'non-conservative'}} \quad . \qquad (4.8)$$

For uniform f, the non-conservative form does indeed give f_x as zero. However, this is not the case with the conservative scheme because the finite-differencing procedure cannot generally guarantee that $y_{\eta\xi}$ and $y_{\xi\eta}$ are identical.

Although the early implementations of boundary-fitted coordinate schemes used the simpler, non-conservative form of derivative transformations (e.g., Thompson et al. (1974), Thames (1975) and Thompson et al. (1977b)), the later work of Johnson (1980, 1982) employed a fully conservative approach because it was intuitively assumed that such a scheme would result in a more accurate discretisation of the partial differential equations. However, the more recent work of Häuser et al. (1985) has shown that this assumption is in fact false. Häuser et al. used both approaches in a study of the oscillatory motions in an annular ring using the

linearised shallow water equations, and found that the non-conservative form was the more accurate method. In view of this, it was decided that the non-conservative approach should be adopted for the purposes of the present numerical study.

4.3.2 Coordinate mapping equations

The method used to generate the boundary-fitted coordinates involves the numerical solution of a pair of Poisson equations with Dirichlet boundary conditions on all perimeters of the flow domain. This is slightly different to the TOMCAT code, where the extensive use of branch cuts, meant that Dirichlet boundary conditions were neither required nor allowed on the re-entrant boundaries. Elliptic transformation equations produce very smooth curvilinear systems and generally ensure that the coordinate lines do not overlap (Thompson et al. 1977b). Since the elliptical properties are required in the physical domain, the Poisson transformation expressions are fundamentally defined in the physical plane:

$$\xi_{xx} + \xi_{yy} = P(\xi,\eta) \qquad (4.9a)$$

$$\eta_{xx} + \eta_{yy} = Q(\xi,\eta) \qquad (4.9b)$$

The functions P and Q are the so-called 'attraction-operators' or 'control functions' which cause the coordinate lines to concentrate as desired. An anomaly in equation (4.9) is that the spatial variations in the attraction-operators are defined as functions of ξ and η rather than the physical coordinates, x and y. Although this makes it more difficult to attract coordinate lines in the interior of the mesh (since the attraction is not to a fixed line or point in the physical plane but to a line or point in the transformed domain), it does reduce the complexity of the computer coding. More recently Thompson (1983) developed a procedure that would allow the attraction-operators to be defined as functions of x and y. However, such techniques are not warranted for the generation of the circular reservoir geometries of the present study.

Referring back to Fig. 4.1, the imposed Dirichlet boundary conditions are of the form:

$$\xi = \xi_A(x,y) \quad \text{and} \quad \eta = \text{constant} = \eta_1 \quad \text{on} \quad \Gamma_A$$

$$\xi = \xi_B(x,y) \quad \text{and} \quad \eta = \text{constant} = \eta_2 \quad \text{on} \quad \Gamma_B$$

$$\eta = \eta_C(x,y) \quad \text{and} \quad \xi = \text{constant} = \xi_1 \quad \text{on} \quad \Gamma_C$$

$$\eta = \eta_D(x,y) \quad \text{and} \quad \xi = \text{constant} = \xi_2 \quad \text{on} \quad \Gamma_D$$

where ξ_A, ..., ξ_D and η_A, ..., η_D are functions that define the boundaries of the domain.

Since all numerical computations are to be performed in the transformed plane, the system of Poisson equations in (4.9) must be rewritten in terms of the computational coordinates (ξ,η). Using the expressions in (4.5), the dependent and independent variables are interchanged, to give

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} + J^2(Px_\xi + Qx_\eta) = 0 \qquad (4.10a)$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} + J^2(Py_\xi + Qy_\eta) = 0 \qquad (4.10b)$$

where

$$\alpha = x_\eta^2 + y_\eta^2$$

$$\beta = x_\xi x_\eta + y_\xi y_\eta$$

$$\gamma = x_\xi^2 + y_\xi^2$$

and J = Jacobian of the transformation $= x_\xi y_\eta - x_\eta y_\xi$.

The transformed boundary conditions take the form,

$$x = x_A(\xi,\eta_1) \quad \text{and} \quad y = y_A(\xi,\eta_1) \quad \text{on} \quad \Gamma_A^*$$

$$x = x_B(\xi,\eta_2) \quad \text{and} \quad y = y_B(\xi,\eta_2) \quad \text{on} \quad \Gamma_B^*$$

$$x = x_C(\xi_1,\eta) \quad \text{and} \quad y = y_C(\xi_1,\eta) \quad \text{on} \quad \Gamma_C^*$$

$$x = x_D(\xi_2,\eta) \quad \text{and} \quad y = y_D(\xi_2,\eta) \quad \text{on} \quad \Gamma_D^*$$

where x_A, ..., x_D and y_A, ..., y_D are functions that define the boundaries of the physical domain, in terms of the transformed coordinates (ξ,η).

The 'quasi-linear' elliptic equations shown in (4.10) give the physical (x,y) coordinates of the mesh points in terms of the transformed coordinates (ξ,η). Although these expressions are more complex than equations (4.9), the new boundary conditions are specified along the straight edges of the computational domain. This permits the accurate use of finite-difference methods for numerically solving the transformed grid generation equations. Furthermore, the subsequent computations for simulating the partial differential equations representing fluid flow are also performed in the transformed plane; these techniques are discussed later in Chapters 5 and 6.

4.3.3 Coordinate control

Before reviewing the expressions used to evaluate the control functions, P and Q, it is beneficial to consider the reasons for adopting the Poisson generating system. Without the inhomogeneous terms, the mapping expressions in (4.9) reduce to the simpler Laplacian system. This form has several advantages over the Poisson equations, including its ability to guarantee a one-to-one mapping and its production of smoother curvilinear meshes. However, the main drawback with the Laplacian system is that it is incapable of providing any means of coordinate control, other than by altering the boundary point distribution.

An important use of the attraction-operators is to overcome the difficulties associated with strongly curved boundaries. Although elliptic grid generation equations tend to form equally spaced mesh cells in the absence of boundary curvature, the coordinate lines will become more closely spaced near convex surfaces and more widely spaced near concave surfaces (as illustrated in Fig. 4.2). This can be demonstrated by considering the Laplace equation for the η-coordinate generation i.e., $\eta_{xx}+\eta_{yy}=0$. In Figure 4.2a, $\eta_{xx}>0$ because of the convex curvature of the lines of constant η (from hereon called η-lines). Thus in order to satisfy the mapping equation, η_{yy} must be negative and hence the spacing between the η-lines increases in the y-direction. A similar argument can be used for a concave boundary, where it can be seen that $\eta_{xx}<0$. This implies that η_{yy} is positive, and hence the spacing of the η-lines decreases outwards from the concave region (Fig.4.2b).

Control of the coordinate lines is exercised by altering the spatial variations of the attraction-operators, P and Q. The effect of these control functions can be demonstrated by similar reasoning to that used to investigate the influence of boundary curvature. For example, a negative value of Q will cause the η-line

spacing in Figure 4.2a to increase more rapidly away from the boundary. Thus, if $Q<0$, the η-lines will be forced to move in the direction of decreasing η (as shown in Figure 4.3a), whilst negative values of the control function, P cause the ξ-lines to move in the direction of decreasing ξ. If boundary conditions prevent the coordinates along the perimeter from altering, the control functions change the intersection angles between the coordinate lines and the boundary. This is demonstrated in Figure 4.3b, where the negative value of P results in the ξ-lines leaning towards the direction of decreasing ξ.

Thompson et al. (1977a) found that a particularly effective coordinate control procedure was obtained by defining the attraction-operators as decaying exponential functions. Thus considering

$$P(\xi) = - a \exp (-d |\xi-\xi_i|) \qquad (4.11)$$

where a and d are constants, and ξ_i is some specified ξ-line, it can be seen that the expression reaches its maximum amplitude on the line $\xi=\xi_i$ and decays away from this line at a rate dependent on the decay factor, d. This particular function would cause the ξ-lines to concentrate towards one side of the $\xi=\xi_i$ line and move away from the opposite side, by an amount dependent upon the 'attraction' amplitude, a. The spatial extent of the coordinate control is influenced by the value of d; Thompson et al. (1977b) quoted that a decay factor of 1.0 would cause the effects to be confined to within a few lines of the attraction source, whilst 0.1 would result in a fairly wide-spread control. By incorporating a sign-changing function, the ξ-lines can be made to congregate on both sides of the attraction line:

$$P(\xi) = - a \, \mathrm{sgn}(\xi-\xi_i) \exp (-d |\xi-\xi_i|) \quad . \qquad (4.12)$$

Repulsion of the coordinate lines is achieved by the specification of a negative attraction amplitude. Furthermore, it is also possible to concentrate the ξ-lines near a specific point (ξ_i,η_i) in the transformed plane. This is accomplished by the use of an exponential expression of the form:

$$P(\xi,\eta) = - a \, \mathrm{sgn}(\xi-\xi_i) \exp (-d [(\xi-\xi_i)^2+(\eta-\eta_i)^2]^{\frac{1}{2}}). \qquad (4.13)$$

The η-coordinate control function, Q, uses analogous expressions to (4.12) and (4.13), but with the ξ and η variables interchanged.

Coordinate concentration can be achieved with more than one line and one

point source, by defining the attraction-operators as the sum of many exponential terms. Thus, the boundary-fitted coordinates are generated as the solution of

$$\xi_{xx} + \xi_{yy} = -\sum_{i=1}^{n} a_i \, \mathrm{sgn}\,(\xi-\xi_i) \exp\left(-c_i \, |\xi-\xi_i| \right)$$

$$-\sum_{j=1}^{m} b_j \, \mathrm{sgn}\,(\xi-\xi_j) \exp\left(-d_j \left[(\xi-\xi_j)^2+(\eta-\eta_j)^2 \right]^{\frac{1}{2}} \right)$$

$$\equiv P(\xi,\eta) \qquad (4.14a)$$

and

$$\eta_{xx} + \eta_{yy} = -\sum_{i=1}^{n'} a_i \, \mathrm{sgn}\,(\eta-\eta_i) \exp\left(-c_i \, |\eta-\eta_i| \right)$$

$$-\sum_{j=1}^{m'} b_j \, \mathrm{sgn}\,(\eta-\eta_j) \exp\left(-d_j \left[(\xi-\xi_j)^2+(\eta-\eta_j)^2 \right]^{\frac{1}{2}} \right)$$

$$\equiv Q(\xi,\eta) \qquad (4.14b)$$

where the amplitude and decay factors need not be the same for both equations.

A major use of coordinate control can be to concentrate lines in viscous boundary layers near solid surfaces (e.g., Thames (1975), Thompson et al. (1977b) and Thames et al. (1977)). Control is also beneficial for improving the grid size variation and skewness when generating boundary-fitted systems in very irregular geometries.

4.4 Computational procedures

4.4.1 Numerical solution of the grid generation equations

The transformed Poisson equations (4.10) are discretised on a uniformly spaced finite-difference mesh, composed of square grid cells. Since the range of the ξ- and η-coordinates is completely arbitrary, the transformed mesh increments $\Delta\xi$ and $\Delta\eta$ are defined, for convenience, as unity. Letting a discretised function $f(\xi,\eta)$ at $\xi=i\Delta\xi=i$, $\eta=j\Delta\eta=j$ be denoted as $f_{i,j}$, the second-order central-difference approximations of equations (4.10) can be written, after rearrangement, as:

$$x_{i,j} = \Big[\alpha\,(x_{i+1,j}+x_{i-1,j}) - \frac{\beta}{2}\,(x_{i+1,j+1}-x_{i+1,j-1}-x_{i-1,j+1}+x_{i-1,j-1}) + \gamma\,(x_{i,j+1}+x_{i,j-1}) + J^2\,(P_{i,j}x_\xi + Q_{i,j}x_\eta) \Big] \frac{1}{2(\alpha+\gamma)} \qquad (4.15a)$$

and

$$y_{i,j} = \Big[\alpha\,(y_{i+1,j}+y_{i-1,j}) - \frac{\beta}{2}\,(y_{i+1,j+1}-y_{i+1,j-1}-y_{i-1,j+1}+y_{i-1,j-1}) + \gamma\,(y_{i,j+1}+y_{i,j-1}) + J^2\,(P_{i,j}y_\xi + Q_{i,j}y_\eta) \Big] \frac{1}{2(\alpha+\gamma)} \qquad (4.15b)$$

where

$$x_\xi = \tfrac{1}{2}\,(x_{i+1,j}-x_{i-1,j}) \quad ; \quad x_\eta = \tfrac{1}{2}\,(x_{i,j+1}-x_{i,j-1})$$

$$y_\xi = \tfrac{1}{2}\,(y_{i+1,j}-y_{i-1,j}) \quad ; \quad y_\eta = \tfrac{1}{2}\,(y_{i,j+1}-y_{i,j-1})$$

and

$$\alpha = x_\eta^2 + y_\eta^2 \; ; \; \beta = x_\xi x_\eta + y_\xi y_\eta \; ; \; \gamma = x_\xi^2 + y_\xi^2 \; ; \; J = x_\xi y_\eta - x_\eta y_\xi .$$

These equations are solved iteratively by successive-over-relaxation (S.O.R.). Numerical experimentation has revealed that the optimum acceleration parameter lies between 1.75 and 1.85 for most forms of mesh geometry (Thompson et al., 1977b). The iteration is continued until the x- and y-values, at all points in the computational field, change by less than predetermined error parameters, ϵ_x and ϵ_y, respectively; the convergence parameters are defined to be a given fraction of the distance scale of the problem being investigated. In certain cases, with very strong coordinate attraction, the iterative solution fails to converge. This can usually be overcome by adding the control functions, $P_{i,j}$ and $Q_{i,j}$ in stages (either linearly or exponentially), thereby allowing a previously converged mesh with small coordinate attraction to be used as the initial solution for a problem involving stronger attraction-operators.

Häuser et al. (1986a) indicate that the numerical solution of the transformed Poisson equations, using conventional S.O.R. techniques, is not very satisfactory for large meshes because of the associated slow convergence rate. Furthermore, the criteria for ending an iteration may often be reached before an adequate coordinate grid has been produced. This is attributed to the fact that the error components of wavelength 2Δ (where Δ is the grid increment in the transformed mesh) are highly

damped by the numerical scheme, whereas the longer wavelength errors are hardly affected and persist for many iterations. The convergence criteria are based upon the localised, short term grid movements, and are therefore dependent upon the short wavelength errors which are damped early in the iterative procedure. Although it is possible to overcome this difficulty by using smaller error parameters, ϵ_x and ϵ_y, the slow rate of convergence in large coordinate meshes then becomes even more critical. Häuser et al. (1986a) propose that multigrid techniques could be beneficial to the grid generation process. This method solves the transformed Poisson expressions in a similar manner to that already presented in equations (4.15), with the exception that the calculations are performed on a progression of finer grids with mesh spacings of ..., 8Δ, 4Δ, 2Δ, Δ; each successive grid eliminating the errors in a certain frequency range.

In the present study, however, a conventional (single grid) approach was adopted because it was believed that the computational savings would not warrant the extra effort in developing a multigrid Poisson equation solver. Indeed, after completing the numerical scheme for the solution of the shallow water equations (Chapter 6), it was found that the computer C.P.U. times required to generate boundary-fitted coordinate schemes were insignificant compared with the times that were necessary to solve the governing hydrodynamic equations.

4.4.2 Boundary data

Although the boundary conditions shown in equation (4.10) appear complex, the functions $x_A \ldots x_D$ and $y_A \ldots y_D$ are only necessary for mathematical completeness. The transformed Poisson equations are solved numerically and hence the boundary conditions need only be applied in a discretised form; the continuous functions describing the edges of the flow domain are replaced by discrete mappings of boundary coordinates (x_k, y_k) onto the equidistant points along the perimeter of the transformed mesh. Numerical grid generation thus involves setting up an appropriately sized computational grid, specifying the (x,y) coordinates at the nodes along the boundaries of the domain, and finally solving the 'quasi-linear' Poisson equations (4.15) at all points in the interior of the mesh.

Most of the flow domains investigated in this thesis possess regular geometries, enabling the boundary point distribution to be specified analytically. However, in order that the computational model can be extended, in the future, to cater for irregular, real-life flow geometries (such as rivers, estuaries, tidal harbours and

shallow service reservoirs), the coordinate generation code was developed with the option of determining the boundary points via cubic-spline interpolation. This can be considered as a means of numerically defining the perimeter functions $x_A,\ldots, x_D$ and $y_A,\ldots, y_D$. Cubic-splines reduce the amount of work required to prepare irregularly curved shapes for input to the code because the data points which define the perimeter need not be the same as the actual boundary nodes utilised for the curvilinear grid generation. Another advantage of the technique is that coordinate meshes of different resolutions can be produced from the *same* boundary data; this is particularly useful when investigating the effects of cell size on the accuracy of flow predictions. Furthermore, the method allows the perimeter to be specified without having to consider the final form of the curvilinear system. It is therefore possible to decide (by numerical experimentation) on the shape of transformed mesh, *after* the boundary data files have been produced. The preparation of data for regions with irregular curves was found to be improved considerably by the use of a digitising tablet. This allowed accurate and rapid conversion of scale drawings, plans or maps into digitised perimeter coordinates, suitable as cubic-spline data for the boundary-fitted coordinate code.

4.4.3 Cubic-spline interpolation

Consider an arbitrary boundary curve in the physical plane, as shown in Figure 4.4a. The most general form of perimeter is multi-valued in both the x- and y-directions and hence a conventional cubic-spline approach, mapping x onto y (or vice versa), is not directly applicable. To overcome this difficulty, the boundary must first be parameterised according to the distance along the curve, s, i.e.,

$$x = X(s) \qquad \text{and} \qquad y = Y(s) \quad . \tag{4.16}$$

As can be seen from Figures 4.4b and 4.4c, this parameterisation reduces the boundary curve to two separate single-valued functions. Chordwise summation is generally adequate for determining s. Thus, with reference to Figure 4.4a:

$$
\left.
\begin{aligned}
s_0 &= 0 \\
s_1 &= [(x_1-x_0)^2+(y_1-y_0)^2]^{\frac{1}{2}} \\
s_2 &= s_1 + [(x_2-x_1)^2+(y_2-y_1)^2]^{\frac{1}{2}} \\
&\vdots \\
s_n &= s_{n-1} + [(x_n-x_{n-1})^2+(y_n-y_{n-1})^2]^{\frac{1}{2}} \quad .
\end{aligned}
\right\} \qquad (4.17)
$$

The cubic-spline approximations of the functions X and Y are determined in a conventional manner (see for example Conte and de Boor, 1980 or Press et al., 1986), and are then used to generate the exact number of boundary points required for the curvilinear grid.

If the data nodes are sparsely situated along the curve, the chordwise representation can be replaced by an iterative technique which determines the exact curve-wise parameterisation; numerical integration along the cubic-splines is used to calculate better estimates for s_0, s_1,..., s_n, and then these new s-values are utilised in the redetermination of the cubic-spline approximations of the functions, X and Y. This process can be repeated several times, in order to obtain a more accurate representation of the boundary curve.

The need to specify first derivatives at the first and last nodes of a cubic-spline can be fulfilled by defining the direction of the curve in the physical plane (with respect to the x-axis). The required first derivatives in the x-s and y-s planes are therefore evaluated as:

$$
\frac{\partial x}{\partial s} = \cos\theta \qquad \text{and} \qquad \frac{\partial y}{\partial s} = \sin\theta \quad . \qquad (4.18)
$$

4.4.4 Shape of the transformed domain

The discussions and mathematical descriptions, so far, have all assumed that the physical domain is mapped onto a rectangular region in the transformed plane. Greater flexibility can be achieved, however, by allowing the transformed mesh to be of any shape (providing, of course, the edges are straight and coincide with the integer ξ- and η-lines of the finite-difference grid). For example, the geometry of a circular reservoir with inlet and outlet channels immediately calls to mind a transformed plane composed of a central square region (corresponding to the circular portion of the physical reservoir), with smaller rectangular areas attached.

Geometrical arrangements of this type are presented later in the chapter, in Figures 4.8, 4.11 and 4.12. Furthermore, complex coordinate systems (as demonstrated in Figure 4.14) can only be produced if the transformed mesh is able to follow the approximate shape of the physical domain.

The boundary-fitted coordinate program was developed to allow the outline of the transformed mesh be of any shape, as indicated schematically in Figure 4.5. The data required to specify such a perimeter is input to the computer code in the form of (ξ,η) coordinates of the corner points in the transformed plane. An array is then established to indicate whether a particular node is inside, on or outside the perimeter of the domain, in a similar manner to the WESCOR code (Thompson, 1983). Whilst this arrangement is entirely adequate for the purposes of grid generation (since the S.O.R. technique is an explicit method), the array does not readily allow the use of implicit procedures to solve the governing hydrodynamic equations. This is because the boundaries of the transformed domain can be multi-valued with respect to both the ξ- and η-directions. In order to avoid the loss of computational efficiency by repeatedly having to search through the field array for the end points of implicit flow calculations, the boundary data is also arranged in a manner more suitable for use with A.D.I. techniques.

Two further arrays, ADI_ξ and ADI_η are specified; each of these is a two-dimensional array containing the end-point data for implicit 'line-wise' calculations. ADI_ξ stores the positions of the vertical walls of the transformed plane and so is used for ξ-direction releases of the A.D.I. algorithm, whilst ADI_η stores the positions of the horizontal boundaries and is therefore used in η-direction releases. The first subscript of each array is used to denote the line along which the 'opposing' coordinate is constant, and the second subscript accesses the list of boundary positions. With reference to Figure 4.5, the elements of the arrays, for the typical multi-valued lines shown in the diagram are

<u>ξ-direction</u> along η = J = 30 :-

$ADI_\xi(J,1) = 1$
$ADI_\xi(J,2) = 31$
$ADI_\xi(J,3) = 45$
$ADI_\xi(J,4) = 61$
$ADI_\xi(J,5) = 9999$

and $ADI_\xi(J,6....)$ undefined

η-direction along ξ = I = 41 :-

$ADI_\eta(I,1) = 11$

$ADI_\eta(I,2) = 25$

$ADI_\eta(I,3) = 47$

$ADI_\eta(I,4) = 61$

$ADI_\eta(I,5) = 9999$

and $ADI_\eta(I,6....)$ undefined

(9999 is used to denote that there are no more sections of the flow domain along a particular line.)

Although the use of these arrays may seem trivial, and not really warranted for inclusion here, they provide the boundary-fitted coordinate code with complete generality in specifying the outer perimeter of flow region. Since it was envisaged that the program would mainly be used to calculate coordinate systems for single bodied flow domains, it cannot, as yet, deal with interior bodies, such as islands. However, it would be a relatively simple task to extend the computer code to allow obstacles in the flow to be modelled as 'slabs' in the transformed plane.

4.4.5 Initial solution

Section 4.4.1 describes the numerical solution of the transformed Poisson equations, but does not indicate the form of the (x,y) coordinates used at the start of the iteration. As stated by Thompson et al. (1977b), equations (4.15) are not completely linear, and so the initial guess must be sufficiently close to the correct solution for the iterative technique to converge. Following Thompson et al., but with modifications for the irregularly-shaped transformed plane, the grid generation code allows the use of several different forms of starting condition:

(a) Weighted average of two boundary nodes: the x- and y-values at each point in the computational field are calculated as weighted averages of the two boundary coordinates having either the same ξ-index or the same η-index (the weights being defined linearly according to the distance to the boundary in the transformed plane). Referring to Figure 4.6, the weight functions are:

$$\left.\begin{aligned} \alpha(i) &= \frac{IMAX - i}{IMAX-IMIN} \quad ; \quad \beta(i) = \frac{i - IMIN}{IMAX - IMIN} \\ \gamma(j) &= \frac{JMAX - j}{JMAX-JMIN} \quad ; \quad \delta(j) = \frac{j - JMIN}{JMAX - JMIN} \end{aligned}\right\} \qquad (4.19)$$

where, unlike the TOMCAT code, IMIN, IMAX, JMIN and JMAX are functions of the node position (i,j).

Interpolation in the ξ-direction then gives

$$\left.\begin{aligned} x_{i,j} &= \alpha(i)x_{IMIN,j} + \beta(i)x_{IMAX,j} \\ \text{and} \\ y_{i,j} &= \alpha(i)y_{IMIN,j} + \beta(i)y_{IMAX,j} \end{aligned}\right\} \qquad (4.20)$$

whilst interpolation in the η-direction gives

$$\left.\begin{aligned} x_{i,j} &= \gamma(j)x_{i,JMIN} + \delta(j)x_{i,JMAX} \\ \text{and} \\ y_{i,j} &= \gamma(j)y_{i,JMIN} + \delta(j)y_{i,JMAX} \quad . \end{aligned}\right\} \qquad (4.21)$$

(b) Weighted average of four boundary nodes: this is accomplished by averaging the resultant coordinates from the ξ- and η-direction interpolations, shown above. Thus, the starting conditions are evaluated as:

$$\left.\begin{aligned} 2x_{i,j} &= \alpha(i)x_{IMIN,j} + \beta(i)x_{IMAX,j} + \gamma(j)x_{i,JMIN} + \delta(j)x_{i,JMAX} \\ \text{and} \\ 2y_{i,j} &= \alpha(i)y_{IMIN,j} + \beta(i)y_{IMAX,j} + \gamma(j)y_{i,JMIN} + \delta(j)y_{i,JMAX} \end{aligned}\right\} \qquad (4.22)$$

(c) Moment projection: in this case, the initial solution is determined by taking moments of the physical (x,y) coordinates along the boundary of the domain. Thus, the x-coordinate at an interior node (i,j) becomes:

$$x_{i,j} = \frac{\left[\sum_{k=1}^{N} d_{i,j,k}\right]\left[\sum_{k=1}^{N} x_k\right] - \left[\sum_{k=1}^{N} d_{i,j,k}\, x_k\right]}{N \sum_{k=1}^{N} d_{i,j,k}}$$

whilst the y-coordinate is:

$$y_{i,j} = \frac{\left[\sum_{k=1}^{N} d_{i,j,k}\right]\left[\sum_{k=1}^{N} y_k\right] - \left[\sum_{k=1}^{N} d_{i,j,k}\, y_k\right]}{N \sum_{k=1}^{N} d_{i,j,k}} \qquad (4.23)$$

where x_k, y_k are the x- and y-values of the boundary points,

$d_{i,j,k} = [(x_{i,j} - x_k)^2 + (y_{i,j} - y_k)^2]^{\frac{1}{2}}$ and is the distance from the k^{th} boundary node to the point (i,j),

and N is the total number of boundary nodes.

The summations are carried out along the entire perimeter of the transformed plane.

Moment projection produces a very concentrated radial form of coordinate system, centred over the centroid of the boundary nodes.

(d) Exponential projection: the x- and y- values at each point in the computational plane are calculated as:

$$x_{i,j} = \frac{\sum_{k=1}^{N} x_k \exp\left(-|IGES|\, d_{i,j,k}/d_0\right)}{\sum_{k=1}^{N} \exp\left(-|IGES|\, d_{i,j,k}/d_0\right)}$$

and

$$y_{i,j} = \frac{\sum_{k=1}^{N} y_k \exp\left(-|IGES|\, d_{i,j,k}/d_0\right)}{\sum_{k=1}^{N} \exp\left(-|IGES|\, d_{i,j,k}/d_0\right)} \qquad (4.24)$$

where $d_{i,j,k}$ is as before for the moment projection,

d_0 is a 'characteristic' length of the the transformed field (usually taken as the largest diagonal),

and IGES is a parameter to alter the strength of the projection. (Thompson et al. (1977b) used values of IGES in the range -4 to -40.)

Again, as for the moment projection, the summations are evaluated over the entire perimeter.

Numerical experimentation has indicated that the 'four-point weighted average' initial solution produces the fastest convergence for the circular reservoir geometries investigated here.

4.5 Truncation errors

An understanding of the types of truncation error that occur with boundary-fitted schemes is important in order to judge the suitability of a particular coordinate mesh for solving the governing hydrodynamic equations. In particular, any numerically generated grid must be inspected for excessive cell size variation between neighbouring grid points and for excessive deviation from orthogonality (skewness). Furthermore, the mesh must also be checked to ensure that the coordinate lines do not overlap at any location in the flow domain (coordinate overlap is indicated by a negative Jacobian, J). This latter defect is most serious since the mapping between the physical domain and the transformed plane then becomes ambiguous; flow calculations based upon such a mesh would be totally unreliable in the region of overlap because the basic requirements to conserve mass and momentum would be grossly violated. If the Jacobian at any point in the computational field is found to be negative (or very close to zero), then the coordinate grid must be regenerated with either a different distribution of boundary nodes or a different set of attraction-operators in the Poisson equations.

The rejection of a boundary-fitted mesh on the grounds of coordinate overlap is easy to accomplish because of the definite, analytical specification that J must be positive. However, the suitability of a grid with respect to cell size variation and skewness can only really be carried out intuitively since, by necessity, boundary-fitted schemes need these features in order to model irregularly curved perimeters. The truncation errors in the finite-difference approximations of the transformed derivatives (equations 4.6) demonstrate the need to place limits on the cell size variation and grid skewness.

Consider the case of a coordinate mesh where all ξ-lines are normal to the x-direction and all η-lines are normal to the y-direction. Therefore,

$$y_\xi = 0 \qquad \text{and} \qquad J = x_\xi y_\eta \ .$$

Substituting these values into equation (4.6a), gives the first derivative of a scalar quantity, f with respect to x as

$$f_x = \frac{y_\eta f_\xi}{x_\xi y_\eta} = \frac{f_\xi}{x_\xi} \ . \tag{4.25}$$

Taylor series expansions of f_{i+1} and f_{i-1} about i result in

$$f_{i+1} - f_{i-1} = (x_{i+1} - x_{i-1}) f_{x_i} + \frac{1}{2}\left[(x_{i+1} - x_i)^2 - (x_i - x_{i-1})^2 \right] f_{xx_i}$$

$$+ \frac{1}{6}\left[(x_{i+1} - x_i)^3 + (x_i - x_{i-1})^3 \right] f_{xxx_i} + \ldots . \tag{4.26}$$

Hence, the central-difference approximation to the derivative shown in equation (4.25) is

$$f_{x_i} = \frac{f_{i+1} - f_{i-1}}{x_{i+1} - x_{i-1}} + T_i \tag{4.27}$$

where T_i is the local truncation error, found by rearranging equation (4.26)

$$T_i = -\frac{1}{2} f_{xx_i} \left[x_{i+1} - 2x_i + x_{i-1} \right]$$

$$- \frac{1}{6} f_{xxx_i} \left[\frac{(x_{i+1} - x_i)^3 - (x_{i-1} - x_i)^3}{(x_{i+1} - x_{i-1})} \right] + \ldots \tag{4.28}$$

It can be seen that the coefficient of f_{xx} is the central difference representation of $x_{\xi\xi}$. Thus, the first term in T_i produces a numerical diffusive effect which is proportional to the second derivative of the physical x-coordinate. In addition to altering the effective Reynolds number of the flow, the numerical diffusion can also have a destabilising effect on any computational scheme used to solve the governing hydrodynamic equations; this is because the numerical diffusivity, unlike physical viscosity, can be negative. Therefore, the rate of change of mesh spacing (i.e. $x_{\xi\xi}$) must be controlled. This is especially important in regions where f also has significant second derivatives.

In a general two-dimensional system, the rates of change of both the ξ- and η-line spacing are important to the truncation error. Wijbenga (1985a) quotes that

the curvilinear mesh should be satisfactory provided adjacent cell lengths do not vary by more than approximately 20%. The present study indicates that the above figure is a very conservative estimate of the permitted variation in grid spacing; tests conducted later in Chapter 7, using a deliberately distorted mesh to simulate uniform flow conditions in a rectangular channel, show that the mesh increment can double between cells without losing significant accuracy. This is probably attributable to the fact that the computational method, developed in Chapter 6, uses a staggered grid arrangement in which the geometrical derivatives (x_ξ, x_η, y_ξ and y_η) are evaluated by central-difference approximations at *every* flow variable position. This avoids the loss of accuracy in Wijbenga's scheme, caused by the need to average the geometrical derivatives at grid points where they are not explicitly calculated.

Considering the limit on non-orthogonality, the truncation error in equation (4.6a), for f_x can be written as

$$T_x = \frac{1}{J} (y_\eta T_\xi - y_\xi T_\eta) \qquad (4.29)$$

where T_ξ and T_η are the truncation errors in the difference approximations of f_ξ and f_η, respectively. Equation (4.29) assumes implicitly that there are no discretisation errors in the evaluation of y_ξ and y_η. Expressing this equation in terms of the angles of inclination (θ_ξ and θ_η) of the tangents to the localised ξ- and η-directions in the physical plane gives

$$T_x = \frac{1}{\sin(\theta_\eta - \theta_\xi)} \left[\sin\theta_\eta \cos\theta_\xi \frac{T_\xi}{x_\xi} - \sin\theta_\xi \cos\theta_\eta \frac{T_\eta}{x_\eta} \right] . \qquad (4.30)$$

Hence the truncation error due to non-orthogonality varies inversely with the sine of the angle between the two sets of coordinate lines. Thompson et al. (1985) state that the errors arising from grid skewness are of little concern, provided $|\theta_\eta - \theta_\xi| \geq 45^\circ$. As for the case of cell size variation, this figure only serves as a rough indication of the suitability of a coordinate mesh. For example, in order to validate the computational procedures developed in Chapter 6, numerical tests were conducted for wind-driven surface elevation effects in elliptical wedge-shaped basins. It was found that the resultant surface elevation contours agreed with theory, even when the two sets of coordinate lines were less than 15° apart!

4.6 Results

By their very nature, the results from the boundary-fitted coordinate generation program need to be presented in the form of diagrams of the non-orthogonal curvilinear physical meshes and the uniform square-celled transformed meshes which represent the mapping process. Although the transformation is normally regarded as mapping the physical plane onto the computational plane, the actual grid generation process is performed essentially in the opposite direction. Thus, it is more logical to indicate the shape of the transformed domain before the physical geometry, since this is the order of operation for the boundary-fitted code.

Figure 4.7 shows a very coarse boundary-fitted system for a circular basin. The transformed or computational mesh is simply a square grid composed of a lattice of 21 x 21 nodes (20 x 20 cells) of unit spacing. This is mapped onto the curvilinear physical mesh illustrated in Figure 4.7b. The corner points of the computational region are mapped to positions on the perimeter of the circle at angles of $\pm \pi/4$ and $\pm 3\pi/4$ to the x-axis, whilst the remaining boundary nodes are distributed equally around the circle between these four positions. Evident in Figure 4.7b is the tendency for the elliptic grid generation equations to produce an increase in coordinate line spacing in regions of concave boundary curvature. However, the use of attraction-operators is not really warranted in this case, because the aspect ratio of even the most distorted cells is judged to be quite satisfactory at 2:1. This particular mesh was used to study wind-driven surface elevation effects in a 10 km diameter basin, and formed part of a whole series of validation tests for the boundary-fitted shallow water equation program. It was found that the convergence parameters, ϵ_x and ϵ_y, for such a coarse coordinate grid needed to be about 10^{-4} times the diameter of the basin (i.e., for a diameter of 10 km, $\epsilon_x = \epsilon_y = 1$ m). However, for more refined meshes (e.g., 100 x 100 nodes), the convergence parameters had to be reduced by several orders of magnitude, in order to generate an adequate coordinate grid. Without this diminution in ϵ_x and ϵ_y, long wave-length errors persisted at the end of the iteration, with the curvilinear lines still showing some of the characteristics of the initial solution.

The basic form of circular basin is extended in Figure 4.8, to include inlet and outlet channels, which are modelled in the transformed plane as two smaller rectangular regions attached to the central square core. In the physical plane, the parallel sided stems are diametrically opposite each other and subtend an angle of $\pi/15$ radians. This represents a geometry loosely based upon that used by Dennis (1974) for numerically studying low Reynolds number jet-forced flow in a circle.

Dennis, however, did not allow any boundary layer effects at the entrance or exit, and so had no reason to consider inlet or outlet stems. Figure 4.8b illustrates the advantages of boundary-fitted coordinate systems; the combination of circular perimeter and parallel sided channels in this case prevents the accurate use of either a single Cartesian or polar mesh, due to boundary non-alignment.

In Chapter 3, the circular reservoir studies were conducted using effectively a non-dimensionalised form of the Navier-Stokes equations. However, the boundary-fitted schemes discussed here are to be used in conjunction with the shallow water equations, and it is therefore more convenient to express the variables in terms of dimensional quantities. The boundary-fitted circular reservoir schemes are all generated with a basin diameter of 1.5 m, since the present study leads eventually to direct comparison with data presented by Falconer (1976, 1980).

The mesh presented in Figure 4.8a has a square core of 77 x 77 nodes, corresponding to the circular portion of the reservoir. Convergence tests were conducted to investigate the optimum starting conditions for the physical geometry shown in Figure 4.8b. For the purpose of these tests, the over-relaxation parameter, Ω was taken to be 1.70 and the error parameters, ϵ_x and ϵ_y were set to 10^{-5} m. Table 4.1 summarises the results of using different initial solutions; the four-point weighted average guess (Section 4.4.5b) is clearly the most effective because it requires the least number of iterations. Figure 4.9 provides a visual demonstration of the various forms of initial coordinate system. It can be seen that the four-point weighted average guess does indeed have the closest coordinate system to the final converged mesh.

Although the exponential projection initial solution shown in Figure 4.9e suffers from coordinate overlap, the numerical scheme still manages to converge, albeit rather slowly, to the correct (non-overlapping) physical mesh. The non-linear transformed elliptic grid generation equations are therefore able to recover from poor starting conditions. This is important for irregular, real-life domains, because it is not always possible to produce an initial guess of the quality shown in Figure 4.9c.

A second series of tests was then performed to determine the optimum acceleration parameter for the successive-over-relaxation scheme. It was found that the fastest convergence was obtained with $\Omega \approx 1.84$. The tests also revealed a very rapid decrease in the convergence rate above this optimum value. Thus, in cases where the optimum acceleration parameter is unknown, it is generally more economical to underestimate Ω.

An alternative boundary-fitted coordinate scheme for a symmetrical circular reservoir is proposed in Figure 4.10. Here, the physical coordinate grid corresponds to a rectangular transformed mesh with 21 x 101 nodes. Although the computational domain has a regular shape, the physical mesh is considered unsuitable for the solution of the governing hydrodynamic equations. The main objections to this particular boundary-fitted system include:

(a) The grid cells in the main portion of the circular reservoir are too coarse in comparison to the high resolution in the inlet and outlet stems.
(b) The occurrence of large discontinuities in the direction of the η-lines as they emerge from the confines of the passageways.
(c) The excessive cell size variation and grid skewness in the vicinity of the intersections of the circular perimeter and the parallel sided stems.
(d) The mesh is computationally inefficient for solving the hydrodynamic equations because any time step restrictions would be based upon the very small mesh increments in the inlet and outlet channels.

In view of these disadvantages, the numerical solution of the hydrodynamic equations was judged to be better suited to the style of boundary-fitted mesh illustrated in Figure 4.8, where the spatial variations in grid increment and curvilinear coordinate direction are less severe.

Figure 4.11 shows a 93 x 93 node (central core) boundary-fitted mesh representing the type of geometry studied by Mills (1977). The circular reservoir has again been given inlet and outlet channels in order to allow for the presence of boundary layer effects. In this case, the stems subtend an angle of $\pi/16$ radians and their centrelines are separated by $7\pi/8$. The vertices of the central square region of the transformed plane are mapped to positions on the circular perimeter at angles of $23\pi/64$, $49\pi/64$, $-41\pi/64$ and $-15\pi/64$ to the x-axis. This gives the resultant physical coordinate grid an axis of symmetry along the line, $\theta = 9\pi/16$. In order to prevent distortion to the η-lines as they emerge from the inlet and outlet passageways, the rectangular regions in the computational domain, representing the stems, were displaced upwards from the centreline. The optimum position for the stems, in the transformed plane, was found by numerical experimentation. Furthermore, it was also necessary to utilise a non-uniform grid spacing along the circular arcs corresponding to the vertical walls of the computational region. A gradual increase in physical mesh increment, between the corners adjacent to the stems and the corners at the singularities, prevented a sudden expansion of the

η-lines.

During the assignment of physical coordinates to the boundary nodes of the transformed plane, the implicit distortion of grid cells near concave boundaries must also be considered. Without the use of attraction-operators, the elliptic grid generation equations tend to produce a cell aspect ratio of approximately 2:1 in a direction normal to the circular perimeter. This must be taken into account when specifying the distribution of the coordinates along the sides of the inlet and outlet passageways; failure to do so, results in bulbous ξ-lines either emerging or entering the stems.

The final circular reservoir geometry to be considered is shown in Figure 4.12 and represents the arrangement used by Falconer (1976, 1980). As for the previous cases, the diameter is 1.5 m, but the inlet and outlet are specified in terms of their widths rather than the angles subtended at the centre of the circle. A narrow inlet, of width 0.08 m, is diametrically opposite a 0.24 m wide outlet. Numerical experimentation indicated that a more suitable physical mesh could be obtained by increasing the number of ξ-lines in the central portion of the reservoir; thus the transformed mesh presented in Figure 4.12a has a central core of 97 x 105 nodes (96 x 104 cells). In order to obtain an adequate number of cells across the narrow entrance, more η-lines were forced to enter the inlet stem than determined from the ratio of the geometrical features of the reservoir*. Therefore, it was again necessary to use a non-uniform distribution of grid points along the circular perimeter of the physical mesh.

The curvilinear meshes discussed so far, have all been used in conjunction with the boundary-fitted shallow water equation solver, described later in Chapter 6. For these geometries, either analytical, alternative numerical predictions or experimental measurements are available to validate the computer model. However, it is finally worthwhile to demonstrate the possibilities of the boundary-fitted scheme by presenting more complex geometries, typical of real-life flow domains.

* $\theta_{inlet} = 2 \sin^{-1} \left[\frac{b_{inlet}}{2 R_0} \right] = 2 \sin^{-1} \left[\frac{0.08}{2 \times 0.75} \right] = 6.1^{\circ}$

$\therefore$ theoretical number of η-cells across the opening = 6.1 / 90 x 96 = 6.5

(cf., actual no. of cells across inlet in Figure 4.12 is 8)

Figure 4.13 shows a boundary-fitted system representative of the Beauly Firth. It can be seen that the two rivers entering the estuary are modelled as smaller rectangular areas attached to the main region of the transformed plane, in a similar manner to that used for the reservoir geometries. The mesh requires the extensive use of attraction-operators in order to avoid excessive grid skewness in the vicinity of headlands.

An even more complex coordinate system is presented in Figure 4.14, which shows a 12 km x 18 km region around a typical sea loch system. The physical coordinates of the perimeter nodes were generated using the curve-wise cubic-spline technique proposed in Section 4.4.3. It can be seen that the boundaries of the transformed mesh are multi-valued in both the ξ- and η-directions and therefore any hydrodynamic calculations would require the implementation of the algorithm described in Section 4.4.4. The coordinate mesh presented in Figure 4.14b has been generated without the use of control functions and can thus only be regarded as an initial attempt at producing a boundary-fitted grid. Excessive grid skewness near headlands and large cell size variation in the vicinity of strongly curved regions of the boundary must be counteracted by the attraction operators, P and Q. The magnitudes and decay factors of the control functions can only really be determined by numerical experimentation.

Although these final curvilinear coordinate systems were not used for simulating hydrodynamic flow phenomena, they indicate the necessity of incorporating an algorithm into the boundary-fitted code which allows implicit hydrodynamic calculations to be performed on arbitrary-shaped transformed domains.

4.7 Concluding remarks

The mathematical procedures for generating non-orthogonal boundary-fitted coordinate systems, described in this chapter, use a pair of Poisson equations to define the transformation, following the methodology proposed by Thompson et al. (1974 onwards). These expressions are solved numerically, by successive-over-relaxation, to produce the curvilinear system. The essential feature of boundary-fitted grids is that coordinate lines coincide with the perimeter, no matter how irregular the shape of the region. This eliminates the problems associated with using finite-difference techniques to solve partial differential equations on awkwardly shaped domains.

The results demonstrate the advantages of implementing a boundary-fitted coordinate system for the investigation of jet-forced reservoir flow. Geometries that could only previously be modelled using finite-elements can now take advantage of finite-difference techniques. Figures 4.8, 4.11 and 4.12 indicate that curvilinear meshes representative of circular reservoirs are viable as regards cell size variation and grid skewness, and accordingly, it is worthwhile continuing the development of the boundary-fitted scheme. The next stage is to transform the governing Cartesian hydrodynamic equations for use in the computational plane.

Table 4.1 Effect of initial solution on mesh generation convergence rate: symmetrical circular reservoir (Dennis' geometry), 77x77 node central core, $\epsilon_x = \epsilon_y = 10^{-5}$ m, $\Omega = 1.70$

Type of initial guess	No. of iterations for convergence	Figure
Two-point weighted average (ξ-direction)	416	4.9a
Two-point weighted average (η-direction)	393	4.9b
Four-point weighted average	258	4.9c
Moment projection	652	4.9d
Exponential projection (IGES = -40)	569	4.9e
Exponential projection (IGES = -4)	545	4.9f

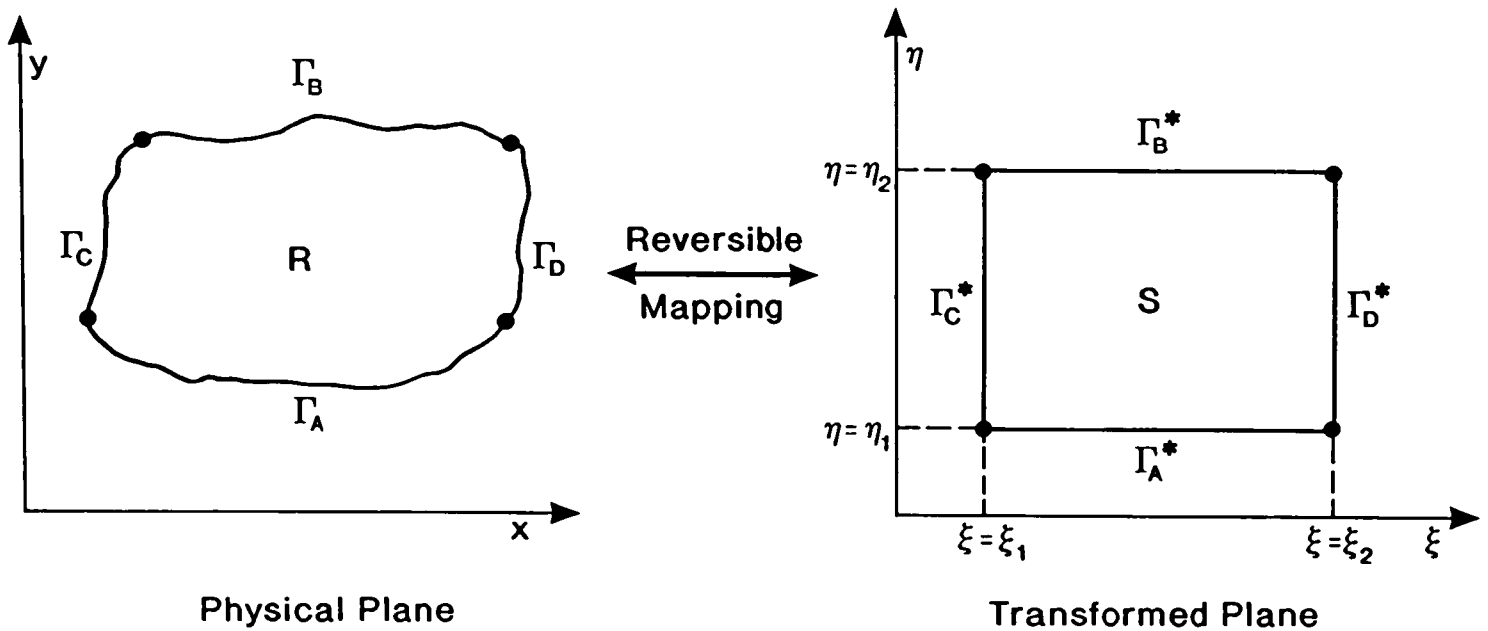


Figure 4.1 Boundary-fitted mapping

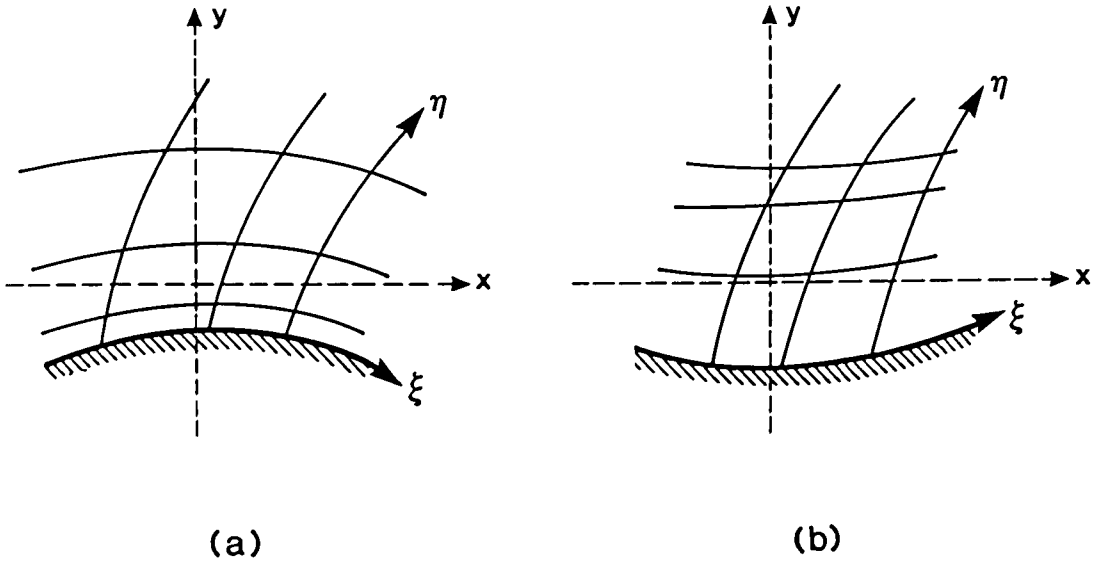


Figure 4.2 Effect of boundary curvature on coordinate line spacing (after Thompson et al., 1985)

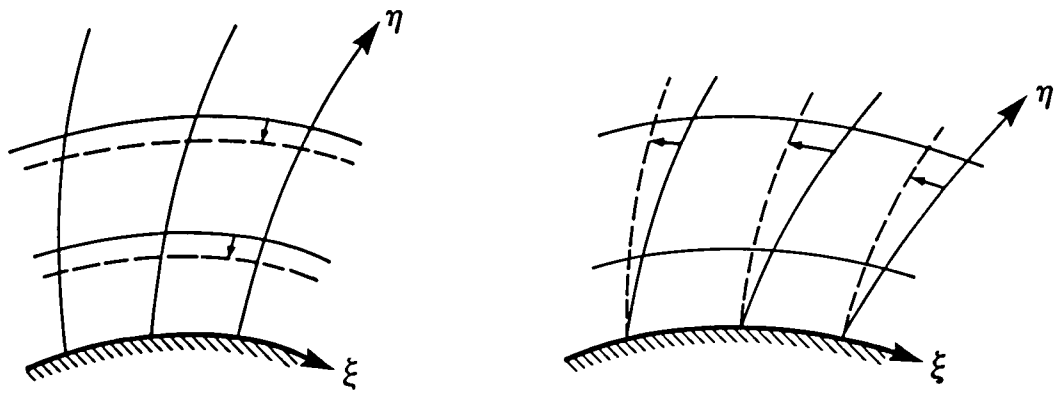


Figure 4.3 Effect of control functions on coordinate line spacing (after Thompson et al., 1985)

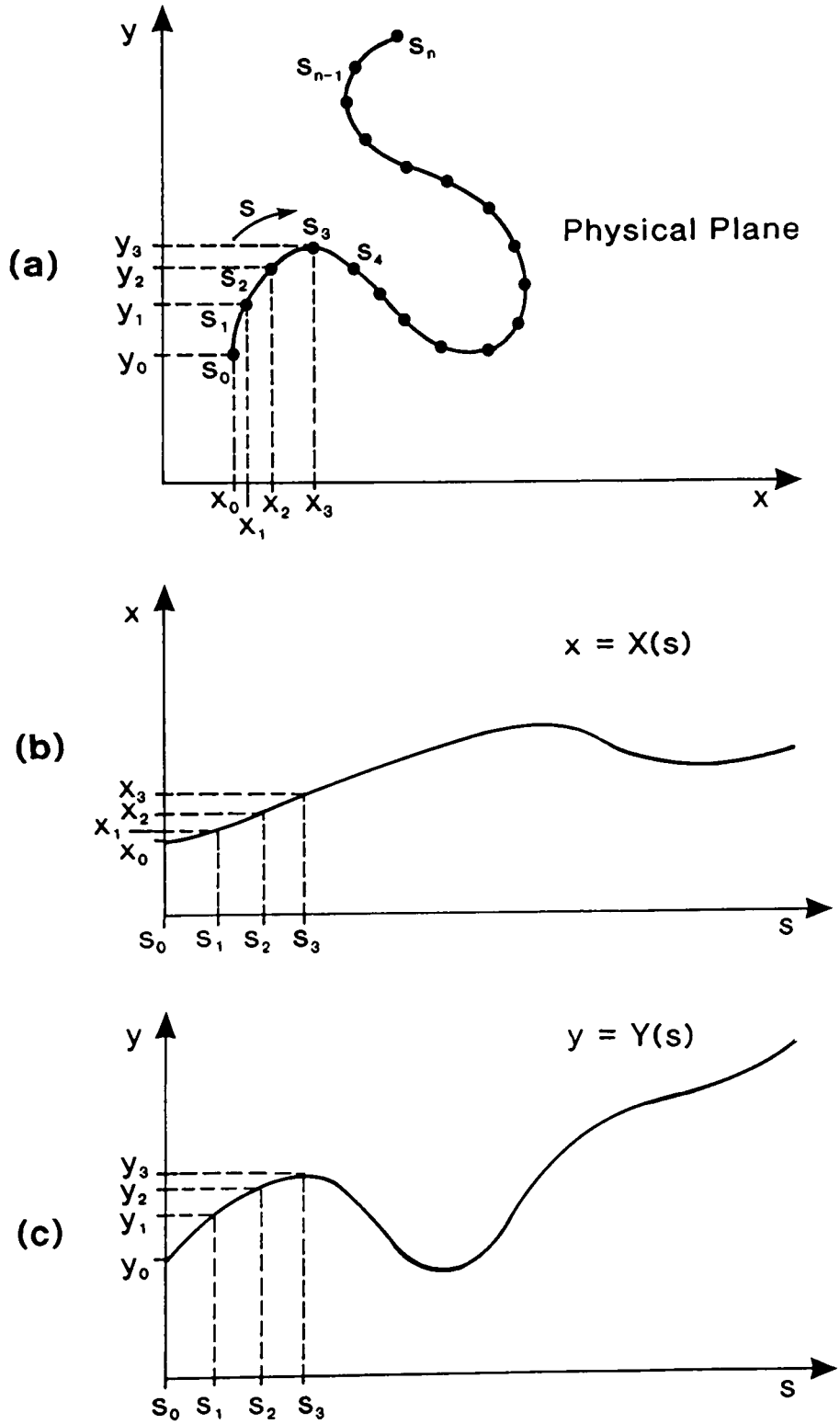


Figure 4.4 Curve-wise parameterisation of an irregular boundary

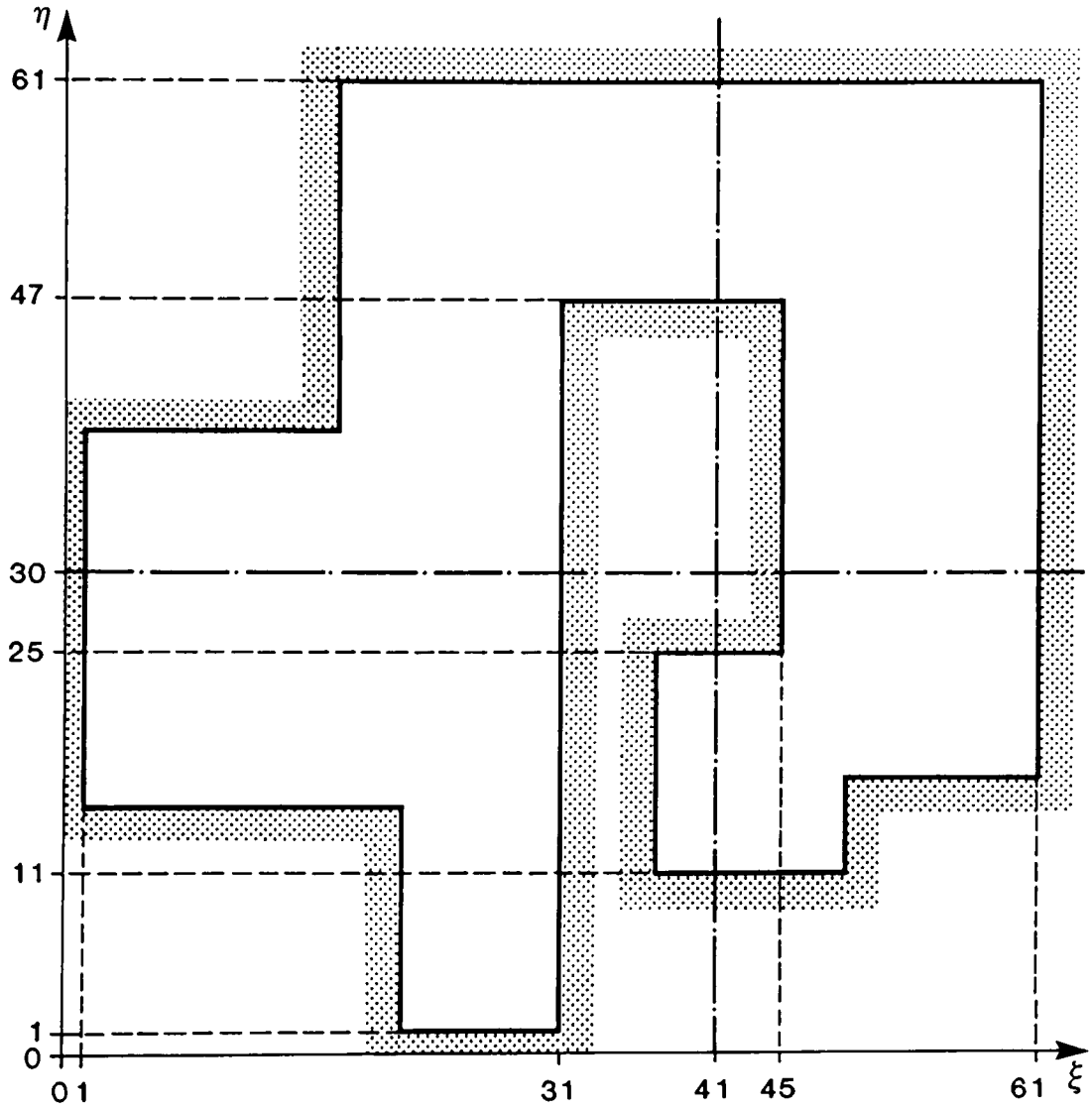
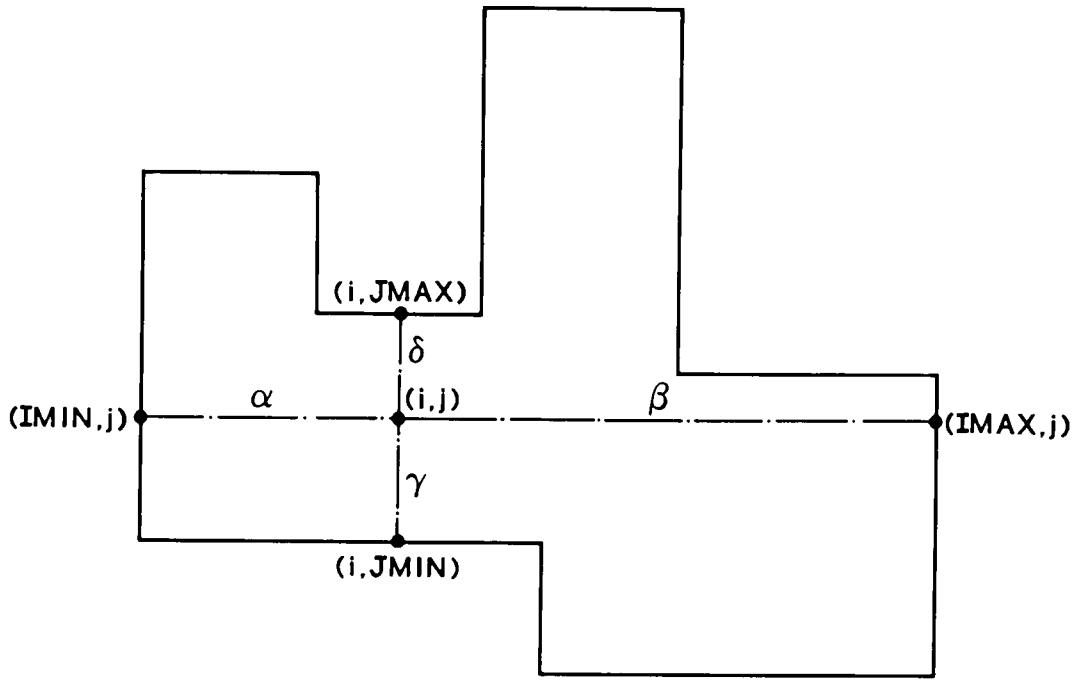


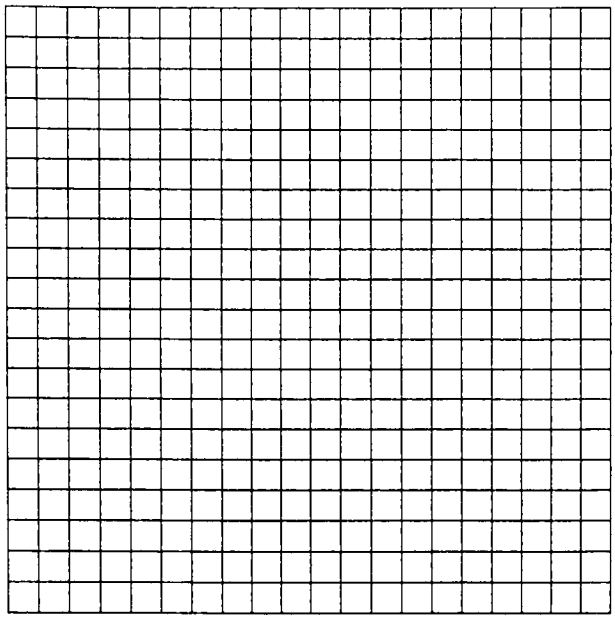
Figure 4.5 Transformed plane: demonstration of end point data stored in arrays ADI_{ξ} and ADI_{η}



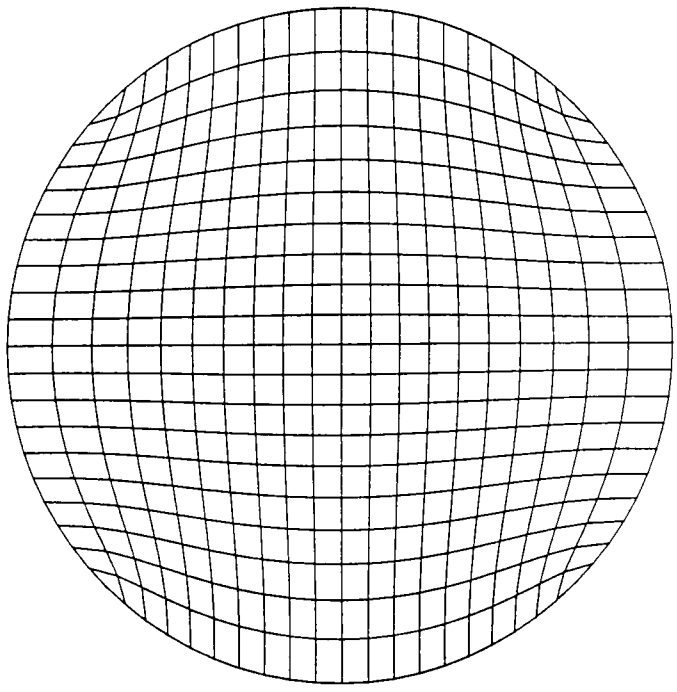
KEY

(i,j)	coordinates of point at which initial guess is required
IMIN, IMAX, JMIN, JMAX	indices of outer boundary of domain [APPERTAINING to the point (i,j)]

Figure 4.6 Weighted average initial guess

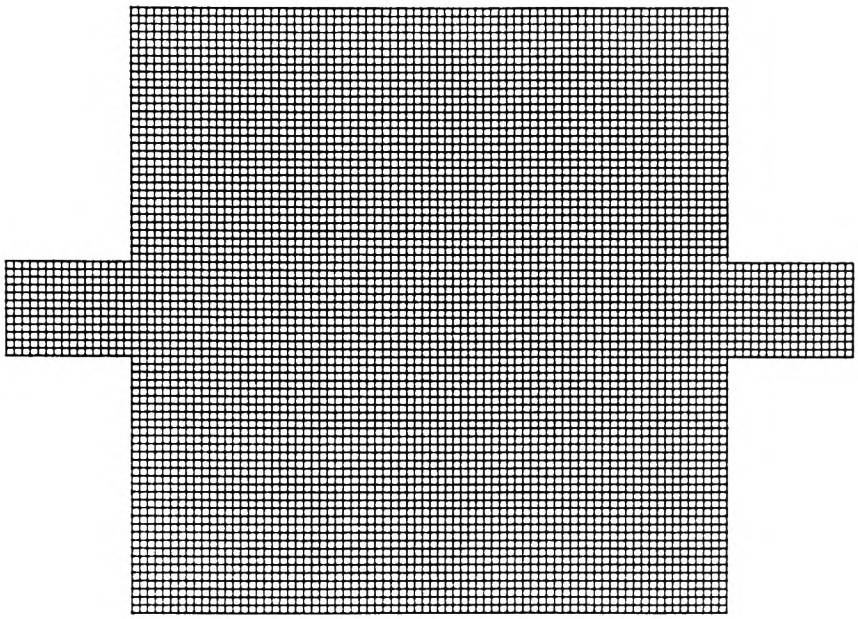


(a) Transformed (ξ, η) plane

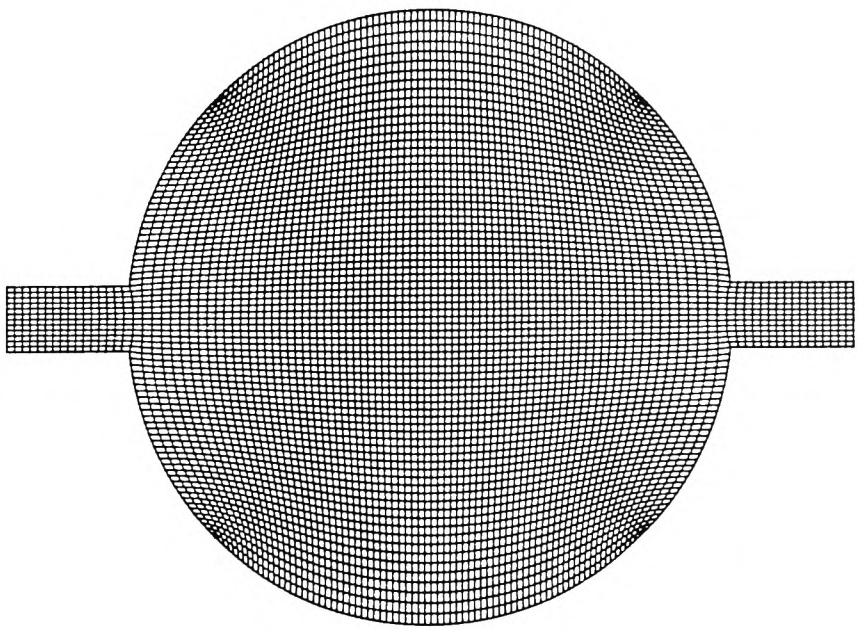


(b) Physical (x,y) plane

Figure 4.7 Boundary-fitted system for a circular basin

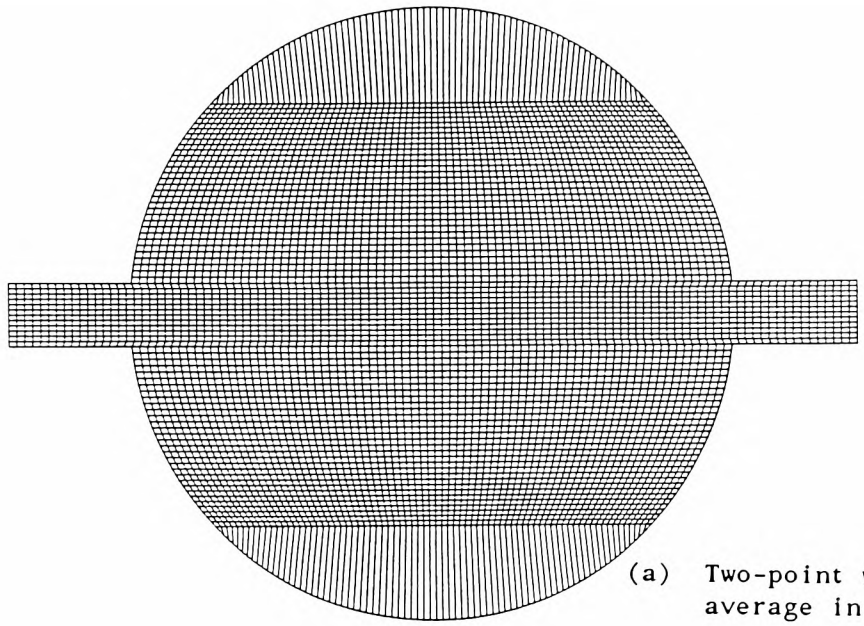


(a) Transformed (ξ, η) plane

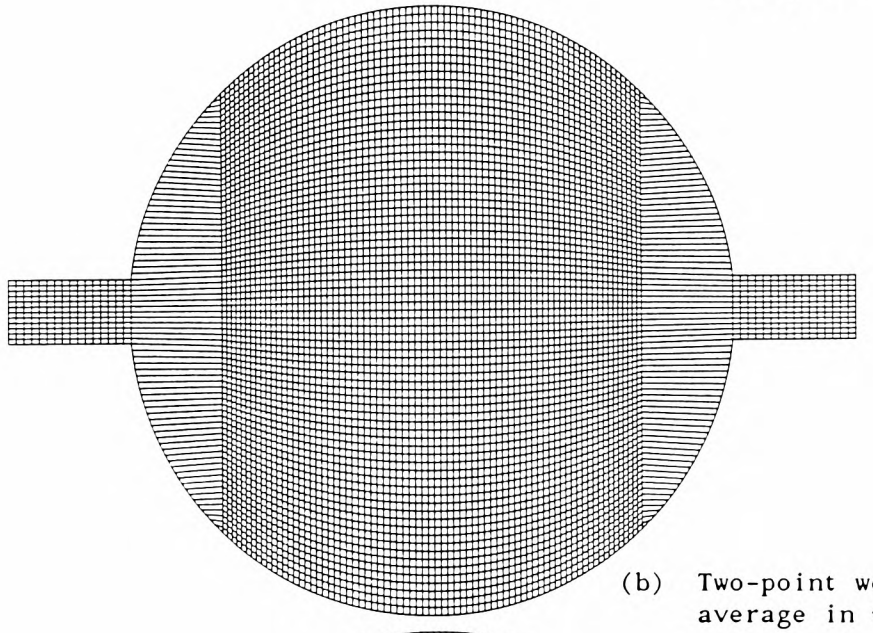


(b) Physical (x,y) plane

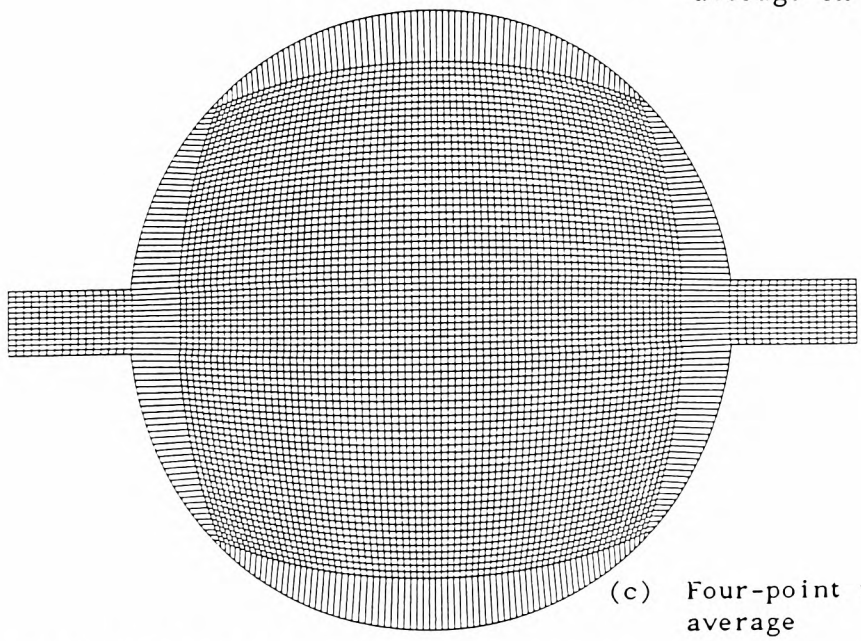
Figure 4.8 Boundary-fitted system for a symmetrical circular reservoir (Dennis' geometry)



(a) Two-point weighted average in ξ-direction

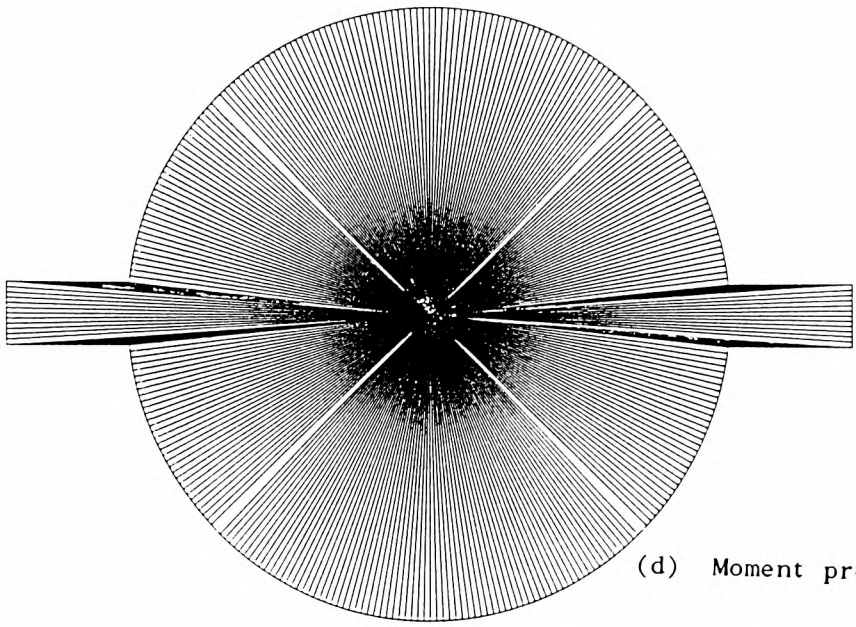


(b) Two-point weighted average in η-direction

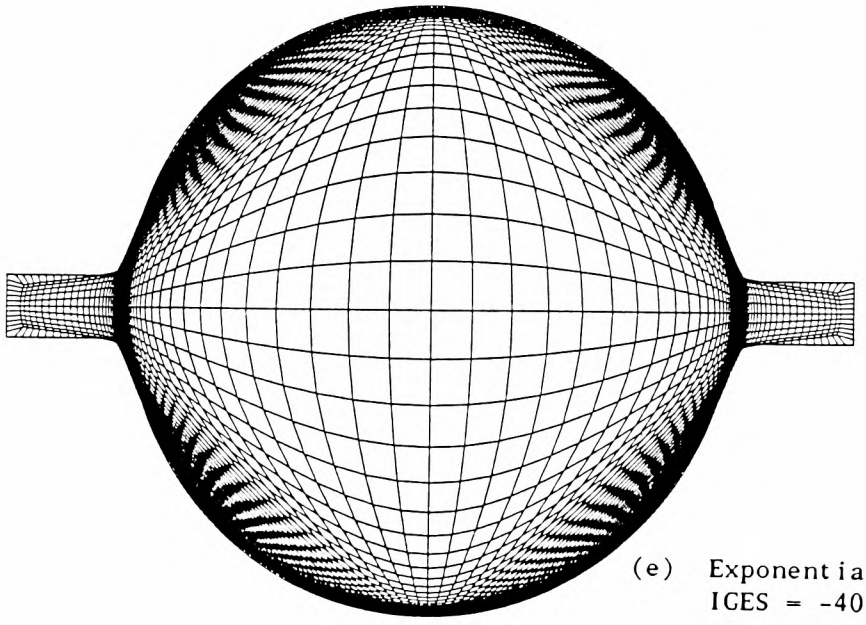


(c) Four-point weighted average

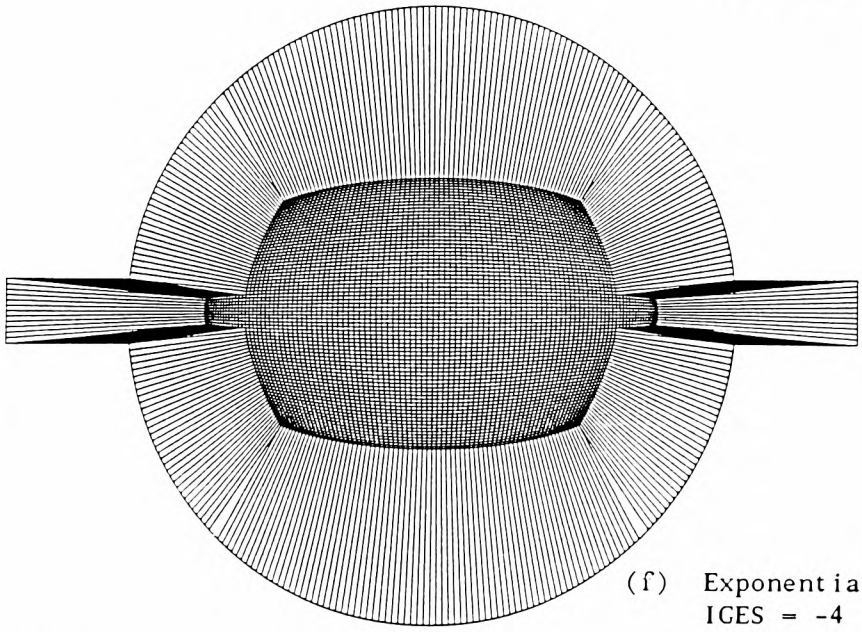
Figure 4.9 Initial solutions for a symmetrical circular reservoir



(d) Moment projection

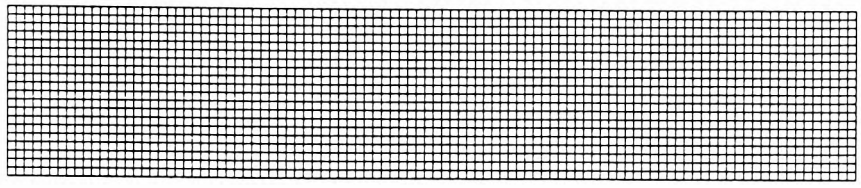


(e) Exponential projection
IGES = -40

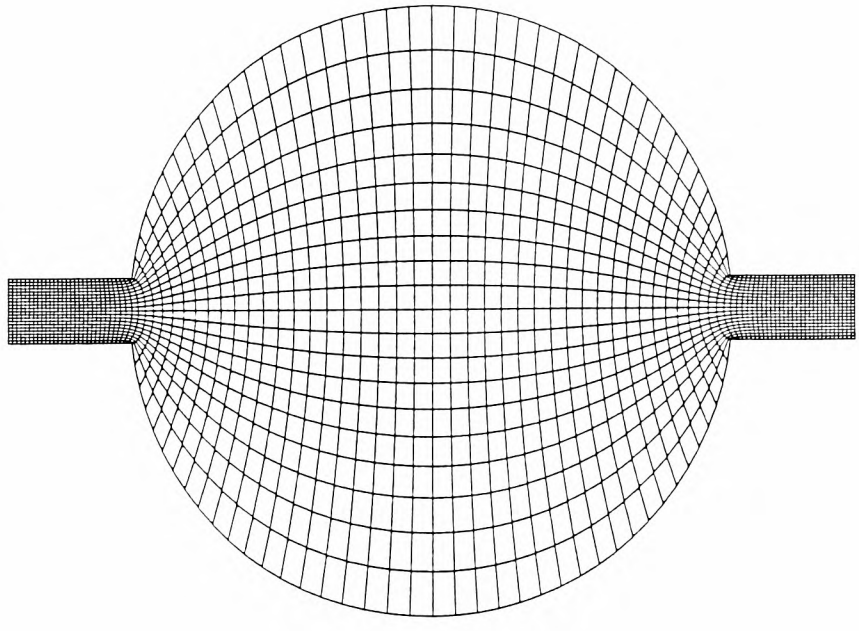


(f) Exponential projection
IGES = -4

Figure 4.9 Continued

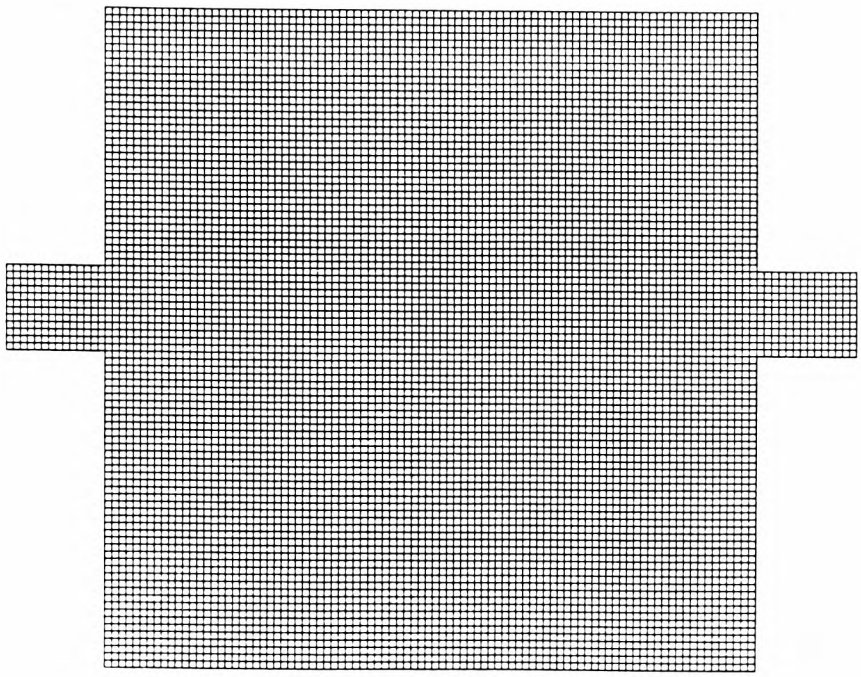


(a) Transformed (ξ, η) plane

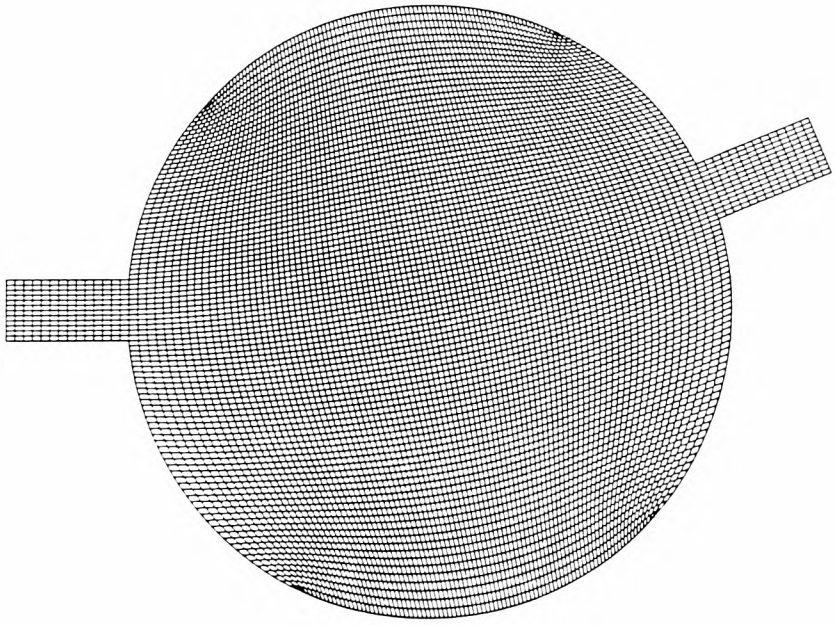


(b) Physical (x,y) plane

Figure 4.10 Alternative boundary-fitted system for a symmetrical circular reservoir

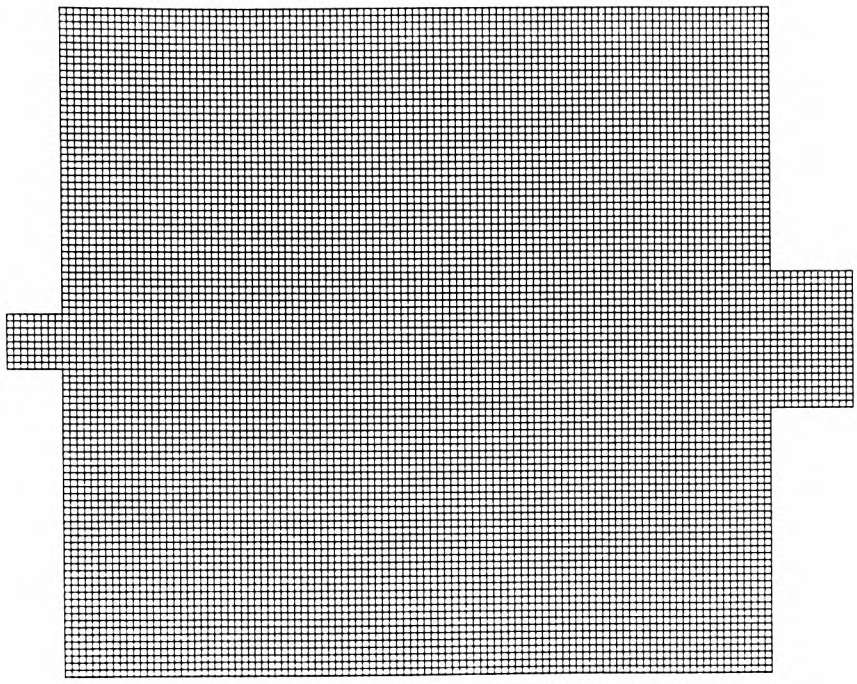


(a) Transformed (ξ, η) plane

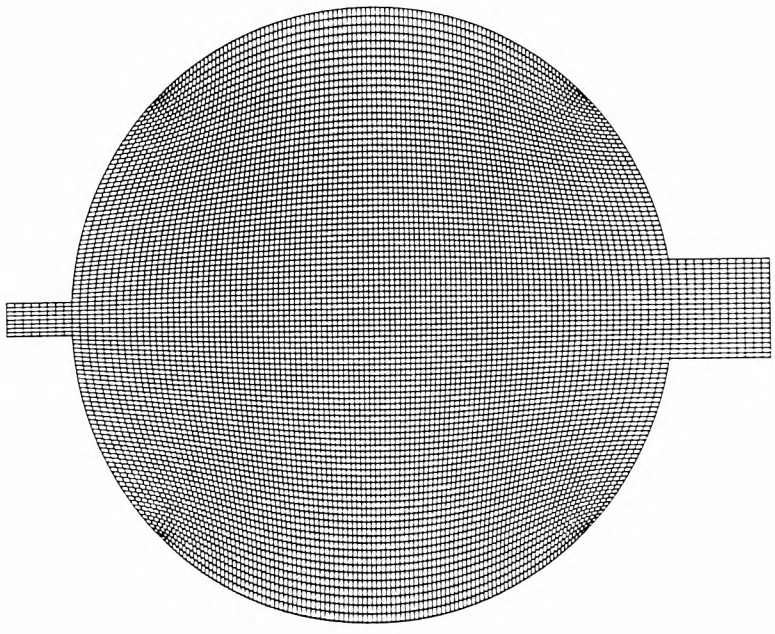


(b) Physical (x,y) plane

Figure 4.11 Boundary-fitted system for Mills' circular reservoir geometry

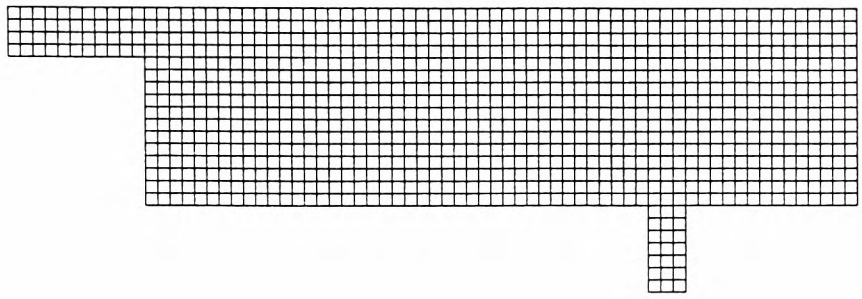


(a) Transformed (ξ, η) plane

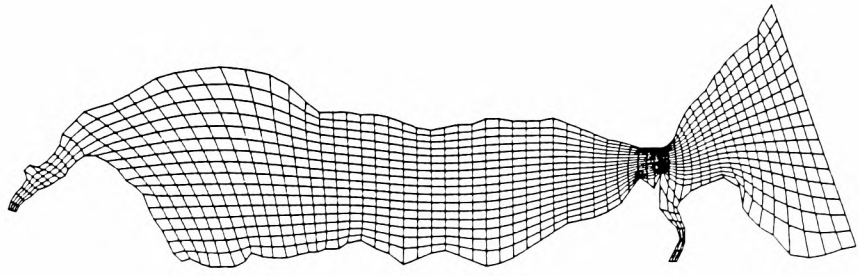


(b) Physical (x,y) plane

Figure 4.12 Boundary-fitted system for Falconer's circular reservoir geometry

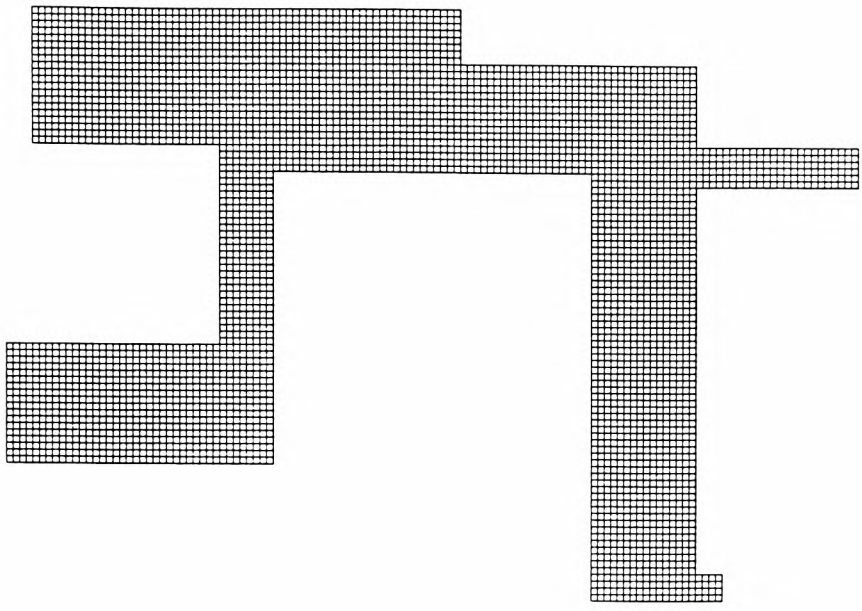


(a) Transformed (ξ, η) plane

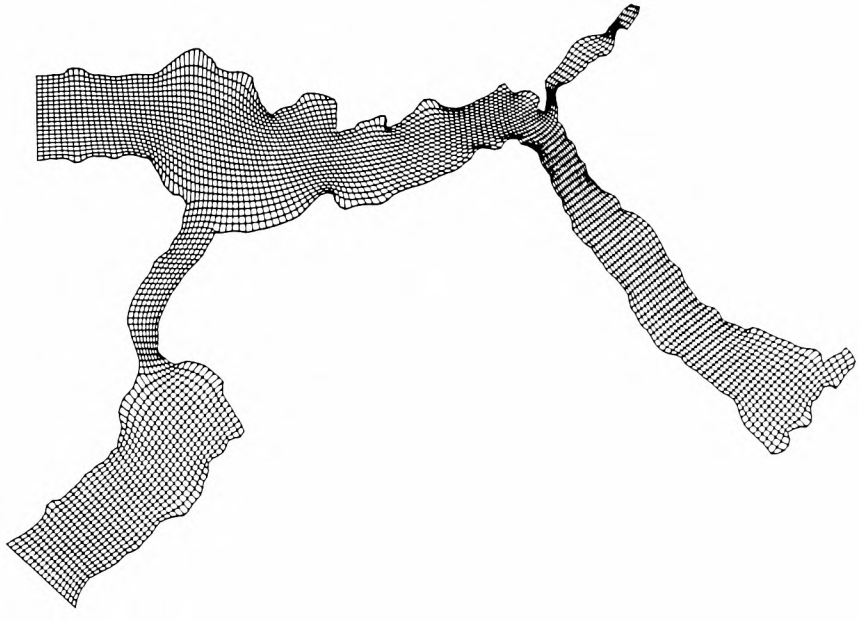


(b) Physical (x,y) plane

Figure 4.13 Boundary-fitted system representative of the Beauly Firth



(a) Transformed (ξ, η) plane



(b) Physical (x,y) plane

Figure 4.14 Boundary-fitted system representative of Loch Duich

CHAPTER 5
DEPTH-AVERAGED EQUATIONS OF MOTION

5.1 Introduction

The numerical simulation of jet-forced reservoir flow described in Chapter 3 utilised a two-dimensional form of the Navier-Stokes equations. Free surface and bed friction effects were ignored and, more importantly, the formulation of the governing hydrodynamic equations assumed a planar flow, with no variation in depth. Although the study provided useful information for validating later reservoir computations, the inability to predict circulation patterns in regions with complex bathymetries severely restricted the practical applications of the model. Whilst it may appear desirable to develop a numerical simulation of the three-dimensional equations of motion, the storage requirements and computing times of such a scheme, are still beyond the reach of present day computers (Rodi, 1984a). So far, three-dimensional models have mainly been applied to 'near-field' problems, which limit the domain to a very localised area around the flow phenomenon of interest. For example, Rastogi and Rodi (1978) investigated the near-field thermal dispersion and buoyancy-induced motion for a discharge of warm water into a rectangular channel. By necessity, the computations only modelled the immediate vicinity around the inlet. Rastogi and Rodi compared the results from their 3-D simulation with predictions from a 2-D depth-averaged procedure and found that the two methods were in agreement for non-smooth beds. This was attributed to strong vertical mixing, caused by the bed roughness, effectively destroying any important velocity and temperature gradients in the vertical direction. Thus, it is not always necessary to resort to three-dimensional hydrodynamic methods, and in many instances, depth-averaged techniques are of sufficient accuracy.

5.2 Stream function/vorticity-transport or primitive equations?

It was decided that the present investigation into momentum-driven flows in shallow circular reservoirs would be better suited to a two-dimensional depth-averaged formulation. Although the vertical motions were neglected, the horizontal flow could then be modelled over larger (or better resolved) regions because the computer limitations on storage capacity and speed of operation would be less prohibitive. Having decided to use a depth-averaged method, the next stage involved choosing the exact format of the governing hydrodynamic equations.

One approach is to utilise a depth-integrated form of the stream function/vorticity-transport (ψ,ω) equations. Codell (1975) employed such a technique for the simulation of flows at power station cooling water intake and outfall structures. The method allows for depth variations and bed resistance, but ignores any free surface effects; the top surface of the flow is assumed to be a horizontal, non-movable plane. A similar approach was used by Ball et al. (1982) to study the resistance effects and subsequent changes in velocity distribution around piled structures in estuaries. Since the surface elevation in a tidal regime is continuously changing, Ball et al. could only model the instantaneous flow at particular states of the tide. Nevertheless, the stream function/vorticity-transport expressions have a number of advantages over other formulations.

It is easier for the stream function method to conserve mass since the flow domain is bounded by constant values of ψ along all solid boundaries. Primitive variable approaches, however, are less able to maintain global mass conservation, because of the cumulative effects of discretisation errors in the finite-difference approximations of the continuity equation. Furthermore, the (ψ,ω) system can be solved on a non-staggered finite-difference grid, eliminating the need to construct a *double* mesh arrangement. This is important in boundary-fitted schemes because the physical (x,y) coordinates of the grid points and geometrical derivatives (x_ξ, x_η, y_ξ and y_η) need to be stored in field arrays; the use of a non-staggered mesh effectively reduces the storage requirements of the coordinate system by a factor of four. A less well defined advantage of the (ψ,ω) approach, applicable to boundary-fitted coordinate systems, is that the numerical solution of the hydrodynamic equations will be less likely to depend on the direction of the curvilinear coordinate lines, because stream function and vorticity are both scalar quantities*. Primitive variable expressions, on the other hand, utilise velocities and therefore boundary-fitted schemes using such formulations should really be written in terms of either the contravariant or covariant velocity components, which follow the directions of the coordinate lines (Sheng and Hirsh, 1984; Sheng, 1986). However, the diffusive terms in such systems are unwieldy and hence the most commonly favoured method for the solution of the primitive equations on curvilinear meshes involves the use of Cartesian velocity components, U and V (Johnson, 1980 and 1982; Johnson et al., 1982; Häuser et al., 1985 and 1986b; Raghunath et al., 1987). Although there does not appear to be any literature to prove that this latter

* Vorticity is strictly a vector but in a two-dimensional system is always perpendicular to the flow plane and can therefore be represented by a scalar.

technique is unsatisfactory, there does seem to be an apparent paradox in using orthogonal velocity directions to solve hydrodynamic equations on non-orthogonal coordinate meshes!

Despite the various advantages of the stream function/vorticity-transport equations, it was decided that a primitive variable method (based upon Cartesian velocity components) would give the numerical model greater flexibility. The main drawback with the (ψ,ω) technique is that it cannot be used to simulate free surface effects and is therefore not readily applicable to situations with time variations in water level. Examples where it would be beneficial to include free surface effects include simulations in estuaries, harbours, tidal river reaches and possibly in service reservoirs which are subjected to water level changes caused by diurnal variations in demand. Furthermore, the stream function/vorticity-transport method has difficulty dealing with flows around internal obstacles, such as islands, because the values of stream function along all solid boundaries must be stipulated a priori. For primitive variable approaches, however, the flow divides itself automatically around the object(s) in accordance with the hydraulics of the situation. In view of the foregoing, and bearing in mind possible future alternative uses of the computer model, it was judged that the depth-averaged Reynolds equations (otherwise known as the shallow water equations) would be better suited to the investigation of reservoir circulation.

Early numerical solutions of the shallow water equations neglected the advective terms in the momentum expressions (Hansen, 1956 and 1962; Reid and Bodine, 1968; Prandle and Crookshank, 1974). This was satisfactory because the computer models considered open seas or wide estuaries and bays, where the flow fields were fairly uniform. Generally, the dominant influences on these numerical simulations came from such features as wind stresses, bed friction, oscillatory surface elevation boundary conditions or the gravitational tide generating forces. Leendertse (1967), in a study of estuaries and seas, found that the non-linear advective terms could be incorporated into the computational scheme without causing any destabilising effects. Whilst this can be attributed partly to the extra stability of the semi-implicit finite-differencing technique proposed by Leendertse, it is also attributable to the fact that the hydraulic conditions modelled were again fairly uniform.

Other studies utilising implicit techniques (e.g., Kuipers and Vreugdenhil, 1973; Vreugdenhil, 1973; Butler, 1978a) found that the advective accelerations could only be used in the simulation provided diffusive terms were also employed. This is because the inclusion of an eddy viscosity tends to control the growth of grid-scale

oscillations in the velocity field (often referred to as 'noodling' or 'flutter' - Hodgins (1977)). The non-linear instabilities are caused by the advective terms transferring turbulence energy from large scale motions through a progression of smaller eddies to eventual dissipation by viscous effects (the so-called energy cascade). The discretisation process, however, prevents the transfer of energy beyond the resolution of the computational mesh and therefore energy accumulates at a wavelength of 2Δ (where Δ is the cell increment). This can eventually lead to an explosive growth of oscillations which completely overwhelms the true numerical solution. Generally, the destabilising effects of the advective acceleration terms are most troublesome when modelling hydraulic regimes with sudden expansions in the cross-sectional area of flow. This is because rapid changes in cross-section usually result in the hydrodynamic equations of motion being dominated by the acceleration components.

Whilst Kuipers and Vreugdenhil, and Butler merely increased the eddy viscosity to a value which suppressed the instabilities, this is erroneous for the numerical solution of jet-forced reservoir flow. Chapter 3 has demonstrated the importance of the viscosity coefficient on the circulation patterns, and therefore the level of diffusion in the computational scheme must be governed by the physics of the situation rather than the requirement to control instability. Falconer (1976) found it necessary to resort to first order upwind differencing of the advective terms in order to achieve stable finite-difference solutions of momentum-driven reservoir flow. For the current investigation, the second order upwind differencing technique proposed by Stelling (1983) is adapted for the purposes of the non-orthogonal boundary-fitted coordinate system. Unlike Falconer's approach, this latter method has the advantage that it can eliminate grid scale oscillations in the velocity field without introducing any appreciable hidden numerical diffusion into the computational scheme.

Kuipers and Vreugdenhil (1973) demonstrated numerically that recirculation regions could only be generated by including the advective accelerations. Thus, any numerical investigation into momentum-driven reservoir flow *must* include the non-linear advective terms, even though they are likely to produce numerical difficulties. Furthermore, Flokstra (1977) used an analytical vorticity balance to prove that secondary circulation patterns could only be set up if the governing hydrodynamic equations contained the so-called effective stress terms. These account for the viscous and turbulent (Reynolds) stresses, as well as the momentum transfer contributions caused by the depth-averaging process (often referred to as *dispersive* terms). Lean and Weare (1979) verified Flokstra's proposal by performing numerical tests on a rectangular channel with a single breakwater extending perpendicularly into

the flow. They concluded that, as well as requiring advective acceleration and effective stress terms, the modelling of momentum transfer also needed a no-slip constraint at the lateral boundaries for the production of recirculation; the application of a perfect slip boundary leads to a flow without any vorticity and therefore without any recirculation. However, Lean and Weare proved that it was possible to produce spurious recirculation behind the breakwater without the inclusion of effective stress terms (contrary to Flokstra's supposition). They attributed this phenomenon to numerical diffusion and found that it was only troublesome in very coarse finite-difference grids; the use of finer meshes reduced the amount of false recirculation. Therefore, in a practical application, the computational grid must have sufficient resolution to ensure that the numerical diffusion is small compared to the physical momentum mixing processes which are being simulated.

From the above discussion, it can be seen that the numerical investigation of reservoir flow must consider the complete non-linear shallow water equations in order to achieve a satisfactory simulation of the hydrodynamic conditions.

The most often quoted disadvantage with the shallow water equations (Wind and Vreugdenhil, 1986; Officier et al., 1986; Bernard, 1989c) is the severe time step restriction caused by the implementation of the Courant stability criterion. For a one-dimensional explicit computational scheme, this takes the form:

$$\sigma = \frac{(gD)^{\frac{1}{2}} \Delta t}{\Delta x} \leq 1 \qquad (5.1)$$

where

σ = Courant number,
g = acceleration due to gravity,
D = local water depth
Δt = time step,
and Δx = mesh increment.

Although there is no theoretical limit on the Courant number for implicit flow calculations, in practice, the non-linear advective acceleration terms and boundary conditions prevent the time step being more than about 3 to 5 times the value quoted above. Furthermore, even though a numerical scheme may be stable with very large time increments, it is likely that free surface wave propagation will be poorly represented as regards amplitude and phase (see Benque et al., 1982). Hence, it is undesirable to use a time step which is significantly larger than the limit prescribed in equation (5.1).

In many flow regimes, the displacement of the free surface from its initial (at rest) state is small, and hence it is possible to replace the free surface calculation by a frictionless lid without loss of accuracy. Indeed, this was found to be the case with the steady state jet-forced reservoir flows investigated later. *Rigid-lid* techniques (McGuirk and Rodi, 1978; Officier et al., 1986; Wind and Vreugdenhil, 1986) use the pressure head $P/\rho g$ exerted on the lid as an approximation of the actual surface elevation. Since the primary computational algorithms developed in this study were concerned with the solution of the free surface shallow water equations, it was more convenient to express the rigid-lid approximation in terms of a head of water, rather than as a pressure, P. In practice, the method updates the values of surface elevation (used in the depth-averaged momentum expressions) at the end of each computational time step but the total depth at each grid point (used for the continuity equation) remains unmodified. This procedure excludes any long wavelength oscillations from the flow solution, and so not only alleviates the Courant time step restriction but may also reduce the time taken to reach steady state; the frictionless lid technique effectively converts the hyperbolic shallow water equations into elliptic Navier-Stokes equations.

The computational scheme, developed in Chapter 6 to solve the shallow water equations on boundary-fitted coordinate systems, includes the option of specifying a rigid-lid approximation. However, this technique must only be employed when the fluctuations in the water level are expected to be small in comparison to the total depth of flow; with large temporal variations in water elevation, it is necessary to resort to the complete free surface computations.

5.3 Derivation of the depth-averaged Reynolds equations (the shallow water equations)

Examination of the literature concerning the development of the two-dimensional shallow water equations reveals a surprising variety of different formulae for the governing Cartesian hydrodynamic equations. Accordingly, it is beneficial to derive the shallow water expressions used in the present investigation. This not only allows an appreciation of the assumptions used in the depth-integration process, but also indicates how alternative versions of the equations are obtained.

The derivation of the depth-averaged equations of motion is accomplished by integrating the three-dimensional Reynolds expressions over the total depth of flow. Consider the reference frame illustrated in Figure 5.1, which is stationary with

respect to the Earth's surface. The conservation of momentum in each of the three Cartesian coordinate directions leads to the standard Reynolds equations:

$$\frac{\partial \bar{u}}{\partial t} + \bar{u}\frac{\partial \bar{u}}{\partial x} + \bar{v}\frac{\partial \bar{u}}{\partial y} + \bar{w}\frac{\partial \bar{u}}{\partial z} = f\,\bar{v} - \frac{1}{\rho}\frac{\partial \bar{P}}{\partial x} + \frac{1}{\rho}\left[\frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z}\right] \qquad (5.2a)$$

$$\frac{\partial \bar{v}}{\partial t} + \bar{u}\frac{\partial \bar{v}}{\partial x} + \bar{v}\frac{\partial \bar{v}}{\partial y} + \bar{w}\frac{\partial \bar{v}}{\partial z} = -f\,\bar{u} - \frac{1}{\rho}\frac{\partial \bar{P}}{\partial y} + \frac{1}{\rho}\left[\frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} + \frac{\partial \tau_{zy}}{\partial z}\right] \qquad (5.2b)$$

$$\frac{\partial \bar{w}}{\partial t} + \bar{u}\frac{\partial \bar{w}}{\partial x} + \bar{v}\frac{\partial \bar{w}}{\partial y} + \bar{w}\frac{\partial \bar{w}}{\partial z} = -g - \frac{1}{\rho}\frac{\partial \bar{P}}{\partial z} + \frac{1}{\rho}\left[\frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \tau_{zz}}{\partial z}\right] \qquad (5.2c)$$

where

x,y,z = Cartesian coordinates of a right handed orthogonal system, with z vertically upwards,

$\bar{u},\bar{v},\bar{w}$ = time-averaged velocity components in the x-, y- and z-directions respectively,

t = time,

$\bar{P}$ = time-averaged pressure,

g = acceleration due to gravity,

ρ = fluid density,

f = Coriolis parameter = $2\omega\sin\varphi$ (where ω = angular velocity of the Earth's rotation and φ = geographical latitude),

and τ_{ij} = fluid deviatoric stresses acting on the plane i = constant, in the j-direction.

The fluid stresses are given by:

$$
\left.
\begin{aligned}
\tau_{xx} &= 2\mu\frac{\partial\bar{u}}{\partial x} - \rho\overline{u'^2} \\
\tau_{xy} = \tau_{yx} &= \mu\left[\frac{\partial\bar{u}}{\partial y} + \frac{\partial\bar{v}}{\partial x}\right] - \rho\overline{u'v'} \\
\tau_{xz} = \tau_{zx} &= \mu\left[\frac{\partial\bar{u}}{\partial z} + \frac{\partial\bar{w}}{\partial x}\right] - \rho\overline{u'w'} \\
\tau_{yy} &= 2\mu\frac{\partial\bar{v}}{\partial y} - \rho\overline{v'^2} \\
\tau_{yz} = \tau_{zy} &= \mu\left[\frac{\partial\bar{v}}{\partial z} + \frac{\partial\bar{w}}{\partial y}\right] - \rho\overline{v'w'} \\
\tau_{zz} &= 2\mu\frac{\partial\bar{w}}{\partial z} - \rho\overline{w'^2}
\end{aligned}
\right\} \qquad (5.3)
$$

where

μ is the coefficient of dynamic viscosity,

and u',v',w' are the fluctuating turbulent velocity components.

The first term in each of the above equations arises from viscous effects, whilst the second term is a Reynolds stress and is caused by the presence of turbulence.

The fluid is assumed to be incompressible and so the continuity equation becomes:

$$\frac{\partial\bar{u}}{\partial x} + \frac{\partial\bar{v}}{\partial y} + \frac{\partial\bar{w}}{\partial z} = 0 \quad . \qquad (5.4)$$

By adding the continuity equation multiplied by $\bar{u}$, $\bar{v}$ and $\bar{w}$ respectively, equations (5.2) can be rewritten as

$$\frac{\partial\bar{u}}{\partial t} + \frac{\partial(\bar{u}^2)}{\partial x} + \frac{\partial(\bar{u}.\bar{v})}{\partial y} + \frac{\partial(\bar{u}.\bar{w})}{\partial z} = f\,\bar{v} - \frac{1}{\rho}\frac{\partial\bar{P}}{\partial x} + \frac{1}{\rho}\left[\frac{\partial\tau_{xx}}{\partial x} + \frac{\partial\tau_{yx}}{\partial y} + \frac{\partial\tau_{zx}}{\partial z}\right] \qquad (5.5a)$$

$$\frac{\partial\bar{v}}{\partial t} + \frac{\partial(\bar{v}.\bar{u})}{\partial x} + \frac{\partial(\bar{v}^2)}{\partial y} + \frac{\partial(\bar{v}.\bar{w})}{\partial z} = -f\,\bar{u} - \frac{1}{\rho}\frac{\partial\bar{P}}{\partial y} + \frac{1}{\rho}\left[\frac{\partial\tau_{xy}}{\partial x} + \frac{\partial\tau_{yy}}{\partial y} + \frac{\partial\tau_{zy}}{\partial z}\right] \qquad (5.5b)$$

$$\frac{\partial\bar{w}}{\partial t} + \frac{\partial(\bar{w}.\bar{u})}{\partial x} + \frac{\partial(\bar{w}.\bar{v})}{\partial y} + \frac{\partial(\bar{w}^2)}{\partial z} = -g - \frac{1}{\rho}\frac{\partial\bar{P}}{\partial z} + \frac{1}{\rho}\left[\frac{\partial\tau_{xz}}{\partial x} + \frac{\partial\tau_{yz}}{\partial y} + \frac{\partial\tau_{zz}}{\partial z}\right] . \qquad (5.5c)$$

The Reynolds equations, presented above, are now in a form more suitable for the depth-integration procedure. Since the depth-averaged technique is used for modelling predominantly horizontal flows, the vertical accelerations due to the hydrodynamic processes are assumed to be small in comparison with the acceleration due to gravity. Furthermore, the shear stresses acting in the z-direction (equation 5.5c) are neglected; this results in a hydrostatic pressure distribution:

$$0 = - g - \frac{1}{\rho} \frac{\partial \bar{P}}{\partial z} \quad . \tag{5.6}$$

The flow conditions are defined in Figure 5.1:-

ζ = distance between the still water level (S.W.L.) and the free surface, measured in the z-direction,

and h = distance between the bed and S.W.L.

Therefore, integrating equation (5.6) with respect to z gives the linear pressure distribution with depth, i.e.

$$\bar{P}(\zeta) - \bar{P}(z) = - \int_{z}^{\zeta} \rho g \, dz = - \rho g(\zeta - z) \quad , \tag{5.7}$$

but the pressure at the free surface is equal to the atmospheric pressure, P_a. Hence,

$$\bar{P}(z) = \rho g(\zeta - z) + P_a \quad . \tag{5.8}$$

The gradients of the pressure in the x- and y-directions of equation (5.5) are replaced by derivatives involving the water elevation, ζ. Thus:

$$\frac{\partial \bar{P}}{\partial x} = \rho g \frac{\partial \zeta}{\partial x} + \frac{\partial P_a}{\partial x} \tag{5.9a}$$

and

$$\frac{\partial \bar{P}}{\partial y} = \rho g \frac{\partial \zeta}{\partial y} + \frac{\partial P_a}{\partial y} \quad . \tag{5.9b}$$

For small scale simulations, the barometric gradients shown in these equations can be neglected because they are insignificant compared to the surface elevation derivatives.

Before beginning the depth-integration process, it is necessary to consider expressions for the kinematic free surface and bed boundary conditions:

(a) Free surface boundary condition: This requires the vertical velocity at the free surface to be equivalent to the total rate of change of surface elevation, i.e.

$$\bar{w}_\zeta = \frac{d\zeta}{dt} = \frac{\partial \zeta}{\partial t} + \bar{u}_\zeta \frac{\partial \zeta}{\partial x} + \bar{v}_\zeta \frac{\partial \zeta}{\partial y} \tag{5.10}$$

where $\bar{u}_\zeta$, $\bar{v}_\zeta$ and $\bar{w}_\zeta$ are the velocity components at the free surface.

(b) Bed boundary condition: Fluid cannot flow through the bed and so

$$\bar{w}_{-h} = \frac{\partial(-h)}{\partial t} + \bar{u}_{-h} \frac{\partial(-h)}{\partial x} + \bar{v}_{-h} \frac{\partial(-h)}{\partial y} \tag{5.11}$$

where $\bar{u}_{-h}$, $\bar{v}_{-h}$ and $\bar{w}_{-h}$ are the velocity components at the bed.

5.3.1 Depth-averaged continuity equation

The two-dimensional depth-averaged continuity equation is obtained by integrating equation (5.4) between the bed (z=-h) and the free surface (z=ζ):

$$\int_{-h}^{\zeta} \frac{\partial \bar{u}}{\partial x} dz + \int_{-h}^{\zeta} \frac{\partial \bar{v}}{\partial y} dz + \int_{-h}^{\zeta} \frac{\partial \bar{w}}{\partial z} dz = 0 \quad . \tag{5.12}$$

In order to remove the partial derivatives from within the integrals, the Leibnitz rule is invoked; namely,

$$\int_a^b \frac{\partial f(x,y)}{\partial y} dx = \frac{\partial}{\partial y} \int_a^b f(x,y)dx - f(b,y)\frac{\partial b}{\partial y} + f(a,y)\frac{\partial a}{\partial y} \quad . \tag{5.13}$$

Thus the continuity equation becomes

$$\frac{\partial}{\partial x} \int_{-h}^{\zeta} \bar{u}\, dz - \bar{u}_\zeta \frac{\partial \zeta}{\partial x} + \bar{u}_{-h} \frac{\partial(-h)}{\partial x} + \frac{\partial}{\partial y} \int_{-h}^{\zeta} \bar{v}\, dz - \bar{v}_\zeta \frac{\partial \zeta}{\partial y}$$

$$+ \bar{v}_{-h} \frac{\partial(-h)}{\partial y} + \bar{w}_\zeta - \bar{w}_{-h} = 0 \quad . \tag{5.14}$$

Using the boundary conditions presented in equations (5.10) and (5.11), this expression may be simplified to give

$$\frac{\partial}{\partial x}\int_{-h}^{\zeta} \bar{u}\, dz + \frac{\partial}{\partial y}\int_{-h}^{\zeta} \bar{v}\, dz + \frac{\partial \zeta}{\partial t} + \frac{\partial h}{\partial t} = 0 \quad . \tag{5.15}$$

The depth-averaged velocity components are defined as

$$U = \frac{1}{(h + \zeta)}\int_{-h}^{\zeta} \bar{u}\, dz = \frac{1}{D}\int_{-h}^{\zeta} \bar{u}\, dz \tag{5.16a}$$

and

$$V = \frac{1}{(h + \zeta)}\int_{-h}^{\zeta} \bar{v}\, dz = \frac{1}{D}\int_{-h}^{\zeta} \bar{v}\, dz \tag{5.16b}$$

where $D = h + \zeta$ and is the total depth of flow.

Assuming the location of the bed is constant with time (i.e. no sediment deposition or erosion) and using the definitions shown in equation (5.16), the depth-averaged continuity equation becomes

$$\frac{\partial \zeta}{\partial t} + \frac{\partial (UD)}{\partial x} + \frac{\partial (VD)}{\partial y} = 0 \quad . \tag{5.17}$$

5.3.2 Depth-averaged momentum equations

The two-dimensional depth-averaged momentum expressions are obtained by integrating equations (5.5a) and (5.5b) between z=-h and z=ζ. After the application of the Leibnitz rule, the x-direction momentum expression has the following form:

$$\frac{\partial}{\partial t}\int_{-h}^{\zeta} \bar{u}\, dz + \frac{\partial}{\partial x}\int_{-h}^{\zeta} \bar{u}^2\, dz + \frac{\partial}{\partial y}\int_{-h}^{\zeta} \bar{u}\,\bar{v}\, dz - \left[\bar{u}_\zeta \frac{\partial \zeta}{\partial t} + \bar{u}_\zeta^2 \frac{\partial \zeta}{\partial x} + \bar{u}_\zeta\, \bar{v}_\zeta \frac{\partial \zeta}{\partial y} \right.$$

$$\left. - \bar{u}_\zeta\, \bar{w}_\zeta \right] + \left[\bar{u}_{-h} \frac{\partial (-h)}{\partial t} + \bar{u}_{-h}^2 \frac{\partial (-h)}{\partial x} + \bar{u}_{-h}\, \bar{v}_{-h} \frac{\partial (-h)}{\partial y} - \bar{u}_{-h}\, \bar{w}_{-h} \right]$$

$$= f\int_{-h}^{\zeta} \bar{v}\, dz - g\int_{-h}^{\zeta} \frac{\partial \zeta}{\partial x}\, dz + \frac{1}{\rho}\frac{\partial}{\partial x}\int_{-h}^{\zeta} \tau_{xx}\, dz + \frac{1}{\rho}\frac{\partial}{\partial y}\int_{-h}^{\zeta} \tau_{yx}\, dz - \frac{1}{\rho}\left[\tau_{xx\zeta} \frac{\partial \zeta}{\partial x} \right.$$

$$\left. + \tau_{yx\zeta} \frac{\partial \zeta}{\partial y} - \tau_{zx\zeta} - \tau_{xx_{-h}} \frac{\partial (-h)}{\partial x} - \tau_{yx_{-h}} \frac{\partial (-h)}{\partial y} + \tau_{zx_{-h}} \right] \quad . \tag{5.18}$$

Letting the wind and bed friction stresses be defined in a similar manner to Kuipers and Vreugdenhil (1973):

$$\tau_{wx} = - \tau_{xx\zeta} \frac{\partial \zeta}{\partial x} - \tau_{yx\zeta} \frac{\partial \zeta}{\partial y} + \tau_{zx\zeta} \qquad (5.19a)$$

and

$$\tau_{bx} = - \tau_{xx-h} \frac{\partial(-h)}{\partial x} - \tau_{yx-h} \frac{\partial(-h)}{\partial y} + \tau_{zx-h} \quad , \qquad (5.19b)$$

replacing $\int_{-h}^{\zeta} \bar{u}\, dz$ and $\int_{-h}^{\zeta} \bar{v}\, dz$ with UD and VD,

and using the kinematic free surface and bed boundary conditions, equation (5.18) may be simplified as

$$\frac{\partial(UD)}{\partial t} + \frac{\partial}{\partial x}\int_{-h}^{\zeta} \bar{u}^2\, dz + \frac{\partial}{\partial y}\int_{-h}^{\zeta} \bar{u}\,\bar{v}\, dz = fVD - g\int_{-h}^{\zeta} \frac{\partial \zeta}{\partial x}\, dz + \frac{\tau_{wx} - \tau_{bx}}{\rho}$$

$$+ \frac{1}{\rho}\frac{\partial}{\partial x}\int_{-h}^{\zeta} \tau_{xx}\, dz + \frac{1}{\rho}\frac{\partial}{\partial y}\int_{-h}^{\zeta} \tau_{yx}\, dz \quad . \qquad (5.20)$$

Noting that $\bar{u}=U+(\bar{u}-U)$ and $\bar{v}=V+(\bar{v}-V)$, it is possible to obtain the advective accelerations in terms of the depth-averaged velocity components, U and V. Thus the x-momentum equation on rearrangement becomes

$$\frac{\partial(UD)}{\partial t} + \frac{\partial(U^2D)}{\partial x} + \frac{\partial(UVD)}{\partial y} = fVD - gD\frac{\partial \zeta}{\partial x} + \frac{\tau_{wx} - \tau_{bx}}{\rho}$$

$$+ \frac{1}{\rho}\left[\frac{\partial}{\partial x}\int_{-h}^{\zeta} \left[\tau_{xx} - \rho(\bar{u}-U)^2\right] dz + \frac{\partial}{\partial y}\int_{-h}^{\zeta} \left[\tau_{yx} - \rho(\bar{u}-U)(\bar{v}-V)\right] dz \right]. \qquad (5.21)$$

The deviatoric stresses and terms accounting for the non-uniformity of the velocity profiles (in the vertical direction) can be grouped together in the manner adopted by Kuipers and Vreugdenhil (1973), Flokstra (1977), Walker (1987) and Dong (1987). Defining the *effective stresses* as

$$T_{xx} = \frac{1}{D}\int_{-h}^{\zeta} \left[\tau_{xx} - \rho(\bar{u}-U)^2\right] dz = \frac{1}{D}\int_{-h}^{\zeta} \left[2\mu\frac{\partial \bar{u}}{\partial x} - \rho\overline{u'^2} - \rho(\bar{u}-U)^2\right] dz \qquad (5.22a)$$

$$T_{xy} = \frac{1}{D}\int_{-h}^{\zeta}\left[\tau_{yx} - \rho(\bar{u}-U)(\bar{v}-V)\right]dz$$

$$= \frac{1}{D}\int_{-h}^{\zeta}\left[\mu\left[\frac{\partial\bar{u}}{\partial y} + \frac{\partial\bar{v}}{\partial x}\right] - \rho\overline{u'v'} - \rho(\bar{u}-U)(\bar{v}-V)\right]dz \qquad (5.22b)$$

$$T_{yy} = \frac{1}{D}\int_{-h}^{\zeta}\left[\tau_{yy} - \rho(\bar{v}-V)^2\right]dz = \frac{1}{D}\int_{-h}^{\zeta}\left[2\mu\frac{\partial\bar{v}}{\partial y} - \rho\overline{v'^2} - \rho(\bar{v}-V)^2\right]dz\ , \qquad (5.22c)$$

the depth-averaged x-momentum equation can be rewritten as

$$\frac{\partial(UD)}{\partial t} + \frac{\partial(U^2D)}{\partial x} + \frac{\partial(UVD)}{\partial y} = fVD - gD\frac{\partial\zeta}{\partial x} + \frac{\tau_{wx} - \tau_{bx}}{\rho} + \frac{1}{\rho}\left[\frac{\partial(DT_{xx})}{\partial x} + \frac{\partial(DT_{xy})}{\partial y}\right] \qquad (5.23a)$$

whilst the y-momentum equation can be similarly derived as

$$\frac{\partial(VD)}{\partial t} + \frac{\partial(UVD)}{\partial x} + \frac{\partial(V^2D)}{\partial y} = -fUD - gD\frac{\partial\zeta}{\partial y} + \frac{\tau_{wy} - \tau_{by}}{\rho} + \frac{1}{\rho}\left[\frac{\partial(DT_{xy})}{\partial x} + \frac{\partial(DT_{yy})}{\partial y}\right]. \qquad (5.23b)$$

The above depth-integrated Reynolds equations are in the so-called conservative form since finite-difference approximations to these expressions can be shown to conserve momentum flux across the individual cells of the coordinate mesh (Roache, 1972). Whilst many hydrodynamic simulations (e.g., Kuipers and Vreugdenhil, 1973; Flokstra, 1981; Falconer, 1976 onwards) have adopted conservative forms of primitive variable equations, the literature indicates that there has also been an extensive use of the simpler advection or non-conservative formulae. These can be derived by expanding the advective acceleration terms in equations (5.23) and employing the depth-averaged continuity equation (5.17) to simplify the resultant expressions. Thus the depth-integrated Reynolds equations can be recast as

$$\frac{\partial U}{\partial t} + U\frac{\partial U}{\partial x} + V\frac{\partial U}{\partial y} = fV - g\frac{\partial\zeta}{\partial x} + \frac{\tau_{wx} - \tau_{bx}}{\rho D} + \frac{1}{\rho D}\left[\frac{\partial(DT_{xx})}{\partial x} + \frac{\partial(DT_{xy})}{\partial y}\right] \qquad (5.24a)$$

and

$$\frac{\partial V}{\partial t} + U\frac{\partial V}{\partial x} + V\frac{\partial V}{\partial y} = -fU - g\frac{\partial\zeta}{\partial y} + \frac{\tau_{wy} - \tau_{by}}{\rho D} + \frac{1}{\rho D}\left[\frac{\partial(DT_{xy})}{\partial x} + \frac{\partial(DT_{yy})}{\partial y}\right]. \qquad (5.24b)$$

Regarding the decision as to which type of governing hydrodynamic equation should be used, Crowley (1968), states:

> ".... In certain coordinate systems, however, proper evaluation of the transport terms may lead to quite complicated algebraic forms that can be time consuming to compute. On the other hand, the advection form may give simpler and thus cheaper, difference equations. Even if one is then forced to use the advection form rather than the conservation form by practical matters, such as limited available computer time, it is possible that the trade-off of accuracy for efficiency will not cause the results to be utterly unreliable. Although the advection form of the equations does not guarantee conservation, it does not follow that the proper quantities will not be approximately conserved."

Many of the computational schemes that are regarded nowadays as 'standard' solutions of the shallow water equations implement the advection form of governing equation (Leendertse, 1967; Abbott et al., 1973; Hodgins, 1977). Many other numerical simulations have also successfully used the advection approach (Butler, 1978a; McClimans and Gjerp, 1978; Chiang and Lee, 1982; Yin and Chen, 1982; Vemulakonda et al., 1985) and therefore, it was decided that the present study should adopt the non-conservative/advective depth-averaged Reynolds expressions presented in equations (5.24). Moreover, this would then coincide with the decision described at the end of Section 4.3.1 to use the non-conservative form of derivative transformation for the boundary-fitted mapping procedure.

5.4 Effective stresses

The effective stresses have already been presented in equations (5.22) and are each composed of three separate components. The first term arises as a result of viscous effects and is analogous to the diffusive portion of the Navier-Stokes expressions. The second term is a Reynolds stress and occurs because of the presence of the turbulent fluctuating velocity components (u' and v'), whilst the third component models the horizontal momentum transfer caused by the depth-averaging procedure (the so-called dispersive terms). For turbulent flows, the viscous stresses are assumed to be much smaller than the Reynolds stresses and are therefore neglected. Furthermore, the dispersive terms caused by the non-uniform velocity profiles in the vertical direction are often neglected (e.g., McGuirk and Rodi, 1978 and 1979; Butler, 1978a; Vemulakonda et al., 1985; Walker, 1987; Dong, 1987).

Falconer (1976 onwards) approximated the dispersion by assuming given distributions of velocity profile in the vertical direction. For example, considering the profile to correspond to a '1/7 power law', then

$$\bar{u}(z) = U_{max} \frac{(h + z)^{1/7}}{(h + \zeta)^{1/7}} = \frac{8}{7} U \frac{(h + z)^{1/7}}{(h + \zeta)^{1/7}} \quad . \qquad (5.25)$$

Using this distribution, the U-velocity dispersion term can be shown to be less than 2% of the corresponding advective acceleration derivative, i.e.

$$\int_{-h}^{\zeta} (\bar{u} - U)^2 \, dz = \int_{-h}^{\zeta} U^2 \left[\frac{8}{7} \frac{(h + z)^{1/7}}{(h + \zeta)^{1/7}} - 1 \right]^2 dz = 0.016 \, U^2 \, (h+\zeta) \qquad (5.26a)$$

and hence,

$$\frac{1}{\rho} \frac{\partial}{\partial x} \int_{-h}^{\zeta} \rho \, (\bar{u} - U)^2 \, dz = 0.016 \, \frac{\partial (U^2 D)}{\partial x} \quad . \qquad (5.26b)$$

Falconer accounted for the dispersion by modifying the advective acceleration terms by a correction factor, α; it can be seen from equation (5.21) that in this particular case, $\alpha = 1.016$. Similarly, if the velocity profile in the vertical plane is represented by a logarithmic distribution of the form

$$\bar{u}(z) = \frac{U_*}{\kappa} \ln (h + z) + \text{constant} \qquad (5.27a)$$

where, U_* = shear velocity = $\frac{(g)^{\frac{1}{2}} U}{C}$ (C is the Chezy coefficient)

and $\kappa = 0.42$ = von Kármán's constant,

then the correction factor, to account for the non-uniformity in velocity profile, can be shown to be

$$\alpha = 1 + \frac{g}{C^2 \kappa^2} \quad . \qquad (5.27b)$$

In both these cases, the correction factors for the advective acceleration terms only differ from the uniform flow distribution (α=1) by a few percent. Furthermore, Rastogi and Rodi (1978) employed a fully three-dimensional computational model for a near field problem of a thermally polluted discharge into a rectangular channel, and showed that the *exact* dispersion terms were generally smaller than the turbulent shear stresses. In regions where the Reynolds stress terms were small, the dispersion momentum fluxes were found to be comparable to or even larger than the turbulent

stresses. However, Rastogi and Rodi indicated that the characteristics of hydraulic regimes originate from regions of strong turbulent shear stresses rather than areas with little turbulence; they observed that their two-dimensional flow predictions (neglecting dispersion) agreed fairly well with the depth-integrated values obtained from the three-dimensional model. Therefore, for the present numerical study, the transport of depth-averaged quantities due to the vertical non-uniformity is neglected.

The effective stress equations (5.22) can thus be rewritten as:

$$T_{xx} = -\frac{1}{D}\int_{-h}^{\zeta} \rho\overline{u'^2}\,dz \quad ; \quad T_{xy} = -\frac{1}{D}\int_{-h}^{\zeta} \rho\overline{u'v'}\,dz \quad ; \quad T_{yy} = -\frac{1}{D}\int_{-h}^{\zeta} \rho\overline{v'^2}\,dz \quad . \tag{5.28}$$

5.5 Evaluation of the Reynolds stresses

Equation (5.28) indicates that the effective stresses are dependent upon the time-averaged correlations between fluctuating velocity components. These correlations, multiplied by the density of the fluid, represent the momentum transport caused by the turbulent motion and are referred to as the turbulent or Reynolds stresses; for example $-\rho\overline{u'v'}$ is the transport of x-momentum in the y-direction due to turbulence. In most flow situations, especially at high Reynolds number, the turbulent stresses are much larger than the equivalent laminar or viscous stresses and hence it is normally satisfactory to neglect the viscous effects from the momentum expressions.

The solution of equations (5.23) or (5.24) can thus only be achieved by determining the fluctuating velocity correlations, and it is this part of the computation which is referred to as the *turbulence model*. There exists a hierarchy of turbulence models, where generally a trade-off is made between the order of mathematical closure (described later) and the ease with which the model can be implemented. This section briefly describes the various approaches to the calculation of turbulent phenomena; the interested reader is referred to the comprehensive review articles of Rodi (1984a) and Markatos (1986) or to the general texts by Hinze (1959), Launder and Spalding (1972) and Bradshaw et al. (1981).

5.5.1 The exact $\overline{u_i' u_j'}$ transport equation.

An exact transport equation for the turbulent stresses, $\overline{u_i' u_j'}$ * can be derived by subtracting the Reynolds equations from the corresponding Navier-Stokes equations for both the x_i- and x_j-directions. The resultant expression for the i-momentum component is then multiplied by the fluctuating velocity u_j', whilst the resulting j-momentum equation is multiplied by u_i'. Summation of the two expressions and time-averaging produces the so-called Reynolds-stress equations (sometimes referred to as the Friedmann-Keller equations):

$$\frac{D(\overline{u_i' u_j'})}{Dt} = \underbrace{\frac{\partial(\overline{u_i' u_j'})}{\partial t}}_{\text{rate of change}} + \underbrace{\bar{u}_l \frac{\partial(\overline{u_i' u_j'})}{\partial x_l}}_{\text{advective transport}} = \underbrace{-\left[\overline{u_i' u_l'} \frac{\partial \bar{u}_j}{\partial x_l} + \overline{u_j' u_l'} \frac{\partial \bar{u}_i}{\partial x_l} \right]}_{\text{stress production}}$$

$$\underbrace{- 2\nu \overline{\frac{\partial u_i'}{\partial x_l} \frac{\partial u_j'}{\partial x_l}}}_{\text{viscous dissipation}} + \underbrace{\overline{\frac{P'}{\rho} \left[\frac{\partial u_i'}{\partial x_j} + \frac{\partial u_j'}{\partial x_i} \right]}}_{\text{pressure-strain correlation}}$$

$$\underbrace{- \frac{\partial}{\partial x_l} \left[\overline{u_i' u_j' u_l'} + \frac{1}{\rho} \left[\delta_{il} \overline{u_j' P'} + \delta_{jl} \overline{u_i' P'} \right] - \nu \frac{\partial(\overline{u_i' u_j'})}{\partial x_l} \right]}_{\text{. . . . diffusive transport . . . .}} \qquad (5.29)$$

where δ_{ij} is the Kronecker delta:-

$$\delta_{ij} = 1 \quad \text{for} \quad i = j$$

$$\delta_{ij} = 0 \quad \text{for} \quad i \neq j$$

and ν is the kinematic viscosity of the fluid ($\nu = \mu/\rho$).

* In order to reduce the complexity of the presented formulae, tensor notation is adopted; vector quantities are written with a subscript attached to the main symbol to define the direction. For example, coordinate directions are all given the symbol x, whilst fluctuating velocity components are given the symbol u'. Moreover, the usual summation convention for tensor notation is adopted; whenever the same index is repeated in a single expression, the sum is taken over all three coordinate directions. Thus, the advective accelerations of the non-conservative Reynolds equations (5.2) can be written as

$$\bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = \sum_{j=1}^{3} \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = \bar{u}_1 \frac{\partial \bar{u}_i}{\partial x_1} + \bar{u}_2 \frac{\partial \bar{u}_i}{\partial x_2} + \bar{u}_3 \frac{\partial \bar{u}_i}{\partial x_3} \; .$$

The physical interpretation of each of the components in this transport equation is shown below the appropriate term. Thus, the rate of change of $\overline{u_i' u_j'}$ is balanced by:

(a) the advective processes due to the mean motion,
(b) the stress production caused by the interaction of the Reynolds stresses with the mean-velocity gradients,
(c) dissipation by viscous action into heat,
(d) 'pressure-strain' correlations (accounting for the effects of the fluctuating pressure and strain rates)
(e) and finally diffusive transport.

It can be seen that the exact transport equation introduces unknown higher order turbulence correlations and hence it is necessary to introduce model assumptions in order to achieve a *closed* solution. The assumptions normally restrict the regime to isotropic turbulence (Rodi, 1984a) so that the same amount of energy is dissipated in each direction whilst algebraic expressions are introduced to model each of the unknown higher order correlations. Even so, the solution of modelled Reynolds-stress equations is still very complex and beyond the scope of the present study. Moreover, since the expressions must be solved in a fully three-dimensional system, the six partial differential equations modelling the transport of the individual Reynolds stresses together with the three momentum equations imply that any numerical scheme using this approach is extremely expensive in terms of computing time.

5.5.2 The eddy viscosity concept

Boussinesq's eddy viscosity concept forms the basis for most turbulence models in use today. The methodology assumes that the Reynolds stresses are proportional to the gradients of the time-averaged velocity components, in an analogous manner to the viscous stresses of laminar regimes. Thus, the fluctuating velocity correlations are determined as

$$- \overline{u_i' u_j'} = \nu_t \left[\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right] - \frac{2}{3} k \, \delta_{ij} \qquad (5.30)$$

where ν_t is the turbulent or eddy viscosity.

Unlike the coefficient of kinematic viscosity, the eddy viscosity is not a physical property of the fluid, but instead depends upon the turbulence characteristics of the

flow; ν_t may vary both spatially and temporally across the hydraulic domain. However, as Rodi (1984a) points out, the introduction of equation (5.30) "does not constitute a turbulence model but only provides the frame-work for constructing such a model; the main problem is now shifted to determining the distribution of ν_t".

The inclusion of the term involving the Kronecker delta, in equation (5.30), allows the formula to be used for the normal stresses (i=j), since it ensures that the sum of the three normal stresses is equal to 2k (where k is the kinetic energy of the fluctuating turbulent motion per unit mass), viz.

$$k = \frac{1}{2}\left[\overline{u_1'^2} + \overline{u_2'^2} + \overline{u_3'^2} \right] \quad . \tag{5.31}$$

The final term of equation (5.30) involving the turbulent kinetic energy, k, can only be used in conjunction with the more advanced turbulence simulations, such as the 'one-' or 'two-equation' models which are described later; with 'zero-equation' models (e.g., constant eddy viscosity, mixing-length or algebraic depth-averaged viscosity simulations) the value of k remains unknown, and it is therefore necessary to omit the $(2/3)k\delta_{ij}$ term from the Reynolds stress.

Implementing the Boussinesq eddy viscosity concept for the depth-averaged equations of motion, allows the effective stresses presented in equation (5.28) to be rewritten as

$$T_{ij} = \rho\, \tilde{\nu}_t \left[\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right] - \frac{2}{3}\rho\, \tilde{k}\, \delta_{ij} \tag{5.32}$$

or in full:

$$T_{xx} = 2\rho\, \tilde{\nu}_t \frac{\partial U}{\partial x} - \frac{2}{3}\rho\, \tilde{k} \tag{5.33a}$$

$$T_{xy} = \rho\, \tilde{\nu}_t \left[\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x} \right] \tag{5.33b}$$

$$T_{yy} = 2\rho\, \tilde{\nu}_t \frac{\partial V}{\partial y} - \frac{2}{3}\rho\, \tilde{k} \tag{5.33c}$$

where the overbar, $\tilde{}$, indicates that the variable is not a true depth-averaged quantity in the sense of the mathematical definition

$$U_i = \frac{1}{(h + \zeta)} \int_{-h}^{\zeta} \bar{u}_i \, dz \quad , \tag{5.16 bis}$$

but is a value which produces the appropriate depth-integrated Reynolds stresses when inserted into equation (5.32).

Since the majority of turbulence models use the eddy viscosity concept, it is convenient to categorize the simulation by the number of *transport* equations used to evaluate the turbulent conditions. Calculations which utilise algebraic expressions for the value of eddy viscosity do not use a transport equation and are therefore referred to as zero-equation models.

5.5.3 Zero-equation turbulence models

(a) Constant eddy viscosity

Many of the early numerical simulations of hydraulic flow phenomena (e.g., Hansen, 1962; Kuipers and Vreugdenhil, 1973; Audunson et al., 1975; Butler, 1978a) employed constant values of eddy viscosity across the entire flow field. The value of eddy viscosity was generally determined by trial-and-error so that the numerical simulation agreed with experimental or real-life observations. As Rodi (1984a) indicates, constant eddy viscosity approaches cannot be regarded as 'true' turbulence models; they are merely a simple method of solving the momentum equations. Nevertheless, the constant eddy viscosity approximation proved useful in the present study for validating the boundary-fitted computational scheme used to solve the shallow water equations because it enabled the results to be compared directly with the laminar flow regimes predicted by the Navier-Stokes simulations described in Chapter 3.

Another popular method for evaluating the turbulent stresses involves neglecting the normal stresses completely and using anisotropic eddy viscosities for the cross-derivative terms. This technique is often used for coastal circulation problems where the momentum mixing processes are markedly different in the alongshore and offshore directions (da Silva Lima, 1981; Vemulakonda et al., 1985; Joynes, 1989). Thus the lateral effective stresses can be calculated from expressions similar in form to

$$T_{xy} = \rho \tilde{\nu}_x \frac{\partial V}{\partial x} + \rho \tilde{\nu}_y \frac{\partial U}{\partial y} \qquad (5.34)$$

where $\tilde{\nu}_x$ and $\tilde{\nu}_y$ are the eddy viscosities in the x- and y-directions.

The eddy viscosity in the offshore direction is often allowed to vary across the flow field according to a mixing-length approximation (see Section 5.5.3b).

A similar anisotropic eddy viscosity approach was employed by Johnson (1980, 1982) and Johnson et al. (1982) for modelling the flow in rivers and estuaries using boundary-fitted coordinate systems. In this case, the normal effective stress components were included in the analysis by defining:

$$\frac{\partial(DT_{xx})}{\partial x} = \frac{\partial}{\partial x}\left[D\,\rho\,\tilde{\nu}_{xx}\frac{\partial U}{\partial x}\right] \qquad (5.35a)$$

$$\frac{\partial(DT_{xy})}{\partial y} = \frac{\partial}{\partial y}\left[D\,\rho\,\tilde{\nu}_{xy}\frac{\partial U}{\partial y}\right] \qquad (5.35b)$$

$$\frac{\partial(DT_{yx})}{\partial x} = \frac{\partial}{\partial x}\left[D\,\rho\,\tilde{\nu}_{yx}\frac{\partial V}{\partial x}\right] \qquad (5.35c)$$

and

$$\frac{\partial(DT_{yy})}{\partial y} = \frac{\partial}{\partial y}\left[D\,\rho\,\tilde{\nu}_{yy}\frac{\partial V}{\partial y}\right] \qquad (5.35d)$$

where $\tilde{\nu}_{xx}$ and $\tilde{\nu}_{yy}$ are the diagonal components of the eddy viscosity tensor,

and $\tilde{\nu}_{xy}$ and $\tilde{\nu}_{yx}$ are the off-diagonal elements of the eddy viscosity tensor.

Johnson expressed the depth-integrated momentum expressions in a form similar to:

$$\frac{\partial(UD)}{\partial t}+\frac{\partial(U^2D)}{\partial x}+\frac{\partial(UVD)}{\partial y}= fVD - gD\frac{\partial \zeta}{\partial x}+\frac{\tau_{wx}-\tau_{bx}}{\rho}+\frac{\partial}{\partial x}\left[D\,\tilde{\nu}_{xx}\frac{\partial U}{\partial x}\right] + \frac{\partial}{\partial y}\left[D\,\tilde{\nu}_{xy}\frac{\partial U}{\partial y}\right] \qquad (5.36a)$$

and

$$\frac{\partial(VD)}{\partial t}+\frac{\partial(UVD)}{\partial x}+\frac{\partial(V^2D)}{\partial y}= -fUD - gD\frac{\partial \zeta}{\partial y}+\frac{\tau_{wy}-\tau_{by}}{\rho}+\frac{\partial}{\partial x}\left[D\,\tilde{\nu}_{yx}\frac{\partial V}{\partial x}\right] + \frac{\partial}{\partial y}\left[D\,\tilde{\nu}_{yy}\frac{\partial V}{\partial y}\right] . \qquad (5.36b)$$

However, the above momentum expressions indicate that Johnson's definition of effective stresses is incomplete because the second order cross-derivative terms have been neglected. Furthermore, the two lateral effective stresses, T_{xy} and T_{yx}, are generally unequal and consequently there is a spurious torque on the fluid elements;

this may be the reason why Johnson set the off-diagonal components of the eddy viscosity tensor to zero. In view of the foregoing, the isotropic eddy viscosity effective stress equations (5.33) were judged to be more appropriate for the present study. Moreover, this approach is imperative if $\tilde{\nu}_t$ is to be determined from a transport model since the one- or two-equation turbulence simulations implicitly assume that the fluctuating turbulent motions are isotropic.

(b) Mixing-length models

In a similar manner to the analytical determination of the molecular viscosity for ideal gases, Prandtl (1925) proposed that the eddy viscosity could be considered as being proportional to the product of a typical characteristic velocity scale (V_c) of the fluctuating turbulent motion and a typical length scale (L_m) of the eddies (which Prandtl called the *mixing-length*). Hence, the eddy viscosity is determined from

$$\nu_t = c \; L_m \; V_c \tag{5.37}$$

where c is an empirical constant.

Although this equation appears trivial, it forms the basis of the more advanced turbulent transport models which are described in Sections 5.5.4 and 5.5.5. Without the use of additional transport expressions to determine the spatial distribution of the velocity and length scales, equation (5.37) can only be used for very simple flows such as boundary-layers, where there is only one significant shear stress and one significant velocity gradient. Prandtl postulated that for a thin-shear layer:

$$V_c = L_m \left| \frac{\partial \bar{u}}{\partial y} \right| \tag{5.38}$$

where $\bar{u}$ is the velocity parallel to the shear-layer
and y is the distance measured normal to the shear-layer.

Substituting equation (5.38) into (5.37) and assuming that the constant of proportionality is unity, Prandtl's mixing length hypothesis can be written as

$$\nu_t = L_m^{\,2} \left| \frac{\partial \bar{u}}{\partial y} \right| \; . \tag{5.39}$$

For simple shear flows, the mixing-length, L_m is assumed to be constant across the layer and can be related to the geometrical thickness of the boundary-layer, δ. Although Prandtl's method can be extended to other hydrodynamic regimes, by

employing the general mixing-length equation (see Rodi, 1984a):

$$\nu_t = L_m^2 \left[\left[\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right] \frac{\partial \bar{u}_i}{\partial x_j} \right]^{\frac{1}{2}} , \qquad (5.40)$$

the technique is of little use for modelling complex flows such as recirculation regions because of the difficulty in specifying the value of the mixing-length, L_m.

(c) Depth-averaged algebraic eddy viscosity models

Following Taylor's investigation (1954) into the turbulent dispersive processes in a pipe, Elder (1959) studied the dispersion of dye in an open channel and concluded that the depth-averaged longitudinal coefficient of diffusivity, K_l could be evaluated as

$$K_l = 5.9\ U_* D \qquad (5.41)$$

where U_* = friction or shear velocity = $[\tau_b/\rho]^{\frac{1}{2}}$

τ_b = bed friction

and D = the depth of flow.

Kuipers and Vreugdenhil (1973) and Vreugdenhil (1973) proposed that the eddy viscosity could be expected to have the same order of magnitude as the above diffusivity. Hence, defining the bed shear stress in terms of the Chezy coefficient:

$$\frac{\tau_b}{\rho} = U_*^2 = \frac{(U^2 + V^2)\ g}{C^2} , \qquad (5.42)$$

leads to the eddy viscosity being defined as

$$\tilde{\nu}_t = \frac{5.9\ D\ \sqrt{g}\ (U^2 + V^2)^{\frac{1}{2}}}{C} . \qquad (5.43)$$

This expression was also utilised by Falconer (1976) for the numerical computation of jet-forced reservoir phenomena; unlike Kuipers and Vreugdenhil, Falconer allowed the eddy viscosity to vary across the flow field in accordance with the local velocity vectors. As Vreugdenhil (1973) points out, the most questionable assumption about the formula arises from the fact that momentum can be transferred by surface elevation gradients, whereas there is no comparable mechanism for the transport of matter in the dispersion equations.

5.5.4 One-equation turbulence model (the k-equation)

Zero-equation models assume that the turbulence is generated and dissipated at the same location in the flow domain. Thus the transport of turbulent phenomena by advective and diffusive processes is neglected and so is the production of turbulence at previous times in the simulation (referred to as 'history effects' by Rodi (1984a)). Consequently, if turbulence is predominantly generated at one location in the flow field and is then transported by the mean velocities to other parts of the domain, simple zero-equation models are unable to predict the flow situation accurately and therefore partial differential *transport* equations must be employed. Most one-equation models calculate the spatial distribution of the turbulent kinetic energy, k and then use $\sqrt{k}$ as a measure of the characteristic velocity scale, V_c. The eddy viscosity equation (5.37) is therefore rewritten as the so-called Kolmogorov-Prandtl expression:

$$\nu_t = c'_\mu \sqrt{k}\, L_m \tag{5.44}$$

where c'_μ is an empirical constant.

The partial differential equation governing the transport of k is derived from the exact Reynolds-stress expression (5.29) by summing the three normal stress formulae (i=j=1,2,3). Thus the exact turbulent energy transport equation can be shown to be

$$\frac{Dk}{Dt} = \frac{\partial k}{\partial t} + \bar{u}_i \frac{\partial k}{\partial x_i} = -\frac{\partial}{\partial x_i}\left[\overline{u'_i\left[\frac{u'_j u'_j}{2} + \frac{P'}{\rho}\right]}\right] - \overline{u'_i u'_j}\frac{\partial \bar{u}_i}{\partial x_j}$$

rate of change; advective transport; diffusive transport due to turbulence; production of k due due to shear

$$- \nu \overline{\frac{\partial u'_i}{\partial x_j}\frac{\partial u'_i}{\partial x_j}} + \nu \frac{\partial^2 k}{\partial x_i^2} \tag{5.45}$$

viscous dissipation (ϵ); diffusive transport due to viscous effects

As before, for the Friedmann-Keller expressions, the physical interpretation of each component of the transport equation is shown below the appropriate term. Thus, the rate of change of k is controlled by :

(a) the advective processes due to the mean motion,

(b) the diffusive transport due to the turbulent velocity and pressure fluctuations,

(c) the production of k due to the interaction of the Reynolds stresses and the mean-velocity gradients,

(d) viscous dissipation into heat,

(e) and finally the diffusive transport due to viscous effects; for high Reynolds number flows this term is usually neglected.

It can be seen that the 'pressure-strain' correlations appearing in the original Reynolds-stress equations are absent from the above turbulent energy transport expression. This is because the summation of the pressure-strain correlations for the three normal stresses, reveals that they do not contribute to the total hydrodynamic energy balance.*

The exact turbulent energy transport equation involves unknown higher order turbulence correlations and therefore it is necessary to utilise model assumptions to obtain a closed set of equations. The diffusive transport of k is assumed to be proportional to the gradient of k, in a similar manner to Boussinesq's eddy viscosity concept (which considers the diffusive transport of momentum as being proportional to derivatives of the mean velocity components). Thus the first term on the right-hand side of equation (5.45) is modelled as

$$-\overline{u_i'\left[\frac{u_j'u_j'}{2}+\frac{P'}{\rho}\right]}=\frac{\nu_t}{\sigma_k}\frac{\partial k}{\partial x_i} \qquad (5.46)$$

where σ_k is an empirical diffusion coefficient for the turbulent kinetic energy (usually taken as unity - see Launder and Spalding, 1972). Furthermore, from dimensional considerations, the viscous dissipation rate, ϵ can be assumed to be

$$\epsilon = c_D \frac{k^{3/2}}{L_m} \qquad (5.47)$$

where c_D is another empirical constant.

Hence the modelled turbulent energy transport expression (the so-called k-equation) can be expressed as

*

$$\sum_{i=j=1}^{3}\overline{\frac{P'}{\rho}\left[\frac{\partial u_i'}{\partial x_j}+\frac{\partial u_j'}{\partial x_i}\right]}=2\overline{\frac{P'}{\rho}\left[\frac{\partial u_1'}{\partial x_1}+\frac{\partial u_2'}{\partial x_2}+\frac{\partial u_3'}{\partial x_3}\right]}=0$$

because of the continuity equation for the fluctuating velocity components.

$$\frac{Dk}{Dt} = \frac{\partial k}{\partial t} + \bar{u}_i \frac{\partial k}{\partial x_i} = \frac{\partial}{\partial x_i}\left[\frac{\nu_t}{\sigma_k} \frac{\partial k}{\partial x_i} \right] + \nu_t \left[\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right] \frac{\partial \bar{u}_i}{\partial x_j} - c_D \frac{k^{3/2}}{L_m} \quad . \tag{5.48}$$

For a specified state of turbulence, as characterised by the magnitudes of k and ϵ, equation (5.47) indicates that the value of the mixing-length, L_m is directly proportional to the constant c_D. Since the empiricism in the Kolmogorov-Prandtl relationship stems from the product of c'_μ and the mixing-length, L_m, the actual values of the constants c'_μ and c_D are unimportant; it is only their product which must be specified (Rodi (1984a) quotes $c'_\mu\, c_D \approx 0.08$).

The main objection to the one-equation transport model arises from the fact that it is still necessary to prescribe a mixing-length for the purposes of determining the eddy viscosity. Hence it is usually more common for the k-equation to be employed with a second transport relationship for evaluating the spatial distribution of the viscous dissipation rate, ϵ (McGuirk and Rodi, 1978 and 1979; Rastogi and Rodi, 1978; Rodi et al., 1981; Walker, 1987; Dong, 1987).

5.5.5 Two-equation turbulence model (the k-ϵ model)

The exact dissipation equation governing the transport of ϵ is obtained by subtracting the Reynolds equations from the Navier-Stokes expressions, i.e.,

Navier-Stokes equation (written with the instantaneous velocities, u_i in the form $u_i = \bar{u}_i + u'_i$):

$$\frac{\partial(\bar{u}_i+u'_i)}{\partial t} + (\bar{u}_j+u'_j) \frac{\partial}{\partial x_j} (\bar{u}_i+u'_i) = -\frac{1}{\rho} \frac{\partial}{\partial x_i} (\bar{P}+P') + \nu \frac{\partial^2(\bar{u}_i+u'_i)}{\partial x_j^2} . \tag{5.49}$$

Reynolds equation:

$$\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \bar{P}}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\nu \frac{\partial \bar{u}_i}{\partial x_j} - \overline{u'_i u'_j} \right] \quad . \tag{5.50}$$

Subtraction of equation (5.50) from (5.49) leads to the momentum relationship

$$\frac{\partial u'_i}{\partial t} + u'_j \frac{\partial(\bar{u}_i+u'_i)}{\partial x_j} + \bar{u}_j \frac{\partial u'_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial P'}{\partial x_i} + \nu \frac{\partial^2 u'_i}{\partial x_j^2} + \frac{\partial}{\partial x_j} (\overline{u'_i u'_j}) \quad . \tag{5.51}$$

It can be shown that differentiation of the above expression with respect to the

coordinate direction x_l, multiplication by $\partial u_i'/\partial x_l$ and time-averaging leads to the exact dissipation equation:

$$\frac{\partial}{\partial t}\left[\frac{1}{2}\left[\frac{\partial u_i'}{\partial x_l}\right]^2\right] + \frac{\partial^2 \bar{u}_i}{\partial x_l \partial x_j}\,\overline{u_j'\frac{\partial u_i'}{\partial x_l}} + \frac{\partial \bar{u}_i}{\partial x_j}\,\overline{\frac{\partial u_i'}{\partial x_l}\frac{\partial u_j'}{\partial x_l}} + \frac{\partial \bar{u}_j}{\partial x_l}\,\overline{\frac{\partial u_i'}{\partial x_l}\frac{\partial u_i'}{\partial x_j}}$$

$$+ \bar{u}_j\frac{\partial}{\partial x_j}\left[\frac{1}{2}\left[\frac{\partial u_i'}{\partial x_l}\right]^2\right] + \overline{\frac{\partial u_i'}{\partial x_l}\frac{\partial u_j'}{\partial x_l}\frac{\partial u_i'}{\partial x_j}} + \overline{u_j'\frac{\partial}{\partial x_j}\left[\frac{1}{2}\left[\frac{\partial u_i'}{\partial x_l}\right]^2\right]}$$

$$= -\frac{1}{\rho}\overline{\frac{\partial u_i'}{\partial x_l}\frac{\partial^2 P'}{\partial x_l \partial x_i}} + \nu\,\overline{\frac{\partial u_i'}{\partial x_l}\left[\frac{\partial^2}{\partial x_j^2}\left[\frac{\partial u_i'}{\partial x_l}\right]\right]} \quad . \qquad (5.52)$$

Whilst the advection and dissipation terms are still in a recognisable form, the remaining components in this equation contain many unknown third-order turbulence correlations. Hence, in order to obtain closure of the k-ϵ model, it is necessary to introduce quite drastic assumptions into the ϵ-equation. Regarding the method of devising a modelled ϵ-equation, Launder (1984) states that the process "while usually nodding in the direction of the exact dissipation equation, can more honestly be said to be the outcome of dimensional analysis, intuition, analogy and the insistence, in assigning empirical coefficients, that certain well established features of a turbulent flow be correctly mimicked". In a similar manner to the modelled k-equation (5.48), the transport expression for ϵ must include a turbulent diffusion term, a production/source term and a destruction/sink component. The diffusive flux of ϵ is assumed to be proportional to the gradient of ϵ, whilst the source term is regarded as being proportional to the rate of generation of turbulent kinetic energy multiplied by ϵ/k, to obtain dimensional homogeneity, i.e.

$$\text{Source of } \epsilon \;\propto\; \frac{\epsilon}{k}\,\nu_t\left[\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right]\frac{\partial \bar{u}_i}{\partial x_j} \quad . \qquad (5.53)$$

Finally, as explained by Launder (1984) the sink term needs to be of the form:

$$\text{Sink of } \epsilon \;\propto\; \frac{\epsilon^2}{k} \qquad (5.54)$$

in order to prevent the turbulent kinetic energy, k from becoming negative in regions with decaying turbulence (for example, behind a grid). Thus the ϵ-equation can be written (Launder and Spalding, 1972 and 1974; Bradshaw et al., 1981; Launder, 1984; Rodi, 1984a) as

$$\frac{D\epsilon}{Dt} = \frac{\partial \epsilon}{\partial t} + \bar{u}_i \frac{\partial \epsilon}{\partial x_i} = \frac{\partial}{\partial x_i}\left[\frac{\nu_t}{\sigma_\epsilon}\frac{\partial \epsilon}{\partial x_i}\right] + c_{1\epsilon}\frac{\epsilon}{k}\nu_t\left[\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right]\frac{\partial \bar{u}_i}{\partial x_j} - c_{2\epsilon}\frac{\epsilon^2}{k} \tag{5.55}$$

diffusion generation destruction

where σ_ϵ is an empirical diffusion coefficient for the viscous dissipation (sometimes referred to as the effective turbulent Prandtl number for the diffusion of ϵ),

and $c_{1\epsilon}$ and $c_{2\epsilon}$ are other empirical coefficients used to calibrate the k–ϵ model.

The implementation of a partial differential transport equation for the viscous dissipation of kinetic energy avoids the need to specify a mixing length, L_m, in the Kolmogorov–Prandtl formula. Therefore, using the relationship presented in equation (5.47), the Kolmogorov–Prandtl expression becomes:

$$\nu_t = c'_\mu \sqrt{k}\,\frac{c_D}{\epsilon}\,k^{3/2} = c'_\mu c_D \frac{k^2}{\epsilon} = c_\mu \frac{k^2}{\epsilon} \quad . \tag{5.56}$$

Furthermore, since the k–ϵ model calculates the spatial distribution of the viscous dissipation (ϵ), the final term of the modelled turbulent energy transport equation (5.48) need no longer be expressed as $(c_D/L_m)k^{3/2}$, but instead, can be simply replaced by ϵ. Thus, the two-equation k–ϵ model may be summarised as

$$\frac{Dk}{Dt} = \frac{\partial k}{\partial t} + \bar{u}_i \frac{\partial k}{\partial x_i} = \frac{\partial}{\partial x_i}\left[\frac{\nu_t}{\sigma_k}\frac{\partial k}{\partial x_i}\right] + \nu_t\left[\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right]\frac{\partial \bar{u}_i}{\partial x_j} - \epsilon \tag{5.57}$$

and

$$\frac{D\epsilon}{Dt} = \frac{\partial \epsilon}{\partial t} + \bar{u}_i \frac{\partial \epsilon}{\partial x_i} = \frac{\partial}{\partial x_i}\left[\frac{\nu_t}{\sigma_\epsilon}\frac{\partial \epsilon}{\partial x_i}\right] + c_{1\epsilon}\frac{\epsilon}{k}\nu_t\left[\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right]\frac{\partial \bar{u}_i}{\partial x_j} - c_{2\epsilon}\frac{\epsilon^2}{k} \tag{5.55 bis}$$

where

$$\nu_t = c_\mu \frac{k^2}{\epsilon} \quad . \tag{5.56 bis}$$

The empirical coefficients in the above equations are determined partly by considering specific types of turbulence (e.g., $c_{2\epsilon}$ can be found by analysing the decay of turbulent kinetic energy behind a grid) and partly by comparing computer predictions with well-documented laboratory observations of turbulent shear flows.

Computer optimisation is used to tune the coefficients so as to achieve the best overall agreement with experimental results; the values recommended by Launder and Spalding (1974) are given in Table 5.1 (Jones and Launder (1972) and Gosman et al. (1979) propose very similar sets of constants).

Table 5.1

The values of the coefficients in the k-ϵ model

c_μ	$c_{1\epsilon}$	$c_{2\epsilon}$	σ_k	σ_ϵ
0.09	1.44	1.92	1.0	1.3

5.6 Depth-averaged k-ϵ model

The turbulence transport models described so far, are only applicable to three-dimensional (and planar two-dimensional) flow systems. Unlike the depth-integration of the continuity and momentum equations (presented in Section 5.3), the development of a depth-averaged k-ϵ model does not possess a strict mathematical proof, but is based upon a more pragmatic approach. It is assumed that the turbulence can be characterised by the depth-averaged turbulent kinetic energy ($\tilde{k}$) and viscous dissipation ($\tilde{\epsilon}$) so that the fluid stresses can be evaluated as

$$T_{ij} = \rho \tilde{\nu}_t \left[\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right] - \frac{2}{3} \rho \tilde{k} \delta_{ij} \qquad (5.32 \text{ bis})$$

with the eddy viscosity, $\tilde{\nu}_t$, being obtained from a depth-averaged form of the k-ϵ Kolmogorov-Prandtl expression:

$$\tilde{\nu}_t = c_\mu \frac{\tilde{k}^2}{\tilde{\epsilon}} \qquad . \qquad (5.58)$$

As stated in Section 5.5.2, the overbar, ~, indicates that the parameter is not a true depth-averaged quantity in the sense of the rigorous definition presented in equation (5.16) for the velocity components U and V, but is a value which produces the appropriate depth-integrated Reynolds stresses when inserted into equation (5.32). Rastogi and Rodi (1978) and McGuirk and Rodi (1978) modified the standard k-ϵ expressions for use in conjunction with depth-averaged computations; they postulated that the spatial variations of $\tilde{k}$ and $\tilde{\epsilon}$ could be obtained from the following transport equations

$$\frac{\partial(U\tilde{k})}{\partial x} + \frac{\partial(V\tilde{k})}{\partial y} = \frac{\partial}{\partial x}\left[\frac{\tilde{\nu}_t}{\sigma_k}\frac{\partial\tilde{k}}{\partial x}\right] + \frac{\partial}{\partial y}\left[\frac{\tilde{\nu}_t}{\sigma_k}\frac{\partial\tilde{k}}{\partial y}\right] + P_h + P_{kv} - \tilde{\epsilon} \qquad (5.59)$$

and

$$\frac{\partial(U\tilde{\epsilon})}{\partial x} + \frac{\partial(V\tilde{\epsilon})}{\partial y} = \frac{\partial}{\partial x}\left[\frac{\tilde{\nu}_t}{\sigma_\epsilon}\frac{\partial\tilde{\epsilon}}{\partial x}\right] + \frac{\partial}{\partial y}\left[\frac{\tilde{\nu}_t}{\sigma_\epsilon}\frac{\partial\tilde{\epsilon}}{\partial y}\right] + c_{1\epsilon}\frac{\tilde{\epsilon}}{\tilde{k}}P_h + P_{\epsilon v} - c_{2\epsilon}\frac{\tilde{\epsilon}^2}{\tilde{k}} \qquad (5.60)$$

where P_h represents the production of turbulent kinetic energy due to the interaction of the Reynolds stresses with the *horizontal* velocity gradients, P_{kv} and $P_{\epsilon v}$ represent the source terms due to the non-uniformity of the vertical profile (discussed later),

and the empirical coefficients c_μ, $c_{1\epsilon}$, $c_{2\epsilon}$, σ_k and σ_ϵ have the same values as the standard k-ϵ model (Table 5.1).

P_h is derived from the general three-dimensional production term,

$$P = \nu_t\left[\frac{\partial\bar{u}_i}{\partial x_j} + \frac{\partial\bar{u}_j}{\partial x_i}\right]\frac{\partial\bar{u}_i}{\partial x_j} \quad , \qquad (5.61)$$

by exchanging the depth-dependent velocity components ($\bar{u}_i$ and $\bar{u}_j$) for depth-averaged quantities (U and V) and summing the contributions from the *two* horizontal coordinate directions. i.e.

$$P_h = \tilde{\nu}_t\left[\underset{i=1,j=1}{\left[\frac{\partial U}{\partial x} + \frac{\partial U}{\partial x}\right]\frac{\partial U}{\partial x}} + \underset{i=1,j=2}{\left[\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x}\right]\frac{\partial U}{\partial y}} + \underset{i=2,j=1}{\left[\frac{\partial V}{\partial x} + \frac{\partial U}{\partial y}\right]\frac{\partial V}{\partial x}} + \underset{i=2,j=2}{\left[\frac{\partial V}{\partial y} + \frac{\partial V}{\partial y}\right]\frac{\partial V}{\partial y}}\right] \qquad (5.62)$$

which may be simplified to

$$P_h = \tilde{\nu}_t\left[2\left[\frac{\partial U}{\partial x}\right]^2 + 2\left[\frac{\partial V}{\partial y}\right]^2 + \left[\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x}\right]^2\right] \quad . \qquad (5.63)$$

The later work of Rodi et al. (1981) and Rodi's review articles (1984a and 1984b) propose slightly different advective acceleration terms:

$$U \frac{\partial \tilde{k}}{\partial x} + V \frac{\partial \tilde{k}}{\partial y} = \frac{\partial}{\partial x}\left[\frac{\tilde{\nu}_t}{\sigma_k} \frac{\partial \tilde{k}}{\partial x} \right] + \frac{\partial}{\partial y}\left[\frac{\tilde{\nu}_t}{\sigma_k} \frac{\partial \tilde{k}}{\partial y} \right] + P_h + P_{kv} - \tilde{\epsilon} \qquad (5.64)$$

and

$$U \frac{\partial \tilde{\epsilon}}{\partial x} + V \frac{\partial \tilde{\epsilon}}{\partial y} = \frac{\partial}{\partial x}\left[\frac{\tilde{\nu}_t}{\sigma_\epsilon} \frac{\partial \tilde{\epsilon}}{\partial x} \right] + \frac{\partial}{\partial y}\left[\frac{\tilde{\nu}_t}{\sigma_\epsilon} \frac{\partial \tilde{\epsilon}}{\partial y} \right] + c_{1\epsilon} \frac{\tilde{\epsilon}}{\tilde{k}} P_h + P_{\epsilon v} - c_{2\epsilon} \frac{\tilde{\epsilon}^2}{\tilde{k}} . \qquad (5.65)$$

The original depth-averaged k-ϵ equations (5.59) and (5.60) can be shown to be equivalent to the expressions presented above, provided the depth derivatives, $\partial D/\partial x$ and $\partial D/\partial y$ are ignored for the purposes of the continuity equation.* Equations (5.64) and (5.65) define the depth-averaged k-ϵ turbulence model adopted for the present numerical study. Although the equations, as presented by Rodi, are only applicable to steady-state conditions, they can easily be modified to account for transient phenomena by adding the time derivatives, $\partial \tilde{k}/\partial t$ and $\partial \tilde{\epsilon}/\partial t$ to the left-hand sides of the appropriate expressions. Hence the turbulence transport relationships used in the current investigation are of the form:

$$\frac{\partial \tilde{k}}{\partial t} + U \frac{\partial \tilde{k}}{\partial x} + V \frac{\partial \tilde{k}}{\partial y} = \frac{\partial}{\partial x}\left[\frac{\tilde{\nu}_t}{\sigma_k} \frac{\partial \tilde{k}}{\partial x} \right] + \frac{\partial}{\partial y}\left[\frac{\tilde{\nu}_t}{\sigma_k} \frac{\partial \tilde{k}}{\partial y} \right] + P_h + P_{kv} - \tilde{\epsilon}$$

and (5.66)

$$\frac{\partial \tilde{\epsilon}}{\partial t} + U \frac{\partial \tilde{\epsilon}}{\partial x} + V \frac{\partial \tilde{\epsilon}}{\partial y} = \frac{\partial}{\partial x}\left[\frac{\tilde{\nu}_t}{\sigma_\epsilon} \frac{\partial \tilde{\epsilon}}{\partial x} \right] + \frac{\partial}{\partial y}\left[\frac{\tilde{\nu}_t}{\sigma_\epsilon} \frac{\partial \tilde{\epsilon}}{\partial y} \right] + c_{1\epsilon} \frac{\tilde{\epsilon}}{\tilde{k}} P_h + P_{\epsilon v} - c_{2\epsilon} \frac{\tilde{\epsilon}^2}{\tilde{k}} . \qquad (5.67)$$

* McGuirk and Rodi (1978, 1979) assumed, for the purposes of the continuity equation, that the depth variations were small in comparison to the velocity derivatives. Furthermore, since they adopted a rigid-lid technique (i.e. $\partial \zeta/\partial t = 0$), their continuity equation could be written as

$$\frac{\partial \zeta}{\partial t} + \frac{\partial (UD)}{\partial x} + \frac{\partial (VD)}{\partial y} = \frac{\partial \zeta}{\partial t} + U \frac{\partial D}{\partial x} + D \frac{\partial U}{\partial x} + V \frac{\partial D}{\partial y} + D \frac{\partial V}{\partial y} \approx \frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} = 0$$

Multiplying the simplified continuity expression by $\tilde{k}$ or $\tilde{\epsilon}$ and adding the result to the appropriate formula presented in equation (5.64) or (5.65) leads to the original depth-averaged k-ϵ model.

The source terms, P_{kv} and $P_{\epsilon v}$, originate from the non-uniformity of the velocity profile in the vertical direction, and represent the generation of turbulent kinetic energy and viscous dissipation not taken into account by the 'horizontal' production term, P_h. Rastogi and Rodi (1978) suggested that the major contributions to these additional terms would come from the interaction of large turbulent shear stresses and velocity gradients in the vicinity of the bed, and consequently the source terms would be strongly influenced by bottom-roughness. Therefore, Rastogi and Rodi related P_{kv} and $P_{\epsilon v}$ to the friction velocity, U_*, by defining

$$P_{kv} = c_k \frac{U_*^3}{D} \quad \text{and} \quad P_{\epsilon v} = c_\epsilon \frac{U_*^4}{D^2} \quad , \tag{5.68}$$

where c_k and c_ϵ are two further empirical coefficients.

5.6.1 Determination of c_k and c_ϵ

The analytical evaluation of the coefficients, c_k and c_ϵ is accomplished by considering the central portion of an infinitely wide, constant depth channel under normal flow conditions. All gradients with respect to x, y and t are equated to zero and therefore the only remaining terms in the k-ϵ expressions are the generation and destruction components. Thus, equations (5.66) and (5.67) reduce to

$$c_k \frac{U_*^3}{D} - \tilde{\epsilon}_u = 0 \tag{5.69}$$

and

$$c_\epsilon \frac{U_*^4}{D^2} - c_{2\epsilon} \frac{\tilde{\epsilon}_u^2}{\tilde{k}_u} = 0 \tag{5.70}$$

where the subscript, u, relates to the values at the *undisturbed* centre of the channel.

Furthermore, the flow is assumed to be entirely in the x-direction, with no bed slope in the y-direction (i.e., V = 0). For normal flow, the rate of energy dissipation due to the turbulent processes can be related to the slope of the energy line, S, via the relationship

$$\tilde{\epsilon}_u = S\ g\ U \quad , \tag{5.71}$$

whilst the friction (or shear) velocity is given by

$$U_* = (S\ g\ D)^{\frac{1}{2}} \quad . \tag{5.72}$$

Fundamentally, the shear velocity can also be determined from the friction factor, c_f (see Section 5.7):

$$U_* = (c_f U^2)^{\frac{1}{2}} \quad . \tag{5.73}$$

Substituting equations (5.71), (5.72) and (5.73) into the k–expression shown in (5.69) leads to

$$S g U \left[c_k (c_f)^{\frac{1}{2}} - 1 \right] = 0 \tag{5.74}$$

or,

$$c_k = \frac{1}{c_f^{\frac{1}{2}}} \quad . \tag{5.75}$$

The determination of c_ϵ is accomplished by applying the depth–averaged Kolmogorov–Prandtl relationship (5.58) to the undisturbed central section of the channel i.e.,

$$\tilde{\nu}_t = c_\mu \frac{\tilde{k}_u^2}{\tilde{\epsilon}_u} \quad . \tag{5.76}$$

The above expression is then employed to eliminate $\tilde{k}_u$ from the denominator of equation (5.70); it can be shown that after rearrangement and simplification, c_ϵ is given by

$$c_\epsilon = \left[\frac{\tilde{\nu}_t}{U_* D} \right]^{-\frac{1}{2}} \frac{c_{2\epsilon}}{c_f^{3/4}} c_\mu^{\frac{1}{2}} \quad . \tag{5.77}$$

Using Laufer's experimental data (1951), Rastogi and Rodi (1978) prescribed

$$\frac{\tilde{\nu}_t}{U_* D} = 0.0765 \quad . \tag{5.78}$$

Substituting this result into equation (5.77) defines the coefficient, c_ϵ as

$$c_\epsilon = 3.6 \frac{c_{2\epsilon}}{c_f^{3/4}} c_\mu^{\frac{1}{2}} \quad . \tag{5.79}$$

5.7 Bottom friction

The components of bed shear stress (τ_{bx} and τ_{by}) are related to the depth-averaged velocity components via a quadratic friction law:

$$\frac{\tau_{bx}}{\rho} = c_f U (U^2 + V^2)^{\frac{1}{2}} \quad ; \quad \frac{\tau_{by}}{\rho} = c_f V (U^2 + V^2)^{\frac{1}{2}} \qquad (5.80)$$

where c_f is an empirical friction coefficient which usually depends upon the bottom roughness. For smooth beds, however, the friction coefficient is determined solely by the Reynolds number, and can be approximated from the formula presented by Schlichting (1968) for channel flow:

$$c_f = 0.027 \left[\frac{\nu}{U R} \right]^{\frac{1}{4}} \qquad (5.81)$$

where ν is the coefficient of kinematic viscosity,
and R is the hydraulic radius of the channel cross-section.

In hydrodynamic situations with near uniform depth, Rodi (1984b) states that it is usually sufficient to employ a constant value of c_f across the entire flow domain.

With rough beds, c_f can be determined either from the Chezy friction law:

$$c_f = \frac{g}{C^2} \quad , \qquad (5.82)$$

or alternatively from Manning's equation:

$$c_f = \frac{n^2 g}{D^{1/3}} \qquad (5.83)$$

where C is the Chezy coefficient,
and n is the Manning roughness factor.

Finally, the friction (or shear) velocity, U_*, used in the evaluation of the k-ϵ source terms, P_{kv} and $P_{\epsilon v}$ is defined as

$$U_* = \left[\frac{\tau_b}{\rho} \right]^{\frac{1}{2}} \qquad (5.84)$$

where τ_b is the magnitude of the bottom stress vector. Since τ_b is given by

$$\tau_b = \left[\tau_{bx}^2 + \tau_{by}^2 \right]^{\frac{1}{2}} \quad , \qquad (5.85)$$

the shear velocity can be expressed in terms of the depth-averaged velocity components, U and V:

$$U_* = \left[c_f \, (U^2 + V^2) \right]^{\frac{1}{2}} \quad . \qquad (5.86)$$

5.8 Wind stresses

The depth-averaged momentum expressions, described in Section 5.3, also contain wind stress components, τ_{wx} and τ_{wy}. Since the aim of the present numerical study is to predict jet-forced (momentum driven) reservoir circulation patterns, wind stresses were ignored for the purposes of the main investigation. Moreover, little additional information concerning the *velocity* structure can be obtained by the inclusion of surface stresses in a depth-averaged model. This is because the depth-integration procedure removes all the important detail appertaining to the velocity profiles in the vertical direction; the modelling of circulation patterns caused by wind stresses must employ a three-dimensional computational scheme (or at least utilise 'layered' hydrodynamic equations) and must also consider the important momentum mixing processes caused by wind-induced surface waves. Nevertheless, if the numerical simulation is concerned with surface elevation effects rather than the velocity structure of the flow, then depth-averaged computations are usually satisfactory (e.g., 2-D storm surge calculations - Hansen, 1962; Reid and Bodine, 1968; Heaps, 1969 and 1977; Wanstrath, 1977). Furthermore, the ability to include wind shear stress components proved particularly useful during the early stages of program development, since computer predicted wind-induced surface elevation gradients could readily be checked against the analytical solution determined from a force balance calculation (see Section 7.2). Although this particular validation was undertaken with specific wind stresses (e.g., τ_{wx} = 0.1 N/m^2 and τ_{wy} = 0.0 N/m^2), it is worthwhile, as an aside, considering the commonly used formulae which relate the stress components to the wind speed.

Van Dorn's method (1953) for evaluating the wind stress was employed by Reid and Bodine (1968) and recommended in the Shore Protection Manual (1977). The stress components are determined from

$$\tau_{wx} = \rho \, K \, W \, W_x \quad ; \quad \tau_{wy} = \rho \, K \, W \, W_y \qquad (5.87)$$

where W is the magnitude of the wind velocity (in ms^{-1}) at an elevation of 10m above the water surface,

W_x and W_y are the components of wind velocity in the x- and y-directions,

ρ is the density of the water (in kgm^{-3}),

and K is an empirical wind stress coefficient, dependent upon the magnitude of the wind velocity:

$$\left.\begin{aligned} K &= K_1 && \text{for } W \leq W_{cr} \\ \text{and} & \\ K &= K_1 + K_2 \left[1 - \frac{W_{cr}}{W} \right]^2 && \text{for } W > W_{cr} \quad . \end{aligned}\right\} \qquad (5.88)$$

W_{cr} is a 'critical' wind speed defined as 7.2 ms^{-1} (14 knots) and the coefficients K_1 and K_2 are taken to be 1.1 x 10^{-6} and 2.5 x 10^{-6}, respectively.

Neumann and Pierson (1966) presented an alternative wind model, which determined the surface shear stresses from

$$\tau_{wx} = \rho_a \, C^* \, W \, W_x \quad ; \quad \tau_{wy} = \rho_a \, C^* \, W \, W_y \qquad (5.89)$$

where ρ_a is the density of the *air* ($\approx$ 1.3 kgm^{-3}),

and C^* is an empirical air-water interface resistance coefficient, given by the formula:

$$C^* = 9.0 \times 10^{-3} \; W^{-\frac{1}{2}} \quad . \qquad (5.90)$$

5.9 Transformation of the depth-averaged hydrodynamic equations

The previous sections of this chapter express the governing depth-averaged hydrodynamic equations in a Cartesian reference frame. However, since the numerical computations are to be performed on a non-orthogonal mesh, it is necessary to convert the Cartesian formulae into 'transformed' equations, written in terms of the boundary-fitted coordinates, ξ and η. Following the work of Johnson (1980, 1982), Johnson et al. (1982), Häuser et al. (1985, 1986b) and Raghunath et al. (1987), only the independent variables (x,y) are transformed. This overcomes the difficulties associated with the unwieldy diffusive terms which are encountered when implementing contravariant/covariant velocity techniques on non-orthogonal coordinate systems (Sheng and Hirsh, 1984). A further advantage in retaining the Cartesian velocity components as dependent variables, arises from the fact that only the partial differential equations must be transformed. Algebraic expressions (e.g., bottom friction and shear velocity relationships) are unaltered.

As discussed in Section 4.3.1, it was decided that a non-conservative derivative transformation would be more suited to the present numerical study; this represents a significant departure from Johnson's pioneering work. Although Häuser et al. (1985, 1986b) and Raghunath et al. (1987) utilised the non-conservative approach, they only examined the linearised shallow water equations (neglecting the advective acceleration, bed friction and diffusive terms). Consequently, the present investigation may be considered an extension of Häuser et al.'s technique. The computational procedure described in the following chapter, however, uses an alternating-direction-implicit method and therefore bears little resemblance to the explicit scheme developed by Häuser et al.

The transformation of the governing partial differential equations of motion is accomplished using the non-conservative derivative relationships presented in Section 4.3.1:

$$f_x = \frac{1}{J} (y_\eta f_\xi - y_\xi f_\eta) \qquad (4.6a\ \text{bis})$$

and

$$f_y = \frac{1}{J} (x_\xi f_\eta - x_\eta f_\xi) \quad . \qquad (4.6b\ \text{bis})$$

Substituting these expressions into all partial derivatives involving x or y, leads to the depth-averaged continuity equation (5.17) being recast as

$$\frac{\partial \zeta}{\partial t} + \frac{1}{J} \left[y_\eta \frac{\partial (UD)}{\partial \xi} - y_\xi \frac{\partial (UD)}{\partial \eta} + x_\xi \frac{\partial (VD)}{\partial \eta} - x_\eta \frac{\partial (VD)}{\partial \xi} \right] = 0 \quad , \qquad (5.91)$$

whilst the non-conservative momentum relationships (5.24a and 5.24b) are rewritten as:

<u>x-momentum</u>

$$\frac{\partial U}{\partial t} + \frac{1}{J} \left[y_\eta U \frac{\partial U}{\partial \xi} - y_\xi U \frac{\partial U}{\partial \eta} + x_\xi V \frac{\partial U}{\partial \eta} - x_\eta V \frac{\partial U}{\partial \xi} \right] - fV + \frac{g}{J} \left[y_\eta \frac{\partial \zeta}{\partial \xi} - y_\xi \frac{\partial \zeta}{\partial \eta} \right]$$

$$- \frac{\tau_{wx} - \tau_{bx}}{\rho D} - \frac{1}{\rho D J} \left[y_\eta \frac{\partial (DT_{xx})}{\partial \xi} - y_\xi \frac{\partial (DT_{xx})}{\partial \eta} + x_\xi \frac{\partial (DT_{xy})}{\partial \eta} - x_\eta \frac{\partial (DT_{xy})}{\partial \xi} \right] = 0 \qquad (5.92a)$$

and

y-momentum

$$\frac{\partial V}{\partial t} + \frac{1}{J}\left[y_\eta U\frac{\partial V}{\partial \xi} - y_\xi U\frac{\partial V}{\partial \eta} + x_\xi V\frac{\partial V}{\partial \eta} - x_\eta V\frac{\partial V}{\partial \xi} \right] + fU + \frac{g}{J}\left[x_\xi\frac{\partial \zeta}{\partial \eta} - x_\eta\frac{\partial \zeta}{\partial \xi} \right]$$

$$- \frac{\tau_{wy} - \tau_{by}}{\rho D} - \frac{1}{\rho DJ}\left[y_\eta\frac{\partial (DT_{xy})}{\partial \xi} - y_\xi\frac{\partial (DT_{xy})}{\partial \eta} + x_\xi\frac{\partial (DT_{yy})}{\partial \eta} - x_\eta\frac{\partial (DT_{yy})}{\partial \xi} \right] = 0 \qquad (5.92b)$$

Since the effective stresses, T_{xx}, T_{xy} and T_{yy} are evaluated from velocity derivatives, it is also necessary to transform equations (5.33a), (5.33b) and (5.33c) into the boundary-fitted coordinate system:

$$T_{xx} = 2\rho\tilde{\nu}_t\frac{1}{J}\left(y_\eta U_\xi - y_\xi U_\eta \right) - \frac{2}{3}\rho\tilde{k} \qquad (5.93a)$$

$$T_{xy} = \rho\tilde{\nu}_t\frac{1}{J}\left(x_\xi U_\eta - x_\eta U_\xi + y_\eta V_\xi - y_\xi V_\eta \right) \qquad (5.93b)$$

$$T_{yy} = 2\rho\tilde{\nu}_t\frac{1}{J}\left(x_\xi V_\eta - x_\eta V_\xi \right) - \frac{2}{3}\rho\tilde{k} \quad . \qquad (5.93c)$$

The partial differential transport equations for the depth-averaged k-ϵ turbulence model are converted into:

k-transport

$$\frac{\partial \tilde{k}}{\partial t} + \frac{1}{J}\left[y_\eta U\frac{\partial \tilde{k}}{\partial \xi} - y_\xi U\frac{\partial \tilde{k}}{\partial \eta} + x_\xi V\frac{\partial \tilde{k}}{\partial \eta} - x_\eta V\frac{\partial \tilde{k}}{\partial \xi} \right] = \frac{1}{J}\left[y_\eta\frac{\partial Kx}{\partial \xi} - y_\xi\frac{\partial Kx}{\partial \eta} + x_\xi\frac{\partial Ky}{\partial \eta} - x_\eta\frac{\partial Ky}{\partial \xi} \right] + P_h + P_{kv} - \tilde{\epsilon} \qquad (5.94a)$$

and

ϵ-transport

$$\frac{\partial \tilde{\epsilon}}{\partial t} + \frac{1}{J}\left[y_\eta U\frac{\partial \tilde{\epsilon}}{\partial \xi} - y_\xi U\frac{\partial \tilde{\epsilon}}{\partial \eta} + x_\xi V\frac{\partial \tilde{\epsilon}}{\partial \eta} - x_\eta V\frac{\partial \tilde{\epsilon}}{\partial \xi} \right] = \frac{1}{J}\left[y_\eta\frac{\partial Ex}{\partial \xi} - y_\xi\frac{\partial Ex}{\partial \eta} + x_\xi\frac{\partial Ey}{\partial \eta} - x_\eta\frac{\partial Ey}{\partial \xi} \right] + c_{1\epsilon}\frac{\tilde{\epsilon}}{\tilde{k}}P_h + P_{\epsilon v} - c_{2\epsilon}\frac{\tilde{\epsilon}^2}{\tilde{k}} \qquad (5.94b)$$

where

$$Kx = \frac{\tilde{\nu}_t}{\sigma_k}\frac{\partial \tilde{k}}{\partial x} = \frac{\tilde{\nu}_t}{\sigma_k}\frac{1}{J}\left[y_\eta \frac{\partial \tilde{k}}{\partial \xi} - y_\xi \frac{\partial \tilde{k}}{\partial \eta} \right] \qquad (5.95a)$$

$$Ky = \frac{\tilde{\nu}_t}{\sigma_k}\frac{\partial \tilde{k}}{\partial y} = \frac{\tilde{\nu}_t}{\sigma_k}\frac{1}{J}\left[x_\xi \frac{\partial \tilde{k}}{\partial \eta} - x_\eta \frac{\partial \tilde{k}}{\partial \xi} \right] \qquad (5.95b)$$

$$Ex = \frac{\tilde{\nu}_t}{\sigma_\epsilon}\frac{\partial \tilde{\epsilon}}{\partial x} = \frac{\tilde{\nu}_t}{\sigma_\epsilon}\frac{1}{J}\left[y_\eta \frac{\partial \tilde{\epsilon}}{\partial \xi} - y_\xi \frac{\partial \tilde{\epsilon}}{\partial \eta} \right] \qquad (5.95c)$$

$$Ey = \frac{\tilde{\nu}_t}{\sigma_\epsilon}\frac{\partial \tilde{\epsilon}}{\partial y} = \frac{\tilde{\nu}_t}{\sigma_\epsilon}\frac{1}{J}\left[x_\xi \frac{\partial \tilde{\epsilon}}{\partial \eta} - x_\eta \frac{\partial \tilde{\epsilon}}{\partial \xi} \right] \qquad (5.95d)$$

and

$$P_h = \tilde{\nu}_t \left[2\left[\frac{\partial U}{\partial x}\right]^2 + 2\left[\frac{\partial V}{\partial y}\right]^2 + \left[\frac{\partial U}{\partial y} + \frac{\partial V}{\partial x}\right]^2 \right] . \qquad (5.63\ \text{bis})$$

The individual velocity gradients in the horizontal production term, P_h are recast as:

$$\left.\begin{aligned} \frac{\partial U}{\partial x} &= \frac{1}{J}\left[y_\eta \frac{\partial U}{\partial \xi} - y_\xi \frac{\partial U}{\partial \eta} \right] , \\ \frac{\partial V}{\partial x} &= \frac{1}{J}\left[y_\eta \frac{\partial V}{\partial \xi} - y_\xi \frac{\partial V}{\partial \eta} \right] , \\ \frac{\partial U}{\partial y} &= \frac{1}{J}\left[x_\xi \frac{\partial U}{\partial \eta} - x_\eta \frac{\partial U}{\partial \xi} \right] , \end{aligned}\right\} \qquad (5.96)$$

and

$$\frac{\partial V}{\partial y} = \frac{1}{J}\left[x_\xi \frac{\partial V}{\partial \eta} - x_\eta \frac{\partial V}{\partial \xi} \right] .$$

Although the transformed expressions are undoubtedly more complex than their Cartesian counterparts, the governing equations of motion can now be discretised on a regular finite-difference mesh composed of uniformly-spaced, square cells. The ability to utilise finite-difference techniques for curved or irregular flow boundaries (in the physical plane) is probably sufficient incentive to tolerate the disadvantages caused by the increased complexity of the governing hydrodynamic equations. The numerical solution of the transformed expressions is accomplished via a semi-implicit computational scheme, detailed in the following chapter.

5.10 Invariance of equation type

Section 5.9 presents the transformed shallow-water and k-ϵ equations which are to be used in conjunction with the non-orthogonal boundary-fitted coordinate systems, described in Chapter 4. Before continuing with further development work, however, it is necessary to check whether the coordinate mapping preserves the equation *type*, since this is considered to be an essential prerequisite for a successful numerical simulation.

Instead of checking each of the governing expressions separately, the preservation of equation type can be inspected for a general, second-order partial differential equation, involving a single scalar quantity, f (in an analogous manner to Thompson et al., 1977b).

Assume that the Cartesian form of the second-order partial differential equation is

$$A\, f_{xx} + B\, f_{xy} + C\, f_{yy} + D\, f_x + E\, f_y + F = 0 \qquad (5.97)$$

where, A, B, C, D, E and F are continuous functions of x, y and possibly f.

The equation type is determined from the coefficients of the second-order derivatives:

$$\left.\begin{array}{lll} B^2 - 4AC < 0 & \Rightarrow & \text{Elliptic} \\ B^2 - 4AC = 0 & \Rightarrow & \text{Parabolic} \\ B^2 - 4AC > 0 & \Rightarrow & \text{Hyperbolic} \end{array}\right\} \qquad (5.98)$$

However, the derivatives, f_{xx}, f_{xy} and f_{yy} can also be expressed in terms of the boundary-fitted coordinates, ξ and η by differentiating equations (4.6a) and (4.6b) with respect to x and y:

$$\begin{aligned} f_{xx} = {} & (y_\eta^2 f_{\xi\xi} - 2y_\xi y_\eta f_{\xi\eta} + y_\xi^2 f_{\eta\eta})/J^2 \\ & + \left[(y_\eta^2 y_{\xi\xi} - 2y_\xi y_\eta y_{\xi\eta} + y_\xi^2 y_{\eta\eta})(x_\eta f_\xi - x_\xi f_\eta)\right. \\ & + \left.(y_\eta^2 x_{\xi\xi} - 2y_\xi y_\eta x_{\xi\eta} + y_\xi^2 x_{\eta\eta})(y_\xi f_\eta - y_\eta f_\xi)\right]/J^3 \end{aligned} \qquad (5.99)$$

$$f_{xy} = \left[(x_\xi y_\eta + x_\eta y_\xi) f_{\xi\eta} - x_\xi y_\xi f_{\eta\eta} - x_\eta y_\eta f_{\xi\xi}\right]/J^2$$

$$+ \left[(x_\xi y_{\eta\eta} - x_\eta y_{\xi\eta})/J^2 + (x_\eta y_\eta J_\xi - x_\xi y_\eta J_\eta)/J^3\right] f_\xi$$

$$+ \left[(x_\eta y_{\xi\xi} - x_\xi y_{\xi\eta})/J^2 + (x_\xi y_\xi J_\eta - x_\eta y_\xi J_\xi)/J^3\right] f_\eta \qquad (5.100)$$

$$f_{yy} = (x_\eta{}^2 f_{\xi\xi} - 2x_\xi x_\eta f_{\xi\eta} + x_\xi{}^2 f_{\eta\eta})/J^2$$

$$+ \left[(x_\eta{}^2 y_{\xi\xi} - 2x_\xi x_\eta y_{\xi\eta} + x_\xi{}^2 y_{\eta\eta})(x_\eta f_\xi - x_\xi f_\eta)\right.$$

$$+ \left.(x_\eta{}^2 x_{\xi\xi} - 2x_\xi x_\eta x_{\xi\eta} + x_\xi{}^2 x_{\eta\eta})(y_\xi f_\eta - y_\eta f_\xi)\right]/J^3 \qquad (5.101)$$

Substitution of the above formulae into equation (5.97), and extensive rearrangement produces the transformed partial differential expression:

$$A^* f_{\xi\xi} + B^* f_{\xi\eta} + C^* f_{\eta\eta} + D^* f_\xi + E^* f_\eta + F^* = 0 \qquad (5.102)$$

where

$$A^* = A\xi_x{}^2 + B\xi_x\xi_y + C\xi_y{}^2$$

$$B^* = 2A\xi_x\eta_x + B(\xi_x\eta_y + \xi_y\eta_x) + 2C\xi_y\eta_y$$

$$C^* = A\eta_x{}^2 + B\eta_x\eta_y + C\eta_y{}^2$$

$$D^* = A\xi_{xx} + B\xi_{xy} + C\xi_{yy} + D\xi_x + E\xi_y$$

$$E^* = A\eta_{xx} + B\eta_{xy} + C\eta_{yy} + D\eta_x + E\eta_y$$

and $$F^* = F \quad .$$

In the transformed plane, the equation type is therefore determined from

$$(B^*)^2 - 4A^*C^* = (2A\xi_x\eta_x + B(\xi_x\eta_y + \xi_y\eta_x) + 2C\xi_y\eta_y)^2$$

$$- 4(A\xi_x{}^2 + B\xi_x\xi_y + C\xi_y{}^2)(A\eta_x{}^2 + B\eta_x\eta_y + C\eta_y{}^2) \quad . \qquad (5.103)$$

Simplification of this relationship leads to

$$(B^*)^2 - 4A^*C^* = (B^2 - 4AC)(\xi_x\eta_y - \xi_y\eta_x)^2$$

$$= (B^2 - 4AC)/J^2 \quad . \qquad (5.104)$$

Since $J^2 > 0$, $(B^*)^2 - 4A^*C^*$ has the same sign as $B^2 - 4AC$ and hence the mapping procedure preserves equation type.

5.11 Concluding remarks

The chapter has reviewed the standard depth-averaged Reynolds expressions (the so-called shallow water equations) and has given a brief resume of the various orders of closure that are currently available for the mathematical modelling of turbulence. For the purposes of the depth-integrated momentum equations, the present study adopts a non-conservative/advective approach since this corresponds with the decision, taken at the end of Section 4.3.1, to use the non-conservative forms of derivative transformation (equations 4.6a and 4.6b).

Regarding the solution of the depth-averaged continuity equation, the latter part of Section 5.2 discusses the possibilities of using a rigid-lid technique to convert the hyperbolic shallow water expressions into elliptic Navier-Stokes equations. Long wavelength surface oscillations are thereby excluded from the simulation, alleviating the severity of the Courant time step restriction. However, detailed comparison of the numerical results from the free-surface and rigid-lid procedures has been deferred until Chapter 7.

Since the implementation of a one-equation turbulence model (Section 5.5.4) requires the specification of a mixing-length parameter, L_m, this approach was not attempted in the present computational study. Instead, after developing and testing the constant/depth-averaged algebraic eddy viscosity models (Sections 5.5.3a and 5.5.3c), the investigation was immediately directed towards the solution of the two-equation ($k-\epsilon$) turbulence model.

The final sections of this chapter have been concerned with the transformation of the governing Cartesian equations into expressions that can be discretised on the computational (ξ,η) plane. Section 5.9, detailing the conversion of the Cartesian formulae, underpins the whole methodology of the boundary-fitted procedure. The next chapter presents numerical methods for solving the transformed depth-averaged

Reynolds equations, and adapts Leendertse's (1967) semi-implicit Cartesian numerical formulation for use with boundary-fitted coordinate systems.

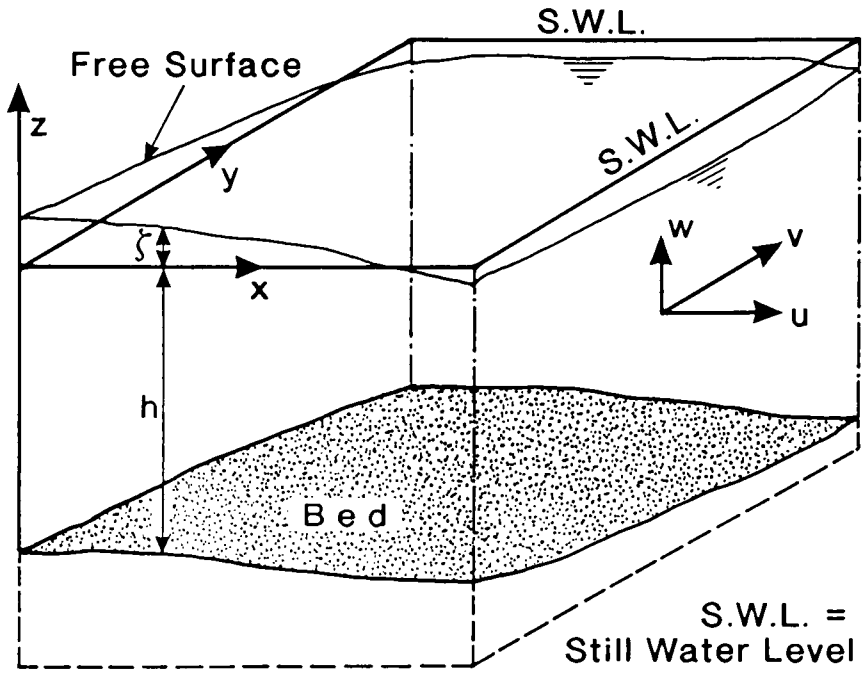


Figure 5.1 Coordinate system definition

CHAPTER 6
NUMERICAL SOLUTION OF THE TRANSFORMED SHALLOW WATER EQUATIONS

6.1 Introduction

Chapter 5 has presented the depth-averaged Reynolds and turbulence transport equations which are applicable to two-dimensional, nearly-horizontal flow. The governing equations ignore the velocity profiles in the vertical direction together with any temperature/density stratification, and are therefore only valid for relatively shallow flow regimes where the water depth is small compared to the lateral dimensions of the hydraulic domain. Since the governing hydrodynamic expressions are non-linear and interdependent, analytical solutions do not exist and consequently numerical methods must be employed.

Although boundary-fitted coordinate systems were developed for use with finite-difference procedures (Thompson et al., 1974 onwards), recent developments in finite-volume methods (Rhie and Chow, 1982; Perić, 1985; Miller and Schmidt, 1988) indicate that these newer techniques are now a plausible alternative to the finite-difference approach. Finite-volume methods discretise the governing expressions by integrating the partial differential equations of motion over a control volume, thereby allowing the discretised equations to conserve the mass and momentum fluxes in an exact manner. Bernard (1989a, 1989b) has presented a hybrid boundary-fitted Navier-Stokes equation solver which uses finite-volume approximations for first derivatives and finite-difference approximations for second derivatives. However, the lack of published research on the use of finite-volume techniques for free-surface computations, the questions concerning the choice of staggered or collocated grids (Perić et al., 1988), and the fact that the implementation of the shallow water equations on boundary-fitted meshes is itself a relatively novel technique, led to the decision to adopt a form of Leendertse's (1967) well-established finite-difference computational procedure.

6.2 Staggered grid arrangement

Figure 6.1 shows the spatial distribution of the variables for a typical cell within the staggered grid system. The layout differs from that presented by Leendertse because the water surface elevation, ζ, and depth, D, are defined at the same position; this approach is therefore similar to the Cartesian staggered grid

arrangement used by Butler (1978a, 1978b), Vemulakonda et al. (1985), Walker (1987) and Dong (1987). Figure 6.1b shows a typical representative cell in the physical domain, whilst Figure 6.1a shows the same element in the orthogonal, transformed finite-difference mesh. All scalar variables (ζ, h, D, k and ϵ) are stored at the centre of the computational cell, whereas the Cartesian velocity components, U and V, are defined at the mid-points of the vertical and horizontal faces. For comparison, the standard Cartesian staggered cell layout utilised by Butler, etc., is presented in Figure 6.1c.

Throughout the numerical scheme, the hydrodynamic variables (U, V, ζ, D, k and ϵ) are required at locations where they are not explicitly stored. This problem is overcome by using either a two- or four-point average of the values surrounding the particular position. In a similar fashion to other boundary-fitted finite-difference procedures (e.g., Johnson, 1980; Häuser et al., 1985, 1986b; Raghunath et al., 1987), the present computational scheme employs an averaging interpolation in the *transformed* plane, even though this is not theoretically compatible with the varying cell-sizes of the physical domain. It is envisaged that a higher order, weighted interpolation, taking into account the variation in the transformation derivatives (x_ξ, x_η, y_ξ and y_η), could be employed to improve the spatial accuracy of the difference equations. However, for the purposes of the numerical scheme presented here, the former procedure is utilised. Provided the cell-size variations (in the x-y plane) are not excessive, the straightforward averaging technique is deemed to be entirely adequate (Bernard, 1989c).

Following Johnson (1980, 1982), Häuser et al. (1985, 1986b) and Raghunath et al. (1987), the partial derivatives, x_ξ, x_η, y_ξ and y_η are generated at every intersection between the solid/dashed mesh lines shown in Figure 6.1a in order to avoid unnecessary interpolation inaccuracies. A diagrammatic layout of the staggered grid system devised for the present study is presented in Figure 6.2; it can be seen that two interrelated spatial indexing systems have been employed:

i and j refer to the U, V, ζ, h, D, k and ϵ variables, and are similar to the indices adopted in Walker's numerical scheme (1987), whilst

I and J define the location of the geometric variables - x, y, x_ξ, x_η, y_ξ and y_η.

With reference to Figure 6.2, the hydrodynamic variables are only stored at locations labelled with a U, V or O, whereas the geometric variables are evaluated and held at all (I,J) points. This split system enables the transformation derivatives to be

defined at the cell centres *and* at the cell mid-faces without over-specifying the array sizes of the hydrodynamic variables; storing U, V, ζ, h, D, k and ϵ on the (I,J) grid would be uneconomical in terms of memory usage because only one quarter of the array locations would be utilised.

A large proportion of the run-time memory requirements of the boundary-fitted shallow water equation program is used for storing data applicable to the numerical mapping. This may be the reason why some authors (for example, Wijbenga, 1985a and 1985b; Willemse et al., 1985) prefer to define the transformation derivatives at the same grid increment as utilised for the U, V and ζ parameters; the storage requirements of the numerical mapping data are thus reduced by a factor of four, although inevitably, interpolation techniques must then be used to calculate the transformation derivatives at certain positions. The results shown in Section 7.3 of the next chapter, however, vindicate the use of the computationally more expensive 'double-mesh' system.

Referring to Figure 6.2, the relationship between the two sets of ξ-direction spatial indices is given by

$$I = 2i - 1 \qquad \text{for the } \zeta, \text{ h, D, k, } \epsilon \text{ or V-velocity positions} \qquad (6.1)$$

and

$$I = 2i \qquad \text{for the U-velocity positions,} \qquad (6.2)$$

whilst the η-indexing formulae are

$$J = 2j - 1 \qquad \text{for the } \zeta, \text{ h, D, k, } \epsilon \text{ or U-velocity positions} \qquad (6.3)$$

and

$$J = 2j \qquad \text{for the V-velocity positions.} \qquad (6.4)$$

For example, if the x-momentum equation is to be updated at the position $U_{i,j}$, then the transformation derivatives applicable to this particular point are stored at the fine grid location: $I = 2i$, $J = 2j - 1$.

Johnson (1980) and Häuser et al. (1985) chose to specify a staggered grid arrangement with the U- and V-velocities defined along every cell face, as illustrated in Figures 6.3a and 6.3b. This grid structure has the advantage that the mass and momentum fluxes across the cells can be determined more accurately, since the discretised equations have only to rely upon two-point averaging techniques (cf., the present numerical scheme requires two- and four-point interpolations). However, the

spatial distribution of hydrodynamic variables, suggested by Johnson, is only suitable for explicit flow calculations; Johnson developed a novel accelerated Gauss-Seidel solution technique for the surface elevation computations and solved the velocity fields using an explicit scheme, whilst Häuser et al. employed a standard, explicit forward-time central-space (F.T.C.S.) numerical scheme for all hydrodynamic variables. Investigations carried out by the present author indicate that the cell structure shown in Figure 6.3 is unsuitable for alternating-direction-implicit (A.D.I.) flow computations because the U- and V- velocity components are defined at the same position. In order to substantiate this claim, it is necessary to consider the nature of the equations generated during a typical implicit flow calculation. Figure 6.4a shows a section of the variables which are evaluated during a 'line-wise' ξ-direction release for the mesh layout proposed in Figure 6.1. As shown later, in Section 6.6.1, the application of the continuity equation to the position at $\zeta_{i,j}$ and the application of the x-momentum equation to the position at $U_{i,j}$ leads to finite-difference expressions of the form:

continuity

$$- a_{i,j}\, U^{n+\frac{1}{2}}_{i-1,j} + b_{i,j}\, \zeta^{n+\frac{1}{2}}_{i,j} - c_{i,j}\, U^{n+\frac{1}{2}}_{i,j} = k_{i,j} \qquad (6.5a)$$

and

x-momentum

$$- a'_{i,j}\, \zeta^{n+\frac{1}{2}}_{i,j} + b'_{i,j}\, U^{n+\frac{1}{2}}_{i,j} - c'_{i,j}\, \zeta^{n+\frac{1}{2}}_{i+1,j} = k'_{i,j} \qquad (6.5b)$$

where a, b, c, k, a', b', c' and k' are coefficients which are themselves composed of finite-difference expressions,

and the superscript, $^{n+\frac{1}{2}}$, is used to denote the time level [i.e., $t = (n+\frac{1}{2})\Delta t$].

Since these discretised expressions are only interrelated by the two nearest neighbouring hydrodynamic variables, the line-wise combination of many such pairs of equations will form a tri-diagonal matrix system. However, if the spatial structure adopted by Johnson (1980) is used for an implicit ξ-direction release (as shown in Figure 6.4b), then the application of the continuity equation to the position at $\zeta_{i,j}$ leads to a finite-difference relationship of the form:

$$- a_{i,j}\, U^{n+\frac{1}{2}}_{i-1,j} - a'_{i,j}\, V^{n+\frac{1}{2}}_{i-1,j} + b_{i,j}\, \zeta^{n+\frac{1}{2}}_{i,j} - c_{i,j}\, U^{n+\frac{1}{2}}_{i,j} - c'_{i,j}\, V^{n+\frac{1}{2}}_{i,j} = k_{i,j} \qquad (6.6)$$

As the above equation is interdependent upon *four* neighbouring values, the

particular arrangement of hydrodynamic variables shown in Figure 6.3 cannot produce a tri-diagonal matrix system and is therefore computationally much more expensive to solve. Accordingly, the present numerical scheme is based upon the fully staggered mesh layout presented in Figures 6.1 and 6.2.

6.3 Evaluation of the transformation derivatives

For the purposes of the hydrodynamic computations, $\Delta\xi$ and $\Delta\eta$ are defined as the distances between the surface elevation positions (see Figure 6.2). Since the geometric variables are held at the nodes of a mesh having twice the resolution of the hydrodynamic grid, the (I,J) points will thus be separated by $\frac{1}{2}\Delta\xi$ and $\frac{1}{2}\Delta\eta$. As previously explained in Chapter 4, the range of the ξ- and η-coordinates is completely arbitrary, and hence the computational mesh increments $\Delta\xi$ and $\Delta\eta$ are specified, for convenience, as unity. Consequently, the geometric variables are defined on a finite-difference mesh with cell increments of 0.5. Although this is not compatible with the boundary-fitted algorithms presented in Chapter 4, the coordinate generation source code and shallow water equation solver are two entirely separate computer programs. The (x,y) arrays, generated by the boundary-fitted coordinate code are stored for subsequent use by the hydrodynamic program, and therefore, the computational mesh increments need not be the same between the two computer programs.

The present system was designed around the fact that the majority of finite-difference expressions need to be written in terms of the hydrodynamic (i,j) grid. It is clearly advantageous to specify the most commonly used mesh with $\Delta\xi=\Delta\eta=1.0$ and to use non-standard differences for the transformation derivatives.*

Considering an arbitrary interior node (I,J), the second-order central-difference approximations to the coordinate derivatives are:

$$\left.\begin{aligned} x_\xi &= x_{I+1,J} - x_{I-1,J} \quad ; \quad y_\xi = y_{I+1,J} - y_{I-1,J} \\ x_\eta &= x_{I,J+1} - x_{I,J-1} \quad ; \quad y_\eta = y_{I,J+1} - y_{I,J-1} \end{aligned}\right\} \qquad (6.7)$$

* The transformation derivatives (x_ξ, x_η, y_ξ and y_η) are invariant with time and therefore need only be evaluated once, at the start of the hydrodynamic simulation.

Furthermore, unlike the boundary-fitted coordinate generation code, where the computer algorithm is only concerned with *interior* nodes, the current numerical scheme requires the transformation derivatives to be evaluated along the perimeter of the hydraulic domain. Hence, second-order forward/backward differences are utilised at the boundary positions. Considering a 'left-hand/vertical' boundary in the computational plane, as shown in Figure 6.5a, the forward difference approximation to the scalar derivative, $\partial f/\partial \xi$ can be shown to be

$$f'_0 = \left.\frac{\partial f}{\partial \xi}\right|_0 = -3\, f_0 + 4\, f_1 - f_2 \quad . \tag{6.8}$$

Hence, the coordinate derivatives $\partial x/\partial \xi$ and $\partial y/\partial \xi$ are evaluated as

$$\left.\begin{aligned} x\xi_0 &= -3\, x_{I,J} + 4\, x_{I+1,J} - x_{I+2,J} \\ \text{and} \\ y\xi_0 &= -3\, y_{I,J} + 4\, y_{I+1,J} - y_{I+2,J} \quad . \end{aligned}\right\} \tag{6.9}$$

Similarly, for a 'right-hand' boundary (Figure 6.5b), the scalar derivative, $\partial f/\partial \xi$ is approximated as

$$f'_0 = \left.\frac{\partial f}{\partial \xi}\right|_0 = 3\, f_0 - 4\, f_1 + f_2 \quad , \tag{6.10}$$

and therefore the transformation derivatives are

$$\left.\begin{aligned} x\xi_0 &= 3\, x_{I,J} - 4\, x_{I-1,J} + x_{I-2,J} \\ \text{and} \\ y\xi_0 &= 3\, y_{I,J} - 4\, y_{I-1,J} + y_{I-2,J} \quad . \end{aligned}\right\} \tag{6.11}$$

For the situations represented in Figures 6.5a and 6.5b, the η-direction coordinate derivatives, $\partial x/\partial \eta$ and $\partial y/\partial \eta$ are evaluated by central-differences.

In an analogous manner, forward/backward differences are used to approximate $\partial x/\partial \eta$ and $\partial y/\partial \eta$ along the 'horizontal' boundaries of the computational domain, whilst $\partial x/\partial \xi$ and $\partial y/\partial \xi$ are calculated from central-difference expressions.

6.4 Control of non-linear instability

The inclusion of the non-linear advective acceleration terms in the x- and y-momentum equations will often make it difficult for the numerical scheme to achieve stability. The destabilising effects of the advective terms are generally most

troublesome when modelling hydraulic regimes with strong recirculation regions and low values of eddy viscosity coefficient. Non-linear instabilities lead to severe grid-scale oscillations of the velocity field (which eventually overwhelm the true numerical solution) and are caused by the discretisation process preventing the transfer of turbulence energy to eddies smaller than the resolution of the computational mesh. Central-difference approximations of the non-linear accelerations are the most unstable form of differencing, and can only be used in conjunction with large values of eddy viscosity coefficient (e.g., Kuipers and Vreugdenhil, 1973; Butler, 1978a; Walker, 1987). As explained in Chapter 5, however, the value of eddy viscosity must be governed by the physics of the situation rather than the requirement to control instability, and therefore the present numerical simulation employs a second-order upwind differencing technique [similar to that proposed by Stelling (1983) and Stelling and Willemse (1984)]. The method eliminates the grid-scale oscillations of the velocity field, without introducing any appreciable hidden numerical diffusion into the computational scheme. In an analogous fashion to the technique presented by Willemse et al. (1985) for an orthogonal boundary-fitted grid, the present study modifies Stelling's numerical procedure for use with non-orthogonal coordinate meshes.

All spatial derivatives in the transformed momentum equations are approximated by central differences except for the *cross-advective* terms; for the x-momentum equation (5.92a) these are:

$$y_\xi \; U \frac{\partial U}{\partial \eta} \qquad \text{and} \qquad x_\xi \; V \frac{\partial U}{\partial \eta} \quad , \tag{6.12a}$$

whilst for the y-momentum equation (5.92b), the cross-advective accelerations are

$$y_\eta \; U \frac{\partial V}{\partial \xi} \qquad \text{and} \qquad x_\eta \; V \frac{\partial V}{\partial \xi} \quad . \tag{6.12b}$$

The terms shown above are evaluated either as weighted central differences or as second-order (i.e., quadratic) upwind differences, depending upon the particular stage of the computational scheme. For example, in the ξ-direction release, (see Step 1 of Section 6.5.1), the cross-advective terms in the x-momentum equation are both calculated using weighted central differences i.e.,

$$y_\xi \; U \frac{\partial U}{\partial \eta} = y_\xi \; U \left[\frac{1}{3} \; D_{2\eta} U \; + \; \frac{2}{3} \; D_{1\eta} U \right] \tag{6.13a}$$

where $D_{1\eta}U$ = 'standard' central difference approximation to $\partial U/\partial \eta$
(evaluated over 2 grid increments),

and $D_{2\eta}U$ = central difference approximation to $\partial U/\partial\eta$ (evaluated over 4 grid increments).

Thus,

$$y_\xi\, U \frac{\partial U}{\partial \eta} = y_\xi\, U^n_{i,j}\left[\frac{1}{3}\left[\frac{U^n_{i,j+2} - U^n_{i,j-2}}{4} \right] + \frac{2}{3}\left[\frac{U^n_{i,j+1} - U^n_{i,j-1}}{2} \right]\right]$$

$$= y_\xi\, U^n_{i,j}\left[\frac{U^n_{i,j+2} - U^n_{i,j-2} + 4\,(\,U^n_{i,j+1} - U^n_{i,j-1}\,)}{12} \right] \qquad (6.13b)$$

and similarly,

$$x_\xi\, V \frac{\partial U}{\partial \eta} = x_\xi\, \bar{V}^n \left[\frac{U^n_{i,j+2} - U^n_{i,j-2} + 4\,(\,U^n_{i,j+1} - U^n_{i,j-1}\,)}{12} \right] . \qquad (6.13c)$$

where the overbar, $\bar{\ }$, indicates that the hydrodynamic variable is obtained by four-point averaging; the subscripts are omitted for the spatially averaged V-velocity, since

$$\bar{V} \neq V_{i,j} \ .$$

During the η-direction release (Step 2), the x-momentum equation is updated explicitly and the advective terms are differenced using a combination of weighted central and quadratic upwind expressions. Since the boundary-fitted coordinate meshes were generated with the y- and η-coordinate axes in the same approximate direction, the cross-advective term $x_\xi V\partial U/\partial\eta$ can use the sign of the V-velocity component to determine the direction of the upwind finite-difference expression for $\partial U/\partial\eta$. On the other hand, the derivative, $y_\xi U\partial U/\partial\eta$ is calculated using a weighted central difference formula because the U-velocity direction is approximately normal to the η-direction. Consequently, the cross-advective derivatives are discretised as:

$$x_\xi\, V \frac{\partial U}{\partial \eta} = \begin{cases} x_\xi\, \bar{V}^{n+1} \left[\frac{3}{2} U^{n+1}_{i,j} - 2\, U^{n+1}_{i,j-1} + \frac{1}{2} U^{n+1}_{i,j-2} \right] & \text{if } \bar{V}^{n+1} > 0 \\ x_\xi\, \bar{V}^{n+1} \left[-\frac{3}{2} U^{n+1}_{i,j} + 2\, U^{n+1}_{i,j+1} - \frac{1}{2} U^{n+1}_{i,j+2} \right] & \text{if } \bar{V}^{n+1} < 0 \end{cases} \qquad (6.14a)$$

and

$$y_\xi\, U \frac{\partial U}{\partial \eta} = y_\xi\, U^{n+1}_{i,j}\left[\frac{U^{n+1}_{i,j+2} - U^{n+1}_{i,j-2} + 4\,(\,U^{n+1}_{i,j+1} - U^{n+1}_{i,j-1}\,)}{12} \right] . \qquad (6.14b)$$

The same principle is also employed for the y-momentum cross-advective components; the explicit computation (Step 1) uses upwind differencing for the acceleration term, $y_\eta U\partial V/\partial\xi$ and weighted central differences for $x_\eta V\partial V/\partial\xi$ whilst the implicit calculation (Step 2) employs weighted central differences for both these terms. Section 6.6 details the discretisation of the momentum and continuity equations and illustrates the solution techniques utilised in the computer code.

6.5 Adaptation of Leendertse's computational scheme for use with non-orthogonal boundary-fitted coordinate systems

Leendertse (1967 and 1970) adapted the A.D.I. finite-difference scheme (proposed by Peaceman and Rachford (1955) and Douglas (1955)) for the solution of the non-linear shallow water equations on staggered mesh structures. Until Leendertse published his implicit numerical procedure, most computer simulations of the shallow water equations employed explicit solution techniques (e.g., Hansen, 1956 and 1962; Reid and Bodine, 1968); these numerical schemes ignored the advective acceleration terms in order to avoid non-linear instabilities. The scheme devised by Leendertse is often referred to as a *semi-implicit* computational procedure because some stages of the simulation are implicit in nature whilst other stages resort to explicit techniques. Leendertse's algorithms allowed the non-linear advective acceleration terms to be included in the discretisation because the implicit solution technique dampened the growth of the grid-scale velocity oscillations. In the present investigation, a straightforward modification of the Leendertse scheme for use with non-orthogonal systems was found to be inadequate since the strong recirculation regions that accompany jet-forced reservoir flows are associated with large advective accelerations. In order to stabilise the numerical scheme, a form of Stelling's second-order upwind differencing (described in the previous section) had to be employed.

The principle of the A.D.I. algorithm is to convert fully-implicit finite-difference approximations of the governing equations, involving large band-width matrices, into tri-diagonal matrices which can be solved very efficiently. This is accomplished by discretising the hydrodynamic equations over two separate steps (or 'releases') and restricting the number of unknown hydrodynamic variables to three per implicit equation. The discretised equations used in the A.D.I. algorithms, however, contain more than three unknowns (this is compounded by the fact that the non-orthogonal transformed hydrodynamic equations contain approximately twice as many individual components as their Cartesian counterparts) and so some terms must

be lagged at the previous time step. In order to increase the temporal accuracy of the computer algorithm, the A.D.I. scheme can be solved by an iterative technique, so that better estimates can be used for the advanced non-implicit variables; this is described in more detail in Section 6.5.1. For flow solutions which tend to steady-state, however, there is little to be gained from the increased computational costs of an iterative technique (especially if the time step, Δt, is small compared to the time necessary to reach equilibrium).

6.5.1 A.D.I. discretisation

With reference to Figure 6.6, the A.D.I. scheme used for the present numerical simulation is accomplished as follows:

Step 1 : ξ-direction release : $t=n\Delta t \rightarrow (n+\frac{1}{2})\Delta t$

All derivatives with respect to ξ are written at the advanced time level, $t = (n+\frac{1}{2})\Delta t$, whereas derivatives involving η are held at $t = n\Delta t$. All non-derivative expressions (e.g., bottom friction and the Coriolis terms) are also evaluated at the advanced time stage. The x-momentum and continuity equations for a given η-line are grouped together to form a tri-diagonal system and are then solved in an implicit fashion (represented by the solid horizontal line in Figure 6.6). Once the advanced U-velocity components and surface elevation values have been evaluated for every η-line, the y-momentum equation is solved at each of the V-velocity positions. Since this particular stage requires the use of a second-order upwind differencing technique (destroying the tri-diagonal character of the matrix representation), the advanced V-velocities are obtained via an explicit computational scheme; this is shown diagrammatically in Figure 6.6 by the horizontal dashed line. At this stage, the process can be repeated in an iterative manner, to obtain better estimates of the hydrodynamic variables that have been lagged by virtue of the computational procedure employed. In a similar manner to the vorticity-transport equation A.D.I. scheme, described in Chapter 3, the values of the flow variables at the half time stage, $t = (n+\frac{1}{2})\Delta t$ do not have a *physical* significance; the flow parameters only become meaningful again, once the second directional release has been completed.

Step 2 : η-direction release : $t=(n+\frac{1}{2})\Delta t \rightarrow (n+1)\Delta t$

For the second directional release, the η-derivatives are expressed at the new time level, $t = (n+1)\Delta t$ whilst the derivatives involving ξ are held at $t = (n+\frac{1}{2})\Delta t$. The solid vertical line depicted in Figure 6.6 symbolizes an implicit, combined y-momentum/continuity equation procedure, whereas the dashed vertical line

represents the explicit, x-momentum discretisation (using quadratic upwind differences for the cross-advective component, $x_\xi V\partial U/\partial\eta$). In order that the combined effects of Steps 1 and 2 are theoretically second order accurate in time, the non-derivative components of the η-direction release are expressed at the old time increment, $t = (n+\frac{1}{2})\Delta t$. As before, for the ξ-direction release, iteration can be incorporated into the numerical algorithms to improve the time accuracy of the scheme.

6.5.2 Mathematical summary of the A.D.I. procedure

The A.D.I. scheme is demonstrated with regards to the transformed x-momentum equation (5.92a). Following the procedures outlined above, the directional releases are formulated as:

Step 1 : ξ-direction release

$$\frac{U^{n+\frac{1}{2}} - U^n}{\frac{1}{2}\Delta t} + \frac{1}{J}\left[y_\eta U^{n+\frac{1}{2}}.\frac{\delta U^{n+\frac{1}{2}}}{\delta\xi} - y_\xi U^n.\frac{\delta U^n}{\delta\eta} + x_\xi V^n.\frac{\delta U^n}{\delta\eta} - x_\eta V^{n+\frac{1}{2}}.\frac{\delta U^{n+\frac{1}{2}}}{\delta\xi}\right]$$

$$- f.V^{n+\frac{1}{2}} + \frac{g}{J}\left[y_\eta\frac{\delta\zeta^{n+\frac{1}{2}}}{\delta\xi} - y_\xi\frac{\delta\zeta^n}{\delta\eta}\right] - \frac{\tau_{wx} - \tau_{bx}^{n+\frac{1}{2}}}{\rho.D^{n+\frac{1}{2}}} - \frac{1}{\rho.J}\left[\frac{1}{D^{n+\frac{1}{2}}}\left[y_\eta\frac{\delta(DT_{xx})^{n+\frac{1}{2}}}{\delta\xi}\right.\right.$$

$$\left.\left. - x_\eta\frac{\delta(DT_{xy})^{n+\frac{1}{2}}}{\delta\xi}\right] - \frac{1}{D^n}\left[y_\xi\frac{\delta(DT_{xx})^n}{\delta\eta} - x_\xi\frac{\delta(DT_{xy})^n}{\delta\eta}\right]\right] = 0 \qquad (6.15a)$$

and

Step 2 : η-direction release

$$\frac{U^{n+1} - U^{n+\frac{1}{2}}}{\frac{1}{2}\Delta t} + \frac{1}{J}\left[y_\eta U^{n+\frac{1}{2}}.\frac{\delta U^{n+\frac{1}{2}}}{\delta\xi} - y_\xi U^{n+1}.\frac{\delta U^{n+1}}{\delta\eta} + x_\xi V^{n+1}.\frac{\delta U^{n+1}}{\delta\eta} - x_\eta V^{n+\frac{1}{2}}.\frac{\delta U^{n+\frac{1}{2}}}{\delta\xi}\right]$$

$$- f.V^{n+\frac{1}{2}} + \frac{g}{J}\left[y_\eta\frac{\delta\zeta^{n+\frac{1}{2}}}{\delta\xi} - y_\xi\frac{\delta\zeta^{n+1}}{\delta\eta}\right] - \frac{\tau_{wx} - \tau_{bx}^{n+\frac{1}{2}}}{\rho.D^{n+\frac{1}{2}}} - \frac{1}{\rho.J}\left[\frac{1}{D^{n+\frac{1}{2}}}\left[y_\eta\frac{\delta(DT_{xx})^{n+\frac{1}{2}}}{\delta\xi}\right.\right.$$

$$\left.\left. - x_\eta\frac{\delta(DT_{xy})^{n+\frac{1}{2}}}{\delta\xi}\right] - \frac{1}{D^{n+1}}\left[y_\xi\frac{\delta(DT_{xx})^{n+1}}{\delta\eta} - x_\xi\frac{\delta(DT_{xy})^{n+1}}{\delta\eta}\right]\right] = 0 \qquad (6.15b)$$

where the symbol, δ represents a second-order (central or quadratic upwind) finite-difference approximation.

Adding the above equations for the two half time steps and multiplying through by $\frac{1}{2}$, produces the expression:

$$\frac{U^{n+1} - U^n}{\Delta t} + \frac{1}{J}\left[y_\eta U^{n+\frac{1}{2}} \cdot \frac{\delta U^{n+\frac{1}{2}}}{\delta \xi} - \frac{y_\xi}{2}\left[U^n \cdot \frac{\delta U^n}{\delta \eta} + U^{n+1} \cdot \frac{\delta U^{n+1}}{\delta \eta} \right] \right.$$

$$\left. + \frac{x_\xi}{2}\left[V^n \cdot \frac{\delta U^n}{\delta \eta} + V^{n+1} \cdot \frac{\delta U^{n+1}}{\delta \eta} \right] - x_\eta V^{n+\frac{1}{2}} \cdot \frac{\delta U^{n+\frac{1}{2}}}{\delta \xi} \right] - f \cdot V^{n+\frac{1}{2}} + \frac{g}{J}\left[y_\eta \frac{\delta \zeta^{n+\frac{1}{2}}}{\delta \xi} \right.$$

$$\left. - \frac{y_\xi}{2}\left[\frac{\delta \zeta^n}{\delta \eta} + \frac{\delta \zeta^{n+1}}{\delta \eta} \right] \right] - \frac{\tau_{wx} - \tau_{bx}^{n+\frac{1}{2}}}{\rho \cdot D^{n+\frac{1}{2}}} - \frac{1}{\rho \cdot J}\left[\frac{1}{D^{n+\frac{1}{2}}} \left[y_\eta \frac{\delta (DT_{xx})^{n+\frac{1}{2}}}{\delta \xi} \right. \right.$$

$$\left. - x_\eta \frac{\delta (DT_{xy})^{n+\frac{1}{2}}}{\delta \xi} \right] - \frac{y_\xi}{2}\left[\frac{1}{D^n} \cdot \frac{\delta (DT_{xx})^n}{\delta \eta} + \frac{1}{D^{n+1}} \cdot \frac{\delta (DT_{xx})^{n+1}}{\delta \eta} \right] + \frac{x_\xi}{2}\left[\frac{1}{D^n} \cdot \frac{\delta (DT_{xy})^n}{\delta \eta} \right.$$

$$\left. \left. + \frac{1}{D^{n+1}} \cdot \frac{\delta (DT_{xy})^{n+1}}{\delta \eta} \right] \right] = 0 \qquad (6.16)$$

Provided the advanced hydrodynamic variables are correctly evaluated, the combined effect of Steps 1 and 2 result in a equation which is fully centred in time; consequently the A.D.I. scheme has a formal time accuracy of (Δt^2). However, if the computational scheme avoids the iteration needed to calculate the advanced non-implicit variables, the time accuracy of the method will fall to approximately (Δt).

6.6 Numerical scheme

The numerical procedures are performed on the staggered computational grid illustrated schematically in Figure 6.6. Since the boundary-fitted mesh is invariant with time, the spatial indices, I and J are sufficient to define the geometric derivatives, x_ξ, x_η, y_ξ and y_η. However, the hydrodynamic variables are time dependent, and therefore as well as using the spatial indices, i and j, it is necessary to introduce a further index, n (as a superscript) to denote the time level.

6.6.1 STEP 1 : ξ–direction release

(a) x–momentum equation (centred at U–velocity positions):

$$I = 2i$$
$$J = 2j - 1$$

The individual terms of equation (5.92a) are discretised as:

$$\frac{\partial U}{\partial t} = \frac{U_{i,j}^{n+\frac{1}{2}} - U_{i,j}^{n}}{\frac{1}{2}\Delta t}$$

$$y_\eta \; U \frac{\partial U}{\partial \xi} = y_{\eta\, I,J} U_{i,j}^{n+\frac{1}{2}} \left[\frac{U_{i+1,j}^{n+\frac{1}{2}} - U_{i-1,j}^{n+\frac{1}{2}}}{2} \right]$$

$$y_\xi \; U \frac{\partial U}{\partial \eta} = y_{\xi\, I,J} U_{i,j}^{n} \left[\frac{U_{i,j+2}^{n} - U_{i,j-2}^{n} + 4\,(U_{i,j+1}^{n} - U_{i,j-1}^{n})}{12} \right]$$

$$x_\xi \; V \frac{\partial U}{\partial \eta} = x_{\xi\, I,J} \left[\frac{V_{i,j}^{n} + V_{i+1,j}^{n} + V_{i+1,j-1}^{n} + V_{i,j-1}^{n}}{4} \right] \cdot \left[\frac{U_{i,j+2}^{n} - U_{i,j-2}^{n} + 4\,(U_{i,j+1}^{n} - U_{i,j-1}^{n})}{12} \right]$$

$$x_\eta \; V \frac{\partial U}{\partial \xi} = x_{\eta\, I,J} \left[\frac{V_{i,j}^{n+\frac{1}{2}} + V_{i+1,j}^{n+\frac{1}{2}} + V_{i+1,j-1}^{n+\frac{1}{2}} + V_{i,j-1}^{n+\frac{1}{2}}}{4} \right] \left[\frac{U_{i+1,j}^{n+\frac{1}{2}} - U_{i-1,j}^{n+\frac{1}{2}}}{2} \right]$$

$$f\, V = f \left[\frac{V_{i,j}^{n+\frac{1}{2}} + V_{i+1,j}^{n+\frac{1}{2}} + V_{i+1,j-1}^{n+\frac{1}{2}} + V_{i,j-1}^{n+\frac{1}{2}}}{4} \right]$$

$$y_\eta \frac{\partial \zeta}{\partial \xi} = y_{\eta\, I,J} \left(\zeta_{i+1,j}^{n+\frac{1}{2}} - \zeta_{i,j}^{n+\frac{1}{2}} \right)$$

$$y_\xi \frac{\partial \zeta}{\partial \eta} = y_{\xi\, I,J} \left[\frac{\zeta_{i,j+1}^{n} + \zeta_{i+1,j+1}^{n} - \zeta_{i,j-1}^{n} - \zeta_{i+1,j-1}^{n}}{4} \right]$$

$$\frac{\tau_{bx}}{\rho D} = \frac{c_f \, U_{i,j}^{n+\frac{1}{2}} \left[\left[U_{i,j}^{n+\frac{1}{2}} \right]^2 + \left[\frac{1}{4} (V_{i,j}^{n+\frac{1}{2}} + V_{i+1,j}^{n+\frac{1}{2}} + V_{i+1,j-1}^{n+\frac{1}{2}} + V_{i,j-1}^{n+\frac{1}{2}}) \right]^2 \right]^{\frac{1}{2}}}{\frac{1}{2} \left[D_{i,j}^{n+\frac{1}{2}} + D_{i+1,j}^{n+\frac{1}{2}} \right]}$$

$$\frac{\tau_{wx}}{\rho D} = \frac{\tau_{wx}}{\rho \tfrac{1}{2}\left[D^{n+\frac{1}{2}}_{i,j} + D^{n+\frac{1}{2}}_{i+1,j} \right]}$$

$$y_\eta \frac{\partial(DT_{xx})}{\partial \xi} = y_{\eta_{I,J}}\left[D^{n+\frac{1}{2}}_{i+1,j} \cdot T^{n+\frac{1}{2}}_{xx_{i+1,j}} - D^{n+\frac{1}{2}}_{i,j} \cdot T^{n+\frac{1}{2}}_{xx_{i,j}} \right]$$

$$y_\xi \frac{\partial(DT_{xx})}{\partial \eta} = y_{\xi_{I,J}} \frac{1}{8}\Big[(D^n_{i,j+1} + D^n_{i+1,j+1})(T^n_{xx_{i,j+1}} + T^n_{xx_{i+1,j+1}}) - (D^n_{i,j-1} + D^n_{i+1,j-1})(T^n_{xx_{i,j-1}} + T^n_{xx_{i+1,j-1}})\Big]$$

$$x_\xi \frac{\partial(DT_{xy})}{\partial \eta} = x_{\xi_{I,J}} \frac{1}{8}\Big[(D^n_{i,j+1} + D^n_{i+1,j+1})(T^n_{xy_{i,j+1}} + T^n_{xy_{i+1,j+1}}) - (D^n_{i,j-1} + D^n_{i+1,j-1})(T^n_{xy_{i,j-1}} + T^n_{xy_{i+1,j-1}})\Big]$$

and

$$x_\eta \frac{\partial(DT_{xy})}{\partial \xi} = x_{\eta_{I,J}}\left[D^{n+\frac{1}{2}}_{i+1,j} \cdot T^{n+\frac{1}{2}}_{xy_{i+1,j}} - D^{n+\frac{1}{2}}_{i,j} \cdot T^{n+\frac{1}{2}}_{xy_{i,j}} \right] .$$

Letting the Jacobian be denoted by JAC in order to avoid confusion with the transformation derivative spatial index, J, equation (5.92a) is formulated in finite-differences as

$$\frac{U^{n+\frac{1}{2}}_{i,j} - U^n_{i,j}}{\frac{1}{2}\Delta t} + \frac{1}{JAC_{I,J}}\Big[y_{\eta_{I,J}} U^{n+\frac{1}{2}}_{i,j} \cdot UD1 - y_{\xi_{I,J}} U^n_{i,j} \cdot UD2 + x_{\xi_{I,J}} VA2 \cdot UD2 - x_{\eta_{I,J}} VA1 \cdot UD1\Big] - f \cdot VA1 + \frac{g}{JAC_{I,J}}\Big[y_{\eta_{I,J}}(\zeta^{n+\frac{1}{2}}_{i+1,j} - \zeta^{n+\frac{1}{2}}_{i,j}) - y_{\xi_{I,J}} ZETAD1\Big]$$

$$+ F_{bx} - F_{wx} - T_x = 0 \qquad (6.17)$$

where

$$UD1 = \frac{U^{n+\frac{1}{2}}_{i+1,j} - U^{n+\frac{1}{2}}_{i-1,j}}{2}$$

$$UD2 = \left[\frac{U^n_{i,j+2} - U^n_{i,j-2} + 4\,(U^n_{i,j+1} - U^n_{i,j-1})}{12} \right]$$

... weighted central difference approximation to $\partial U/\partial \eta$.

$$VA1 = \left[\frac{V^{n+\frac{1}{2}}_{i,j} + V^{n+\frac{1}{2}}_{i+1,j} + V^{n+\frac{1}{2}}_{i+1,j-1} + V^{n+\frac{1}{2}}_{i,j-1}}{4} \right]$$

$$VA2 = \left[\frac{V^n_{i,j} + V^n_{i+1,j} + V^n_{i+1,j-1} + V^n_{i,j-1}}{4} \right]$$

$$ZETAD1 = \left[\frac{\zeta^n_{i,j+1} + \zeta^n_{i+1,j+1} - \zeta^n_{i,j-1} - \zeta^n_{i+1,j-1}}{4} \right]$$

$$D1 = \frac{D^{n+\frac{1}{2}}_{i,j} + D^{n+\frac{1}{2}}_{i+1,j}}{2}$$

$$D2 = \frac{D^n_{i,j} + D^n_{i+1,j}}{2}$$

$$SD1 = D^n_{i,j+1} + D^n_{i+1,j+1}$$

$$SD2 = D^n_{i,j-1} + D^n_{i+1,j-1}$$

$$F_{bx} = \frac{c_f\, U^{n+\frac{1}{2}}_{i,j} \left[\left[U^{n+\frac{1}{2}}_{i,j} \right]^2 + \left[VA1 \right]^2 \right]^{\frac{1}{2}}}{D1}$$

$$F_{wx} = \frac{\tau_{wx}}{\rho . D1}$$

$$T1 = y_{\eta_{I,J}} \left[D^{n+\frac{1}{2}}_{i+1,j} . T^{n+\frac{1}{2}}_{xx_{i+1,j}} - D^{n+\frac{1}{2}}_{i,j} . T^{n+\frac{1}{2}}_{xx_{i,j}} \right]$$

$$T2 = -y_{\xi_{I,J}} \frac{1}{8} \left[SD1 (T^n_{xx_{i,j+1}} + T^n_{xx_{i+1,j+1}}) - SD2 (T^n_{xx_{i,j-1}} + T^n_{xx_{i+1,j-1}}) \right]$$

$$T3 = x_{\xi_{I,J}} \frac{1}{8}\left[SD1(T^n_{xy_{i,j+1}} + T^n_{xy_{i+1,j+1}}) - SD2(T^n_{xy_{i,j-1}} + T^n_{xy_{i+1,j-1}})\right]$$

$$T4 = -x_{\eta_{I,J}}\left[D^{n+\frac{1}{2}}_{i+1,j} . T^{n+\frac{1}{2}}_{xy_{i+1,j}} - D^{n+\frac{1}{2}}_{i,j} . T^{n+\frac{1}{2}}_{xy_{i,j}} \right]$$

and

$$T_x = \frac{1}{\rho . JAC_{I,J}}\left[\frac{(T1 + T4)}{D1} + \frac{(T2 + T3)}{D2} \right]$$

Rearranging the terms in equation (6.17) leads to the x-momentum equation:

$$- \frac{g\Delta t}{2} . \frac{y_{\eta_{I,J}}}{JAC_{I,J}} \zeta^{n+\frac{1}{2}}_{i,j} + \left[1.0 + \frac{\Delta t}{2} . \frac{y_{\eta_{I,J}}}{JAC_{I,J}} . UD1 \right] U^{n+\frac{1}{2}}_{i,j} + \frac{g\Delta t}{2} . \frac{y_{\eta_{I,J}}}{JAC_{I,J}} \zeta^{n+\frac{1}{2}}_{i+1,j}$$

$$= U^n_{i,j} + \frac{\Delta t}{2}\left[\frac{1}{JAC_{I,J}}\left[y_{\xi_{I,J}} U^n_{i,j} . UD2 - x_{\xi_{I,J}} VA2 . UD2 + x_{\eta_{I,J}} VA1 . UD1 \right.\right.$$

$$\left.\left. + g . y_{\xi_{I,J}} ZETAD1 \right] + f . VA1 - F_{bx} + F_{wx} + T_x \right] . \qquad (6.18)$$

The right-hand side of this expression contains advanced U- and V-velocity components in the terms UD1, VA1 and F_{bx}, together with advanced effective stress components in T_x. Furthermore, the coefficient of $U^{n+\frac{1}{2}}_{I,J}$ involves UD1 which is also at the new time level. Consequently, the computer code allows the option of determining these advanced non-implicit variables via an iteration technique which is carried out in conjunction with the explicit y-momentum equation, shown later.

(b) Continuity equation (centred at ζ/h/D positions):

$$I = 2i - 1$$
$$J = 2j - 1$$

The individual terms of the transformed depth-averaged continuity equation (5.91) are approximated as

$$\frac{\partial \zeta}{\partial t} = \frac{\zeta^{n+\frac{1}{2}}_{i,j} - \zeta^n_{i,j}}{\frac{1}{2}\Delta t}$$

$$y_\eta \frac{\partial(UD)}{\partial\xi} = y_{\eta_{I,J}}\left[U^{n+\frac{1}{2}}_{i,j}\left[\frac{D^{n+\frac{1}{2}}_{i,j} + D^{n+\frac{1}{2}}_{i+1,j}}{2} \right] - U^{n+\frac{1}{2}}_{i-1,j}\left[\frac{D^{n+\frac{1}{2}}_{i-1,j} + D^{n+\frac{1}{2}}_{i,j}}{2} \right] \right]$$

$$y_\xi \frac{\partial(UD)}{\partial\eta} = y_{\xi_{I,J}}\left[\frac{(U^n_{i-1,j+1} + U^n_{i,j+1})D^n_{i,j+1} - (U^n_{i-1,j-1} + U^n_{i,j-1})D^n_{i,j-1}}{4} \right]$$

$$x_\xi \frac{\partial(VD)}{\partial\eta} = x_{\xi_{I,J}}\left[V^n_{i,j}\left[\frac{D^n_{i,j} + D^n_{i,j+1}}{2} \right] - V^n_{i,j-1}\left[\frac{D^n_{i,j-1} + D^n_{i,j}}{2} \right] \right]$$

and

$$x_\eta \frac{\partial(VD)}{\partial\xi} = x_{\eta_{I,J}}\left[\frac{(V^{n+\frac{1}{2}}_{i+1,j} + V^{n+\frac{1}{2}}_{i+1,j-1})D^{n+\frac{1}{2}}_{i+1,j} - (V^{n+\frac{1}{2}}_{i-1,j} + V^{n+\frac{1}{2}}_{i-1,j-1})D^{n+\frac{1}{2}}_{i-1,j} }{4} \right]$$

The above expressions have different discretisation errors since $y_\eta \partial(UD)/\partial\xi$ and $x_\xi \partial(VD)/\partial\eta$ are approximated over distances of $\Delta\xi$ and $\Delta\eta$, whilst the other spatial derivatives, $y_\xi \partial(UD)/\partial\eta$ and $x_\eta \partial(VD)/\partial\xi$ are evaluated over $2\Delta\eta$ and $2\Delta\xi$, respectively. This is unavoidable in the fully staggered cell layout shown in Figure 6.1 and can only be overcome by using Johnson's (1980) or Häuser et al.'s (1985) cell definition (Figure 6.3). However, the additional stability of an implicit solution technique (compared with the explicit schemes that are necessary for the grid structure shown in Figure 6.3) led to the decision to permit the above mentioned spatial discretisation anomalies, and to continue with the present staggered mesh layout. This can be corroborated by considering the 'anomalies' that occur in standard Cartesian meshes; often the velocity derivatives are evaluated over two grid increments, whilst the surface elevation gradients, $\partial\zeta/\partial x$ and $\partial\zeta/\partial y$ are discretised over a single cell length!

Thus the transformed continuity equation is written as

$$\frac{\zeta^{n+\frac{1}{2}}_{i,j} - \zeta^n_{i,j}}{\frac{1}{2}\Delta t} + \frac{1}{JAC_{I,J}}\Big[y_{\eta_{I,J}} U^{n+\frac{1}{2}}_{i,j}.D1 - y_{\eta_{I,J}} U^{n+\frac{1}{2}}_{i-1,j}.D3 - y_{\xi_{I,J}} UDD + x_{\xi_{I,J}} VDD1 - x_{\eta_{I,J}} VDD2 \Big] = 0 \qquad (6.19)$$

where

$$D1 = \frac{D^{n+\frac{1}{2}}_{i,j} + D^{n+\frac{1}{2}}_{i+1,j}}{2}$$

$$D3 = \frac{D^{n+\frac{1}{2}}_{i-1,j} + D^{n+\frac{1}{2}}_{i,j}}{2}$$

$$UDD = \left[\frac{(U^{n}_{i-1,j+1} + U^{n}_{i,j+1})D^{n}_{i,j+1} - (U^{n}_{i-1,j-1} + U^{n}_{i,j-1})D^{n}_{i,j-1}}{4}\right]$$

$$VDD1 = V^{n}_{i,j}\left[\frac{D^{n}_{i,j} + D^{n}_{i,j+1}}{2}\right] - V^{n}_{i,j-1}\left[\frac{D^{n}_{i,j-1} + D^{n}_{i,j}}{2}\right]$$

and

$$VDD2 = \left[\frac{(V^{n+\frac{1}{2}}_{i+1,j} + V^{n+\frac{1}{2}}_{i+1,j-1})D^{n+\frac{1}{2}}_{i+1,j} - (V^{n+\frac{1}{2}}_{i-1,j} + V^{n+\frac{1}{2}}_{i-1,j-1})D^{n+\frac{1}{2}}_{i-1,j}}{4}\right] .$$

After rearrangement, the discretised continuity equation (6.19) becomes

$$-\frac{\Delta t}{2}\cdot\frac{y_{\eta_{I,J}}}{JAC_{I,J}}.D3.U^{n+\frac{1}{2}}_{i-1,j} + \zeta^{n+\frac{1}{2}}_{i,j} + \frac{\Delta t}{2}\cdot\frac{y_{\eta_{I,J}}}{JAC_{I,J}}.D1.U^{n+\frac{1}{2}}_{i,j} = \zeta^{n}_{i,j}$$

$$+ \frac{\Delta t}{2}\cdot\frac{1}{JAC_{I,J}}\left[y_{\xi_{I,J}}UDD - x_{\xi_{I,J}}VDD1 + x_{\eta_{I,J}}VDD2\right] \qquad (6.20)$$

(c) y–momentum equation (centred at V–velocity positions):

$$I = 2i - 1$$
$$J = 2j$$

The individual terms of equation (5.92b) are written:

$$\frac{\partial V}{\partial t} = \frac{V^{n+\frac{1}{2}}_{i,j} - V^{n}_{i,j}}{\frac{1}{2}\Delta t}$$

$$y_\eta\ U\ \frac{\partial V}{\partial \xi} = y_{\eta_{I,J}}\left[\frac{U^{n+\frac{1}{2}}_{i,j} + U^{n+\frac{1}{2}}_{i,j+1} + U^{n+\frac{1}{2}}_{i-1,j+1} + U^{n+\frac{1}{2}}_{i-1,j}}{4}\right].VD1A$$

where VD1A is the quadratic upwind approximation to $\partial V/\partial \xi$, (see later)

$$y_\xi\ U\ \frac{\partial V}{\partial \eta} = y_{\xi_{I,J}}\left[\frac{U^{n}_{i,j} + U^{n}_{i,j+1} + U^{n}_{i-1,j+1} + U^{n}_{i-1,j}}{4}\right]\left[\frac{V^{n}_{i,j+1} - V^{n}_{i,j-1}}{2}\right]$$

$$x_\xi \; V \frac{\partial V}{\partial \eta} = x_{\xi_{I,J}} V^n_{i,j} \left[\frac{V^n_{i,j+1} - V^n_{i,j-1}}{2} \right]$$

$$x_\eta \; V \frac{\partial V}{\partial \xi} = x_{\eta_{I,J}} V^{n+\frac{1}{2}}_{i,j} . \text{VD1B}$$

where VD1B is the weighted central difference approximation to $\partial V/\partial \xi$, (see later)

$$f \; U = f \left[\frac{U^{n+\frac{1}{2}}_{i,j} + U^{n+\frac{1}{2}}_{i,j+1} + U^{n+\frac{1}{2}}_{i-1,j+1} + U^{n+\frac{1}{2}}_{i-1,j}}{4} \right]$$

$$x_\xi \frac{\partial \zeta}{\partial \eta} = x_{\xi_{I,J}} (\zeta^n_{i,j+1} - \zeta^n_{i,j})$$

$$x_\eta \frac{\partial \zeta}{\partial \xi} = x_{\eta_{I,J}} \left[\frac{\zeta^{n+\frac{1}{2}}_{i+1,j+1} + \zeta^{n+\frac{1}{2}}_{i+1,j} - \zeta^{n+\frac{1}{2}}_{i-1,j+1} - \zeta^{n+\frac{1}{2}}_{i-1,j}}{4} \right]$$

$$\frac{\tau_{by}}{\rho D} = \frac{c_f \; V^{n+\frac{1}{2}}_{i,j} \left[\left[\frac{1}{4} (U^{n+\frac{1}{2}}_{i,j} + U^{n+\frac{1}{2}}_{i,j+1} + U^{n+\frac{1}{2}}_{i-1,j+1} + U^{n+\frac{1}{2}}_{i-1,j}) \right]^2 + \left[V^{n+\frac{1}{2}}_{i,j} \right]^2 \right]^{\frac{1}{2}}}{\frac{1}{2} \left[D^{n+\frac{1}{2}}_{i,j} + D^{n+\frac{1}{2}}_{i,j+1} \right]}$$

$$\frac{\tau_{wy}}{\rho D} = \frac{\tau_{wy}}{\rho \; \frac{1}{2} \left[D^{n+\frac{1}{2}}_{i,j} + D^{n+\frac{1}{2}}_{i,j+1} \right]}$$

$$y_\eta \frac{\partial (DT_{xy})}{\partial \xi} = y_{\eta_{I,J}} \frac{1}{8} \Big[(D^{n+\frac{1}{2}}_{i+1,j+1} + D^{n+\frac{1}{2}}_{i+1,j}) (T^{n+\frac{1}{2}}_{xy_{i+1,j+1}} + T^{n+\frac{1}{2}}_{xy_{i+1,j}})$$

$$- (D^{n+\frac{1}{2}}_{i-1,j+1} + D^{n+\frac{1}{2}}_{i-1,j}) (T^{n+\frac{1}{2}}_{xy_{i-1,j+1}} + T^{n+\frac{1}{2}}_{xy_{i-1,j}}) \Big]$$

$$y_\xi \frac{\partial (DT_{xy})}{\partial \eta} = y_{\xi_{I,J}} \left[D^n_{i,j+1} \cdot T^n_{xy_{i,j+1}} - D^n_{i,j} \cdot T^n_{xy_{i,j}} \right]$$

$$x_\xi \frac{\partial (DT_{yy})}{\partial \eta} = x_{\xi_{I,J}} \left[D^n_{i,j+1} \cdot T^n_{yy_{i,j+1}} - D^n_{i,j} \cdot T^n_{yy_{i,j}} \right]$$

$$x_\eta \frac{\partial (DT_{yy})}{\partial \xi} = x_{\eta_{I,J}} \frac{1}{8} \Big[(D^{n+\frac{1}{2}}_{i+1,j+1} + D^{n+\frac{1}{2}}_{i+1,j}) (T^{n+\frac{1}{2}}_{yy_{i+1,j+1}} + T^{n+\frac{1}{2}}_{yy_{i+1,j}})$$

$$- (D^{n+\frac{1}{2}}_{i-1,j+1} + D^{n+\frac{1}{2}}_{i-1,j}) (T^{n+\frac{1}{2}}_{yy_{i-1,j+1}} + T^{n+\frac{1}{2}}_{yy_{i-1,j}}) \Big]$$

Hence the finite-difference approximation to the y-momentum equation can be formulated as

$$\frac{V_{i,j}^{n+\frac{1}{2}} - V_{i,j}^{n}}{\frac{1}{2}\Delta t} + \frac{1}{JAC_{I,J}}\Big[y_{\eta_{I,J}}\text{UA1}.\text{VD1A} - y_{\xi_{I,J}}\text{UA2}.\text{VD2} + x_{\xi_{I,J}}V_{i,j}^{n}.\text{VD2}$$

$$- x_{\eta_{I,J}}V_{i,j}^{n+\frac{1}{2}}.\text{VD1B}\Big] + f.\text{UA1} + \frac{g}{JAC_{I,J}}\Big[x_{\xi_{I,J}}\text{ZETAD2} - x_{\eta_{I,J}}\text{ZETAD3}\Big]$$

$$+ F_{by} - F_{wy} - T_{y} = 0 \qquad (6.21)$$

where

$$\text{UA1} = \left[\frac{U_{i,j}^{n+\frac{1}{2}} + U_{i,j+1}^{n+\frac{1}{2}} + U_{i-1,j+1}^{n+\frac{1}{2}} + U_{i-1,j}^{n+\frac{1}{2}}}{4}\right]$$

$$\text{UA2} = \left[\frac{U_{i,j}^{n} + U_{i,j+1}^{n} + U_{i-1,j+1}^{n} + U_{i-1,j}^{n}}{4}\right]$$

$$\text{VD2} = \frac{V_{i,j+1}^{n} - V_{i,j-1}^{n}}{2}$$

$$\text{VD1A} = \begin{cases} \frac{3}{2}V_{i,j}^{n+\frac{1}{2}} - 2V_{i-1,j}^{n+\frac{1}{2}} + \frac{1}{2}V_{i-2,j}^{n+\frac{1}{2}} & \text{if UA1} > 0 \\ -\frac{3}{2}V_{i,j}^{n+\frac{1}{2}} + 2V_{i+1,j}^{n+\frac{1}{2}} - \frac{1}{2}V_{i+2,j}^{n+\frac{1}{2}} & \text{if UA1} < 0 \end{cases}$$

... quadratic upwind difference approximation to $\partial V/\partial\xi$.

$$\text{VD1B} = \left[\frac{V_{i+2,j}^{n+\frac{1}{2}} - V_{i-2,j}^{n+\frac{1}{2}} + 4\,(V_{i+1,j}^{n+\frac{1}{2}} - V_{i-1,j}^{n+\frac{1}{2}})}{12}\right]$$

... weighted central difference approximation to $\partial V/\partial\xi$.

$$\text{ZETAD2} = \zeta_{i,j+1}^{n} - \zeta_{i,j}^{n}$$

$$\text{ZETAD3} = \left[\frac{\zeta_{i+1,j+1}^{n+\frac{1}{2}} + \zeta_{i+1,j}^{n+\frac{1}{2}} - \zeta_{i-1,j+1}^{n+\frac{1}{2}} - \zeta_{i-1,j}^{n+\frac{1}{2}}}{4}\right]$$

$$D4 = \frac{D^{n+\frac{1}{2}}_{i,j} + D^{n+\frac{1}{2}}_{i,j+1}}{2}$$

$$D5 = \frac{D^{n}_{i,j} + D^{n}_{i,j+1}}{2}$$

$$SD3 = D^{n+\frac{1}{2}}_{i+1,j+1} + D^{n+\frac{1}{2}}_{i+1,j}$$

$$SD4 = D^{n+\frac{1}{2}}_{i-1,j+1} + D^{n+\frac{1}{2}}_{i-1,j}$$

$$F_{by} = \frac{c_f \, V^{n+\frac{1}{2}}_{i,j} \left[[UA1]^2 + [V^{n+\frac{1}{2}}_{i,j}]^2 \right]^{\frac{1}{2}}}{D4}$$

$$F_{wy} = \frac{\tau_{wy}}{\rho . D4}$$

$$T5 = y_{\eta_{I,J}} \frac{1}{8} \left[SD3 (T^{n+\frac{1}{2}}_{xy_{i+1,j+1}} + T^{n+\frac{1}{2}}_{xy_{i+1,j}}) - SD4 (T^{n+\frac{1}{2}}_{xy_{i-1,j+1}} + T^{n+\frac{1}{2}}_{xy_{i-1,j}}) \right]$$

$$T6 = -y_{\xi_{I,J}} \left[D^{n}_{i,j+1} . T^{n}_{xy_{i,j+1}} - D^{n}_{i,j} . T^{n}_{xy_{i,j}} \right]$$

$$T7 = x_{\xi_{I,J}} \left[D^{n}_{i,j+1} . T^{n}_{yy_{i,j+1}} - D^{n}_{i,j} . T^{n}_{yy_{i,j}} \right]$$

$$T8 = -x_{\eta_{I,J}} \frac{1}{8} \left[SD3 (T^{n+\frac{1}{2}}_{yy_{i+1,j+1}} + T^{n+\frac{1}{2}}_{yy_{i+1,j}}) - SD4 (T^{n+\frac{1}{2}}_{yy_{i-1,j+1}} + T^{n+\frac{1}{2}}_{yy_{i-1,j}}) \right]$$

and

$$T_y = \frac{1}{\rho . JAC_{I,J}} \left[\frac{(T5 + T8)}{D4} + \frac{(T6 + T7)}{D5} \right] .$$

Rearranging the terms in equation (6.21) leads to the explicit y-momentum equation for the ξ-direction release:

$$V_{i,j}^{n+\frac{1}{2}} = V_{i,j}^{n} + \frac{\Delta t}{2}\left[\frac{1}{JAC_{I,J}}\left[-y_{\eta_{I,J}}UA1.VD1A + y_{\xi_{I,J}}UA2.VD2 - x_{\xi_{I,J}}V_{i,j}^{n}.VD2\right.\right.$$

$$\left. + x_{\eta_{I,J}}V_{i,j}^{n+\frac{1}{2}}.VD1B - x_{\xi_{I,J}}g.ZETAD2 + x_{\eta_{I,J}}g.ZETAD3\right]$$

$$\left. - f.UA1 - F_{by} + F_{wy} + T_y\right] \qquad (6.22)$$

When the above equation is solved, the advanced U-velocity and surface elevation values will have already been calculated by the implicit x-momentum/continuity procedure and therefore the terms, ZETAD3 and UA1 will be correctly centred in time. However, the right-hand side of equation (6.22) also contains expressions involving advanced V-velocities (i.e., VD1A, VD1B, F_{by} and T_y) which are unknown and have to be lagged. Consequently, for an accurate time-dependent solution, this explicit stage must also use an iterative technique; the algorithm devised for the present study couples the y-momentum iteration to the implicit U/ζ determination (equations 6.18 and 6.20). A flow chart illustrating the ξ-direction release is presented later in the chapter, in Figure 6.13.

6.6.2 Solution technique

The discretised x-momentum and continuity equations, (6.18) and (6.20) for the ξ-direction release can be written as

<u>continuity</u>

$$-a_{i,j}U_{i-1,j}^{n+\frac{1}{2}} + b_{i,j}\zeta_{i,j}^{n+\frac{1}{2}} - c_{i,j}U_{i,j}^{n+\frac{1}{2}} = k_{i,j} \qquad (6.23a)$$

and

<u>x-momentum</u>

$$-a'_{i,j}\zeta_{i,j}^{n+\frac{1}{2}} + b'_{i,j}U_{i,j}^{n+\frac{1}{2}} - c'_{i,j}\zeta_{i+1,j}^{n+\frac{1}{2}} = k'_{i,j} \qquad (6.23b)$$

where

$$a_{i,j} = \frac{\Delta t}{2}\cdot\frac{y_{\eta_{I,J}}}{JAC_{I,J}}.D3 \quad ,$$

$$b_{i,j} = 1.0 \quad ,$$

$$c_{i,j} = -\frac{\Delta t}{2}.\frac{y_{\eta_{I,J}}}{JAC_{I,J}}.D1 \quad ,$$

$$k_{i,j} = \zeta^{n}_{i,j} + \frac{\Delta t}{2}.\frac{1}{JAC_{I,J}}\left[y_{\xi_{I,J}}UDD - x_{\xi_{I,J}}VDD1 + x_{\eta_{I,J}}VDD2 \right]$$

and $\left\{ \begin{array}{l} I = 2i - 1 \\ J = 2j - 1 \end{array} \right\}$ for the above (continuity) coefficients,

whilst

$$a'_{i,j} = \frac{g\Delta t}{2}.\frac{y_{\eta_{I,J}}}{JAC_{I,J}} \quad ,$$

$$b'_{i,j} = 1.0 + \frac{\Delta t}{2}.\frac{y_{\eta_{I,J}}}{JAC_{I,J}}.UD1 \quad ,$$

$$c'_{i,j} = -\frac{g\Delta t}{2}.\frac{y_{\eta_{I,J}}}{JAC_{I,J}} = -a'_{i,j} \quad ,$$

$$k'_{i,j} = U^{n}_{i,j} + \frac{\Delta t}{2}\left[\frac{1}{JAC_{I,J}}\left[y_{\xi_{I,J}}U^{n}_{i,j}.UD2 - x_{\xi_{I,J}}VA2.UD2 + x_{\eta_{I,J}}VA1.UD1 \right.\right.$$

$$\left.\left. + g.y_{\xi_{I,J}}ZETAD1 \right] + f.VA1 - F_{bx} + F_{wx} + T_{x} \right]$$

and $\left\{ \begin{array}{l} I = 2i \\ J = 2j - 1 \end{array} \right\}$ for the above (x-momentum) coefficients.

Combining equations (6.23a) and (6.23b) along a given η-line produces the tri-diagonal matrix system:

$$\begin{bmatrix} b'_{1,j} & -c'_{1,j} & & & & \\ -a_{2,j} & b_{2,j} & -c_{2,j} & & & \\ & -a'_{2,j} & b'_{2,j} & -c'_{2,j} & & \\ & & \ddots & \ddots & \ddots & \\ & & -a_{n_i-1,j} & b_{n_i-1,j} & -c_{n_i-1,j} & \\ & & & -a'_{n_i-1,j} & b'_{n_i-1,j} & \end{bmatrix} \begin{Bmatrix} U^{n+\frac{1}{2}}_{1,j} \\ \zeta^{n+\frac{1}{2}}_{2,j} \\ U^{n+\frac{1}{2}}_{2,j} \\ \vdots \\ \zeta^{n+\frac{1}{2}}_{n_i-1,j} \\ U^{n+\frac{1}{2}}_{n_i-1,j} \end{Bmatrix} = \begin{Bmatrix} K'_{1,j} \\ K_{2,j} \\ K'_{2,j} \\ \vdots \\ K_{n_i-1,j} \\ K'_{n_i-1,j} \end{Bmatrix} \tag{6.24}$$

where $K'_{1,j} = k'_{1,j} + a'_{1,j} \cdot \zeta^{n+\frac{1}{2}}_{1,j}$

$K_{2,j} = k_{2,j}$

$K'_{2,j} = k'_{2,j}$

$\vdots$

$K_{n_i-1,j} = k_{n_i-1,j}$

and $K'_{n_i-1,j} = k'_{n_i-1,j} + c'_{n_i-1,j} \cdot \zeta^{n+\frac{1}{2}}_{n_i,j}$.

It is unnecessary to lag the values of -

$$\zeta^{n+\frac{1}{2}}_{1,j} \quad \text{and} \quad \zeta^{n+\frac{1}{2}}_{n_i,j}$$

in the codomain since the advanced surface elevations at the boundaries are determined from explicit finite-difference discretisations of the continuity equation (see Sections 6.7.1 and 6.7.2).

The tri-diagonal system presented in equation (6.24), assumes that the unknown hydrodynamic variables are required between boundary walls at i=1 and i=n_i, and is therefore compatible with the perimeter locations detailed in Figure 6.6. However,

as explained in Chapter 4, the boundary-fitted coordinate generation code allows the outline of the transformed mesh to be of any shape and consequently the left-hand perimeter subscript is not necessarily i=1. Furthermore, the A.D.I. algorithms are coded to allow an arbitrary number of implicit line-wise flow calculation segments along each η-line (utilising the boundary location array, ADI_ξ, described in Section 4.4.4); this gives the shallow water equation program the additional flexibility needed for use with complex boundary-fitted grids of the type illustrated in Figure 4.14. Similarly, the second directional release utilises the boundary array, ADI_η to enable an arbitrary number of V/ζ line-wise determinations to be performed along each particular ξ-line.

The interior U-velocities and surface elevations, at $t=(n+\frac{1}{2})\Delta t$, are obtained by solving equation (6.24) using Gaussian elimination, for all U/ζ line segments (i.e., $j = 2, 3, \ldots, n_{j-1}$). Once the advanced U and ζ values have been evaluated along every η-line, the y-momentum equation is used to update the V-velocities to the new time level. This stage of the numerical simulation is performed using the explicit finite-difference formula presented in equation (6.22). The explicit nature of the Step 1, y-momentum discretisation does not, however, destroy the computational scheme's ability to employ time steps larger than the Courant stability criterion (Section 6.9) and therefore operates in a similar manner to the Cartesian implicit/explicit algorithm developed by Leendertse. The stability of the explicit stage, for time steps above the Courant limit, can be attributed to the coupling of the U- and V-velocity components via the surface elevation field, which is determined implicitly over both directional releases.

Before implementing the numerical scheme, it was not known whether the non-orthogonality and the associated increase in the number of non-implicit expressions in the discretisation would cause a destabilising effect on the computational procedures (possibly requiring a very restrictive time step, Δt). Numerical experimentation has revealed, however, that even though the simulation requires the implementation of a digital recursive filter (described in Section 6.10) when using low values of eddy viscosity coefficient, the non-orthogonal scheme devised for the present investigation is still able to utilise time steps with Courant numbers of approximately 3. This justifies the decision to use a computationally more expensive* implicit numerical scheme, instead of the explicit approach adopted by Häuser et al. (1985, 1986b) and Raghunath et al. (1987).

During the solution procedure, it is necessary to modify some of the previously presented finite-difference approximations when considering nodes which are close to

flow boundaries. For example, in the discretisation of the x-momentum equation (for the ξ-direction release), the weighted central difference approximation to $\partial U/\partial\eta$ must be rewritten as an ordinary central difference expression, when either $U_{i,j+2}$ or $U_{i,j-2}$ lies outside the hydraulic domain. The previously specified weighted central difference expression is

$$UD2 = \frac{\partial U}{\partial \eta} = \left[\frac{U^n_{i,j+2} - U^n_{i,j-2} + 4\,(U^n_{i,j+1} - U^n_{i,j-1})}{12} \right] \qquad (6.25a)$$

.... when $U_{i,j+2}$ and $U_{i,j-2}$ are both defined within/on the flow boundaries,

whilst

$$UD2 = \frac{U^n_{i,j+1} - U^n_{i,j-1}}{2} \qquad (6.25b)$$

.... when $U_{i,j+2}$ or $U_{i,j-2}$ lies outside the flow region.

In a similar manner, the quadratic upwind and weighted central differences, VD1A and VD1B, for the y-momentum equation (ξ-direction release) also need to be reformulated. The modifications, in this case, are required when the V-velocity positions are adjacent to ξ=constant flow perimeters. For example, considering a V-velocity node, (i,j) next to a 'left-hand/ξ=constant' boundary, then $V_{i-1,j}$ exists but $V_{i-2,j}$ is undefined. Consequently, the cross-advective formulae presented in Section 6.6.1c are recast as:

$$VD1A = \begin{cases} V^{n+\frac{1}{2}}_{i,j} - V^{n+\frac{1}{2}}_{i-1,j} & \text{if } UA1 > 0 \\ -\frac{3}{2} V^{n+\frac{1}{2}}_{i,j} + 2\, V^{n+\frac{1}{2}}_{i+1,j} - \frac{1}{2} V^{n+\frac{1}{2}}_{i+2,j} & \text{if } UA1 < 0 \end{cases} \qquad (6.26a)$$

* Weare (1976a) quotes the relative costs of Cartesian shallow water equation finite-difference simulations as follows:

(a) Explicit methods:- require approximately
40 units of work/grid point/time step.

(b) Implicit (A.D.I.) schemes:-
50 units of work/grid point/time step.

and

$$VD1B = \frac{V^{n+\frac{1}{2}}_{i+1,j} - V^{n+\frac{1}{2}}_{i-1,j}}{2} \quad . \tag{6.26b}$$

For V-velocities adjacent to 'right-hand/ξ=constant' flow boundaries, the formula presented in equation (6.26b) remains unaltered, but VD1A becomes

$$VD1A = \begin{cases} \frac{3}{2} V^{n+\frac{1}{2}}_{i,j} - 2 V^{n+\frac{1}{2}}_{i-1,j} + \frac{1}{2} V^{n+\frac{1}{2}}_{i-2,j} & \text{if } UA1 > 0 \\ \\ V^{n+\frac{1}{2}}_{i+1,j} - V^{n+\frac{1}{2}}_{i,j} & \text{if } UA1 < 0 \end{cases} \quad . \tag{6.26c}$$

Similar methods are also employed during the η-direction release.

As explained in Section 6.5.1, the computational scheme used for each directional release can employ an iterative technique to obtain better estimates of the lagged hydrodynamic variables. A flow chart clarifying the iteration involved in the ξ-direction release is presented in Figure 6.13, later in the chapter, once the remaining numerical procedures and boundary conditions have been described.

6.6.3 STEP 2 : η-direction release

The η-direction release, $[t=(n+\frac{1}{2})\Delta t \rightarrow t=(n+1)\Delta t]$, is discretised in an analogous manner to that shown above for the ξ-direction. In this case, η-derivatives are expressed at the new time level, $t=(n+1)\Delta t$, whilst gradients with respect to ξ and non-derivative expressions are held at the old time stage, $t=(n+\frac{1}{2})\Delta t$. The y-momentum and continuity equations are grouped together along each ξ-line to form tri-diagonal matrix systems which are solved implicitly, whereas the x-momentum conservation is satisfied using a straightforward explicit technique. Since the discretised equations for the second direction release are very similar to the finite-difference approximations presented in Sections 6.6.1 and 6.6.2, they are not detailed herein.

6.7 Boundary conditions

6.7.1 Walls

The boundary conditions devised for the present non-orthogonal coordinate system, can be explained by firstly considering the methodologies implemented in conventional Cartesian numerical approaches. Staggered grid structures require special treatment along the perimeters of the flow domain since velocity components and surface elevations need to be defined *outside* the boundary walls (Leendertse, 1967; Kuipers and Vreugdenhil, 1973; Falconer, 1976). Figure 6.7 illustrates a section of perimeter on a typical Cartesian shallow water equation staggered mesh. The 'extrapolated' or 'external' velocity components depend upon whether a no-slip or a free-slip boundary condition is implemented. For example, in the no-slip situation, the boundary velocities depicted in Figure 6.7 are:

$$\left.\begin{aligned} U_{i_w,j} &= 0 \\ V_{i_w+1,j} &= -V_{i_w,j} \\ V_{i,j_w} &= 0 \\ \text{and} \quad U_{i,j_w+1} &= -U_{i,j_w} \quad , \end{aligned}\right\} \qquad (6.27)$$

whilst for the slip condition, the corresponding velocities become:

$$\left.\begin{aligned} U_{i_w,j} &= 0 \\ V_{i_w+1,j} &= V_{i_w,j} \\ V_{i,j_w} &= 0 \\ \text{and} \quad U_{i,j_w+1} &= U_{i,j_w} \quad . \end{aligned}\right\} \qquad (6.28)$$

Surface elevation boundary conditions are usually obtained by stipulating a zero gradient normal to the wall, i.e.,

$$\left.\begin{aligned} &\frac{\partial \zeta}{\partial x} = 0 \qquad \text{along 'vertical/x=constant' walls} \\ &\text{and} \\ &\frac{\partial \zeta}{\partial y} = 0 \qquad \text{along 'horizontal/y=constant' walls.} \end{aligned}\right\} \quad (6.29)$$

With reference to Figure 6.7, the above derivative expressions are formulated in finite-differences as

$$\left.\begin{aligned} &\zeta_{i_w+1,j} = \zeta_{i_w,j} \\ &\text{and} \\ &\zeta_{i,j_w+1} = \zeta_{i,j_w} \quad . \end{aligned}\right\} \quad (6.30)$$

The most questionable aspects of the above boundary conditions are the surface elevation gradient assumptions, as these will clearly be in error if the simulation is to include wind stress phenomena. Furthermore, the definition of the effective stresses, T_{xx}, T_{xy} and T_{yy} at the surface elevation nodes of the present non-orthogonal scheme would create additional problems at the boundaries; the specification of a zero normal gradient for these terms is theoretically unacceptable. Therefore, it was decided that an entirely new boundary approach should be adopted.

In the present numerical investigation, the boundary curve passes through the surface elevation positions and the values of ζ on the perimeter are calculated from an explicit discretisation of the continuity equation at the start of every half time step. This technique has several advantages over the previously considered approach:

(a) The usual procedure of setting the normal derivative of surface elevation to zero at the boundaries is avoided (this would be difficult to accomplish properly in non-orthogonal systems).

(b) Effective stresses can be correctly evaluated along the perimeter of the flow field via second order forward/backward finite-difference expressions (Section 6.8).

(c) The values of surface elevation along the boundary will obey the continuity equation during transient flow phenomena.

(d) Surface elevation gradients at the wall will automatically adjust themselves to suit the hydraulics of the situation and consequently, wind stress effects can be correctly modelled (see Section 7.2).

Since the accurate simulation of momentum transfer in recirculating flows

requires the use of no-slip constraints at the lateral boundaries (Lean and Weare, 1979), it was decided that the present computational scheme should employ no-slip conditions along all solid perimeters. With reference to Figure 6.8, which represents a perimeter section of the staggered mesh structure used for the non-orthogonal numerical scheme, the velocity boundary conditions are

$$\left.\begin{aligned} U_{i_w,j} &= -U_{i_w-1,j} \\ V_{i_w,j} &= 0 \\ U_{i,j_w} &= 0 \\ V_{i,j_w} &= -V_{i,j_w-1} \end{aligned}\right\} \qquad (6.31)$$

The surface elevations along the no-flow boundaries of the hydraulic domain are calculated from the continuity equation. For example, along a ξ=constant boundary:

$$\frac{\partial U}{\partial \eta} = 0 \qquad \text{and} \qquad \frac{\partial V}{\partial \eta} = 0 \quad , \qquad (6.32)$$

and therefore the transformed continuity expression (5.91) becomes

$$\frac{\partial \zeta}{\partial t} + \frac{D}{J}\left[y_\eta \frac{\partial U}{\partial \xi} - x_\eta \frac{\partial V}{\partial \xi} \right] = 0 \quad . \qquad (6.33)$$

Similarly, for η=constant perimeter walls:

$$\frac{\partial U}{\partial \xi} = 0 \qquad \text{and} \qquad \frac{\partial V}{\partial \xi} = 0 \quad , \qquad (6.34)$$

and thus the continuity equation can be written as

$$\frac{\partial \zeta}{\partial t} + \frac{D}{J}\left[-y_\xi \frac{\partial U}{\partial \eta} + x_\xi \frac{\partial V}{\partial \eta} \right] = 0 \quad . \qquad (6.35)$$

Equations (6.33) and (6.35) are discretised using an explicit approach. For instance, referring to the 'vertical' wall illustrated in Figure 6.8, the finite-difference approximation to equation (6.33) is

$$\zeta_{i_w,j}^{n+\frac{1}{2}} = \zeta_{i_w,j}^{n} - \frac{\frac{1}{2}\Delta t\, D_{i_w,j}}{JAC_{I,J}}\left[y_{\eta_{I,J}} UXI^n - x_{\eta_{I,J}} VXI^n \right] \qquad (6.36)$$

where $I = 2i_w - 1$ and $J = 2j - 1$,

$$UXI^n = U^n_{i_w,j} - U^n_{i_w-1,j} \qquad \text{central difference approximation}$$

$$VXI^n = \frac{3}{2} V^n_0 - 2 V^n_1 + \frac{1}{2} V^n_2 \qquad \text{backward difference approximation}$$

$$V^n_0 = \frac{V^n_{i_w,j} + V^n_{i_w,j-1}}{2}$$

$$V^n_1 = \frac{V^n_{i_w-1,j} + V^n_{i_w-1,j-1}}{2}$$

and

$$V^n_2 = \frac{V^n_{i_w-2,j} + V^n_{i_w-2,j-1}}{2} .$$

Two-point averages are used to centre the V-velocities at the surface elevation/depth nodes.

The only modification required for 'left-hand/vertical' boundaries is the use of a forward difference quadratic approximation for VXI.

An analogous finite-difference discretisation is employed for determining the advanced surface elevations along η=constant perimeters.

6.7.2 Inflow and outflow boundaries

The boundary conditions across the inflow and outflow openings must be designated so as to 'drive' the hydrodynamic regime. In many shallow water equation simulations, the flows are stipulated entirely via the surface elevation. For example, estuarial and tidal harbour computations apply known sea-level oscillatory conditions across the open boundaries of the finite-difference mesh in order to model the resultant circulation patterns. Although jet-forced reservoir flows could be considered from the view point of the surface elevation gradient, it has been found more convenient to specify the hydraulic conditions according to the inflow velocities. Consequently, the flow boundary conditions utilised in the present investigation can be summarised as:

(a) Inflow conditions

The inlet boundary conditions stipulate the required depth-averaged velocity

components, U and V across the inflow opening. For example, if the inflow is parallel to the x-axis *, then

$$\left.\begin{aligned} V_{inlet} &= 0 \\ \text{and} & \\ U_{inlet} &= \text{constant} \quad . \end{aligned}\right\} \qquad (6.37)$$

The specification of a uniform inflow velocity distribution is common to other shallow water equation numerical schemes (e.g. Kuipers and Vreugdenhil, 1973; Lean and Weare, 1979). This present approach is clearly not as flexible as the stream function/vorticity-transport simulation, described in Chapter 3, which allows the inlet velocity, V_r to develop its own boundary-layer distribution. However, the hydrodynamic regime in a (ψ,ω) approach is 'driven' primarily via the difference in stream function values between the walls rather than the velocity boundary conditions. The only method of implementing a more flexible inlet velocity boundary condition for the primitive variable discretisation, would be to drive the flow by prescribing the difference in surface elevation between inflow and outflow; this technique is rejected in the present study, however, because neither the discharge nor the inlet velocity can then be specified *a priori*. Instead, the surface elevations along the inlet boundary are updated using an explicit finite-difference approximation of the continuity equation. Assuming V=0 and x_η=0 (i.e., the flow perimeter is parallel to the y-axis), then the transformed continuity equation can be written as

$$\frac{\partial \zeta}{\partial t} + \frac{1}{J}\left[y_\eta\left[U\frac{\partial D}{\partial \xi} + D\frac{\partial U}{\partial \xi} \right] - y_\xi\left[U\frac{\partial D}{\partial \eta} + D\frac{\partial U}{\partial \eta} \right] \right] = 0 \; . \qquad (6.38)$$

This expression is discretised in a similar manner to the boundary continuity equations presented in Section 6.7.1.

(b) Outflow conditions

The downstream surface elevation condition is simply

$$\zeta_{outlet} = \text{constant} \quad . \qquad (6.39)$$

* The boundary-fitted coordinate systems are generated so that the inflow direction is parallel to the x-axis.

Whilst this may appear unrealistic, it can be considered the result of an unmodelled control structure (such as a weir, etc.) downstream of the outlet. The specification of an inflow velocity and a downstream surface elevation is very similar to the methodology adopted by Wijbenga (1985a, 1985b) for modelling river flows; Wijbenga chose to stipulate the discharge along the upstream boundary and the water depths ($D=h+\zeta$) on the downstream perimeter.

Finally, in order to avoid over-specifying the downstream boundary conditions, the velocity components across the outflow are calculated in a similar manner to that proposed by Stelling and Willemse (1984):

$$\left.\begin{aligned} U_{i_o,j} &= 2\,U_{i_o-1,j} - U_{i_o-2,j} \\ \text{and} \\ V_{i_o,j} &= 2\,V_{i_o-1,j} - V_{i_o-2,j} \end{aligned}\right\} \qquad (6.40)$$

where the subscript i_o refers to the outlet perimeter.

The above velocity conditions are designed to make the outlet boundary 'non-reflective' so as to reduce the oscillatory nature of free-surface computations.

6.7.3 Corner points

The numerical schemes used to update the surface elevations along the boundary walls and inflow opening were found to be entirely satisfactory. However, the explicit continuity equation discretisations cannot be utilised at the corner points illustrated in Figure 6.9 on account of the breakdown in the validity of the velocity derivative finite-difference expressions at these positions. Therefore, the computational procedures have to adopt interpolative/extrapolative methods for evaluating the advanced values of ζ at these irregular nodes. For example, at the intersections between the circular perimeter and inlet/outlet stems (Figure 6.9a), the value of surface elevation can be found from a two-point interpolation in the direction of flow, viz.

$$\zeta^{n+\frac{1}{2}}_{i_w,j_w} = \frac{\zeta^{n+\frac{1}{2}}_{i_w-1,j_w} + \zeta^{n+\frac{1}{2}}_{i_w+1,j_w}}{2} \qquad (6.41)$$

At 'concave' corner points (Figure 6.9b), the surface elevation is determined by extrapolating along the ξ=constant and η=constant perimeter walls:

$$\left.\begin{aligned} \zeta_1 &= 2\,\zeta^{n+\frac{1}{2}}_{i_w, j_w-1} - \zeta^{n+\frac{1}{2}}_{i_w, j_w-2} \\ \zeta_2 &= 2\,\zeta^{n+\frac{1}{2}}_{i_w+1, j_w} - \zeta^{n+\frac{1}{2}}_{i_w+2, j_w} \\ \text{and} \quad \zeta^{n+\frac{1}{2}}_{i_w, j_w} &= \frac{\zeta_1 + \zeta_2}{2} \quad . \end{aligned}\right\} \qquad (6.42)$$

Whilst the interpolative/extrapolative techniques presented in equations (6.41) and (6.42) may appear unacceptable from a theoretical standpoint, they offer a pragmatic approach for dealing with the corner nodes of the computational mesh.

6.8 Evaluation of the effective stresses

The transformed effective stress equations, (5.93a), (5.93b) and (5.93c) are discretised at the surface elevation/depth positions of the staggered mesh (i.e., I = 2i - 1, J = 2j - 1). Hence the finite-difference approximations to the effective stresses are formulated as

$$T^n_{xx_{i,j}} = 2\,\rho\,\tilde{\nu}^{\,n}_{t_{i,j}} \frac{1}{JAC_{I,J}} \left[y_{\eta_{I,J}} UXI^n - y_{\xi_{I,J}} UETA^n \right] - \frac{2}{3}\,\rho\,\tilde{k}^{\,n}_{i,j} \qquad (6.43a)$$

$$T^n_{xy_{i,j}} = \rho\,\tilde{\nu}^{\,n}_{t_{i,j}} \frac{1}{JAC_{I,J}} \left[x_{\xi_{I,J}} UETA^n - x_{\eta_{I,J}} UXI^n + y_{\eta_{I,J}} VXI^n - y_{\xi_{I,J}} VETA^n \right] \qquad (6.43b)$$

and

$$T^n_{yy_{i,j}} = 2\,\rho\,\tilde{\nu}^{\,n}_{t_{i,j}} \frac{1}{JAC_{I,J}} \left[x_{\xi_{I,J}} VETA^n - x_{\eta_{I,J}} VXI^n \right] - \frac{2}{3}\,\rho\,\tilde{k}^{\,n}_{i,j} \qquad (6.43c)$$

where

$$UXI^n = U^n_{i,j} - U^n_{i-1,j}$$

$$VETA^n = V^n_{i,j} - V^n_{i,j-1}$$

$$UETA^n = \left[\frac{U^n_{i-1,j+1} + U^n_{i,j+1} - U^n_{i,j-1} - U^n_{i-1,j-1}}{4} \right]$$

and

$$VXI^{n} = \left[\frac{V^{n}_{i+1,j} + V^{n}_{i+1,j-1} - V^{n}_{i-1,j} - V^{n}_{i-1,j-1}}{4} \right] .$$

The advanced effective stresses are obtained via similar finite-difference formulae to those presented above (the only differences being the superscript index, n). For example, during the ξ-direction release [$t = n\Delta t \rightarrow (n+\frac{1}{2})\Delta t$], the advanced T_{xx} effective stress becomes

$$T^{n+\frac{1}{2}}_{xx_{i,j}} = 2\,\rho\,\tilde{\nu}_{t_{i,j}} \frac{1}{JAC_{I,J}} \left[y_{\eta_{I,J}} UXI^{n+\frac{1}{2}} - y_{\xi_{I,J}} UETA^{n+\frac{1}{2}} \right] - \frac{2}{3}\,\rho\,\tilde{k}^{n+\frac{1}{2}}_{i,j} \qquad (6.44)$$

where

$$UXI^{n+\frac{1}{2}} = U^{n+\frac{1}{2}}_{i,j} - U^{n+\frac{1}{2}}_{i-1,j}$$

and

$$UETA^{n+\frac{1}{2}} = \left[\frac{U^{n+\frac{1}{2}}_{i-1,j+1} + U^{n+\frac{1}{2}}_{i,j+1} - U^{n+\frac{1}{2}}_{i,j-1} - U^{n+\frac{1}{2}}_{i-1,j-1}}{4} \right] .$$

The remaining advanced effective stress and velocity derivative terms are evaluated in a similar fashion. If the A.D.I. scheme does not employ an iterative technique to calculate the advanced non-implicit variables, the effective stresses must be lagged at the old time level, i.e.

ξ-direction release:

$$T^{n+\frac{1}{2}}_{xx} = T^{n}_{xx} \quad ; \quad T^{n+\frac{1}{2}}_{xy} = T^{n}_{xy} \quad ; \quad T^{n+\frac{1}{2}}_{yy} = T^{n}_{yy} \qquad (6.45a)$$

and

η-direction release:

$$T^{n+1}_{xx} = T^{n+\frac{1}{2}}_{xx} \quad ; \quad T^{n+1}_{xy} = T^{n+\frac{1}{2}}_{xy} \quad ; \quad T^{n+1}_{yy} = T^{n+\frac{1}{2}}_{yy} \quad . \qquad (6.45b)$$

The velocity derivative finite-differences, presented so far, have assumed that the node is in the interior of the grid, where second-order central difference approximations can be utilised. At the boundaries of the flow domain, however, the formulae for UETA and VXI must be recast as second-order forward/backward differences. For example, at a surface elevation node, (i_w,j) along a 'left-hand/vertical' wall of the computational plane, the V-velocity gradient in the

ξ-direction is written as a quadratic forward difference:

$$\frac{\partial V^n}{\partial \xi} = VXI^n = -\frac{3}{2} V_0^n + 2 V_1^n - \frac{1}{2} V_2^n \qquad (6.46a)$$

where

$$V_0^n = \frac{V^n_{i_w,j} + V^n_{i_w,j-1}}{2}$$

$$V_1^n = \frac{V^n_{i_w+1,j} + V^n_{i_w+1,j-1}}{2}$$

$$V_2^n = \frac{V^n_{i_w+2,j} + V^n_{i_w+2,j-1}}{2}$$

Two-point averages are used to centre the V-velocities at the surface elevation/depth nodes.

In a similar fashion, VXI must be written as a second-order backward difference expression along the 'right-hand/ξ=constant' boundaries of the computational mesh:

$$\frac{\partial V^n}{\partial \xi} = VXI^n = \frac{3}{2} V_0^n - 2 V_1^n + \frac{1}{2} V_2^n \qquad (6.46b)$$

where

$$V_0^n = \frac{V^n_{i_w,j} + V^n_{i_w,j-1}}{2}$$

$$V_1^n = \frac{V^n_{i_w-1,j} + V^n_{i_w-1,j-1}}{2}$$

$$V_2^n = \frac{V^n_{i_w-2,j} + V^n_{i_w-2,j-1}}{2} \quad .$$

The velocity derivative, U_ξ, however, can still be evaluated as a central difference expression due to the fact that the staggered hydrodynamic mesh defines U-velocities at positions $\frac{1}{2}\Delta\xi$ *outside* the ξ=constant boundaries (see Figures 6.2 and 6.6). Hence, the finite-difference approximation to U_ξ remains unaltered on the perimeter of the flow field.

Second-order forward/backward differences are similarly employed to calculate the U-velocity derivatives in the η-direction along η=constant perimeters.

6.9 Time step restrictions

For explicit numerical schemes, the spatial grid increment and the time step employed in the computational procedure are interrelated via the so-called Courant-Friedricks-Lewy stability criterion. In a one-dimensional Cartesian simulation, this may be expressed as:

$$\sigma = \frac{(gD)^{\frac{1}{2}} \Delta t}{\Delta x} \leq 1 \qquad (6.47)$$

where σ = Courant number,

g = acceleration due to gravity,

D = local water depth = $h + \zeta$,

Δt = time step,

and Δx = mesh increment.

The above formula is usually obtained from a Fourier analysis of the damping of error components in the linearised momentum equations (a von Neumann stability analysis); for a detailed exposition on the subject of stability, the reader is referred to Richtmeyer and Morton (1967). Equation (6.47) restricts the time step, Δt, so that the effects of a free surface gravity wave do not propagate more than one grid increment per time step. Since the Courant stability criterion is derived from consideration of the linearised equations, it is reasonable to expect that the time step may need to be further reduced if the non-linear advective acceleration terms are included in the computational scheme. Consequently, explicit non-linear approaches often require very small time steps in order to maintain numerical stability (e.g., da Silva Lima, 1981). The extension of the Courant time step constraint to two-dimensional simulations is rather ambiguous; for example, da Silva Lima chose to define the Courant number as

$$\sigma = \frac{(gD)^{\frac{1}{2}} \Delta t}{(\Delta x^2 + \Delta y^2)^{\frac{1}{2}}} \leq 1 \quad , \qquad (6.48)$$

whilst Johnson (1980) indicated that two separate one-dimensional restrictions can be utilised i.e.,

$$\sigma = \frac{(gD)^{\frac{1}{2}} \Delta t}{\Delta x} \leq 1 \quad \text{and} \quad \sigma = \frac{(gD)^{\frac{1}{2}} \Delta t}{\Delta y} \leq 1 \; . \qquad (6.49)$$

With implicit numerical methods, a von Neumann analysis indicates that linearised systems are unconditionally stable. The inclusion of the non-linear

acceleration terms and boundary conditions, however, will introduce instabilities into the computational scheme, so that in practice the Courant number generally never exceeds about 5. Even if the numerical method is stable beyond this value, it is likely that the free surface wave propagation and velocity fields will be poorly modelled (see Benque et al., 1982). Consequently, the present computational scheme restricts the Courant number to approximately 3. (The stability criteria for the transformed shallow water momentum equations (5.92a) and (5.92b) have not actually been derived; instead, Cartesian time step restrictions are applied to the cell increments of the physical (x,y) coordinate mesh).

As discussed in Section 5.2, the numerical solution may employ a rigid-lid technique in situations where the fluctuations in water level are expected to be small in comparison to the total depth of flow, D. The implementation of a rigid-lid approach reduces the severity of the time-step restriction, because the constraint on Δt is then dependent upon the fluid velocity rather than the free surface wave celerity; i.e., the time step restriction for an explicit scheme is then given by

$$\Delta t \leq \frac{\Delta x}{U} \quad \text{and} \quad \Delta t \leq \frac{\Delta y}{V} \quad . \tag{6.50}$$

Numerical experimentation has revealed, however, that for the rigid-lid mode of operation, the present computational scheme can only achieve a time step of approximately one third that presented in equation (6.50). Nevertheless, the utilisation of a rigid-lid technique allows the model to employ larger time steps and may help to reduce the excessive seiching motions which often occur at the start of free surface computations.

6.10 Digital filtering

The occurrence of instabilities in boundary-fitted shallow water equation simulations appears to be a common fault of the method. For example, Johnson (1980) had to use large values of eddy viscosity coefficient in order to prevent an unbounded growth of grid-scale oscillations swamping the numerical solution. Häuser et al. (1986b), in a numerical study of the *linearised* shallow water equations for a boundary-fitted mesh representative of a tidal harbour, found that their computational scheme became unstable after approximately 15000 time steps (equivalent to about 24 hours of simulation). In this latter case, the destabilising influences could not be attributed to the advective acceleration terms since they were not included in the discretisation. Häuser et al. were unable to offer an explanation for the instabilities

and merely stated that they could not be eliminated by spatial filtering techniques (see Shapiro, 1970, 1971 and 1975).

For the present jet-forced reservoir flow investigations, large advective terms were anticipated in the shear layers between the recirculation and through-flow regions. Consequently, the numerical scheme adopted Stelling's second order upwind differencing for the advective discretisation with the aim of preventing numerical instabilities. Although grid scale velocity oscillations ('flutter') were eradicated by the finite-differencing procedures, the numerical simulation still encountered stability problems in the form of a gradual growth in surface elevation oscillations for low values of eddy viscosity coefficient; these grid scale variations in the surface elevation were eventually transmitted to oscillations in the velocity field. At first, this instability was attributed to the imperfect time centering of the advective acceleration terms. However, the problem could not be eliminated by the use of an iteration technique to calculate the advanced non-implicit variables and therefore it was concluded that the instabilities were caused either by the boundary conditions or the inability to use second order upwind differences for all cross-advective derivatives.

Since Häuser et al. (1986b) had little success with spatial filtering procedures, a second-order recursive digital *temporal* filter was employed to eliminate the high-frequency surface elevation noise. The scheme adopted by Butler (1978a) to control non-linear instabilities in a Cartesian shallow water equation simulation, was found to be particularly effective for the present investigation; this involves applying a *low-pass* smoothing filter to damp the high frequency oscillations at the end of every half time step. In the present scheme, the temporal filter is applied to all primary hydrodynamic variables, i.e., ζ, U, V, ϵ and k. With reference to the surface elevation (ζ), the recursive digital filter can be expressed as

$$\zeta^{n+\frac{1}{2}} = a\ \bar{\zeta}^{n+\frac{1}{2}} + b\ \zeta^{n} + c\ \zeta^{n-\frac{1}{2}} \qquad (6.51)$$

where $\bar{\zeta}^{n+\frac{1}{2}}$ represents the latest water level calculated by the A.D.I. algorithm,

$\zeta^{n+\frac{1}{2}}$ is the filtered surface elevation value which is used in subsequent flow calculations,

and a, b, c are coefficients which are chosen to damp the high-frequency oscillations, such that:

$$a + b + c = 1 \quad . \qquad (6.52)$$

In order to allow the longer period wave motions to remain almost undisturbed, the coefficient, a, should be as close to unity as possible. Numerical experimentation can be used to select the values of the coefficients in the recursive filter. After studying a wide range of parameters, ranging from (a, b, c) = (0.89, 0.1, 0.01) to (0.4, 0.4, 0.2), Butler chose the coefficients to be

$$a = 0.6 \quad , \quad b = 0.3 \quad \text{and} \quad c = 0.1 \quad . \tag{6.53}$$

Unless otherwise stated, the same parameters are also used herein.

The filter shown in equation (6.51) undoubtedly introduces false diffusion into the numerical scheme during transient flow phenomena. If the hydrodynamic solution tends to a steady state, however, the effects of the recursive filter become less significant since the hydrodynamic variables will change ever more slowly with time as the equilibrium condition is approached. Moreover, Butler tested a linearised Cartesian shallow water equation numerical scheme with and without the filter and concluded that the procedure could be satisfactorily utilised for modelling transient flows such as tidal regimes. Provided the time step is small compared to the tidal cycle (which is usually the case because of the Courant stability criterion), the filter introduces no appreciable differences in the flow simulation.

6.10.1 Filter stability

Although the coefficients, a, b and c are determined partly by trial and error, they must be chosen so as to maintain the stability of the filter. Consider a general second order digital filtering equation:

$$y_i = b\, y_{i-1} + c\, y_{i-2} + g(x_i) \tag{6.54}$$

where y_i are the filtered values,

x_i are the unfiltered data,

and g is a linear function of x_i :

$$g(x_i) = (1 - b - c)\, x_i = a\, x_i \quad . \tag{6.55}$$

Otnes and Enochson (1972) indicate that the transfer function for the filter presented above is given by

$$H(f) = \frac{a}{1 - b.\exp(-j2\pi\Delta t f) - c.\exp(-j4\pi\Delta t f)} \tag{6.56}$$

where Δt is the filtration interval (not to be confused with the time step used for the A.D.I. procedure),

f is the frequency,

and $j = \sqrt{-1}$.

The stability is determined by considering the denominator of the transfer function; values of b and c within the triangle, illustrated in Figure 6.10, result in stable filters, whilst values outside the triangular region produce unstable filters (see Otnes and Enochson).

An alternative/additional stability criterion can be found by considering the 'unit response function' of the filter, i.e., the output from the filter when the input signal is a single delta function at time t = 0, viz.

$$x_i = \begin{cases} 1 \ , & i = 0 \\ 0 \ , & i \neq 0 \end{cases} \quad . \tag{6.57}$$

The filter will only be stable if the unit response function is also stable. Letting the amplitude reduction factor for each application of the filter be r, and denoting p successive filtrations by

$$y_{i+p} = r^p \, y_i \tag{6.58}$$

where r is a real number, then the unit response will be stable if and only if

$$|r| \leq 1 \quad . \tag{6.59}$$

Starting from the current value of y_k, it is possible to extrapolate backwards to obtain:

$$\left.\begin{aligned} y_{k-1} &= r^{-1} \, y_k \\ \text{and} \quad y_{k-2} &= r^{-2} \, y_k \quad . \end{aligned}\right\} \tag{6.60}$$

Using the unit response input, shown in equation (6.57), the recursive filter can therefore be expressed as

$$y_k - b \, r^{-1} \, y_k - c \, r^{-2} \, y_k = 0 \qquad \text{for} \quad k \neq 0 \tag{6.61}$$

which may be rearranged to give

$$r^2 - b\,r - c = 0 \quad . \tag{6.62}$$

For the filter to be stable, the roots of the above expression must not violate the condition prescribed in equation (6.59).

6.11 Determination of stream function contours

The results from the boundary-fitted shallow water equation program are usually presented in the form of velocity vectors at the scalar-variable positions. This is achieved using two-point averages for both the U- and V-velocity interpolations, i.e., considering a scalar-variable node, (i,j), the velocity vector to be plotted is evaluated from:

$$\bar{U} = \frac{U_{i,j} + U_{i-1,j}}{2} \qquad \text{and} \qquad \bar{V} = \frac{V_{i,j} + V_{i,j-1}}{2} \quad . \tag{6.63}$$

In most cases, the depiction of the flow field using velocity vectors is sufficient to illustrate the salient features of the hydrodynamic regime. However, in certain instances, the flow structure may be better represented as contours of stream function. The advantages of this approach include:

(a) The elimination of optical illusions caused by plotting the velocity vectors at nodes along the curvilinear (ξ,η) coordinate system. (Optical illusions are clearly exhibited in Figures 7.5b and 7.6b where the velocity vectors give the impression of curved flow, even though the actual flow structure is uniform throughout the channel).

(b) Ease of comparison of reservoir circulation patterns with results from the stream function/vorticity-transport discretisation, presented in Chapter 3.

(c) Weak secondary recirculation structures become more discernible when stream function contours are drawn.

Since the computational procedures use a (U,V,ζ) approach, the values of ψ must be determined numerically by integrating the velocity/depth fields across the hydraulic domain. The additional truncation errors in the integration process inevitably lead to a reduction in the accuracy of the stream function evaluation, compared with the (ψ,ω) technique.

With reference to Figure 6.11, showing an elemental line segment AB, let the

stream function at A be ψ, and at B be $\psi+\delta\psi$. The definition of stream function applicable to depth-averaged computations is:

$$U = \frac{1}{D}\frac{\partial\psi}{\partial y} \qquad \text{and} \qquad V = -\frac{1}{D}\frac{\partial\psi}{\partial x} \tag{6.64}$$

where D is the local water depth at the point being considered.

Consequently, the change in stream function from point A to point B can be expressed as

$$\delta\psi \approx D\,[\,U\delta y - V\delta x\,] \tag{6.65}$$

which may be rewritten in terms of the elemental chord length, δs as

$$\delta\psi \approx D\,[\,U\sin\theta - V\cos\theta\,]\,\delta s \tag{6.66}$$

where θ is the angle of inclination of the line, AB with respect to the x-axis.

Hence the discharge between two points, P_1 and P_2 joined by a curve, s, is

$$Q = \Delta\psi = \int_{s=s(P_1)}^{s=s(P_2)} D\,[\,U\sin\theta - V\cos\theta\,]\,ds \quad . \tag{6.67}$$

The above s-parameterisation allows the numerical integration to be performed along the curvilinear coordinate lines of the non-orthogonal physical mesh. Considering the reservoir boundary-fitted grids presented in Chapter 4 (e.g., Figures 4.8, 4.11 and 4.12), it can be seen that the primary flow will be in the ξ-direction. The integral shown in equation (6.67) can therefore be determined more accurately by choosing the path approximately normal to the major flow direction; thus, the numerical integration is performed along lines of constant ξ (ξ-lines).

6.11.1 Stream function numerical integration procedure

Figure 6.12 shows a section of the hydrodynamic variables along a line of constant ξ. The evaluation of stream function at the scalar-variable mesh positions is accomplished by:

(a) Determining the scalar-variable node distances, s along the ξ=constant curve (measured from the lower flow perimeter):

$$\left.\begin{aligned} s_0 &= 0 \\ s_1 &= \left[(x_{\frac{1}{2}}-x_0)^2+(y_{\frac{1}{2}}-y_0)^2\right]^{\frac{1}{2}} + \left[(x_1-x_{\frac{1}{2}})^2+(y_1-y_{\frac{1}{2}})^2\right]^{\frac{1}{2}} \\ s_2 &= s_1 + \left[(x_{1\frac{1}{2}}-x_1)^2+(y_{1\frac{1}{2}}-y_1)^2\right]^{\frac{1}{2}} + \left[(x_2-x_{1\frac{1}{2}})^2+(y_2-y_{1\frac{1}{2}})^2\right]^{\frac{1}{2}} \\ &\vdots \end{aligned}\right\} \quad (6.68)$$

i.e., the chord lengths are evaluated over the high resolution, geometric variable finite-difference mesh.

(b) Evaluating, at each surface elevation node:

$$f = D\left[\bar{U}\sin\theta - \bar{V}\cos\theta\right] \quad (6.69)$$

where θ is the angle of inclination of the ξ-line with respect to the x-axis. Hence, θ can be evaluated from the transformation derivatives, y_η and x_η as follows:

$$\theta_{I,J} = \tan^{-1}\left[\frac{\partial y}{\partial \eta} \div \frac{\partial x}{\partial \eta}\right]_{I,J} = \tan^{-1}\left[\frac{YETA_{I,J}}{XETA_{I,J}}\right] . \quad (6.70)$$

(c) Using a cubic spline to approximate the function, F where,

$$f = F(s). \quad (6.71)$$

(d) Integrating F between successive scalar-variable nodes, using a standard numerical integration technique (for example, Simpson's Rule) to obtain the stream function increment, $\Delta\psi$ between each mesh point.

By assigning the lower boundary of the reservoir a constant ψ-value (usually 0), the stream function at the interior nodes along the ξ-line can be obtained by successive additions of the stream function increment, $\Delta\psi$, calculated in (d).

The entire process is repeated for each ξ-line in the computational plane. For the presentation of the circular reservoir stream function contours, shown in Chapter 7, it is convenient to scale the values of ψ, so that the stream function along the upper wall has the same value as that used for the Navier-Stokes simulations, described in Chapter 3. This permits a direct visual comparison of the recirculation patterns predicted by the two different numerical approaches.

6.12 Concluding remarks

This chapter has described the numerical algorithms devised for solving the depth-averaged momentum and continuity equations on a boundary-fitted coordinate system. The solution technique employs a semi-implicit numerical scheme based upon Leendertse's well established Cartesian algorithms (1967, 1970). In order to suppress non-linear instabilities when modelling flows with low values of eddy viscosity coefficient, two additional numerical mechanisms have to be utilised. The first involves discretising the cross-advective acceleration terms in the momentum expressions using a second-order upwind differencing technique (Stelling, 1983) whilst the second procedure uses a second-order digital low-pass recursive filter to eliminate high frequency oscillations from the primitive variable fields (Butler, 1978a).

During the implicit stages of the numerical algorithm it is necessary to restrict the number of unknown variables in each flow equation to three in order to implement a tri-diagonal solution technique. Since the non-orthogonal hydrodynamic equations contain approximately twice as many individual components as their Cartesian counterparts, it is inevitable that many expressions at the future time level need to be lagged. Furthermore, the explicit momentum stages will also require a lagging procedure for certain advanced variables. In order to improve the time accuracy of the computer model, the user has the option of calculating the advanced non-implicit expressions via an iteration technique. This involves solving the implicit and explicit flow equations for a given directional release several times; the flowchart illustrated in Figure 6.13 summarises the computational methods employed to update the hydrodynamic variables during the ξ-direction release. Numerical experimentation indicates, however, that the iteration technique is computationally expensive and not really warranted when modelling flows which tend to a steady-state. Therefore, the examples presented in Chapter 7, ignore the iterative procedure and use lagged values for all advanced non-implicit parameters.

The solution of the shallow water equations on boundary-fitted coordinate systems is a relatively novel technique and therefore extensive validation must be conducted on the proposed numerical algorithms. This involves running the computer code for flow cases which have analytical solutions. Furthermore, additional tests with complex recirculating flows should be compared with data from small-scale experimental models and alternative numerical approaches. The results of the procedures used to validate the boundary-fitted hydrodynamic algorithms are presented in the following chapter.

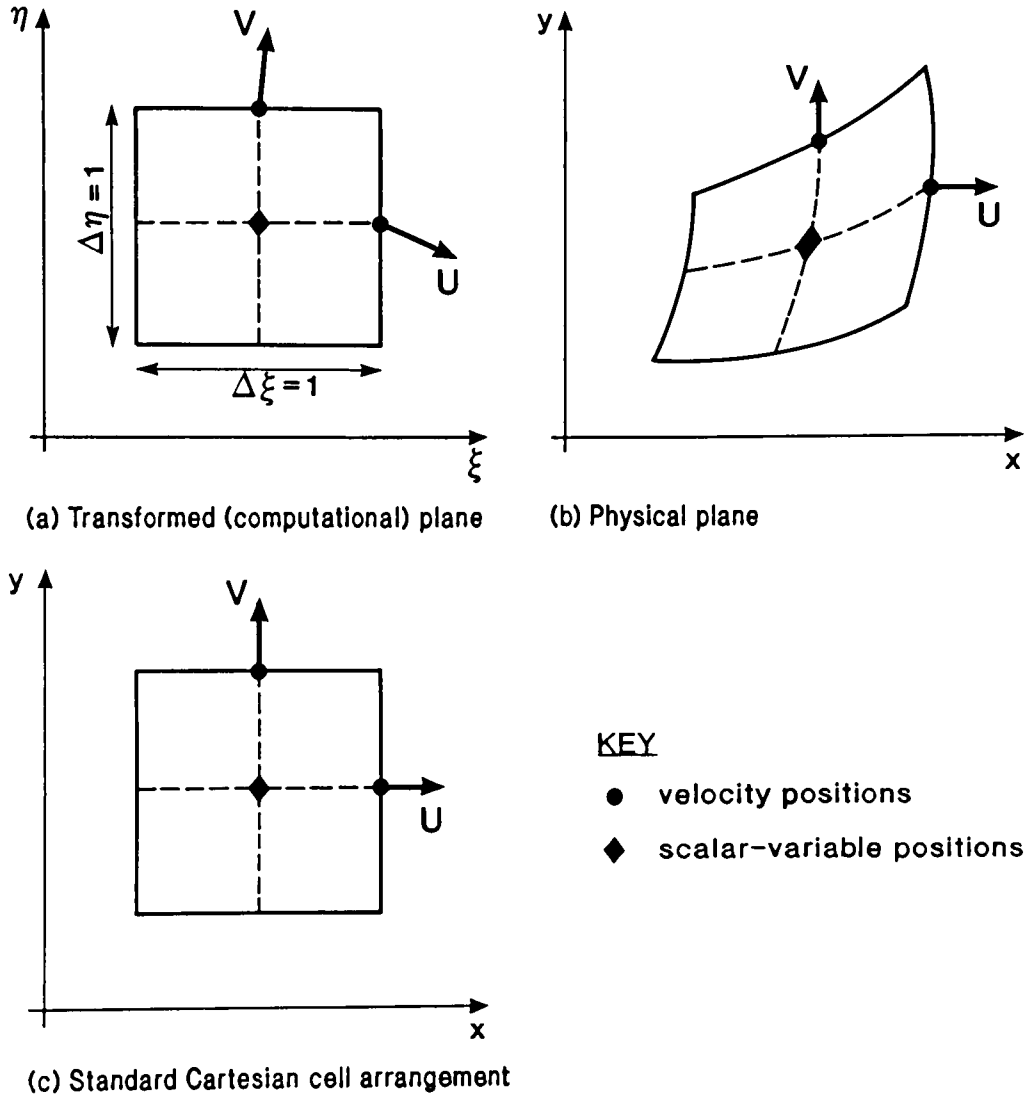
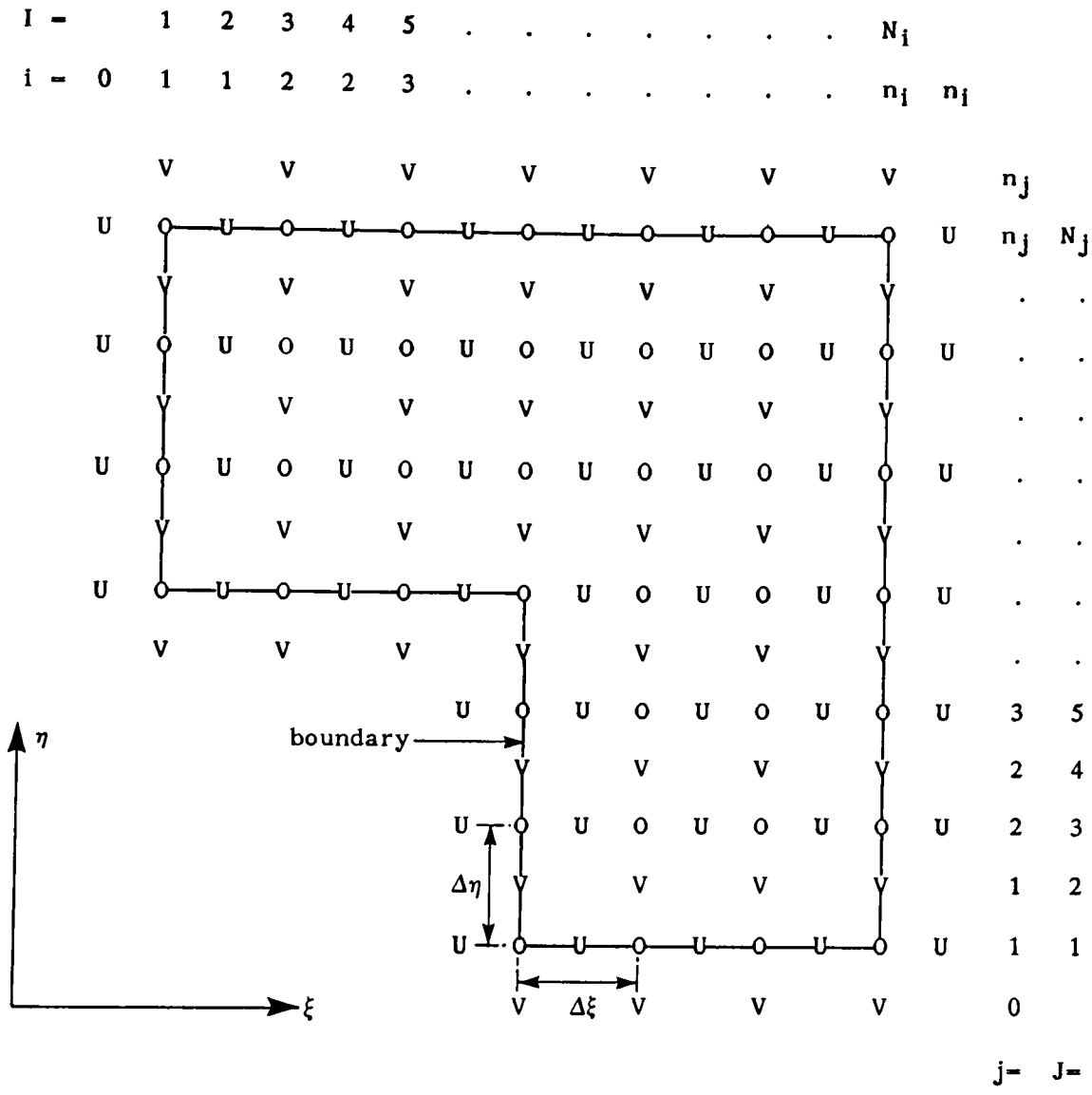


Figure 6.1 Cell definition



KEY

I,J : geometric spatial indices.

i,j : hydrodynamic spatial indices.

O : scalar-variable positions (ζ, h, D, etc.).

U : U-velocity positions.

V : V-velocity positions.

Figure 6.2 Staggered grid system : transformed (ξ,η) plane.

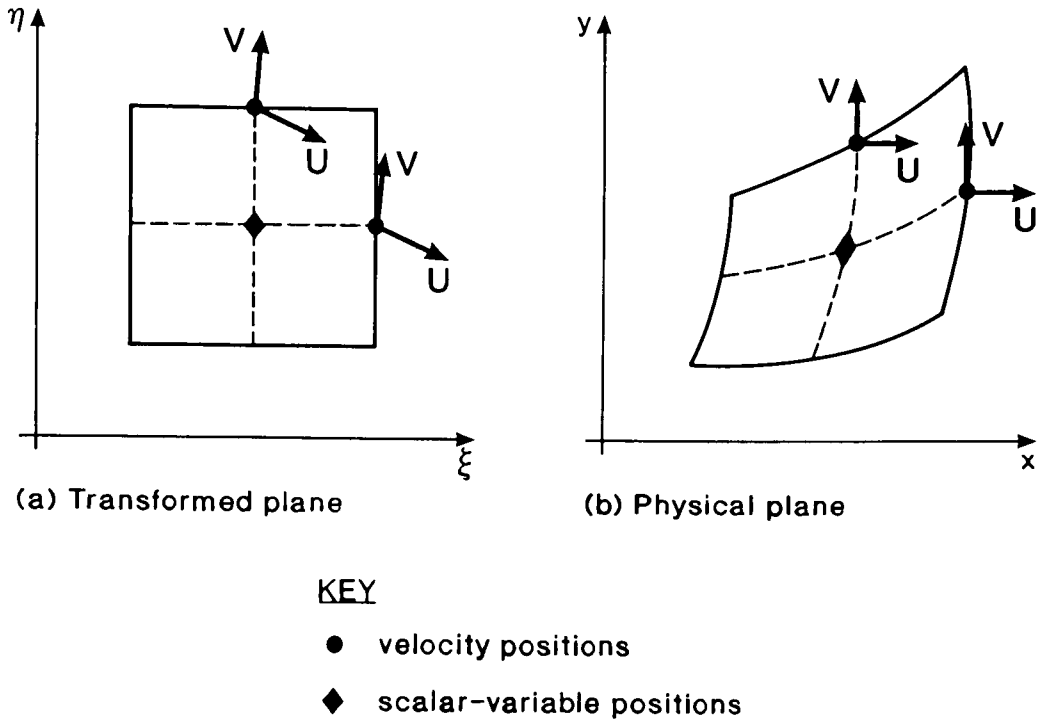
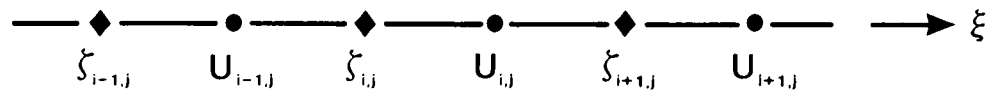
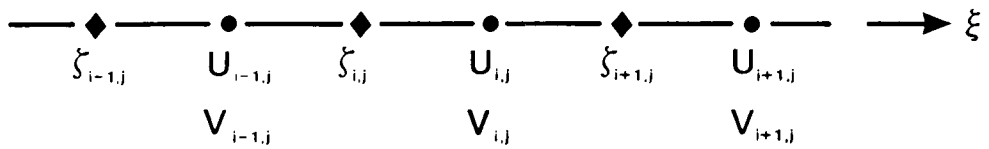


Figure 6.3 Johnson's/Häuser et al.'s cell definition



(a) Grid layout adopted for the current investigation



(b) Grid layout utilised by Johnson (1980) and Häuser et al. (1985)

Figure 6.4 Diagrammatic representation of 'line-wise' ξ-direction releases

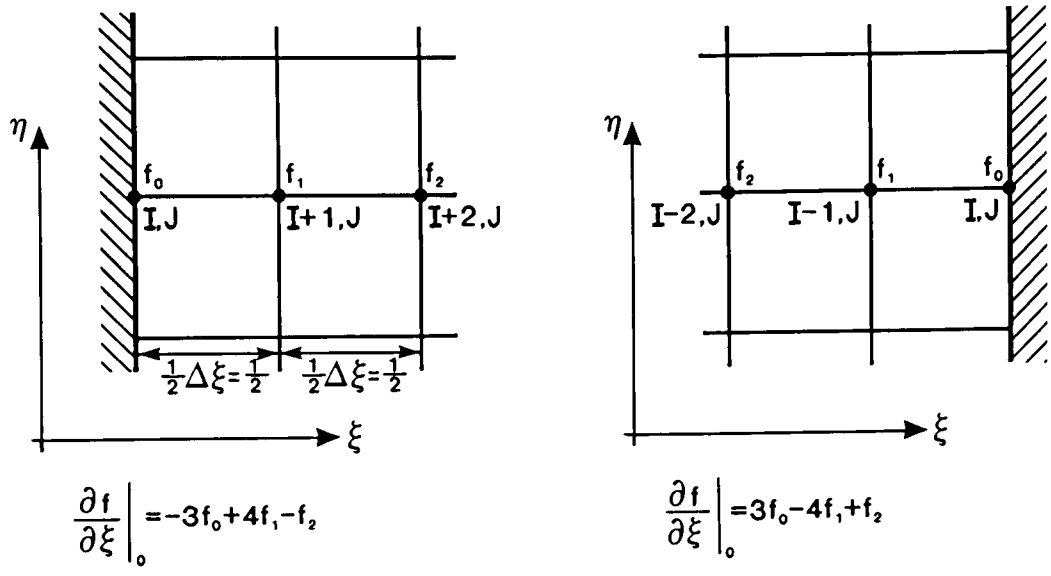
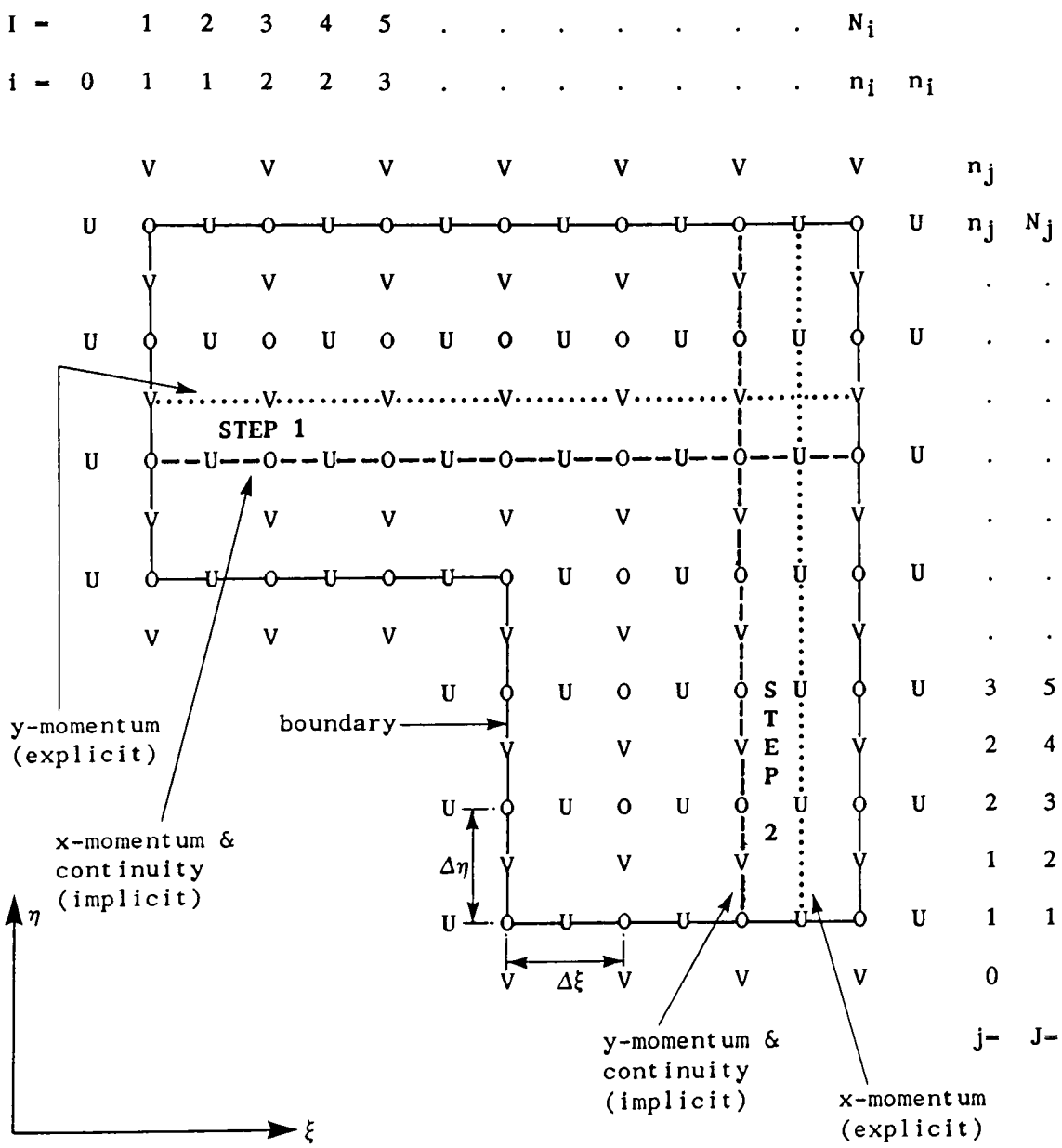


Figure 6.5 Evaluation of ξ-direction spatial derivatives along the 'vertical' boundaries of the transformed plane



KEY

—— : boundary.

– – – : represents an implicit line-wise flow calculation.

..... : symbolizes explicit computations.

STEP 1 : ξ-direction release : $t=n\Delta t \rightarrow (n+\frac{1}{2})\Delta t$.

STEP 2 : η-direction release : $t=(n+\frac{1}{2})\Delta t \rightarrow (n+1)\Delta t$.

Figure 6.6 Diagrammatic representation of semi-implicit numerical scheme : transformed (ξ,η) plane.

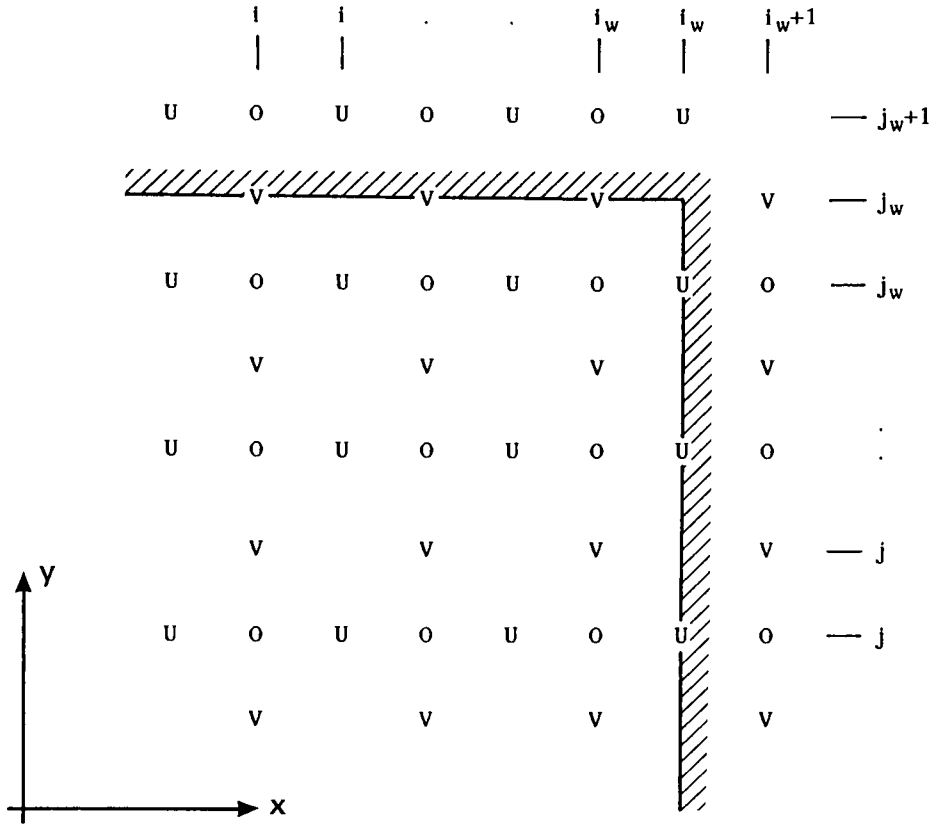


Figure 6.7 Conventional (Cartesian) treatment of boundaries

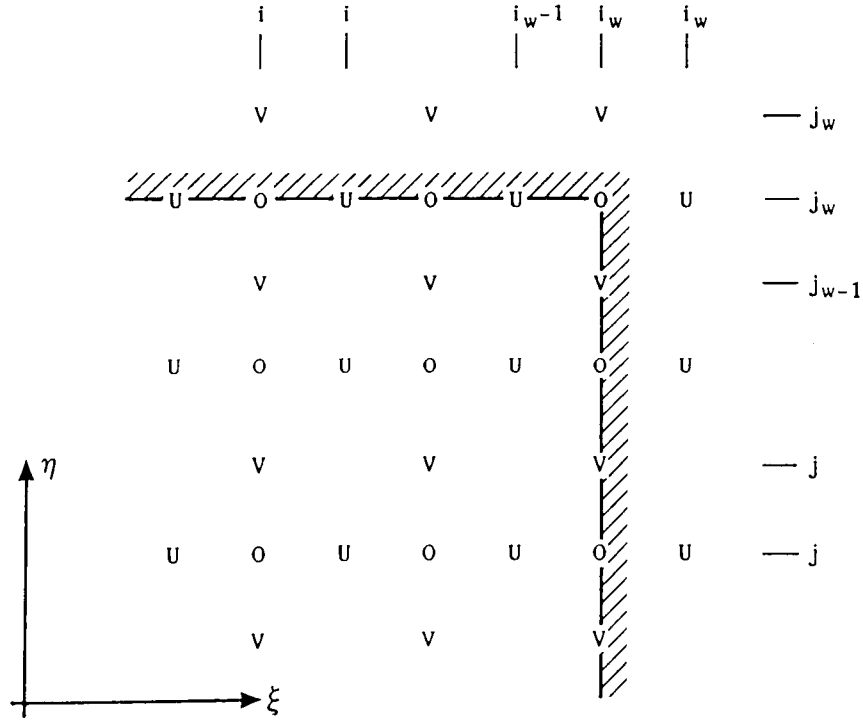


Figure 6.8 Boundary locations for non-orthogonal scheme

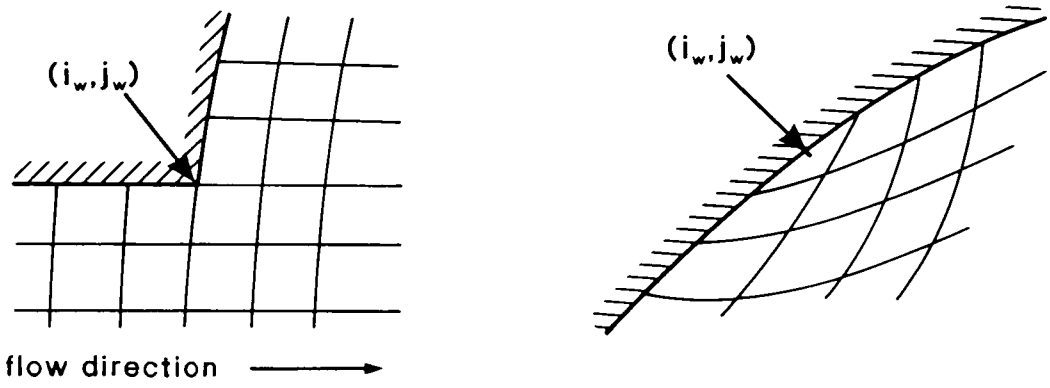


Figure 6.9 Irregular node positions

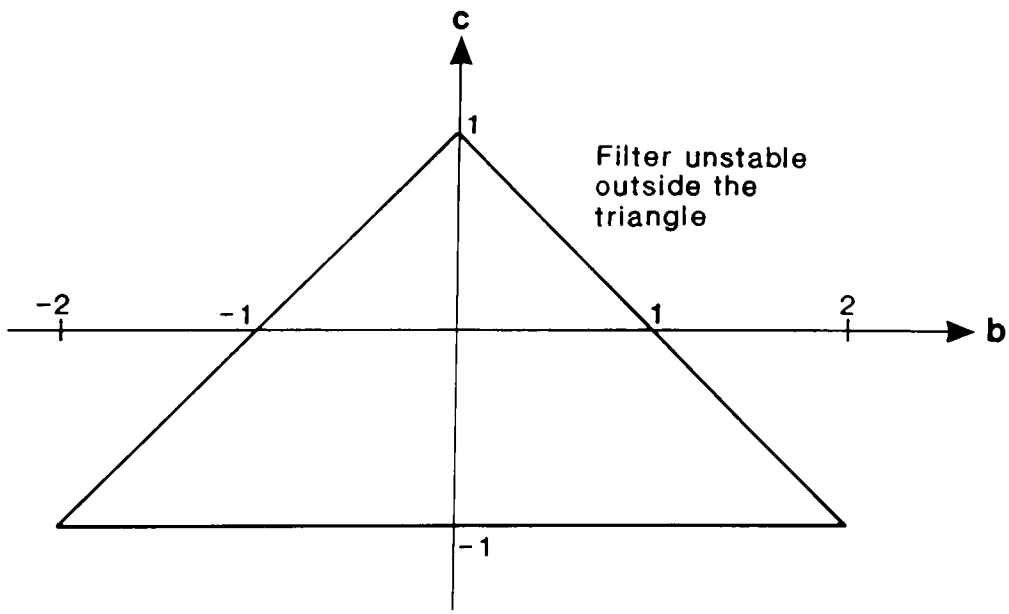


Figure 6.10 Filter stability in terms of the coefficients, b and c (after Otnes and Enochson, 1972)

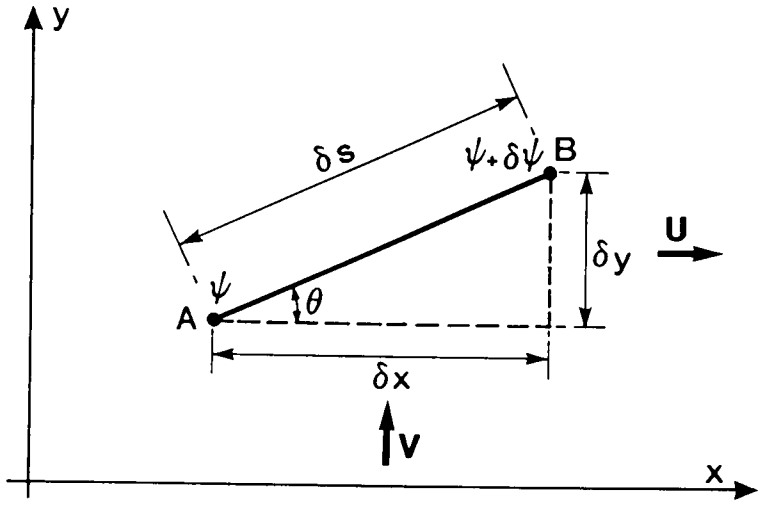


Figure 6.11 Elemental line segment used for the definition of stream function

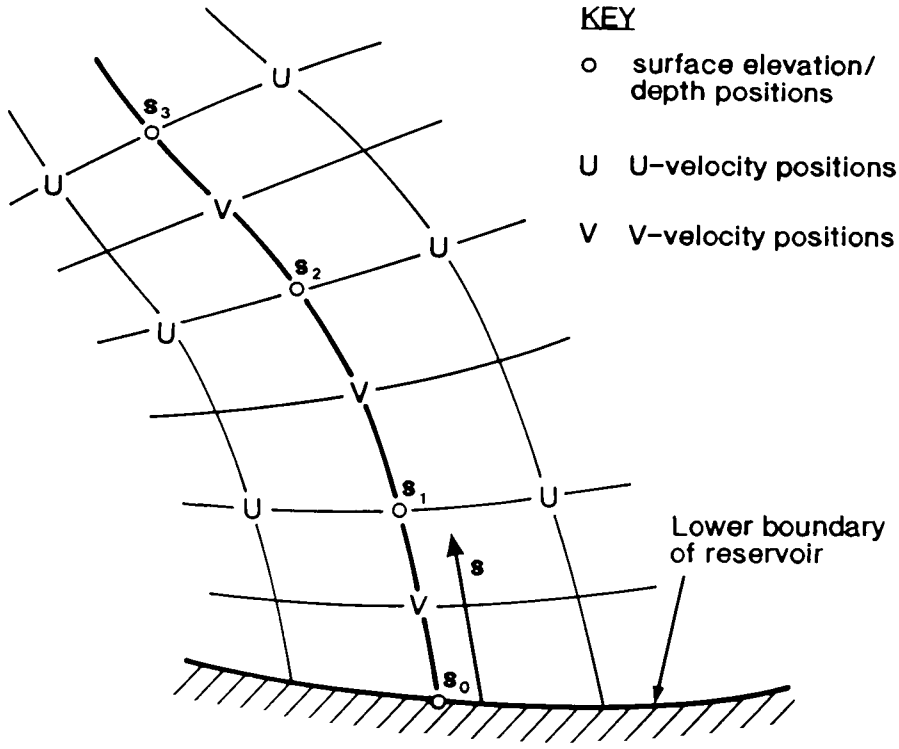


Figure 6.12 Diagrammatic representation of ξ-line stream function numerical integration

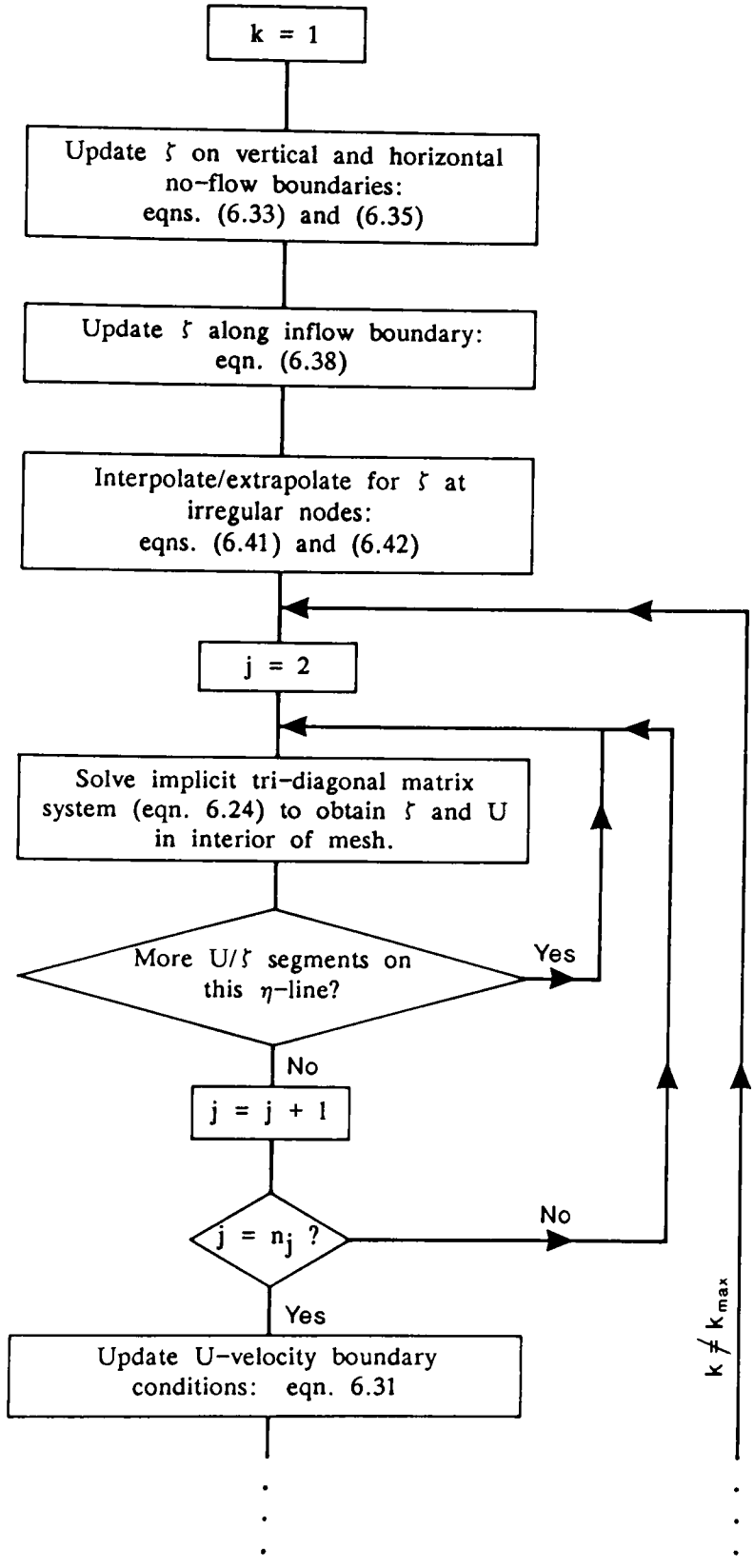


Figure 6.13a Flow chart illustrating computational procedures of ξ-direction release

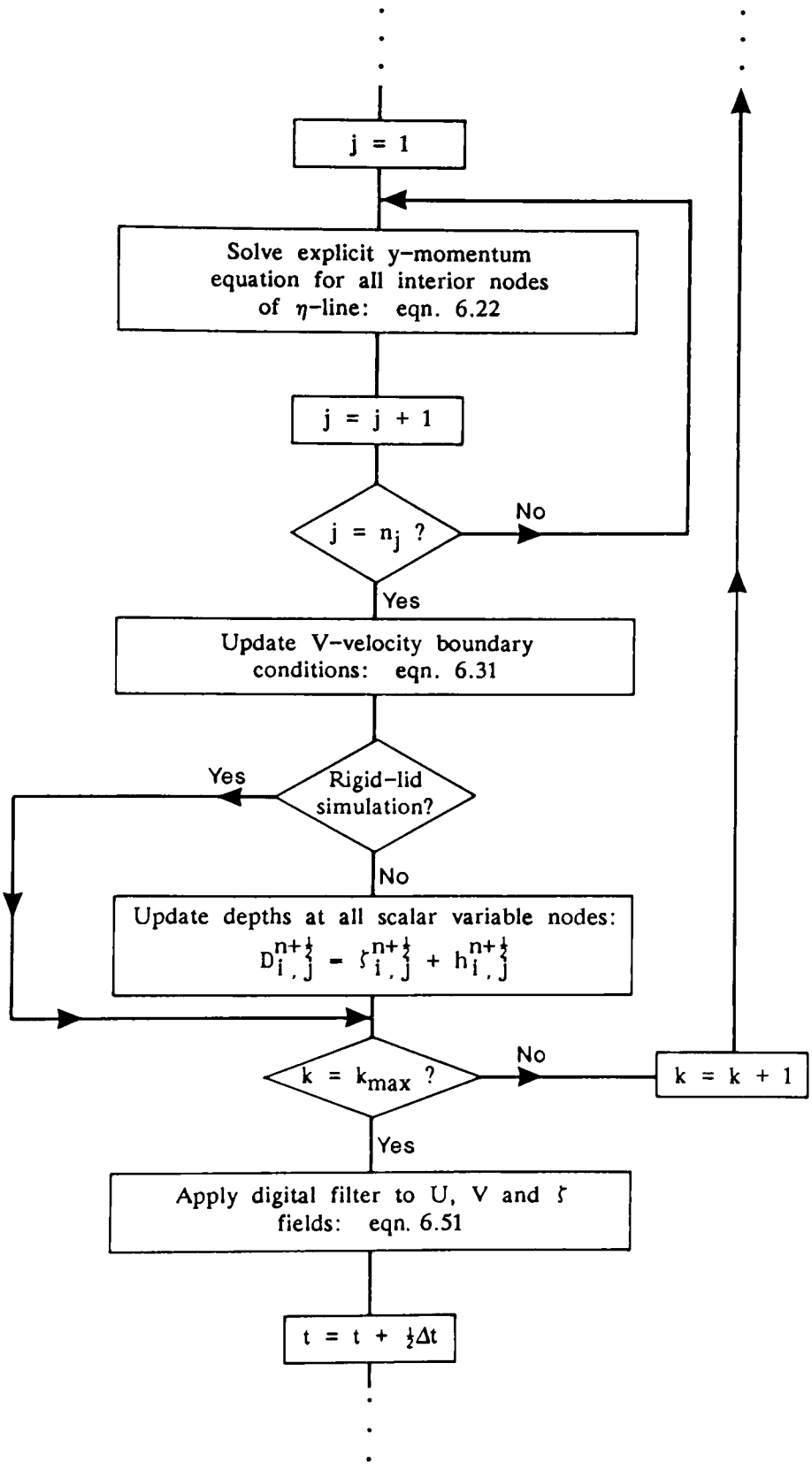


Figure 6.13b Flow chart illustrating computational procedures of ξ-direction release (continued)

CHAPTER 7
RESULTS

7.1 Introduction

In order to validate the computational procedures which have been proposed in the previous chapter, it is necessary to compare flow predictions from the numerical model with accurate comparative data from independent studies. As with many other types of engineering fluid flow, reliable benchmark test data of jet-forced reservoir circulation patterns are rather scarce. Ali et al. (1978), Ali and Pateman (1981) and Ali (1983, 1985) have conducted extensive laboratory-sized experimental studies into reservoirs of varying geometries ranging from flat-bottomed circular basins to reservoirs which have boundaries and bathymetries representative of prototypical situations. Although the above-mentioned studies are probably the most accurate source of benchmark data currently available for momentum driven reservoir flow, they are unsuitable for the validation of the boundary-fitted shallow water equation solver because of the small size of the inlet jet in proportion to the reservoir area. For example, Ali and Pateman (1981) and Ali (1983) studied radial and tangential jet-forced flow in a circular container of radius 1.5 m. Since the jet width was only 2 mm, the geometry would require a prohibitively fine curvilinear mesh in order to resolve the velocity structure in the shear layers of the jet.

Field data from river and estuary flows have also been reported widely in the literature, yet accurate velocity measurements in regions of recirculation appear to have received little attention. For the purposes of the research studies presented herein, data from analytical, alternative numerical and small-scale experimental investigations are compared with predictions from the computational model. The validation exercise is presented in three sections; the first deals with wind-induced surface elevation effects in shallow bodies of water, the second deals with steady uniform flow in an open rectangular channel, and the third (and major) validation section is concerned with jet-forced flow in flat-bottomed circular reservoirs. The jet-forced flow comparisons are conducted using data obtained from the Navier-Stokes simulation presented in Chapter 3, and from the small-scale laboratory investigation and coarse Cartesian grid numerical model studied by Falconer (1976, 1980). The validation sections, therefore, provide a rigorous test of the numerical algorithms proposed in Chapter 6. In particular, the direct comparison of momentum driven circulation data obtained from the boundary-fitted shallow water equation scheme with that from the Navier-Stokes simulation of Chapter 3, allows an assessment to be made of the artificial (numerical) diffusion introduced by the second

order upwind differences used to discretise the cross-advective terms of the curvilinear hydrodynamic equations.

The final section of the chapter demonstrates the versatility of the boundary-fitted coordinate system by examining the flow in an arbitrary-shaped reservoir with a highly contorted perimeter. Although the circulation patterns are uncorroborated by experimental data, the results show that the transformed shallow water equation solver is capable of predicting large-scale flow phenomena in irregular geometries which are typically encountered in the strategic management of river basins and service reservoirs.

7.2 Validation against wind-induced surface elevation effects

For the first series of validation tests, the program simulated wind-induced surface elevation effects in circular and elliptical wedge-shaped basins. As already explained in Section 5.8, the depth-averaging process causes details of the vertical velocity structure to be lost and therefore the two-dimensional shallow water equations are unable to model wind-driven circulation currents. Nevertheless, wind-induced free surface effects *can* be analysed by the depth-averaged hydrodynamic equations and consequently wind shear stresses may be utilised as a means of checking the validity of the numerical scheme.

7.2.1 Circular wedge-shaped basin

Figure 7.1a shows a coordinate mesh composed of 21 x 21 nodes (10 x 10 flow-variable cells) which was used to simulate wind-induced free surface effects in a 10 m deep flat-bottomed wedge-shaped basin. The mesh lines were constructed algebraically from a cylindrical polar coordinate reference frame so that the perimeters of the basin were defined by $5 \leq r \leq 10$ km and -0.7 rad $\leq \theta \leq 0.7$ rad. A surface wind stress of 0.1 N/m^2 was imposed in the eastward direction (i.e. τ_{wx} = 0.1 N/m^2 and τ_{wy} = 0.0 N/m^2), the Coriolis parameter was zero, the coefficient of eddy viscosity, $\tilde{\nu}_t$, was set equal to 1000 m^2/s and the fluid density was taken as 1000 kg/m^3. Using a time step of 30 s, the flow simulation converged to equilibrium after approximately 15,000 s of damped seiching motion. Figure 7.1b illustrates the steady-state surface elevation contours at t = 30,000 s, and shows identical agreement with predictions obtained by Sheng and Hirsh (1984) using a boundary-fitted systems model based upon the contravariant velocity components.

The contour pattern consists of straight parallel lines; this is completely in accordance with an analysis of the force balance, which indicates that the surface elevation gradients are given by

$$\frac{\partial \zeta}{\partial x} = \frac{\tau_{wx}}{\rho\, g\, D} \qquad \text{and} \qquad \frac{\partial \zeta}{\partial y} = \frac{\tau_{wy}}{\rho\, g\, D} \tag{7.1}$$

where τ_{wx} and τ_{wy} are the x- and y-direction wind stress components, ρ is the fluid density, g the acceleration due to gravity and D is the water depth.

7.2.2 Elliptical wedge-shaped basins

Following Sheng and Hirsh (1984), wind-induced surface effects were also simulated in elliptical wedge-shaped basins. In this case, the mesh nodes were obtained by transforming the circular wedge investigated in Section 7.2.1, using the coordinate transformation

$$\left.\begin{aligned} x &= P \cos \theta \\ \text{and} \\ y &= Q \sin \theta \end{aligned}\right\} \tag{7.2}$$

where P and Q are the scale factors for the major and minor axes of the ellipse. Two geometries were investigated, one with (P,Q) = (3,1) shown in Figure 7.2 and the other with (P,Q) = (10,1), illustrated in Figure 7.3. In each case, the wind stresses and flow parameters were identical to those utilised for the circular wedge but the stretching process described by equation (7.2) allowed the time step to be increased to 100 s. Figure 7.2b presents the steady-state surface elevation contours at t = 75,000 s for the first elliptical wedge, whereas Figure 7.3b illustrates the free surface contours at t = 200,000 s for the more distorted geometry. As for the case of the cylindrical wedge-shaped domain, the exact solution for the surface elevation is a linear function of x and therefore the numerical scheme produces excellent agreement with the analytical solution.

The increased simulation time necessary for the elliptical basins to reach equilibrium can be attributed to the fact that the mapping process results in much larger dimensions for the flow domains. For example, whilst the circular wedge shown in Figure 7.1 is 5 km across, the meshes presented in Figures 7.2 and 7.3 are 15 km and 50 km in length, respectively.

Perhaps the most important distinguishing feature between the circular and

elliptical wedge simulations is the fact that the circular validation exercise uses an orthogonal mesh whereas the elliptical test cases are conducted with non-orthogonal coordinate lines. The satisfactory simulation of wind-induced surface elevation effects on the highly skewed coordinate mesh illustrated in Figure 7.3a demonstrates the feasibility of solving the transformed shallow water equations on non-orthogonal meshes. As discussed in Section 4.5, boundary-fitted coordinate systems are usually generated with the two sets of coordinate lines intersecting at angles greater than 45° in order to reduce the truncation errors associated with grid skewness. Since the surface elevation contours agree with theory, even when the coordinate lines are less than 15° apart (Figure 7.3a), it may be concluded that a skewness constraint of 45°, discussed in Section 4.5, is probably a very conservative estimate of the non-orthogonality limit.

7.2.3 Circular basin

The final series of wind stress tests considered surface elevation gradients in a 10 m deep flat-bottomed circular basin of diameter 10 km. Figure 7.4a presents a 21 x 21 node boundary-fitted grid generated using the numerical techniques described in Chapter 4. It is evident that the elliptic grid generation equations tend to increase the coordinate line spacing in regions of concave boundary curvature, leading to some of the cells along the circular perimeter having aspect ratios of approximately 2:1. Figure 7.4b illustrates the steady-state surface elevation contours for a wind stress of 0.1 N/m^2 in an eastward direction; again the contour pattern is completely in accordance with the analytical force balance equation, despite the cell-size variation and grid skewness in the vicinity of the singularities on the circular perimeter. The flow computation utilised a time step of 30 s and required approximately 40,000 s of simulation time before the transient free surface oscillations fully disappeared.

A second test was conducted using a surface wind stress of 0.1 N/m^2 in a north-eastward direction (i.e., $\tau_{wx} = \tau_{wy} = 0.07071$ N/m^2). Figure 7.4c depicts the steady-state surface elevation contours at t = 75,000 s and demonstrates that the accuracy of the numerical scheme is uninfluenced by the direction of the applied wind stress. Furthermore, the ± 5 mm contours next to the singularity positions at $\theta = \pi/4$ and $\theta = -3\pi/4$ show no deviation from the theoretical solution and therefore the approximate extrapolative scheme presented in equation (6.42) appears to be quite satisfactory for determining the surface elevations at 'concave' corner positions.

7.3 Validation against steady uniform flow in a rectangular channel

After a period of damped seiching motion, the test cases studied in Section 7.2 all settled to a state of *static* equilibrium with the depth-averaged velocity components tending to zero. Consequently, it was judged that a further series of validation exercises were warranted which would test the numerical scheme in a dynamic flow situation. The test devised by Wijbenga (1985a) for examining the accuracy of the Delft Hydraulics Laboratory orthogonal curvilinear shallow water equation program, WAQUA, was considered to be ideal for the present validation purposes. This involves simulating steady uniform flow in a rectangular channel using a *deliberately* distorted numerically generated coordinate mesh. The accuracy of the simulation can easily be estimated by comparing the predicted velocities with the Chezy friction law for open channel flow.

Following Wijbenga (1985a), the program simulated steady uniform flow in a rectangular channel 1000 m long by 240 m wide, with a water depth of 5.0 m at the downstream boundary. The channel had a constant bed slope of 1 in 1000 and a Chezy roughness coefficient of 41.42 $m^{\frac{1}{2}}/s$. The eddy viscosity coefficient, $\tilde{\nu}_t$, was set to zero, and in contrast to all other simulations investigated in this chapter, *slip* boundary conditions (as utilised by Wijbenga) were applied to the side walls of the flow domain. Besides altering the velocity boundary conditions, the specification of slip walls along the sides of the channel required the finite-difference discretisation of the continuity boundary condition to be modified. Assuming that the channel walls are formed from η=constant lines and providing the coordinate system is generated with the slip boundaries parallel to the x-axis, then the following conditions can be invoked along the side walls of the flow field:

$$V = 0 \quad , \quad \frac{\partial V}{\partial \xi} = 0 \quad \text{and} \quad \frac{\partial y}{\partial \xi} = 0 \quad . \tag{7.3}$$

Substituting the above expressions into the transformed continuity equation (5.91) leads to the surface elevation boundary condition being recast as

$$\frac{\partial \zeta}{\partial t} + \frac{1}{J}\left[y_\eta \left[U\frac{\partial D}{\partial \xi} + D\frac{\partial U}{\partial \xi} \right] + x_\xi D\frac{\partial V}{\partial \eta} \right] = 0 \quad . \tag{7.4}$$

Equation (7.4) was discretised using an explicit forward-time finite-difference technique similar to that already described in Section 6.7.1 for the no-slip constraint. Inflow and outflow velocity conditions were stipulated as $\partial U/\partial x = 0$ and $\partial V/\partial x = 0$.

Figures 7.5a and 7.6a illustrate the 81 x 21 node (40 x 10 flow cell) coordinate

meshes that were used in the validation exercise. The coordinate systems are representative of those utilised by Wijbenga, but are generated so that the grids lines are non-orthogonal. The flow simulations were started from an initial condition of $U = V = 0$, but in order to reduce the computation time, the initial surface elevation profile was specified as having the same slope as the bed, in order to avoid large transient free surface oscillations. Using a time step of 5 s, the numerical scheme converged to equilibrium after approximately 5000 s of simulation time. The velocity vectors in Figures 7.5b and 7.6b show that the flow field is uniform throughout the channel even though at first inspection there is an optical illusion of curved flow due to the fact that the velocity vectors are plotted at the grid points of the curvilinear mesh. In fact, the flow features may be better represented by plotting interpolated contours of surface elevation, as indicated in Figures 7.5c and 7.6c. Here, the contours are again equi-spaced parallel lines, indicating a linear drop in surface elevation along the channel (with the same slope as that of the bed). This is completely in accordance with normal flow conditions. Furthermore, the velocities predicted by the numerical simulation show excellent agreement with the Chezy friction law for normal channel flow:

$$U = C\sqrt{R\,S_0} = 41.42 \times \sqrt{5 \times 10^{-3}} = 2.9288 \text{ m/s} \qquad (7.5)$$

where C is the Chezy coefficient,

R is the hydraulic radius of the channel $\approx$ flow depth in this case,

and S_0 is the bed slope.

In order to demonstrate the truncation errors in the numerical scheme, Figures 7.7 and 7.8 present the transverse surface elevation profiles and U-velocity profiles at the section halfway along the channel (i.e. 500 m from the upstream inflow boundary). The diagrams indicate that the discretisation errors become larger as the coordinate mesh becomes more distorted. However, even for the more distorted of the two meshes, the maximum error in surface elevation at the halfway section is less than 0.05 mm whilst the maximum error in the U-velocity component is less than 0.005 m/s. These errors are considerably smaller than those presented by Wijbenga (1985a) for his orthogonal curvilinear shallow water equation solver. This is probably attributable to the fact that the numerical scheme proposed in Chapter 6 uses a staggered grid arrangement in which the geometrical derivatives (x_ξ, x_η, y_ξ and y_η) are evaluated at every flow variable position whereas Wijbenga's scheme requires some of the geometrical derivatives to be found from two-point interpolation. The main disadvantage with the present double-mesh system is that the numerical mapping data requires four times the storage capacity of Wijbenga's algorithms. Nevertheless, the increased spatial accuracy of the proposed

non-orthogonal scheme vindicates the decision to use the computationally more expensive split indexing system.

Finally, it should be noted that the small discretisation errors depicted in Figures 7.7 and 7.8 imply that it is satisfactory to generate boundary-fitted meshes which have quite large cell-size variations between adjacent nodes.

7.4 Validation against jet-forced flow in circular reservoirs

The tests presented in Sections 7.2 and 7.3 have indicated that the non-orthogonal numerical scheme is capable of modelling wind-induced free surface effects and gravity driven flows in open channels. However, although the validation exercises have proved that the geometrical transformations used in the numerical mapping have been correctly derived and coded, the previously described flow examples were unable to test the discretisation of the non-linear advective acceleration and diffusive effective stress terms. It was judged that the prediction of jet-forced flow patterns in flat-bottomed circular reservoirs with vertical side walls would provide an excellent opportunity to test the non-linear terms of the transformed shallow water equations. In this case, two-dimensional analytical solutions are available for jet-forced flow into a circular cylinder as well as alternative numerical studies and experimental data. Three separate circular reservoir geometries were investigated; Dennis' (1974) symmetrical circular container, Mills' (1977) asymmetrical geometry and the small-scale laboratory investigation and coarse Cartesian grid numerical model studied by Falconer (1976, 1980). Details of the dimensions and geometrical properties of the boundary-fitted coordinate systems for these cases have already been presented in Section 4.6.

In jet-forced hydrodynamic regimes, the non-linear advective acceleration terms generally dominate the flow and consequently the studies detailed in this section should provide a severe test of the capabilities of the non-orthogonal numerical scheme. The Navier-Stokes simulations reported in Chapter 3 are a valuable source of independent data for strongly recirculating flow conditions and are therefore utilised for the validation of the boundary-fitted shallow water equation program. The decision to discretise the advective terms of the vorticity-transport equation using central-differences proved to be ideal for the purposes of providing reliable benchmark tests. This is because the central difference advective scheme for the vorticity-transport equation should produce less numerical diffusion than the second order upwind difference algorithms implemented in the boundary-fitted primitive

variable approach. Consequently, direct comparison of predictions from the stream function/vorticity-transport and primitive variable discretisations should indicate whether the second order upwind scheme gives rise to a significant increase in the level of numerical diffusion.

7.4.1 Dennis' geometry

For the initial series of reservoir circulation tests, a relatively coarse boundary-fitted mesh, having a central core of 61 x 61 nodes (30 x 30 flow-variable cells) was generated, as shown in Figure 7.9. The parallel sided inlet and outlet channels were placed diametrically opposite each other and were created so that the openings into the circular reservoir subtended an angle of $\pi/15$ radians (as measured from the centre of the circle). This is representative of the geometry used by Dennis (1974) for the numerical investigation of low Reynolds number jet-forced flow inside a circle. Dennis, however, did not allow for boundary layer effects at the entrance or exit and so had no reason to consider the inlet or outlet channels. The small number of mesh points was primarily chosen so that the various test exercises could be completed using a relatively small amount of computer C.P.U. time.

The circular basin was assumed to have a diameter of 1.5 m, a depth of approximately 0.1 m and an inflow velocity of 0.1 m/s. No-slip boundary conditions were applied to the perimeter walls of the flow domain, and for the first series of tests, the bed friction coefficient, c_f, was set to zero so that the results were independent of the base boundary layer shear stresses. This allowed the numerical model to simulate the planar flow conditions which were studied in Chapter 3. Furthermore, the removal of bottom stress terms provided a stringent stability test for the boundary-fitted numerical procedures. This is because the discretisation of the shallow water equations becomes more unstable when the damping influence of bed friction is removed (see Weare, 1976b). The time step employed in the numerical scheme was 0.05 s, corresponding to an approximate Courant number of 2.5.* A constant eddy viscosity of 0.000784 m^2/s was utilised across the whole flow

* Applying the 'one-dimensional' definition of Courant number, presented in equation (6.47), to the flow cell at the centre of the circular reservoir gives

$$\sigma = \frac{(gD)^{\frac{1}{2}} \Delta t}{\Delta x} = \frac{(9.81 \times 0.10)^{\frac{1}{2}} \times 0.05}{0.02} \approx 2.5 \quad .$$

domain, giving an inlet Reynolds number (as defined in Chapters 2 and 3) of Re_I=10.0; this allowed a direct comparison with the Navier-Stokes simulation illustrated in Figure 3.5.

It was found that the initial gravity wave formed by the impulsive start of the flow computation led to an uncontrollable growth in surface elevation oscillations at grid points near to the outlet opening. These amplified clapotis effects are attributed to the initial transient free surface wave (travelling in the positive x-direction) being reflected back towards the inflow by the vertical sides of the reservoir, causing a very disturbed surface elevation field; the instability is exacerbated in the present case because the computational scheme ignores the damping effects of the bottom shear stress. However, numerical experimentation revealed that the clapotis effects could be eliminated by building up the flow velocity at the inlet boundary over a period of time. Instead of using a linear ramp to prescribe the increase in the inflow velocity with time, the following cubic function was found to be more suitable in avoiding shocks:

$$\left.\begin{aligned} U_{in}(t) &= \left[-2\left[\frac{t}{T_r} \right]^3 + 3\left[\frac{t}{T_r} \right]^2 \right] . U_{ss} \qquad t \leq T_r \\ U_{in}(t) &= U_{ss} \qquad\qquad\qquad\qquad\qquad\qquad t > T_r \end{aligned}\right\} \qquad (7.6)$$

where T_r is the length of the ramp in seconds,

U_{ss} is the desired inflow velocity at *steady-state*,

and $U_{in}(t)$ is the inflow velocity at time, t.

The advantage in employing the above function, instead of a linear variation in time, arises from the fact that dU/dt is zero at both t = 0 and t = T_r, thereby avoiding the discontinuities in velocity derivative which occur with a linear profile. Trial-and-error was used to find a suitable value for the length of the ramp; a value of T_r = 100 s was found to provide a sufficiently long build-up period to prevent the occurrence of spurious surface elevation effects.

Figure 7.10a shows the steady-state velocity vectors at t = 600 s (i.e. after 12,000 time steps). A visual comparison of the velocity fields presented in Figures 3.5c and 7.10a indicates that the Navier-Stokes simulation and primitive variable boundary-fitted numerical scheme produce very similar flow solutions. Moreover, Figure 7.10a demonstrates that the procedures developed for solving the transformed

curvilinear shallow water equations are capable of predicting highly non-linear flow regimes*. The boundary-fitted program utilised approximately 1.6 seconds of C.P.U. time per iteration on a PR1ME 9955 computer and therefore required a total of 320 minutes C.P.U. time for the computation to reach steady-state. Although the total run-time may seem excessive since the central core of the boundary-fitted mesh was only composed of 30 x 30 flow cells, the last 300 seconds (of simulation time) were in fact used to ascertain whether the remnants of the initial free surface disturbance had fully disappeared. The complete convergence of the surface elevation field is particularly important whenever surface elevation effects are to be plotted because small changes in ζ can lead to very large changes in the positions of the surface contours. Consequently, it is probably rather ambiguous to compare the computational costs of the present scheme with those of the stream function/vorticity-transport discretisation presented in Chapter 3 since the Navier-Stokes simulations do not allow for free surface effects and are therefore unaffected by the seiching motions which occur in the (U,V,ζ) formulation.

Figure 10b presents the surface elevation contours at steady-state (t = 600 s) and illustrates that the water depth inside the circular reservoir basin is almost uniform. The flow in the narrow inlet and outlet stems is ramped with a rapid but almost linear change in surface elevation with distance. However, the gradients of the free surface in the narrow feeder channels are considerably larger than expected. These excessive surface gradients can be attributed to the fact that the boundary-fitted mesh has insufficient resolution within the confined passageways to be able to simulate the large velocity derivatives which occur near the no-slip perimeter walls. An approximate method of reducing the surface gradients caused by the poor mesh resolution would be to specify *slip* boundary conditions along the side walls of the flow stems; however, this would contradict Flokstra's (1977) recommendation to use no-slip boundary conditions whenever secondary circulation patterns are modelled. Consequently, the only accurate method of overcoming the problem is to refine the computational mesh. The constraint on the amount of computer C.P.U. time that

* Until the completion of this initial simulation, it was not known whether the numerical scheme would be able to model the strongly recirculating hydrodynamic conditions which are associated with jet-forced reservoir flow. Furthermore, the excessive cell-size variations and grid non-orthogonality near to the corner point singularity positions were considered to be a probable source of numerical instability; fortunately, the flow computations showed no evidence of any destabilising effects at these particular locations.

could be utilised, however, led to the decision to continue the validation exercise using the relatively coarse boundary-fitted coordinate mesh presented in Figure 7.9.

In order to investigate the flow predictions from the Navier-Stokes and primitive variable numerical schemes in a more systematic manner, the computed transverse profiles of U-velocity across the mid-section of the reservoir are compared. By necessity, the flow velocities and coordinates from the two systems have to be non-dimensionalised to enable a direct graphical comparison. Defining the average inflow velocity, $U_{av(inlet)}$, as the characteristic velocity scale and the radius of the basin, R_0, as the characteristic length scale, the non-dimensionalised U-velocity profiles across the mid-section of the reservoir (i.e., along $x = 0$) are presented in Figure 7.11. The solid curved line depicts the velocity distribution computed from the stream function/vorticity-transport discretisation, whereas the crosses indicate the profile predicted by the boundary-fitted primitive variable scheme. Apart from the discrepancies in the velocities at the centreline of the reservoir, Figure 7.11 shows good agreement between the two numerical approaches, especially in the recirculation zones either side of the main throughflow. In fact, the agreement is remarkable when it is considered that the boundary-fitted mesh uses approximately one-eighth the number of grid points of the (ψ,ω) formulation.* However, the erroneous central throughflow velocities predicted by the primitive variable approach is further evidence to suggest that an increase in the resolution of the coordinate mesh is warranted.

Referring back to Figure 7.10b, it can be seen that the displacement of the reservoir free surface from its initial (at rest) state was only about 1.2 mm in the central region of the flow domain. Since this change in ζ was only a small fraction of the total water depth (100 mm), it should be possible to replace the free surface calculation by the rigid-lid procedure, described in Section 5.2, without loss of accuracy. The frictionless-lid approach updates the surface elevation using the depth-averaged momentum expressions in the usual manner, but the total depth at each grid point (utilised in the continuity equation) remains unmodified, thereby effectively converting the hyperbolic shallow water equations into a more stable elliptic formulation. Rigid-lid techniques are generally utilised whenever the free

* The Navier-Stokes simulation depicted in Figure 3.5, utilises a polar mesh with 180 x 40 coordinate lines (i.e. 7200 nodal points); the present boundary-fitted grid employs a central core of 30 x 30 flow-variable cells (i.e. approximately 900 grid points).

surface is expected to undergo little vertical movement (McGuirk and Rodi, 1978; Officier et al., 1986; Wind and Vreugdenhil, 1986). Of course the main disadvantage with the approach is the violation of the exact depth-averaged continuity equation:

$$\frac{\partial \zeta}{\partial t} + \frac{\partial (UD)}{\partial x} + \frac{\partial (VD)}{\partial y} = 0 \quad . \qquad (5.17)\ \text{bis}$$

Instead, the procedure solves the approximate depth-averaged continuity equation given by

$$\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} = 0 \quad . \qquad (7.7)$$

Figure 7.12a illustrates the steady-state velocity vectors in the reservoir after 600 s of rigid-lid simulation (the flow parameters were identical to those utilised for the free surface calculation, described earlier in the section). The surface elevation contours presented in Figure 7.12b, are very similar to the results obtained from the free surface computation (Figure 7.10b). Although the *rigid-lid* 1.2 mm contour in the circular central region appears to deviate considerably from the position occupied by the *free-surface* 1.2 mm contour, it must be remembered that ζ is almost constant across the circular basin. Consequently, slight discrepancies in the value of ζ can lead to large changes in the position of surface elevation contours.

The velocity profile across the mid-section of the reservoir is presented in Figure 7.13 and is almost identical with the distribution obtained from the free surface computation illustrated in Figure 7.11. It can therefore be concluded that the rigid-lid technique is acceptable for numerically simulating the nearly-horizontal flow regimes associated with steady-state jet-forced reservoir circulation. However, the boundary-fitted shallow water equation solver was also developed to allow a third mode of operating the surface elevation computation. This final technique can be referred to as a *semi-rigid-lid* procedure since the methodology lies mid-way between the free surface calculation and the rigid-lid constraint. In principle, the method employs the same stabilising influence as the rigid-lid approach, i.e. the water depths at grid points are held constant despite changes to the free surface elevation. However, in order to account for the variations in free surface from the initial still water level, it was found that the water depths could be updated periodically without losing the additional stability of the frictionless-lid constraint. Numerical experimentation was used to find the optimum time increment between the modifications to the depths; for the present boundary-fitted mesh it was discovered that the interval should be at least 50 seconds of simulation time (corresponding to

1000 time steps).

Figure 7.14 presents the steady-state surface elevation contours determined using the semi-rigid-lid approach. Comparison with the free surface computation, illustrated in Figure 7.10b, reveals that there is almost an exact agreement in the positions of the contour lines. Furthermore, although the semi-rigid-lid simulation was run to t = 600 s, in order for the calculation to be consistent with the free surface approach, it was found that steady-state conditions were actually obtained as early as t = 400 s. Therefore, the semi-rigid-lid technique is considered to be the most suitable mode for operating the boundary-fitted shallow water equation solver because the computational costs of the simulation can be reduced without having to sacrifice the accuracy of the free surface calculation. As a consequence, all subsequent flow computations described in this chapter adopt the semi-rigid-lid procedure. Finally, in order to demonstrate that the mode of operation for the surface calculation has little effect upon the resultant surface elevation predictions, Figure 7.15 compares the longitudinal profile of ζ obtained from the various numerical approaches. The graph presents the variation of water level along the reservoir's longitudinal axis of symmetry (i.e. y = 0). It can be seen that the distributions of ζ obtained from the three approaches are extremely similar and consequently there is little to be gained by selecting the computationally more expensive free surface technique in this case.

The reservoir circulation tests described up to this point have ignored the bed shear stress terms in order to simulate the planar flow conditions modelled in Chapter 3. This section of the validation procedure investigates the effect of introducing a realistic bed friction coefficient into the numerical scheme. The reservoir was assumed to have a smooth bed, and therefore the bottom friction coefficient was approximated using the formula presented by Schlichting (1968):

$$c_f = 0.027 \left[\frac{\nu}{U R} \right]^{\frac{1}{4}} \qquad (5.81 \text{ bis})$$

where ν is the coefficient of kinematic viscosity,

U is the magnitude of the depth-averaged velocity vector,

and R is the hydraulic radius of the channel $\approx$ flow depth for wide flow regimes.

The bottom friction coefficient for smooth beds thus depends upon the local depth-averaged velocity vector. However, as Rodi (1984b) points out, it is generally acceptable to employ a constant value of c_f across the entire flow domain. This

can be substantiated by computing typical values of bottom friction coefficient for different parts of the jet-forced velocity field. For example:

$$\left.\begin{aligned} &\underline{\text{At the inlet: } U = 0.1 \text{ m/s}} \\ &\therefore\ c_f = 0.027 \left[\frac{1.14 \times 10^{-6}}{0.1 \times 0.1} \right]^{\frac{1}{4}} = 0.00279 \\ &\text{and} \\ &\underline{\text{In the centre of the reservoir : } U \approx 0.05 \text{ m/s}} \\ &\therefore\ c_f = 0.027 \left[\frac{1.14 \times 10^{-6}}{0.05 \times 0.1} \right]^{\frac{1}{4}} = 0.00332 \quad . \end{aligned}\right\} \qquad (7.8)$$

The above calculations illustrate Rodi's supposition that the value of c_f in small-scale laboratory investigations is typically 0.003.

Figure 7.16 illustrates the steady-state transverse U-velocity profile across the mid-section of the reservoir obtained from a semi-rigid-lid simulation employing a constant bed friction coefficient of 0.003. In a similar manner to the previously presented U-velocity profiles, the solid curved line in Figure 7.16 represents the benchmark velocity distribution calculated from the Navier-Stokes discretisation (i.e., without bed friction). The velocity profile computed by the boundary-fitted primitive variable scheme (depicted by the crosses) is remarkably similar to the distributions illustrated in Figures 7.11 and 7.13 for the zero bed friction simulations. It can therefore be concluded that bottom friction is relatively unimportant when modelling the circulation patterns in small-scale circular reservoirs.

A second boundary-fitted coordinate system, also representative of Dennis' circular geometry was generated as shown in Figure 7.17. The mesh had the same sized central core as the earlier curvilinear grid (i.e., 61 x 61 nodes) but the length of the inlet channel was reduced in order to investigate the effects of a more linear velocity profile across the inflow jet. It was anticipated that the shorter length of the no-slip side walls in the feeder channel would prevent the build-up of a parabolic laminar velocity distribution across the inflow jet and would therefore better represent the Re_I=10.0 simulation depicted in Figure 3.5.* The steady-state velocity vectors (at t = 550 s) calculated using the semi-rigid-lid approach are shown in

* The lower Reynolds number Navier-Stokes simulations, presented in Chapter 3, employ a uniform velocity distribution across the inlet opening into the circular cylinder in order to model the boundary constraints stipulated by Dennis (1974) and Mills (1977).

Figure 7.18 whilst Figure 7.19 illustrates the transverse U-velocity profile across the mid-section of the reservoir. Figure 7.19 indicates that the shortened inlet boundary-fitted coordinate system reduces the U-velocity at the centre of the reservoir and therefore improves the velocity distribution in the throughflow jet. However, an inspection of the velocities in the zones of recirculation either side of the throughflow shows the boundary-fitted scheme underestimating the strength of the secondary flow compared with the Navier-Stokes simulation. A comparison of Figures 7.11 and 7.19 demonstrates that the amount of recirculation in the reservoir is crucially dependent upon the velocity profile in the shear layers of the inflow jet. As a corollary to this finding, it can be concluded that the numerical simulation of jet-forced flow in prototypical storage reservoirs should only be attempted when the finite-difference mesh has sufficient resolution to model the velocity distribution across the inflow jet.

The final validation exercise using Dennis' circular geometry employed a much finer boundary-fitted grid having a central core of 121 x 121 nodes (60 x 60 flow-variable cells); this represents a B.F.G. with twice the resolution of the previously investigated circular reservoir coordinate systems. Figure 7.20 presents the refined boundary-fitted system and shows that the reservoir utilises the original long inlet geometry. The convergence parameters for ending the numerical grid generation were set to $\epsilon_x = \epsilon_y = 10^{-7}$ in order to prevent the occurrence of long wavelength errors in the coordinate lines at the end of the iteration procedure. As a consequence, the numerical grid generation was rather costly, requiring approximately 40 minutes of C.P.U. time on a PRIME 9955 computer. Multigrid Poisson equation solvers (Häuser et al., 1986a) reduce the computational costs of generating curvilinear coordinate systems and therefore it is recommended that such techniques are incorporated in future versions of the mesh-generation code. However, without the introduction of a multigrid shallow water equation solver*, the possible savings achieved during mesh production are considered to be insignificant compared with the costs of the hydrodynamic simulation. This statement can be illustrated by considering the computational costs of using the present *single-grid* boundary-fitted shallow water equation scheme to solve the jet-forced flow for the mesh illustrated in Figure 7.20. Since the coordinate system has twice the resolution of earlier reservoir meshes, the time step must be reduced by a factor of 2. Furthermore, each iteration in the computational procedure requires approximately

* It is envisaged that future research will also be directed towards multigrid techniques for the discretisation of the hyperbolic shallow water equations.

four times the number of arithmetic operations. Consequently the numerical scheme utilises about eight times the amount of computer C.P.U. time in order to reach steady-state. Since the earlier (61 x 61 node) boundary-fitted reservoir meshes need approximately 5.5 hours of computer time in order for the flow to reach equilibrium, the present 121 x 121 node coordinate system requires about 44 hours of computer time. Therefore, the 40 minutes of C.P.U. time required to generate the mesh only represents about 1.5% of the overall cost of the numerical simulation!

The flow calculation employed a constant bed friction coefficient of 0.003, a time step of 0.025 s and an eddy viscosity of 0.000784 m^2/s (equivalent to an inlet Reynolds number of Re_I=10.0). Steady-state was attained after approximately 20,000 time steps (equivalent to 500 seconds of simulation time). Instead of depicting the reservoir circulation using the velocity vector distribution, the boundary-fitted mesh was sufficiently refined to enable contours of numerically integrated depth-averaged stream function to be plotted; this was accomplished using the computational techniques described in Section 6.11 of the previous chapter. In order to allow the stream function contours predicted by the primitive variable discretisation to be compared directly with the ψ-contours obtained from the Navier-Stokes simulation, it was necessary to scale the values of depth-averaged stream function obtained from the boundary-fitted approach. This was achieved by considering the ratio of the discharge through the circular reservoir, i.e.

$$Q_{Navier\text{-}Stokes} = \Delta\psi = 2.0 \quad \text{.. depth not considered in } (\psi,\omega) \text{ formulation.}$$

whilst

$$Q_{boundary\text{-}fitted} = U_{av(inlet)} \times B \times D$$

where B and D are the width and flow depth of the inlet channel. (7.9)

Thus,

$$Q_{boundary\text{-}fitted} = 0.100 \times 2 \times 0.75 \times \sin(\pi/30) \times 0.102$$
$$= 1.60 \times 10^{-3}\ m^3/s.$$

Figure 7.21a illustrates the scaled stream function contours at steady-state as determined by the boundary-fitted primitive variable scheme whilst Figure 7.21b presents an enlargement of the circulation pattern depicted in Figure 3.5a, computed using the (ψ,ω) formulation. Apart from the innermost stream function contour at the centre of each gyre, the contours depicted in Figure 7.21a have the same values as those plotted for the Navier-Stokes simulation. The refined mesh boundary-fitted

model predicts slightly stronger recirculation zones in comparison to the Navier-Stokes discretisation. This slight discrepancy is considered to be a function of the parabolic velocity profile across the inflow jet formed as the flow travels down the feeder channel. Comparison of the circulation patterns reveals that there is almost an exact agreement in the throughflow stream function contours. Moreover, the positions of reattachment of the separation streamlines (ψ = 0.0 and ψ = 2.0) are also very similar. It can therefore be concluded that the quadratic upwind differencing methodology introduced to discretise the curvilinear transformed shallow water equations causes very little additional numerical diffusion compared with the central difference advective scheme used to solve the vorticity-transport equation.

The final results presented in this sub-section depict the velocity distributions around the entrance and exit of the circular cylinder as computed by the 121 x 121 node boundary-fitted mesh. Figure 7.22a illustrates the flow as it emerges from the confines of the inlet channel and shows the formation of the divergent throughflow jet. The diagram also indicates that the velocities in the recirculation zones are extremely small compared with those in the main throughflow. Figure 7.22b depicts the bulbous flow pattern at the exit, caused by the throughflow jet having to accelerate in order to squeeze into the confined outlet channel. The velocity distribution immediately upstream of the exit opening closely resembles a radial sink-like potential flow. Downstream of the exit opening, the no-slip side walls of the outlet stem quickly re-establish a laminar boundary-layer velocity profile.

The validation exercises described in this section have demonstrated that the boundary-fitted shallow water equation solver is capable of predicting the complex non-linear hydrodynamic conditions which are associated with jet-forced reservoir flows. In particular, the excellent agreement between the boundary-fitted curvilinear (U,V,ζ) procedure and the orthogonal (ψ,ω) discretisation indicates that the numerical scheme proposed in Chapter 6 is a viable solution technique for the shallow water equations in irregular flow domains.

7.4.2 Mills' geometry

The boundary-fitted coordinate system utilised for the study of jet-forced flow in Mills' (1977) asymmetrical circular geometry has already been described in detail in Section 4.6. For the present validation purposes, results from the boundary-fitted shallow water equation program are compared with the Re_I=25.0 flow prediction obtained from the stream function/vorticity-transport discretisation, described in

Chapter 3. Since the higher Reynolds number Navier-Stokes simulations employed inlet and outlet channels to avoid the unrealistic entrance and exit velocity constraints specified by Mills, the boundary-fitted coordinate system was generated with flow passageways as shown in Figure 4.11. In the physical (x,y) plane, the inlet and outlet channels subtend an angle of $\pi/16$ radians and their centrelines are separated by $7\pi/8$, whereas in the computational (ξ,η) plane, the flow stems are diametrically opposite each other. The rectangular regions in the computational domain which represent the stems are displaced upwards from the centreline of the square core to prevent distortion of the η-lines as they emerge from the confines of the flow channels.

The fineness of the boundary-fitted grid was selected using experience gained during the previous validation tests with Dennis' symmetrical geometry. It was judged that the 61 x 61 node systems employed for the initial studies in Section 7.4.1 did not have sufficient resolution to produce detailed flow simulations, whereas the 121 x 121 node coordinate system utilised in the final validation test required an excessive amount of computer C.P.U. time to reach steady-state. Consequently, it was necessary to make a compromise, and so, the boundary-fitted coordinate system was generated with a central core of 93 x 93 nodes (46 x 46 flow-variable cells).

The circular basin was assumed to have a diameter of 1.5 m, a depth of approximately 0.1 m and an inflow velocity of 0.1 m/s. A constant eddy viscosity of 0.000292 m^2/s was employed across the whole flow domain, giving an inlet Reynolds number of Re_I=25.0. The time step was set to 0.075 s and the inflow velocity allowed to build up to the desired steady-state value of 0.1 m/s over a period of 250 s. For the purposes of the validation test, the bed friction coefficient was set to zero in order to simulate the planar flow conditions modelled in Chapter 3. It was found that the instabilities caused by reflections of the initial transient free surface wave in the vertical walls of the reservoir were more troublesome than those encountered during the simulation of jet-forced flow in Dennis' symmetrical geometry. This can partly be attributed to the fact that the outlet channel is not directly opposite the inlet feeder passageway, thereby causing a higher proportion of the initial free surface disturbance to be reflected back into central circular region of the reservoir. However, the main reason for the increased occurrence of destabilising amplified clapotis effects was thought to be caused by the lower value of the eddy viscosity coefficient. In the absence of bed friction, the only mechanism to stabilise the flow field is the combination of the diffusive influence of the effective stress terms acting in conjunction with the no-slip wall velocity boundary conditions. Lowering the value of eddy viscosity reduces the *coupling* between the interior of the

flow field and the no-slip side walls and hence increases the possibility of spurious surface elevation effects.

Numerical experimentation revealed that it was more cost effective to increase the damping of the temporal filter, described in Section 6.10, than to lengthen the build-up ramp used to specify the variation of inflow velocity. As a consequence, the flow simulations for Mills' geometry employed the following modified filter coefficients:

$$a = 0.4 \quad , \quad b = 0.5 \quad \text{and} \quad c = 0.1 \quad . \qquad (7.10)$$

Comparison with the standard parameters of (a, b, c) = (0.6, 0.3, 0.1), recommended by Butler (1978a), indicates that the filter defined by the coefficients in equation (7.10) will offer increased damping in the initial stages of the hydrodynamic simulation. However, as previously explained in Section 6.10, the effects of the recursive digital filtering technique will always tend to zero as the hydrodynamic regime approaches equilibrium. Therefore, the increased numerical damping of the filter should have no effect upon the final steady-state flow predictions.

Figure 7.23a illustrates the velocity vectors at t = 1275 s (i.e. after 17000 time steps). The boundary-fitted numerical scheme utilised approximately 3.7 seconds of C.P.U. time per iteration on a PRIME 9955 computer and therefore required a total of about 1100 minutes C.P.U. time for the flow calculation to reach equilibrium. A visual comparison of the velocity fields presented in Figures 3.7c and 7.23a once again demonstrates that the Navier-Stokes (ψ,ω) formulation and the primitive variable boundary-fitted numerical scheme produce very similar flow predictions. This can be confirmed by comparing contours of numerically integrated depth-averaged stream function, depicted in Figure 7.23b, with the ψ-contours from the Navier-Stokes simulation (shown in Figure 3.7a). Again it was necessary to scale the values of stream function computed using the boundary-fitted method in order for the contours from the two numerical approaches to be directly equivalent. The circulation patterns in Figures 3.7a and 7.23b reveal that there is almost an exact agreement in the stream function contours for the throughflow jet and upper gyre structure. However, the boundary-fitted numerical scheme underpredicts the strength of the lower recirculation zone; i.e., the $\psi = -0.8$ contour is completely absent from the flow depicted in Figure 7.23b and the $\psi = -0.6$ and $\psi = -0.4$ contours enclose considerably smaller areas than the equivalent lines presented in Figure 3.7a. Although the two models show discrepancies in the positions of the stream function

contours for the lower recirculation zone, it must be remembered that ψ is almost constant in the central region of the gyre and thus slight discrepancies in the predicted values of stream function can lead to large changes in the position of the contours.

Figure 7.24 shows the non-dimensionalised velocity distribution along the axis of symmetry of the reservoir (i.e., along the line $\theta = + 9\pi/16$) and employs the same characteristic length and velocity scales as used in Section 7.4.1. The only change in the method of presentation between Figure 7.24 and the earlier non-dimensionalised velocity profiles, utilised for Dennis' geometry, lies in the introduction of primes in the velocity and coordinate variables, U' and y'. These variables are defined with respect to a second Cartesian coordinate system, (x',y') inclined at $\theta = + \pi/16$ radians to the primary (x,y) coordinates. In a similar manner to the earlier graphical velocity comparisons, the solid curved line in Figure 7.24 shows the velocity profile calculated using the stream function/vorticity-transport discretisation, whereas the crosses indicate the distribution computed by the boundary-fitted primitive variable scheme. It can be seen that the discrepancies between the predicted velocities in the lower recirculation zone are in fact relatively small compared with the errors in the central region of the throughflow jet. Figure 7.24 therefore demonstrates that the large disparity in the positions of the stream function contours in the lower gyre are not as serious as they might appear. It is interesting to note that the ψ-contours from the two numerical models appear to be more accurate in the central throughflow zone, yet the profiles presented in Figure 7.24 indicate that the longitudinal jet is actually the region of largest velocity error. This effect is closely related to the earlier discussion concerning the inaccuracy of stream function contours in the recirculation zones; the main throughflow will be characterised by large transverse stream function gradients and therefore even fairly large errors in ψ will lead to relatively small changes in the positions of the stream function contours.

Comparison of Figure 7.24 with the velocity profiles from the earlier Re_I=10.0 flow computations confirms that the present higher Reynolds number simulation produces a much faster and narrower throughflow jet. For example, at the centre of the reservoir, the maximum throughflow velocity presented in Section 7.4.1 was approximately 50% of the inflow velocity. In the present flow simulation (Re_I=25.0), the maximum throughflow velocity at the mid-section of the reservoir is approximately 90% of the the inflow velocity. Another interesting feature of the transverse U'-velocity profile shown in Figure 7.24 is the significant change in velocity gradient in the *lower* shear layer of the throughflow jet (i.e., at

approximately $y'/R_0 = -0.2$). This phenomenon is exhibited by both the Navier-Stokes discretisation and the primitive variable boundary-fitted scheme, and is therefore unlikely to be caused by errors in the numerical differencing procedures. The author is unable to offer an exact physical explanation for this feature, but it is probably the remnant of the discontinuity in velocity profile created at the lower inflow convex corner point.

Although the velocity vectors presented in Figure 7.23a give a reasonable indication of the overall circulation pattern in the reservoir, the distribution of flow arrows near the entrance and exit of the circular cylinder is rather ambiguous. Therefore, it is convenient to replot the velocity patterns in the vicinity of the inflow and outflow stems using a larger scale. Figure 7.25a depicts the flow emerging from the inlet channel and demonstrates that the jet has less of a tendency to diverge immediately after entering the reservoir in comparison to the lower Reynolds number simulations of Section 7.4.1 (cf. Figure 7.22a). The velocity distribution in the vicinity of the outlet passageway, shown in Figure 7.25b, illustrates the characteristic sink-like bulbous flow upstream of the exit opening and also provides a clearer representation of the stagnation points at the centres of the recirculation zones.

The validation test has once again proved that the boundary-fitted shallow water equation solver is capable of simulating non-linear jet-forced flow regimes. Furthermore, the strong correlation between the velocity profiles depicted in Figure 7.24, implies that the second order upwind differencing technique utilised in the discretisation of the cross-advective terms of the transformed shallow water equations introduces very little additional numerical diffusion above the level associated with a standard central difference advective scheme.

7.4.3 Falconer's geometry

This section presents results from a simulation of jet-forced flow in a flat bottomed circular reservoir representative of the small-scale laboratory investigation and coarse Cartesian numerical model studied by Falconer (1976, 1980). Using Falconer's experimental geometry, the reservoir diameter was specified as 1.5 m, and a narrow inlet, of width 0.08 m, was positioned diametrically opposite a 0.24 m wide outlet channel. Further details concerning the generation of the boundary-fitted coordinate system may be found in Section 4.6. Numerical experimentation indicated that the layout of the coordinate lines in the physical mesh could be improved by

increasing the proportion of ξ-lines in the central region of the reservoir. Therefore the boundary-fitted coordinate system was generated with a *rectangular* core of 105 x 97 nodes (52 x 48 flow-variable cells). Figure 4.12a illustrates the transformed (ξ,η) domain, whilst Figure 4.12b presents the corresponding physical (x,y) non-orthogonal grid system. In a similar manner to previous reservoir circulation studies, the water depth was assumed to be approximately 0.1 m and the inflow velocity was again specified as 0.1 m/s.

Following Falconer (1976), the eddy viscosity coefficient was determined from the algebraic depth-averaged turbulent dispersive model proposed by Elder (1959):

$$K_l = \frac{5.9\ D\ \sqrt{g}\ (U^2 + V^2)^{\frac{1}{2}}}{C} \approx \tilde{\nu}_t \quad . \qquad (5.43\ bis)$$

The above formula assumes that the depth-averaged longitudinal coefficient of diffusivity, K_l and the depth-averaged eddy viscosity, $\tilde{\nu}_t$ have the same order of magnitude. Kuipers and Vreugdenhil (1973) and Vreugdenhil (1973) point out that this is only an approximate relationship since momentum can be transferred by surface elevation gradients, whereas there is no comparable mechanism for the transport of mass in the dispersion equations*. Nevertheless, the algebraic relationship presented above should provide a better estimate of momentum diffusion than the assumption of a constant eddy viscosity across the entire flow domain.

The initial validation test consisted of simulating jet-forced circulation for a Chezy roughness coefficient of 438 $m^{\frac{1}{2}}/s$ (as quoted by Falconer, 1980). A time step of 0.06 s was utilised, and the inflow velocity was allowed to build up to the desired steady-state value of 0.1 m/s over a ramp of 300 s. The temporal filter coefficients used to stabilise the numerical scheme were chosen to be the same as those employed for the Mills' geometry simulation, described in Section 7.4.2 (i.e., a = 0.4, b = 0.5 and c = 0.1). Figure 7.26 presents the steady-state velocity vectors at t = 1920 s (after 32000 time steps) obtained using the semi-rigid-lid procedure. The boundary-fitted scheme utilised approximately 5.27 seconds of C.P.U. time per iteration on a PRIME 9955 computer and therefore required a total of 2800 minutes C.P.U. time to reach equilibrium. The attainment of steady-state for the surface elevation field required an excessive number of iterations in comparison with the previous reservoir studies of Sections 7.4.1 and 7.4.2. This can be attributed to the

* Elder (1959) calculated the coefficient of proportionality in equation (5.43) by measuring the dispersion of dye in open channel flow conditions.

fact that the spatially varying eddy viscosity coefficient in most parts of the flow field is considerably smaller than the constant values utilised in earlier studies. Consequently, the seiching motion formed as the flow is built up is subjected to less damping and therefore the transient free surface waves persist for a greater period of time.

Figure 7.27 reproduces the experimental and computational reservoir circulation patterns presented by Falconer (1980); the experimentally measured velocity field is depicted in Figure 7.27a whilst Figure 7.27b illustrates the results from Falconer's coarse 19 x 19 flow cell Cartesian numerical model. Reasonable agreement may be noted between Figure 7.26 and Falconer's results, thereby confirming the validity of the boundary-fitted shallow water equation solver. However, it is evident from Figure 7.26 that the non-orthogonal scheme predicts two pairs of counter-rotating gyres either side of the fast throughflow, rather than the single pair observed by Falconer. This discrepancy is consistent with the fact that the refined mesh boundary-fitted scheme has a smaller amount of artificial numerical diffusion compared with the coarse Cartesian grid depicted in Figure 7.27b. The velocity vectors in the lower secondary eddy are detailed in Figure 7.28 which illustrates the complex flow structure in the zone of interaction between the counter-rotating gyres.

The flow arrows presented in Figure 7.26 may be difficult to interpret correctly because the velocity vectors are positioned at grid points on the curvilinear mesh. In fact, the flow features may be better represented by plotting interpolated contours of numerically integrated depth-averaged stream function as shown in Figure 7.29. Here, the streamlines indicate the slight radial divergence of the inflow jet close to the inlet, the more or less direct throughflow between inlet and outlet and the bulbous flow pattern near the outlet as the throughflow squeezes into the exit passageway in a sink-like manner. Two primary eddies, of opposite sense of rotation, are located either side of the throughflow jet close to the outlet. In addition, two zones of separation containing secondary gyres may be discerned near the inlet. It is possible that there is a further pair of eddies in the vicinity of the reservoir walls adjacent to the secondary gyres, but the numerical grid is not sufficiently refined to verify this.

Figure 7.30 presents the steady-state surface elevation contours calculated using the boundary-fitted approach, and illustrates that the flow depth inside the reservoir is almost uniform. As would be expected, surface elevation minima occur at the centres of the primary eddies. Although not contoured, the flow in the narrow inlet stem is ramped with a rapid, but almost linear change in surface elevation with

distance. As previously explained in Section 7.4.1, this phenomenon can be attributed to the fact that the finite-difference mesh has insufficient resolution in the inlet channel to model the large velocity gradients created at the no-slip walls. Figure 7.31 depicts the longitudinal surface elevation profile along the reservoir's axis of symmetry (i.e., y = 0) and illustrates the almost uniform flow depth within the circular basin. Also evident from Figure 7.31 is the absence of a steep ζ profile in the outlet passageway ($x/R_0>1.0$), demonstrating that the greater number of flow cells across the outlet stem is more able to resolve the velocity gradients imposed by the wall boundary conditions. The noticeable increase in surface elevation from $x/R_0=+0.25$ to $x/R_0=+1.0$ corresponds to the deceleration of the throughflow stream as it approaches the wide outlet, and is in accordance with Bernoulli's equation. Figure 7.32 shows the transverse surface elevation profile across the mid-section of the reservoir (i.e., along the line x = 0) and illustrates that the centres of the primary gyres correspond to surface elevation minima.

Figure 7.33a depicts the flow emerging from the inlet channel and clearly shows the stagnation points at the centres of the secondary gyres together with the characteristic *lobes* created when the primary recirculating flow stream divides into two separate directions. The velocity distribution near the outlet, presented in Figure 7.33b, provides an excellent visualisation of the flow in the primary eddies. Furthermore, it is evident that the bulbous flow pattern at the entrance to the outlet channel is less pronounced than in earlier reservoir circulation studies. This is due to the fact that the wide outlet stem does not require such a rapid acceleration of the throughflow jet in order for the flow to enter the outlet. Figure 7.33b also demonstrates that the velocity profile in the outlet passageway is considerably more uniform than the velocity distributions computed in the earlier narrow outlet reservoir geometries. Consequently, it may be concluded that this flow simulation is probably more representative of turbulent open channel flow. The increased resolution of the finite-difference mesh in the outflow channel is the main reason for the enhanced realism of the flow computation, but another significant factor is the determination of the eddy viscosity coefficient from the magnitude of the local velocity vector. This helps prevent the occurrence of the characteristic parabolic laminar velocity distribution found in earlier constant eddy viscosity simulations.

The flow geometry described in this section has allowed predictions from the boundary-fitted shallow water equation solver to be compared directly with results from a totally independent numerical model and data from a small-scale physical model. The similarity in the circulation patterns depicted in Figures 7.26 and 7.27 demonstrates that the primitive variable boundary-fitted scheme is capable of

producing acceptable flow solutions in jet-forced hydrodynamic regimes, thus concluding the series of validation tests. However, before leaving Falconer's reservoir geometry, it is worthwhile considering the results from two further flow computations. Figure 7.34 presents the circulation pattern resulting from a Chezy roughness coefficient of 438 $m^{\frac{1}{2}}/s$ and a *constant* eddy viscosity of $\tilde{\nu}_t$ = 0.000292 m^2/s (as used in Section 7.4.2). The flow is characterised by a single pair of recirculating eddies rather than the double pair of gyres found in the previous simulation. This confirms that the formation of the weaker secondary gyres depends upon the magnitude of the eddy viscosity coefficient and is not a feature of the narrow inlet geometry.

The final flow simulation considered the effect of the Chezy friction coefficient on the level of momentum diffusion within the reservoir. Although Falconer (1980) clearly stated that he used a Chezy value of 438 S.I. units, the present author considers that a value of 43.8 $m^{\frac{1}{2}}/s$ would be more appropriate to a timber-lined laboratory-sized reservoir. For example, equating the Chezy and Manning friction expressions shown in equations (5.82) and (5.83) leads to

$$C = \frac{D^{1/6}}{n} \qquad (7.11)$$

where C is the Chezy coefficient,
n is the Manning roughness factor
and D is the local water depth.

In the present situation, D $\approx$ 0.1 m and n $\approx$ 0.015 $s/m^{1/3}$, and therefore the Chezy roughness should be approximately 45.4 $m^{\frac{1}{2}}/s$.

Figure 7.35 presents the flow obtained using the Elder depth-averaged eddy viscosity relationship in conjunction with a Chezy roughness coefficient of 43.8 $m^{\frac{1}{2}}/s$ (i.e. ten times smaller than that employed by Falconer). The reservoir circulation depicted in Figure 7.35 is markedly different to Falconer's experimental flow measurements, presented in Figure 7.27a. In the present numerical simulation, the throughflow jet diverges rapidly causing the recirculation zones to contract towards the inlet. Furthermore, the velocities in the eddies are smaller and a bulbous flow pattern, characteristic of low Reynolds number regimes, is clearly visible at the exit. In fact, the overall appearance of the circulation pattern closely resembles the Re_I=10.0 simulation presented in Section 7.4.1. The flow computation therefore demonstrates that the depth-averaged eddy viscosity expression shown in equation

(5.43) only provides a very approximate estimate of the spatial variation of $\tilde{\nu}_t$. This may explain why Falconer chose to define the Chezy number as 438 $m^{\frac{1}{2}}/s$ rather than use the theoretical value.

7.5 Arbitrary river geometry

So far, the boundary-fitted model has considered geometries and conditions which are directly comparable with existing flow data. The various validation exercises have provided a rigorous test for the numerical algorithms proposed in Chapter 6 and the excellent agreement between the boundary-fitted flow predictions and the test data have confirmed that the shallow water equation solver has been correctly coded. This final section demonstrates the versatility of the curvilinear coordinate system by examining the flow in an arbitrary-shaped watercourse with a highly contorted perimeter. Contrary to earlier simulations, the results discussed in this section are not substantiated against alternative data and therefore the flows must be regarded as *predictions* rather than as validation tests.

Figure 7.36 illustrates the 121 x 31 node (60 x 15 flow cell) boundary-fitted coordinate system utilised for the present investigation. In order to demonstrate that the boundary-fitted shallow water equation solver is capable of predicting *large-scale* jet-forced flow phenomena, typically encountered in real-life studies, the physical mesh represents a 1.5 km long section of river. The (x,y) coordinates of the boundary nodes were generated using the curve-wise cubic-spline technique proposed in Section 4.4.3, whilst two-point weighted averaging in the η-direction was employed in the production of the coordinates for the initial physical mesh. Numerical experimentation demonstrated that it was beneficial to generate the boundary-fitted system using the control-functions, P and Q, described in Section 4.3.3. Without the use of attraction-operators, the elliptic grid generation equations tended to produce excessively large grid cells in regions of strong concave boundary curvature (e.g., bays) and very small cells in the vicinity of strong convex boundary curvature (e.g., headlands). Table 7.1 summarises the attraction-operators used for the physical mesh presented in Figure 7.36b.

The irregular bathymetry employed in the study is illustrated in Figure 7.37, which shows the bed contours in metres below the horizontal datum. It can be seen that the depths vary from 0.5 m at the no-slip perimeters to a maximum of approximately 3.0 m in the centre of the flow domain. The velocity of the narrow inflow jet was set to 0.5 m/s, the Chezy roughness coefficient was assumed to be 45

$m^{\frac{1}{2}}/s$ across the entire flow field and the Coriolis parameter was specified as f = + 1.19 x 10^{-4}, corresponding to a latitude of 55° N. A time step of 1.0 s was utilised and the inflow velocity was allowed to build up to the stipulated steady-state value over a period of 1000 s. Despite the highly contorted shape of the perimeter, it was found that the original temporal filter coefficients of (a, b, c) = (0.6, 0.3, 0.1) were able to stabilise the scheme. Furthermore, numerical experimentation revealed that a ramped inflow velocity variation was unnecessary and that an impulsive start could also be employed. This can be attributed to the fact that the initial transient free surface wave is less likely to be reflected by the side walls * and therefore a higher proportion of the wave will travel down the watercourse and exit the flow domain via the *non-reflective* velocity boundary conditions defined by equation (6.40).

In a similar manner to the jet-forced reservoir circulation studies of Section 7.4, the magnitude of the eddy viscosity coefficient, $\tilde{\nu}_t$, utilised in the numerical model is the critical parameter in determining the resultant circulation patterns. As well as controlling the size and strength of secondary flow phenomena, the value of eddy viscosity affects the transfer of momentum between the main river channel and the shallower flow areas near the banks, and therefore influences the transverse velocity profile across the river. Vreugdenhil and Wijbenga (1982), in a numerical study of the flow patterns in a fast flowing river with a wide flood plain, considered eddy viscosities in the range 5.0 m^2/s to 1.0 m^2/s, and calibrated their model by comparing the computed longitudinal surface elevation profile against experimental field data. On the other hand, Ogink (1985) attempted to estimate the magnitude of $\tilde{\nu}_t$ by considering the increase in flow depth due to the horizontal momentum transfer from the river channel to the flood plain. Having estimated the increase in normal flow depth from an empirical momentum transfer analysis, Ogink then ran a two-dimensional numerical model with different values of eddy viscosity until the predicted flow depths were in agreement. This procedure was applied to parts of the River Rhine, and it was found that during flood conditions, the eddy viscosity coefficient was approximately 0.5 m^2/s. In the present simulation, the flow velocities are an order of magnitude smaller than those considered by Vreugdenhil and

* The reduction in depth near the perimeter walls results in an increase in flow velocities associated with the transient free surface phenomena. This in turn leads to an increase in the frictional damping of the seiching motion, and consequently the reflected free surface wave usually has a smaller amplitude and smoother profile than the incident wave.

Wijbenga (1982) and therefore the eddy viscosity is expected to range between 0.5 m^2/s and 0.1 m^2/s. Three separate flow simulations were performed, with constant eddy viscosity coefficients of 1.0, 0.5 and 0.25 m^2/s.

Figure 7.38 illustrates the predicted steady-state velocity distributions (at t = 30,000 s) across the entire flow domain, whilst Figure 7.39 details the circulation patterns in the vicinity of the inflow jet. It is evident that the magnitude of the eddy viscosity coefficient has a dramatic effect upon the level of recirculation in the river. For example, at an eddy viscosity of 1.0 m^2/s, Figure 7.39a indicates that the lower recirculation zone is very weak and only extends about 200 m downstream of the separation point. However, with an eddy viscosity of 0.25 m^2/s (Figure 7.39c), the lower gyre is much more prominent and stretches approximately 500 m from the separation point. The increased size of the recirculating eddies is closely related to the reduced divergence of the fast inflow jet.

A further flow simulation using an eddy viscosity of 0.1 m^2/s was attempted, but it was found that the numerical scheme suffered from severe surface elevation instabilities, even when the coefficients of the digital filter had been altered to (a, b, c) = (0.2, 0.5, 0.3). Furthermore, increasing the length of the ramp, used to define the variation of inflow velocity with time, had no effect. It was thought that the instabilities were due to excessive cell Reynolds numbers in the inflow jet, and consequently the only method to investigate flows with lower values of eddy viscosity is by increasing the resolution of the finite-difference mesh.

In order to demonstrate that the lower limit of the eddy viscosity coefficient is not too severe compared with other numerical schemes, it is worthwhile considering the cell Reynolds number (Peclet number) at the centre of the inflow boundary. In this case,

$$U = 0.5 \text{ m/s} \quad , \quad \Delta x \approx 20 \text{ m}$$

and, therefore, the longitudinal cell Reynolds number for the stable ($\tilde{\nu}_t = 0.25$ m^2/s) simulation is

$$R_c = \frac{U \Delta x}{\tilde{\nu}_t} \approx \frac{0.5 \times 20}{0.25} = 40 \quad . \qquad (7.12)$$

Similarly, the unstable ($\tilde{\nu}_t = 0.1$ m^2/s) computation is associated with a Peclet number of approximately 100. The upper limit of $R_c \approx 40$ in the present scheme compares favourably with other numerical discretisations using curvilinear grids (e.g.

Officier et al., 1986; Bernard, 1989a).

The results described in this section have demonstrated the versatility of the boundary-fitted method by indicating that the non-orthogonal shallow water equation solver is capable of predicting large-scale flow phenomena in highly contorted geometries. It should be noted that the simulations are more stable than the previous jet-forced flow computations in small-scale circular reservoirs, despite the irregular nature of the finite-difference mesh, and so, the boundary-fitted technique appears to be ideally suited to the type of flow problem encountered in the management of river basins.

7.6 Conclusions

This chapter has described the various validation exercises which were used to test the numerical algorithms employed in the discretisation of the non-orthogonal curvilinear shallow water equations. The validation tests were designed to verify comprehensively the component parts of the numerical procedure. For example, the modelling of wind-induced surface elevation phenomena in circular and elliptical wedge-shaped basins provided an effective test for the geometrical transformations utilised in the curvilinear mapping technique. At steady-state, results from the numerical scheme are completely in accordance with an analysis of the force balance, and therefore it is concluded that the linear terms of the transformed momentum equations have been correctly discretised. The second series of validation tests simulated steady uniform flow in a prismatic open channel. As well as checking the numerical scheme in a dynamic flow situation, these particular tests proved that the boundary-fitted computational procedures are capable of solving the hydrodynamic equations on fairly distorted physical meshes. The third, and major, results section, demonstrates the use of the boundary-fitted concept for predicting momentum-driven circulation patterns in small-scale flat-bottomed circular reservoirs; this provided a rigorous test for the discretisation of the non-linear advective acceleration and diffusive effective stress terms. Benchmark data from alternative numerical studies and experimental flow investigations indicate that the second order upwind differencing employed in the discretisation of the cross-advective terms of the transformed depth-averaged hydrodynamic equations introduces very little additional artificial diffusion into the scheme. The combination of the reservoir circular perimeter wall and the parallel sided flow channels serves as a good example of how the boundary-fitted technique is able to produce a grid system with coordinate lines coincident with all boundaries of the flow domain. This is also demonstrated in

Section 7.5, which illustrates the use of the boundary-fitted methodology for a highly contorted flow geometry, typical of a real-life flow domain.

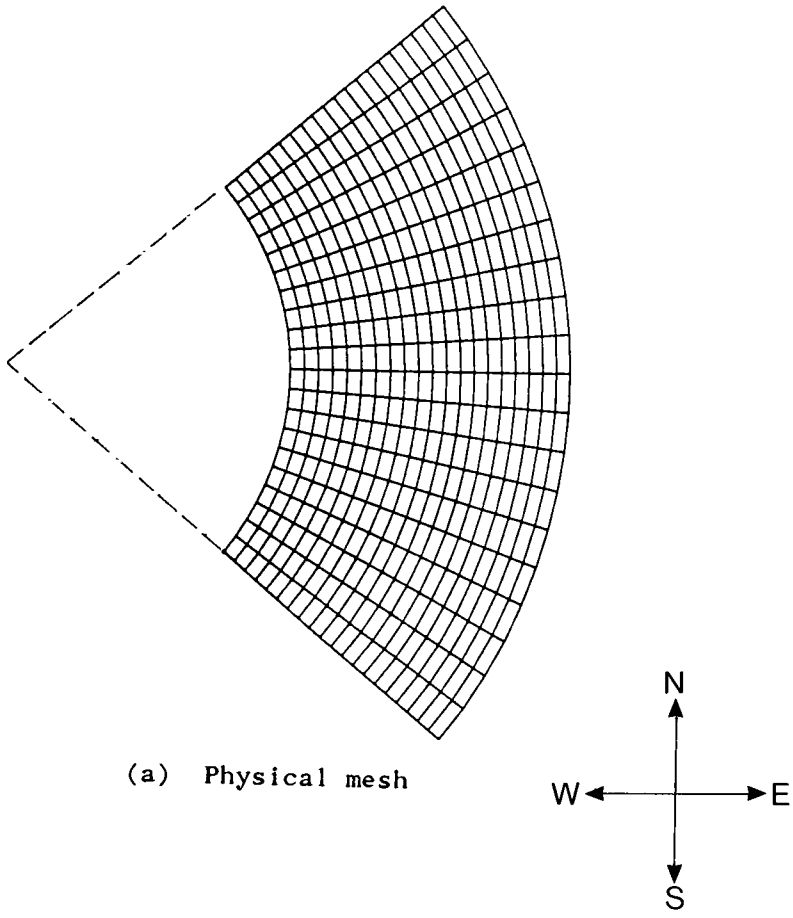
The chapter has demonstrated that the non-orthogonal shallow water equation solver, proposed in Chapter 6, is a viable technique for simulating nearly-horizontal flows. Provided the user is prepared to accept the increased complexity of the transformed hydrodynamic equations, the boundary-fitted approach offers the considerable advantage over conventional Cartesian numerical schemes of being able to simulate flow conditions in irregular geometries. The proposed non-orthogonal scheme uses the *advective* form of the momentum expressions and employs the non-conservative geometrical derivative relationships presented by Thompson et al. (1974). Consequently, the present scheme represents the most basic form of discretisation in the hierarchy of boundary-fitted models. Other more complex schemes would involve using the conservative momentum and geometric derivative equations, the use of contravariant or covariant velocities as primitive variables (in place of Cartesian components) and the use of a better discretisation procedure such as the finite-volume or multigrid technique. There are a large number of possible variations in the use of boundary-fitted methods for solving the shallow water equations, and it is envisaged that it will be some time before a *definitive* solution procedure (equivalent to Leendertse's (1967) Cartesian A.D.I. scheme for rectangular meshes) is available for non-orthogonal curvilinear systems. Nevertheless, the results presented in this chapter indicate that even the most basic form of boundary-fitted discretisation represents a considerable advance over the more conventional Cartesian finite-difference shallow water equation solvers.

Finally, the jet-forced flow simulations in Sections 7.4 and 7.5 have demonstrated that the resultant circulation is influenced by the magnitude of the eddy viscosity coefficient. This is borne out by Figure 7.39, for the arbitrary river geometry, which illustrates that a change in the global eddy viscosity from 1.0 m^2/s to 0.25 m^2/s produces a dramatic increase in the amount of secondary flow. Therefore, it is vital that the computational flow results are compared and calibrated against experimental field data (i.e., hindcasting), before the numerical model is used to *predict* flows for future unknown conditions such as higher discharge capacities, dredging operations or the construction of engineering works. In view of the difficulties and theoretical objections in assigning a constant value for the eddy viscosity coefficient in complex flow situations, the boundary-fitted scheme is extended, in the next chapter, to include a depth-averaged $k-\epsilon$ model to predict the spatial distribution of turbulence. However, for reasons which are explained in Chapter 8, the discretisation of the depth-averaged $k-\epsilon$ equations was found to be

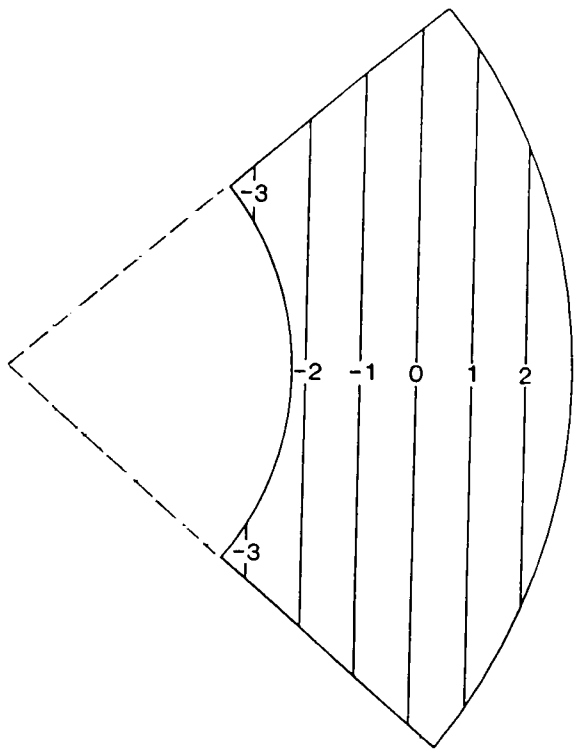
unsatisfactory and therefore the boundary-fitted shallow water equation solver currently operates using either the assumption of a constant eddy viscosity or the depth-averaged algebraic relationship presented in equation (5.43).

Table 7.1 Attraction-operators employed in the generation of the arbitrary river geometry coordinate mesh

Type of attraction	Coordinates of attraction point (ξ_i, η_i)	Amplitude	Decay factor
η-line	(97, 1)	0.00075	0.1
η-line	(97,31)	0.00075	0.1
η-line	(78,31)	0.00075	0.1
η-line	(33,31)	0.00100	0.1

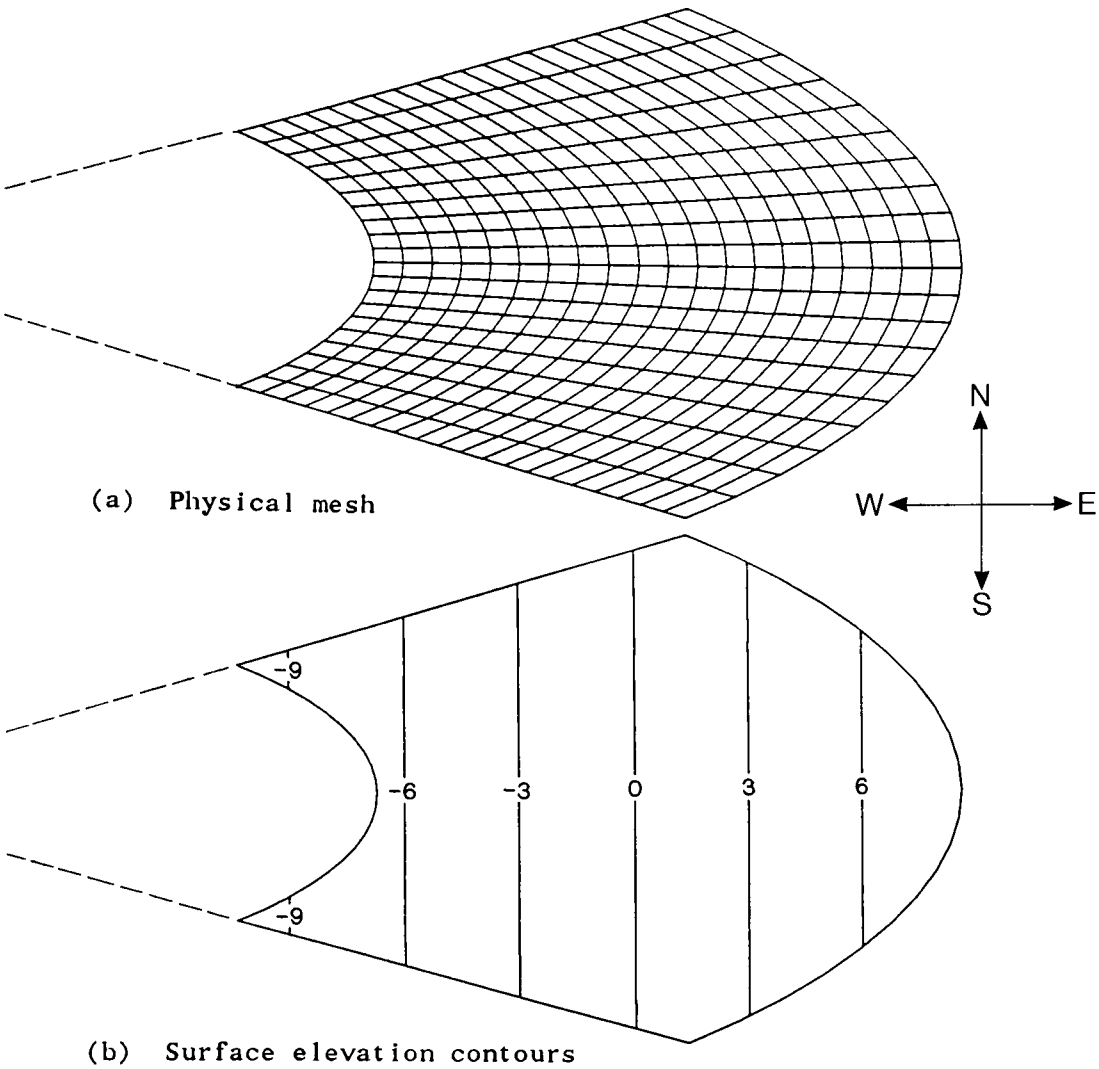


(a) Physical mesh



(b) Surface elevation contours
(mm above undisturbed level)

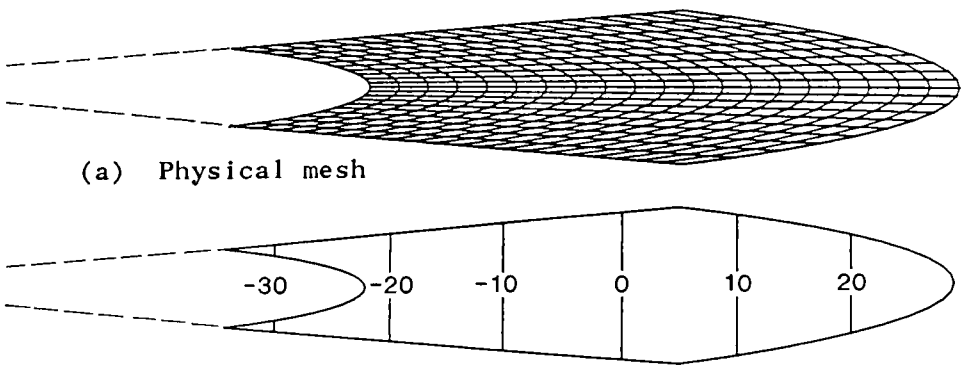
Figure 7.1 Steady-state wind-induced surface elevation effects in a circular wedge-shaped basin



(a) Physical mesh

(b) Surface elevation contours
(mm above undisturbed level)

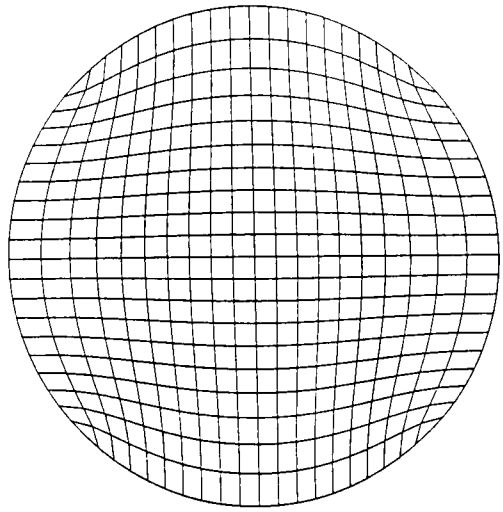
Figure 7.2 Steady-state wind-induced surface elevation effects in an elliptical wedge-shaped basin of length 15 km



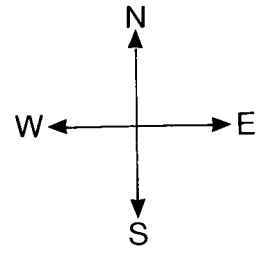
(a) Physical mesh

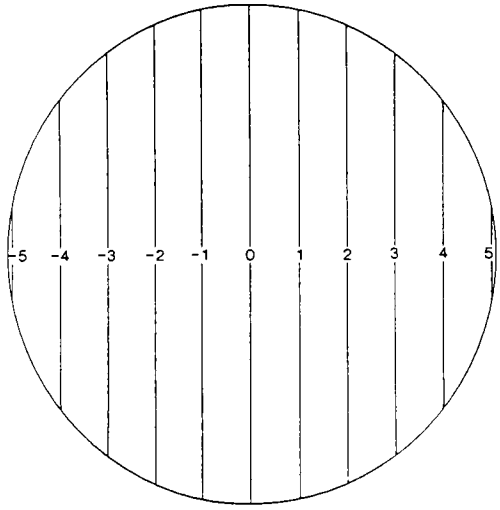
(b) Surface elevation contours
(mm above undisturbed level)

Figure 7.3 Steady-state wind-induced surface elevation effects in an elliptical wedge-shaped basin of length 50 km

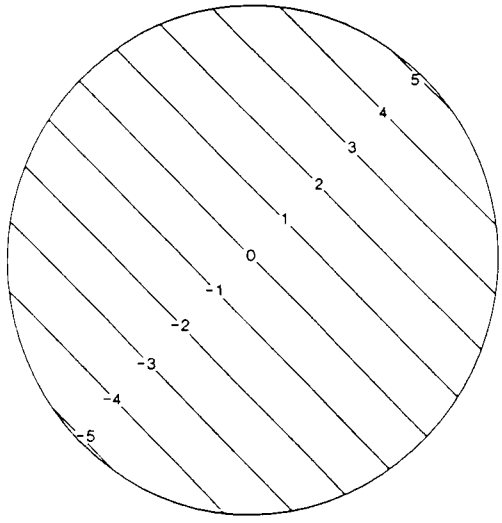


(a) Physical mesh



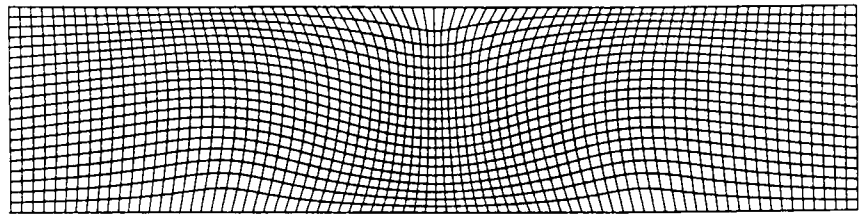


(b) Surface elevation contours for a wind-stress of 0.1 N/m^2 in an eastward direction (mm above undisturbed level)

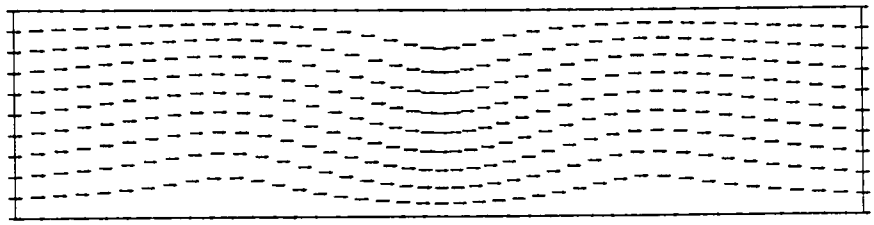


(c) Surface elevation contours for a wind-stress of 0.1 N/m^2 in a north-eastward direction (mm above undisturbed level)

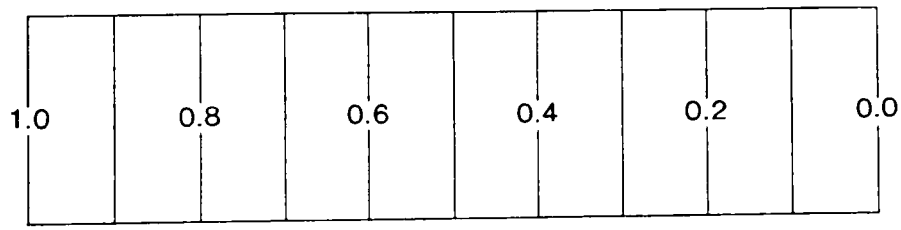
Figure 7.4 Steady-state wind-induced surface elevation effects in a 10 km diameter circular basin



(a) Physical mesh

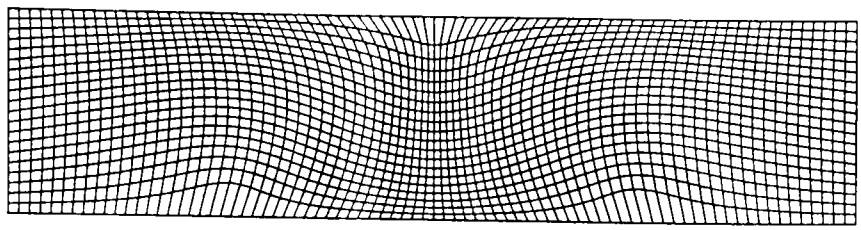


(b) Velocity vectors at steady-state

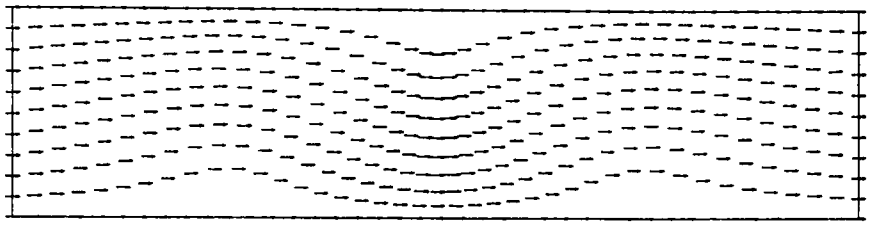


(c) Surface elevation contours at steady-state
(metres above downstream boundary)

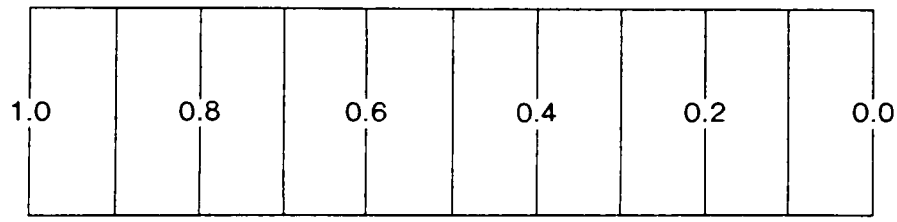
Figure 7.5 Uniform flow in a rectangular channel
(after Wijbenga, 1985a)



(a) Physical mesh

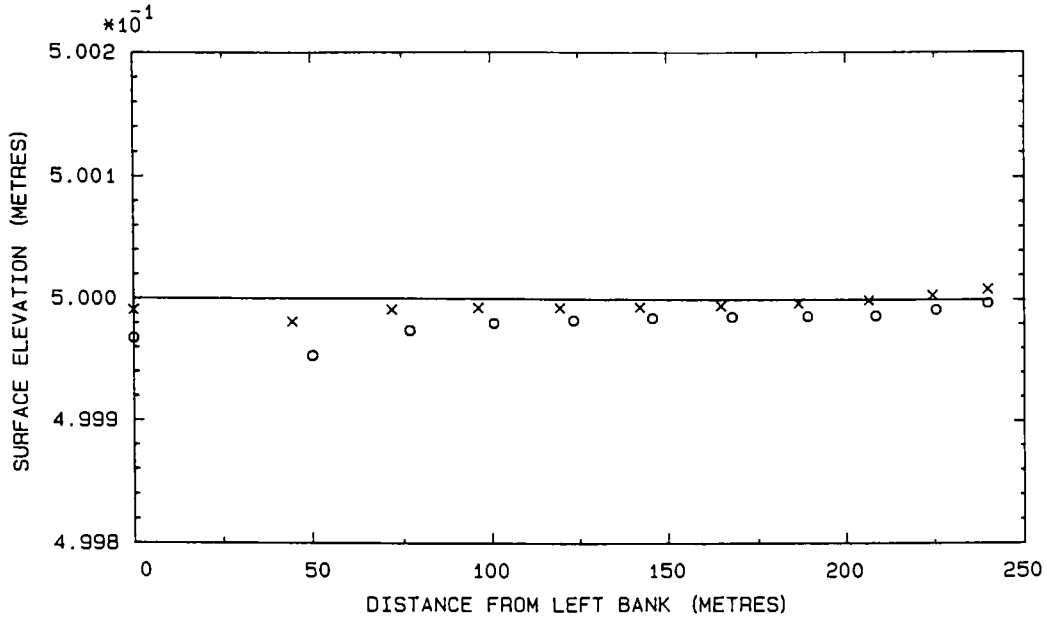


(b) Velocity vectors at steady-state



(c) Surface elevation contours at steady-state
(metres above downstream boundary)

Figure 7.6 Uniform flow in a rectangular channel for a more distorted non-orthogonal mesh



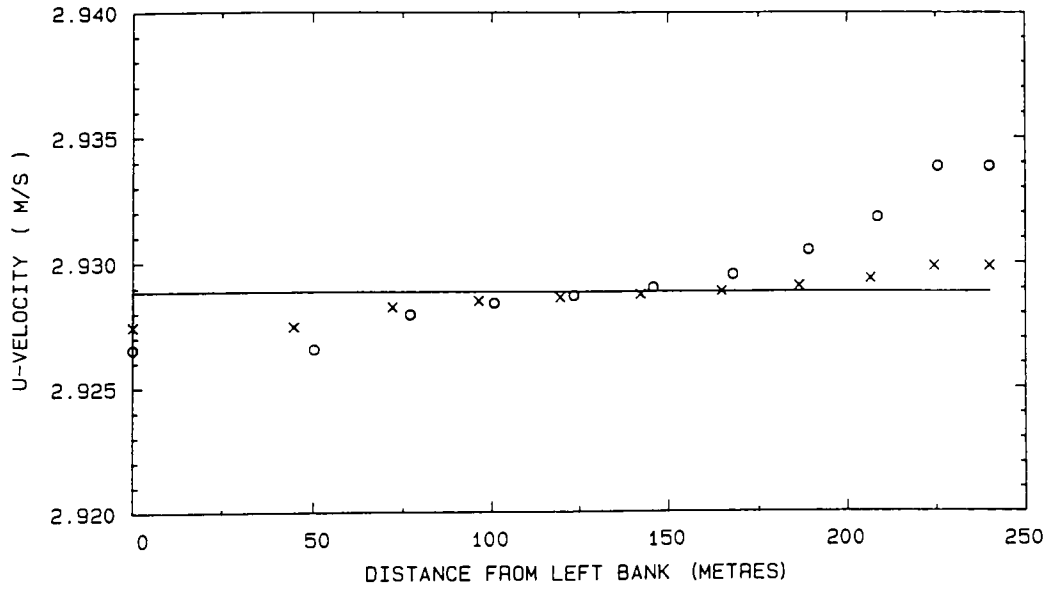
KEY

—— theoretical value (0.500 m)

× distribution calculated using B.F.G. illustrated in Figure 7.5a

o distribution calculated using B.F.G. illustrated in Figure 7.6a

Figure 7.7 Transverse surface elevation profile at section halfway along channel



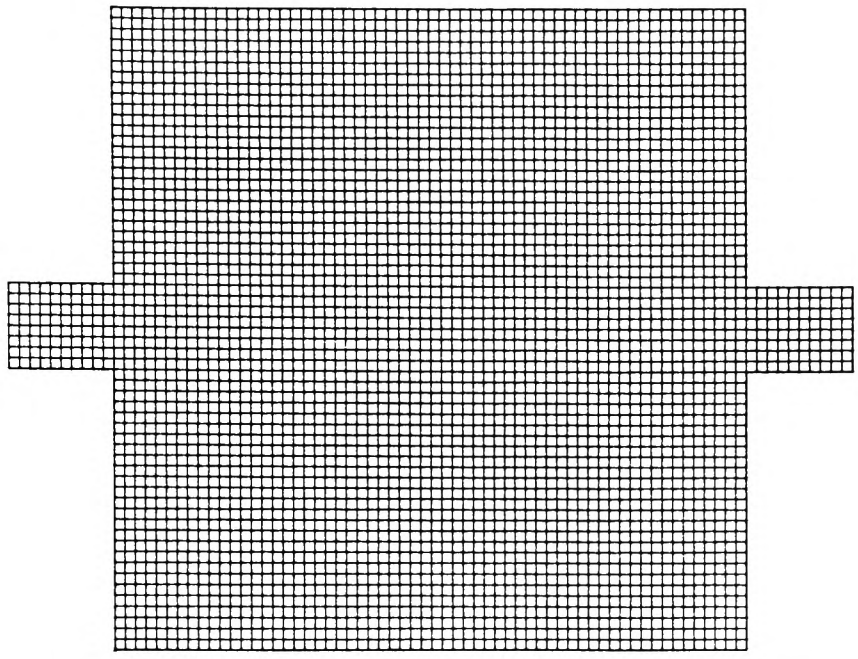
KEY

—— theoretical value (2.9288 m/s)

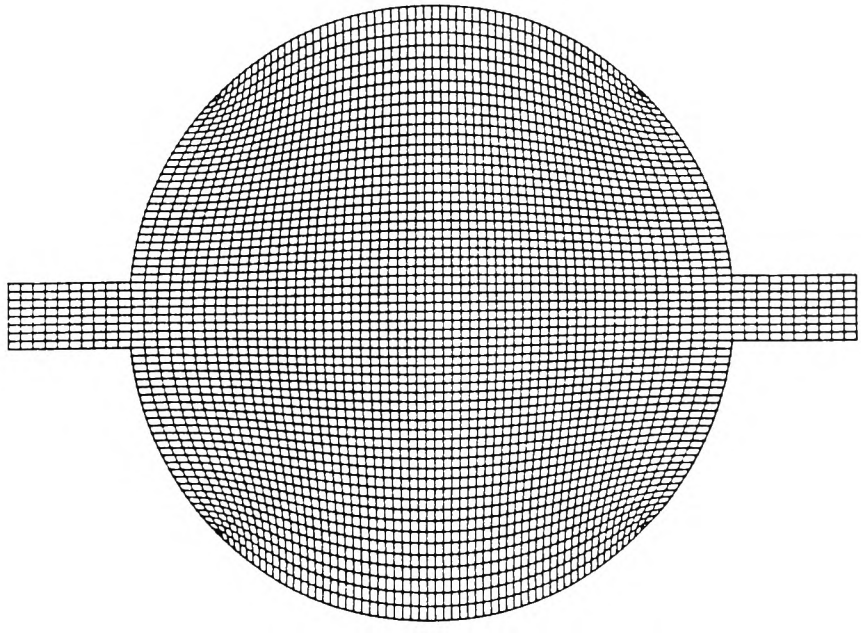
× distribution calculated using B.F.G. illustrated in Figure 7.5a

o distribution calculated using B.F.G. illustrated in Figure 7.6a

Figure 7.8 Transverse U-velocity profile at section halfway along channel

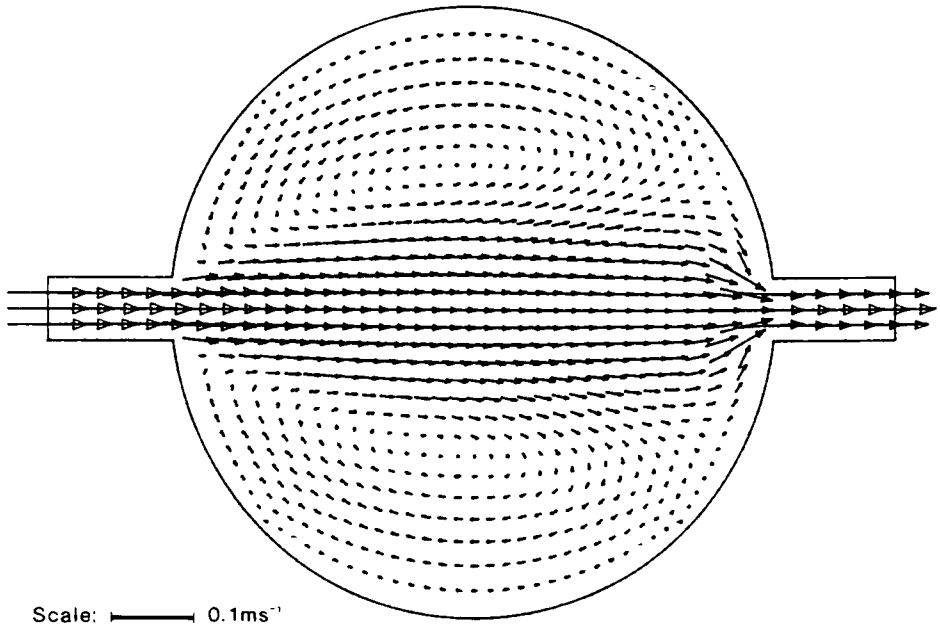


(a) Transformed (ξ, η) plane

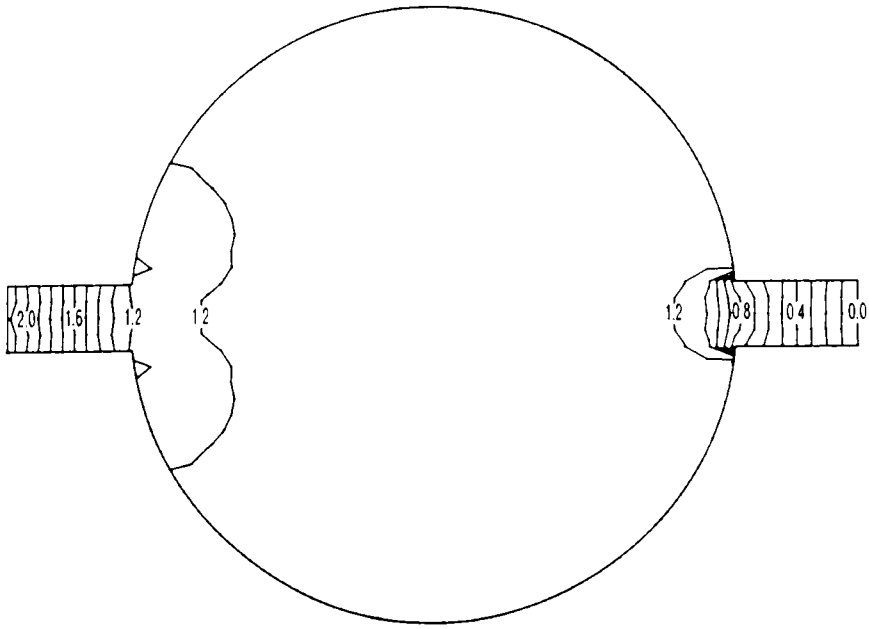


(b) Physical (x,y) plane

Figure 7.9 Boundary-fitted system for a symmetrical circular reservoir (Dennis' geometry)

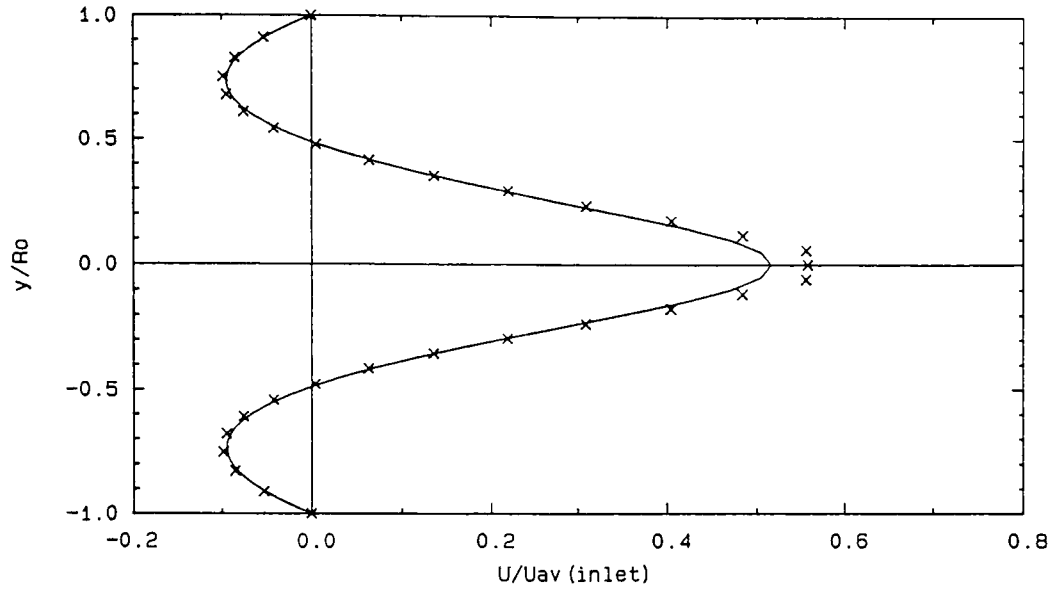


(a) Velocity vectors at steady-state



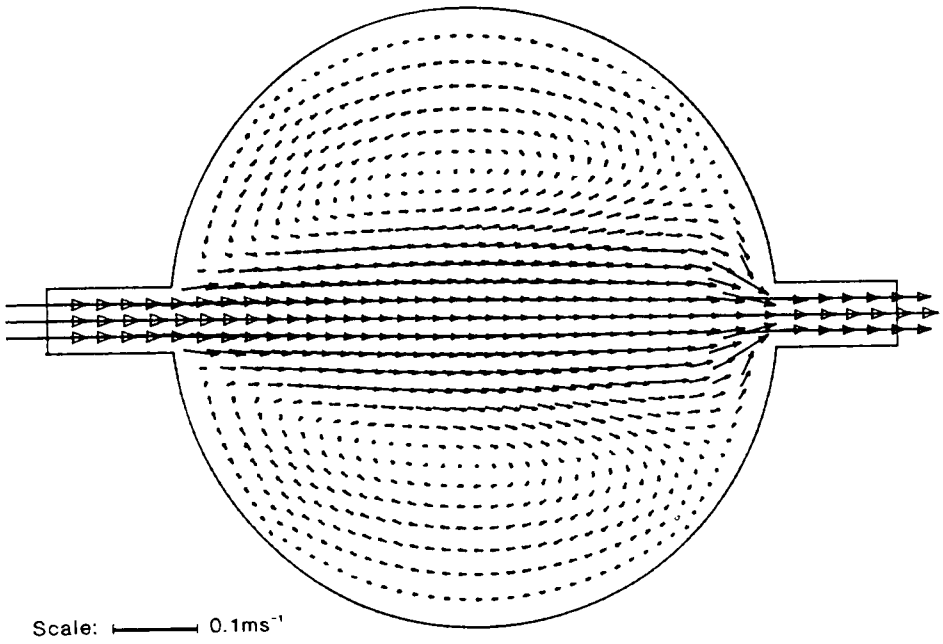
(b) Surface elevation contours at steady-state
(mm above downstream boundary)

Figure 7.10 Jet-forced flow in a symmetrical circular reservoir (free surface calculation; $c_f = 0.000$)

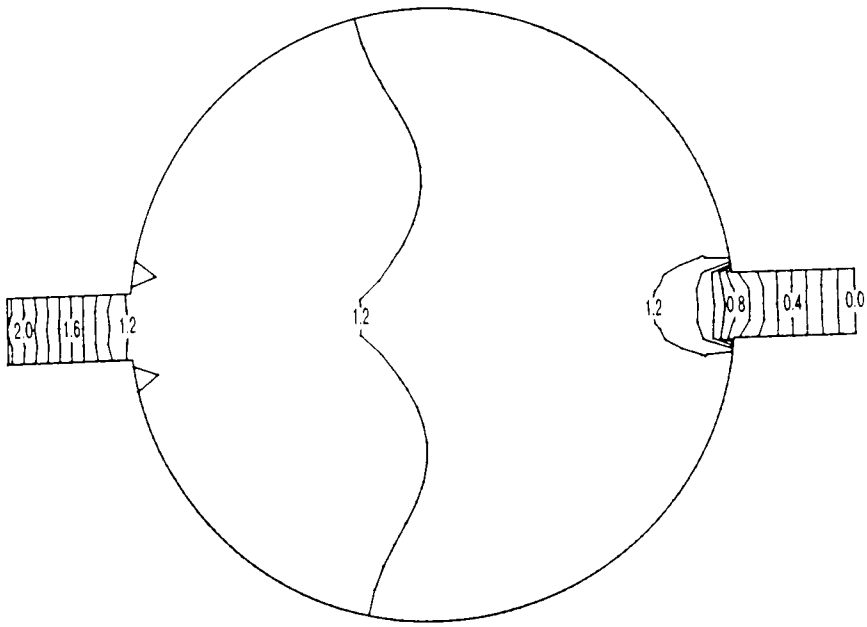


KEY —— (ψ,ω) discretisation
× boundary-fitted (U,V,ζ) scheme

Figure 7.11 Transverse U-velocity profile across mid-section of reservoir (free surface calculation; c_f = 0.000)

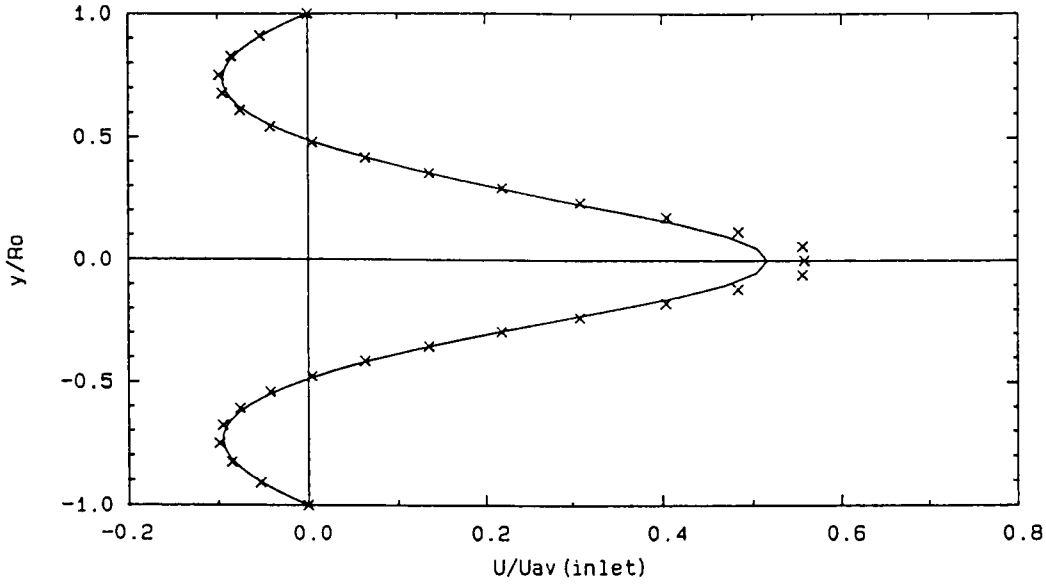


(a) Velocity vectors at steady-state



(b) Surface elevation contours at steady-state
(mm above downstream boundary)

Figure 7.12 Jet-forced flow in a symmetrical circular reservoir (rigid-lid approximation; $c_f = 0.000$)



KEY —— (ψ,ω) discretisation
× boundary-fitted (U,V,ζ) scheme

Figure 7.13 Transverse U-velocity profile across mid-section of reservoir (rigid-lid approximation; $c_f = 0.000$)

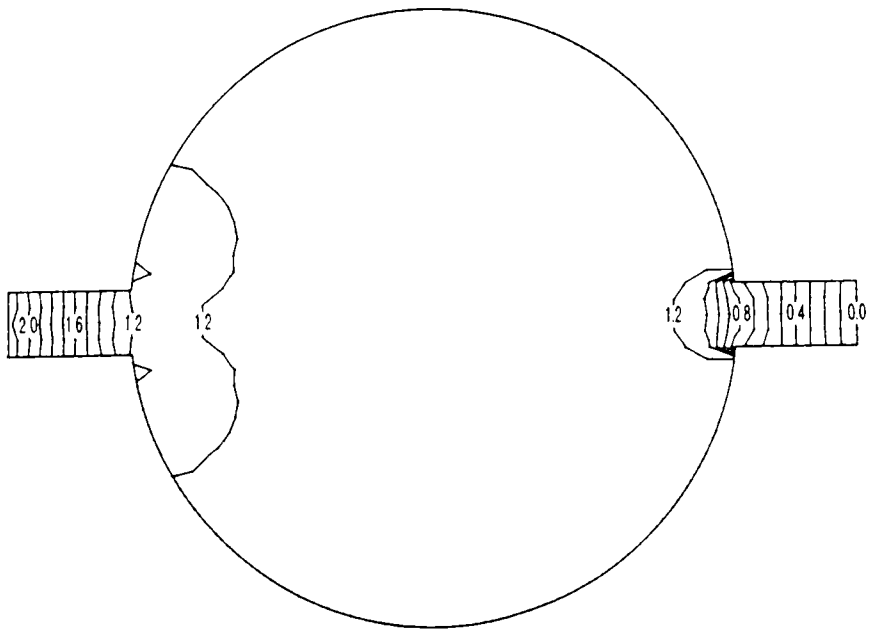
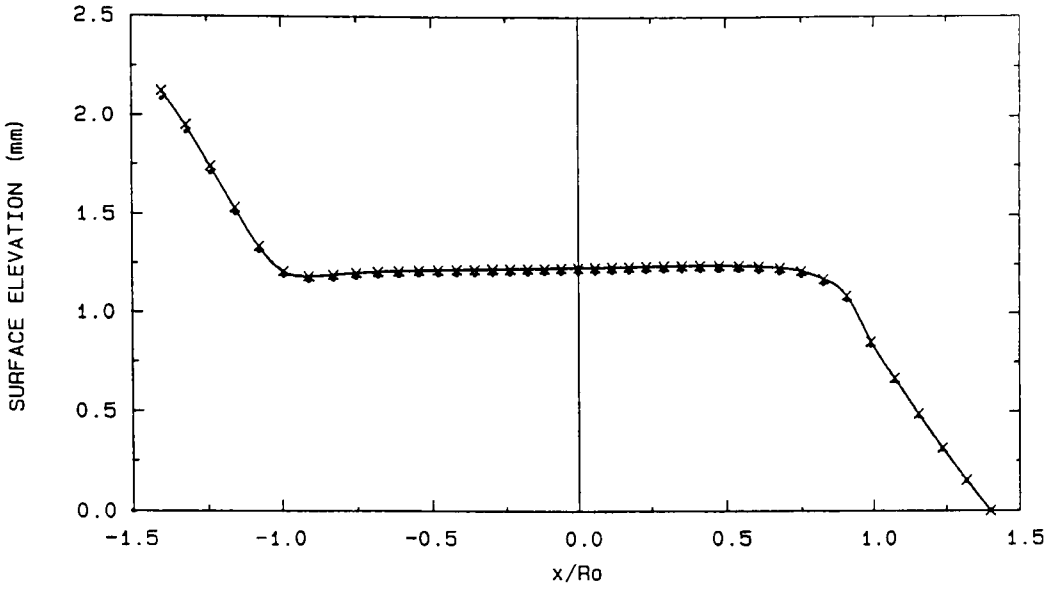


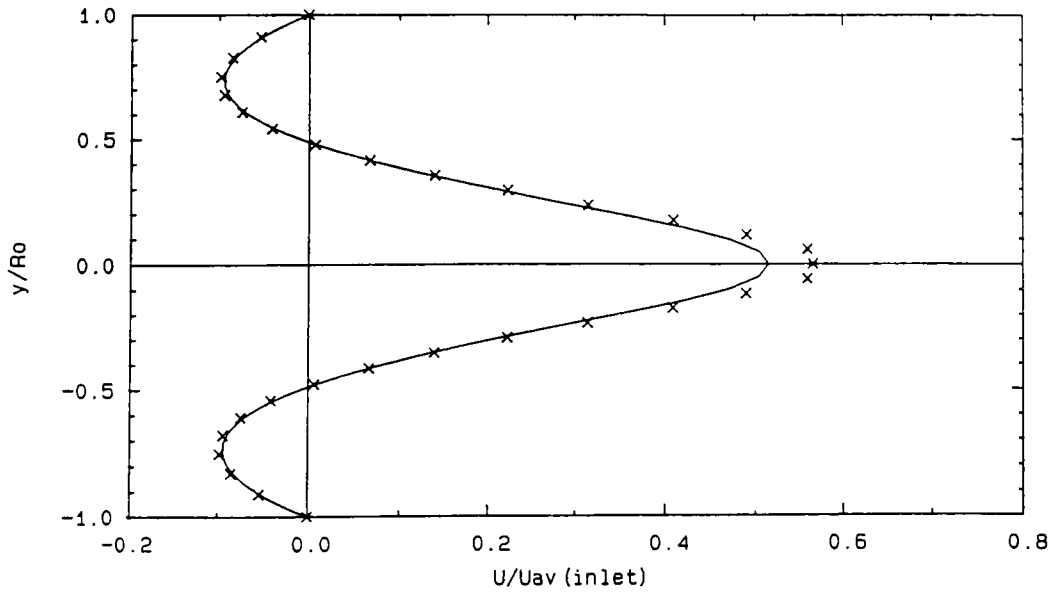
Figure 7.14 Surface elevation contours at steady-state: semi-rigid-lid procedure (mm above downstream boundary)



KEY

—— free surface calculation
• rigid-lid calculation
× semi-rigid-lid calculation

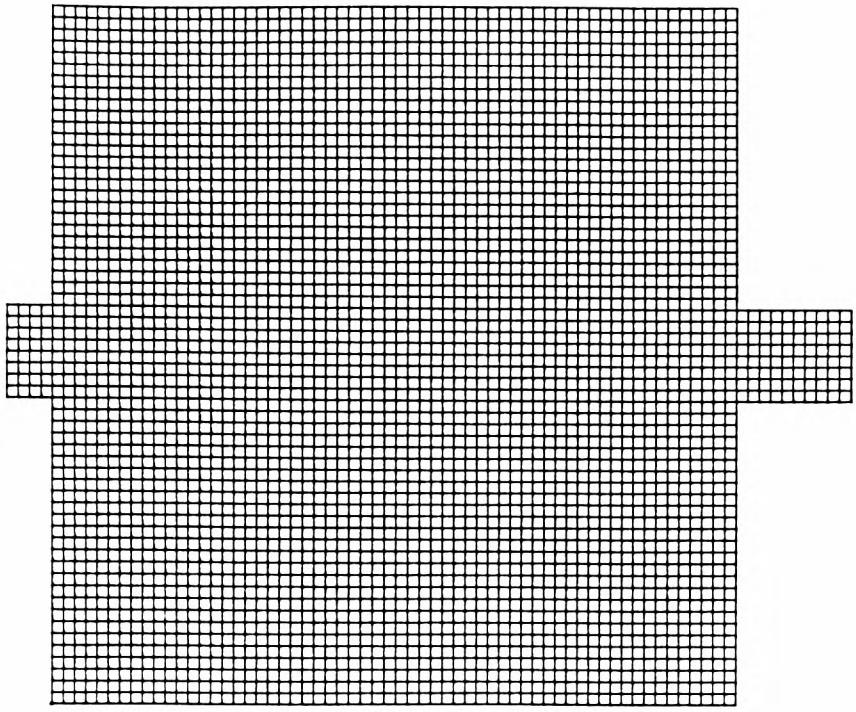
Figure 7.15 Longitudinal surface elevation profile along y = 0



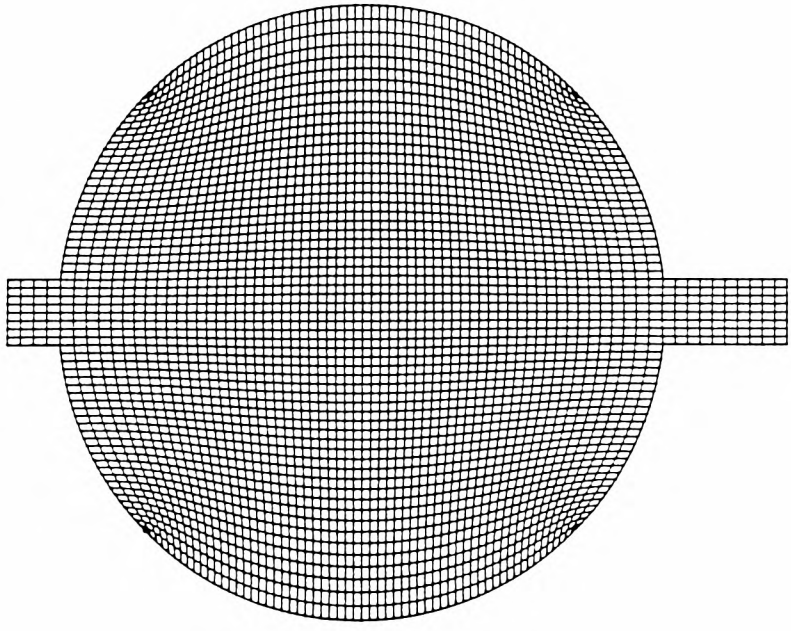
KEY

—— (ψ,ω) discretisation
× boundary-fitted (U,V,ζ) scheme

Figure 7.16 Transverse U-velocity profile across mid-section of reservoir (semi-rigid-lid procedure; c_f = 0.003)



(a) Transformed (ξ, η) plane



(b) Physical (x,y) plane

Figure 7.17 Boundary-fitted coordinate system representative of Dennis' symmetrical geometry (shortened inlet stem)

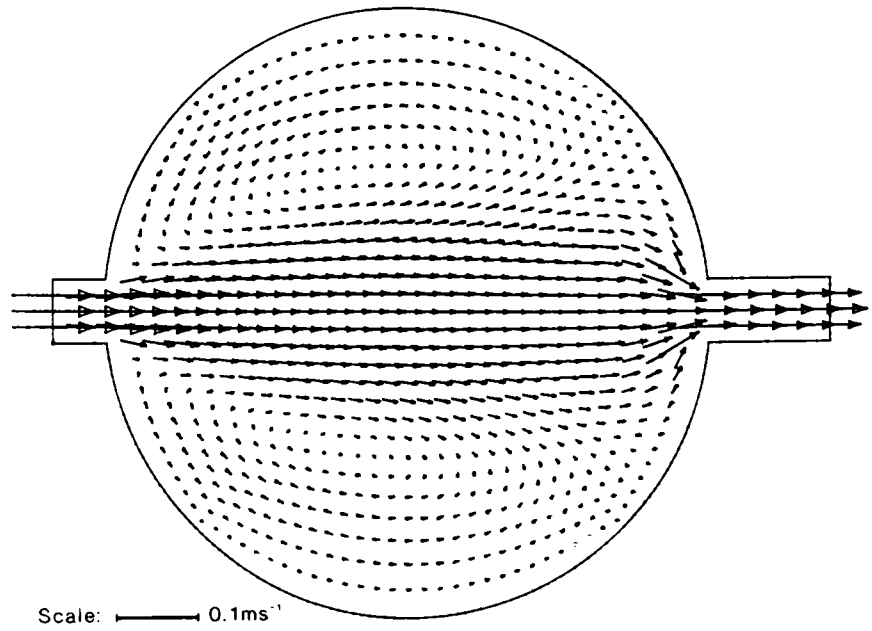
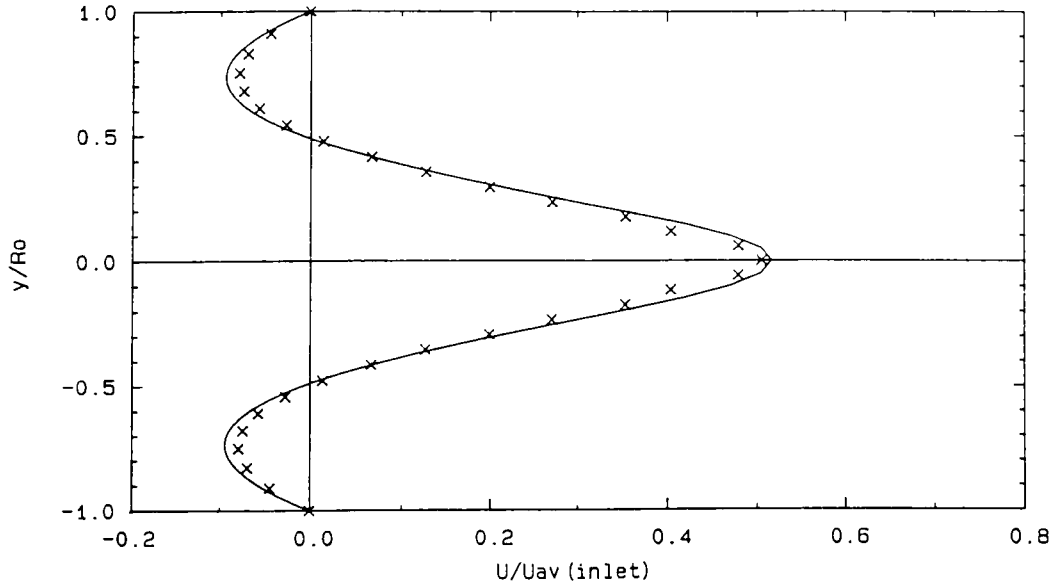
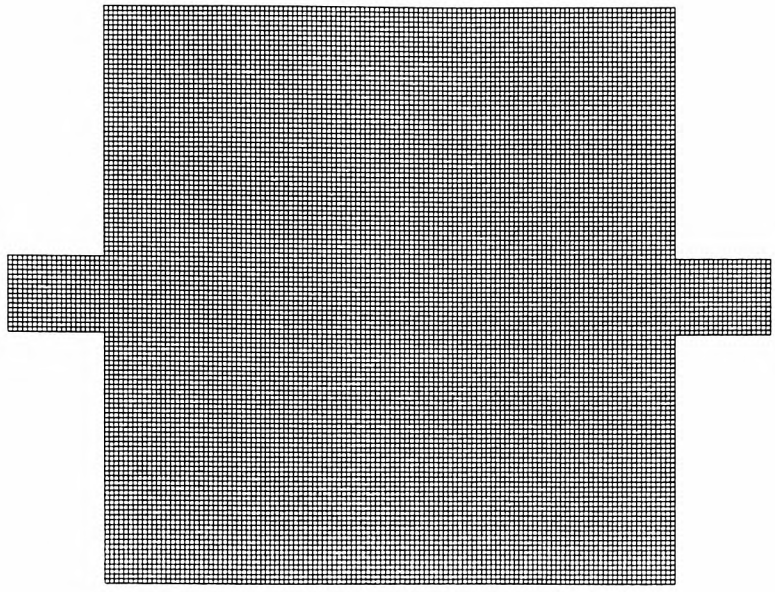


Figure 7.18 Velocity vectors at steady-state (semi-rigid-lid procedure; c_f = 0.000; shortened inlet stem)

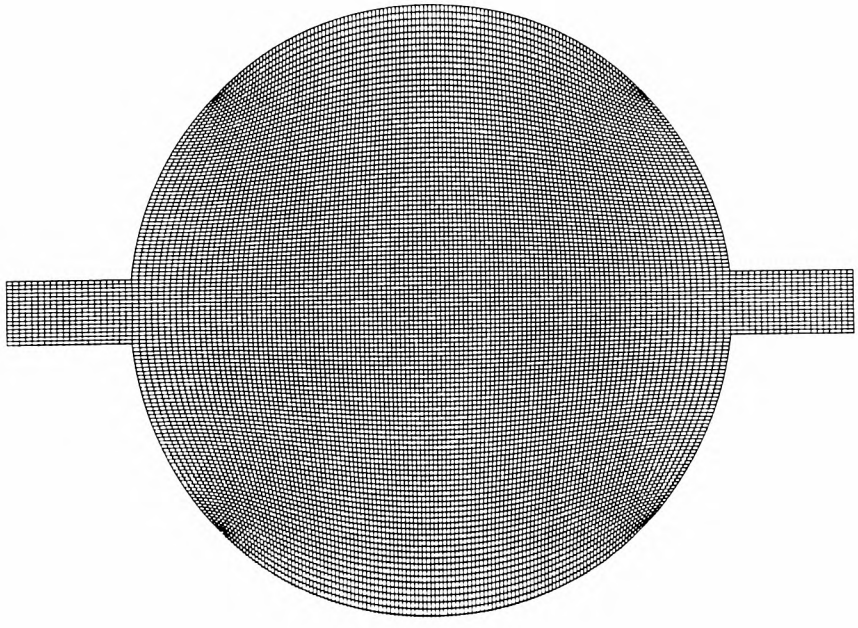


KEY —— (ψ,ω) discretisation
× boundary-fitted (U,V,ζ) scheme

Figure 7.19 Transverse U-velocity profile across mid-section of reservoir (semi-rigid-lid procedure; c_f = 0.000; shortened inlet stem)

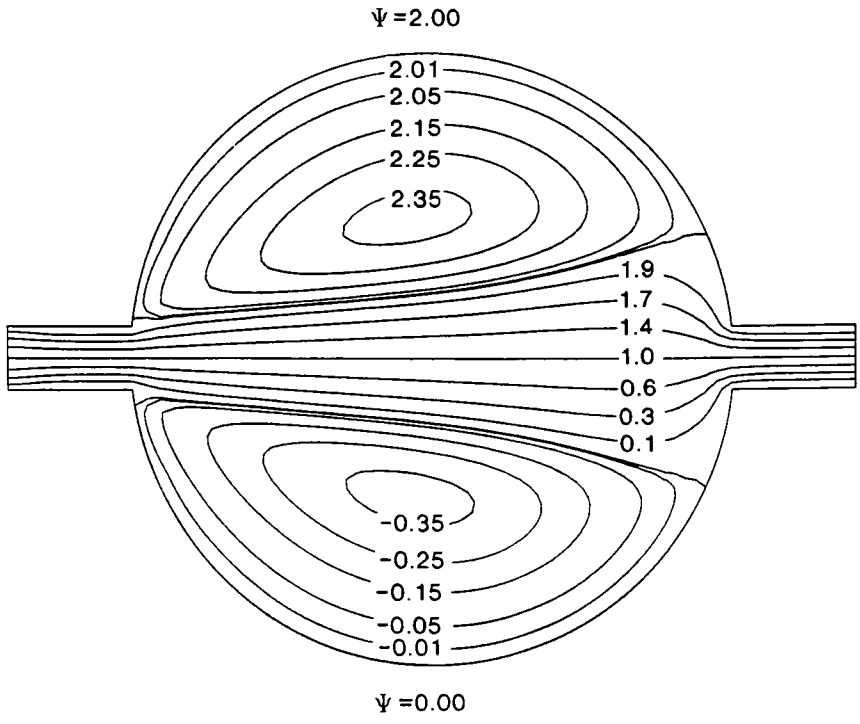


(a) Transformed (ξ, η) plane

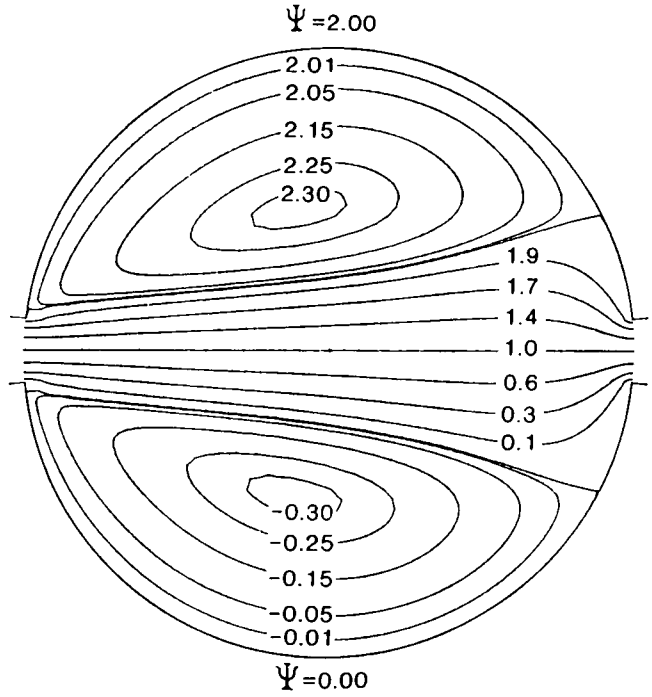


(b) Physical (x,y) plane

Figure 7.20 Boundary-fitted coordinate system representative of Dennis' symmetrical geometry (121 x 121 node refined mesh)

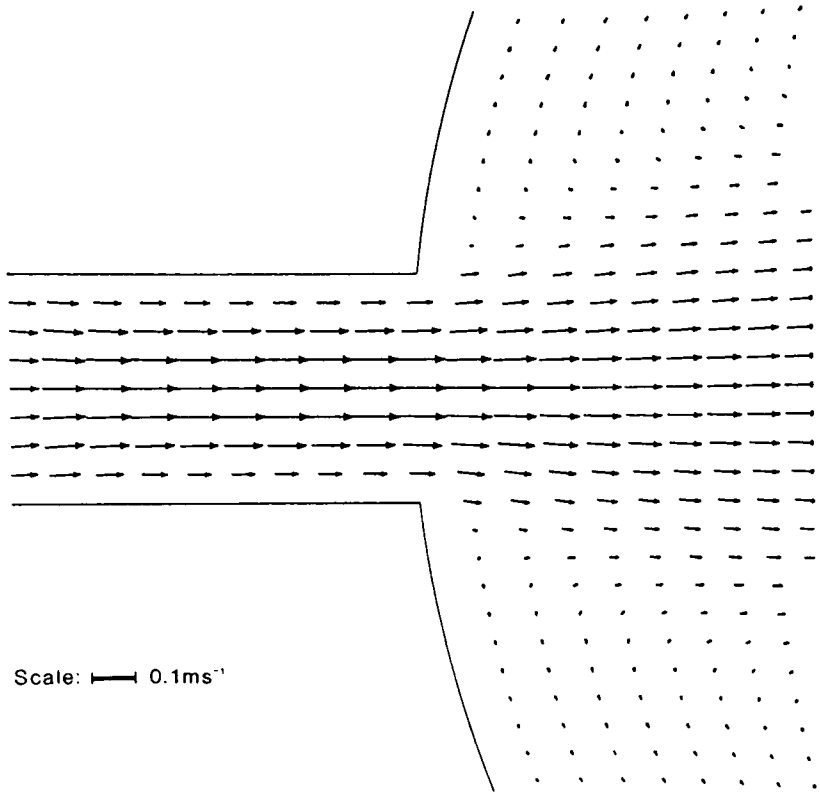


(a) Boundary-fitted (U,V,ζ) approach

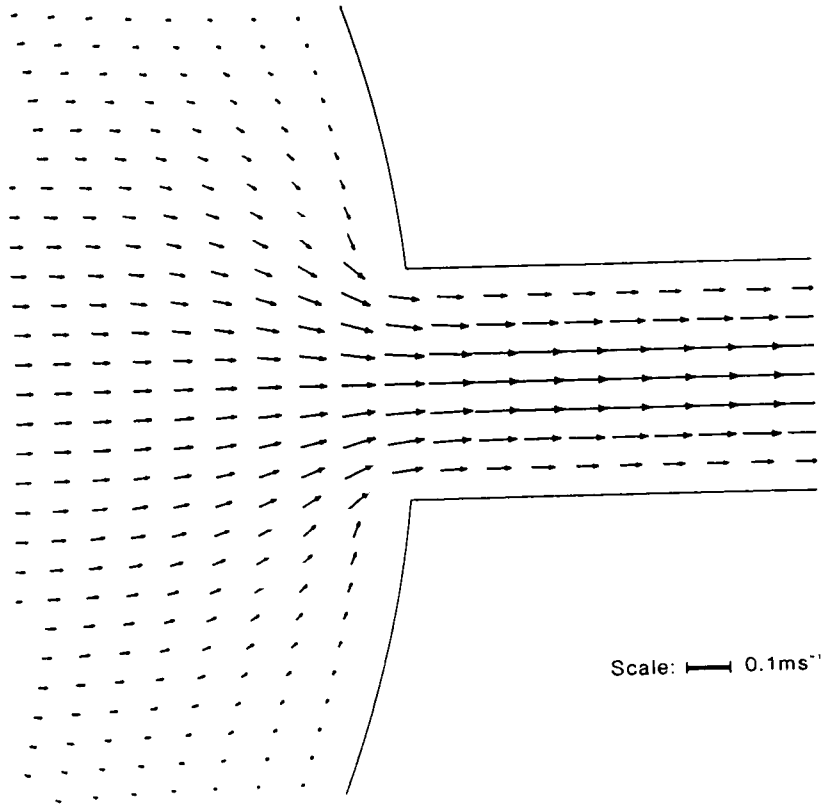


(b) (ψ,ω) discretisation

Figure 7.21 Stream function contours at steady-state (Re_I = 10.0)

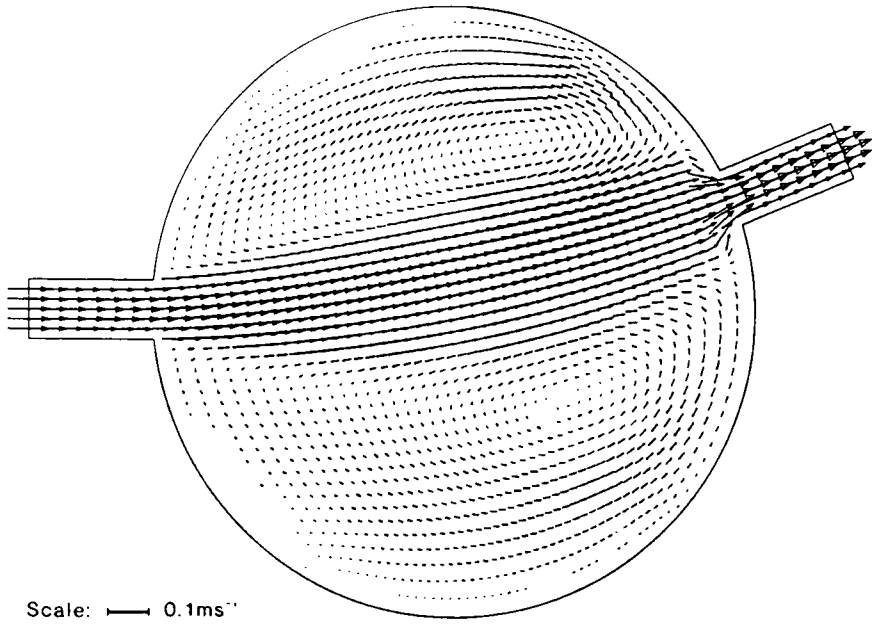


(a) Reservoir inlet

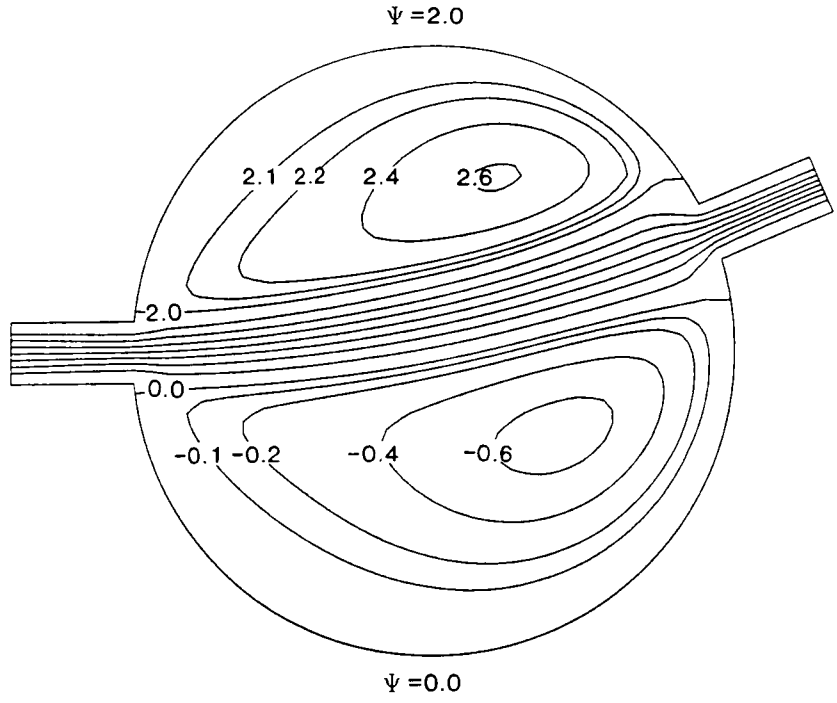


(b) Reservoir outlet

Figure 7.22 Detail of velocity vectors at steady-state (121 x 121 node boundary-fitted system)

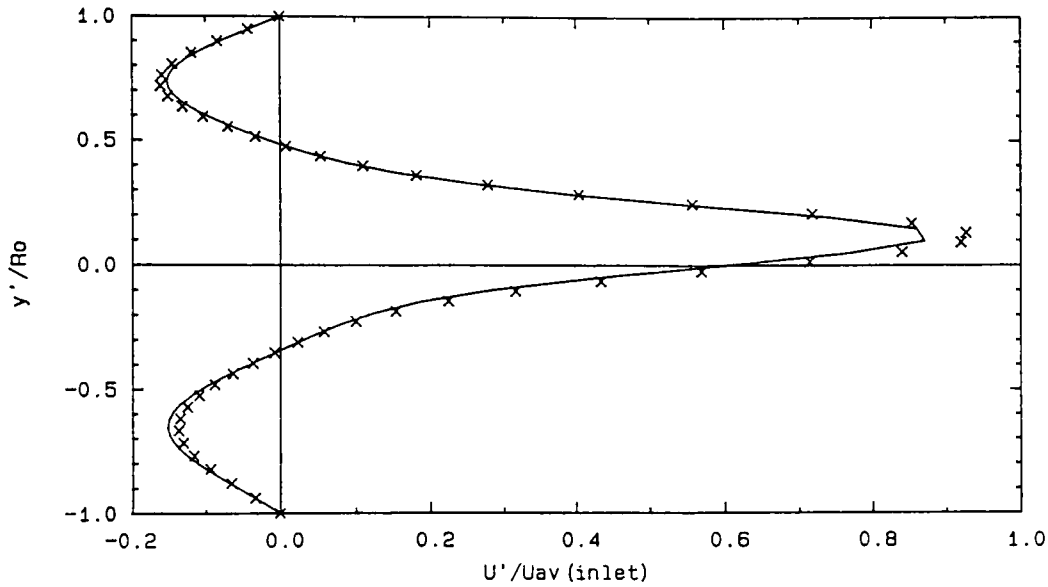


(a) Velocity vectors at steady-state



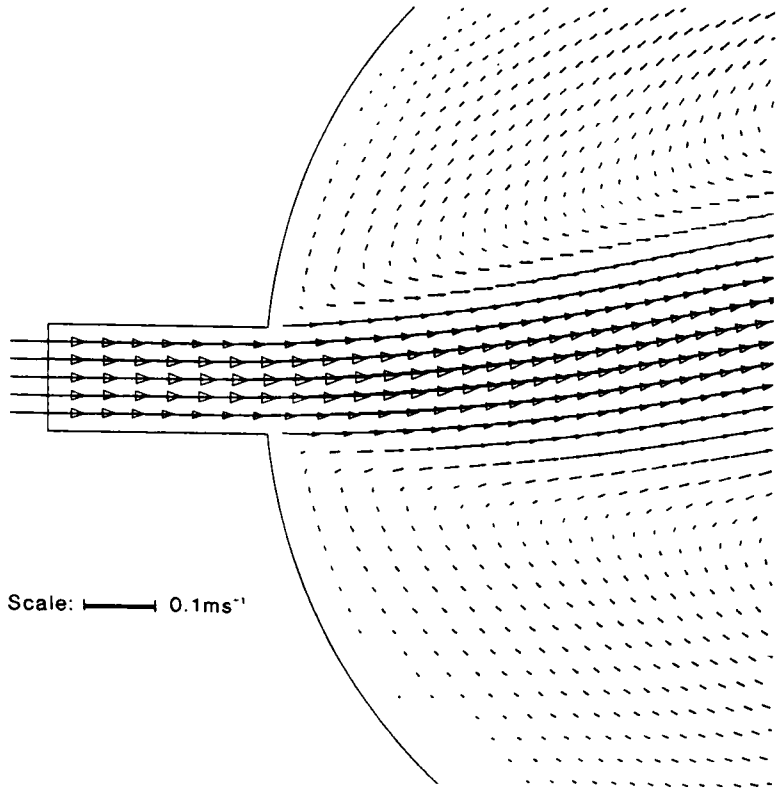
(b) Stream function contours at steady-state

Figure 7.23 Jet-forced flow in an asymmetrical circular reservoir representative of Mills' geometry (Re_I = 25.0)

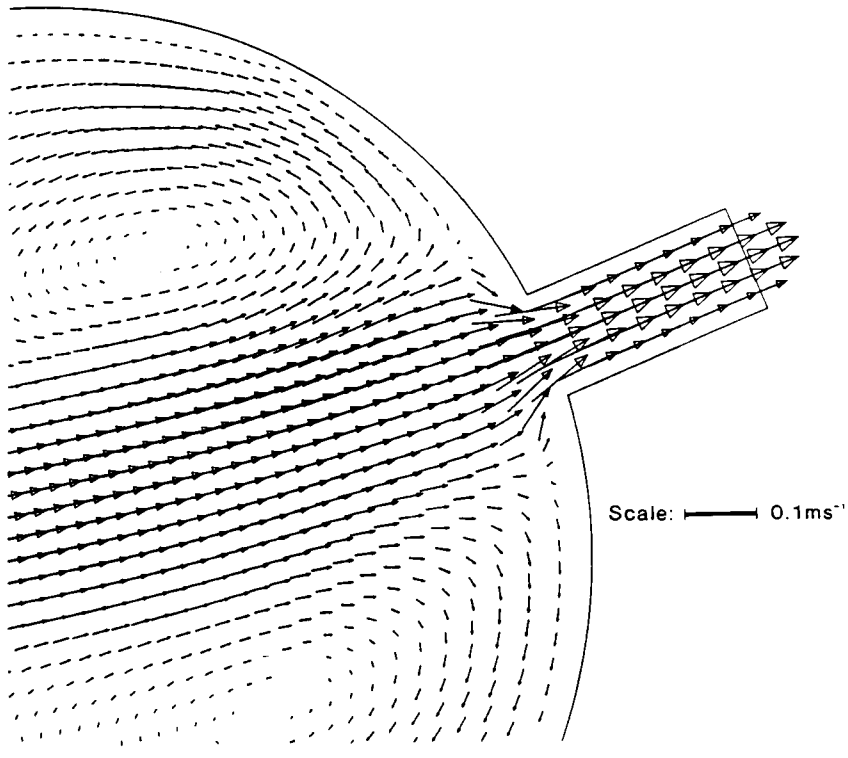


KEY — (ψ,ω) discretisation
x boundary-fitted (U,V,ζ) scheme

Figure 7.24 Transverse U'-velocity profile across mid-section of reservoir (semi-rigid-lid procedure; c_f = 0.000)



(a) Reservoir inlet



(b) Reservoir outlet

Figure 7.25 Detail of velocity vectors at steady-state

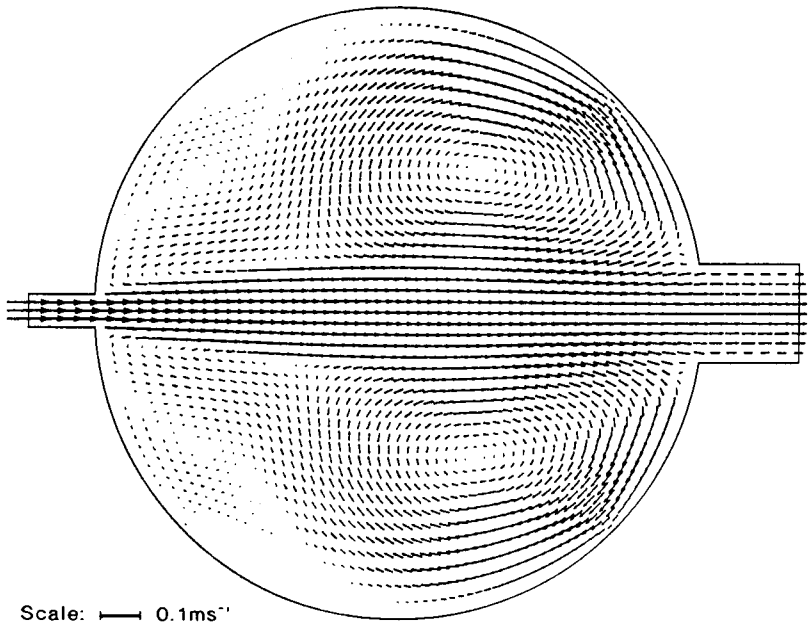
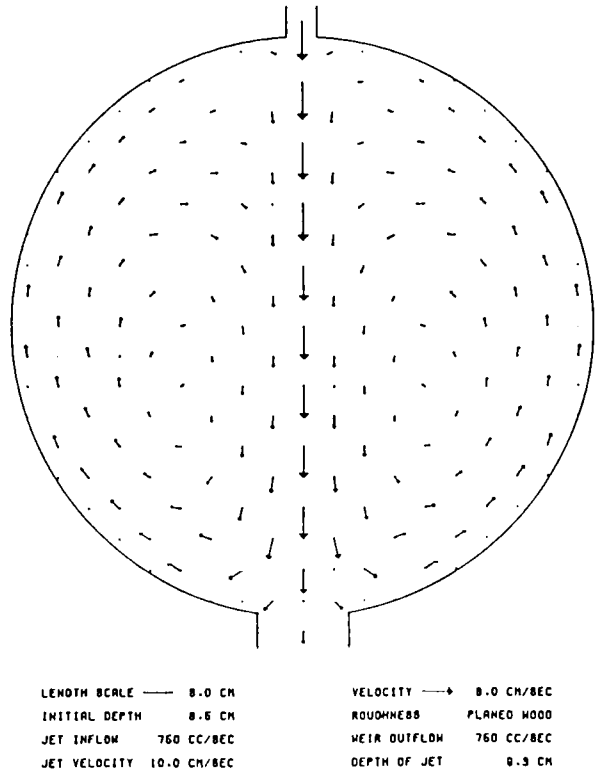
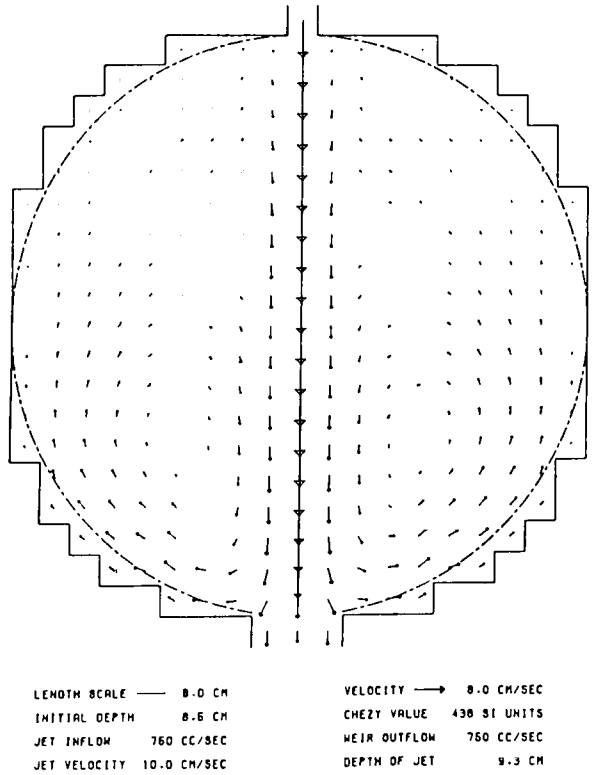


Figure 7.26 Velocity vectors at steady-state (Falconer's circular reservoir geometry; semi-rigid-lid procedure; $C = 438\ m^{\frac{1}{2}}/s$)



(a) Experimentally measured velocity field



(b) Computed velocity field

Figure 7.27 Jet-forced flow in a circular reservoir (after Falconer, 1980)

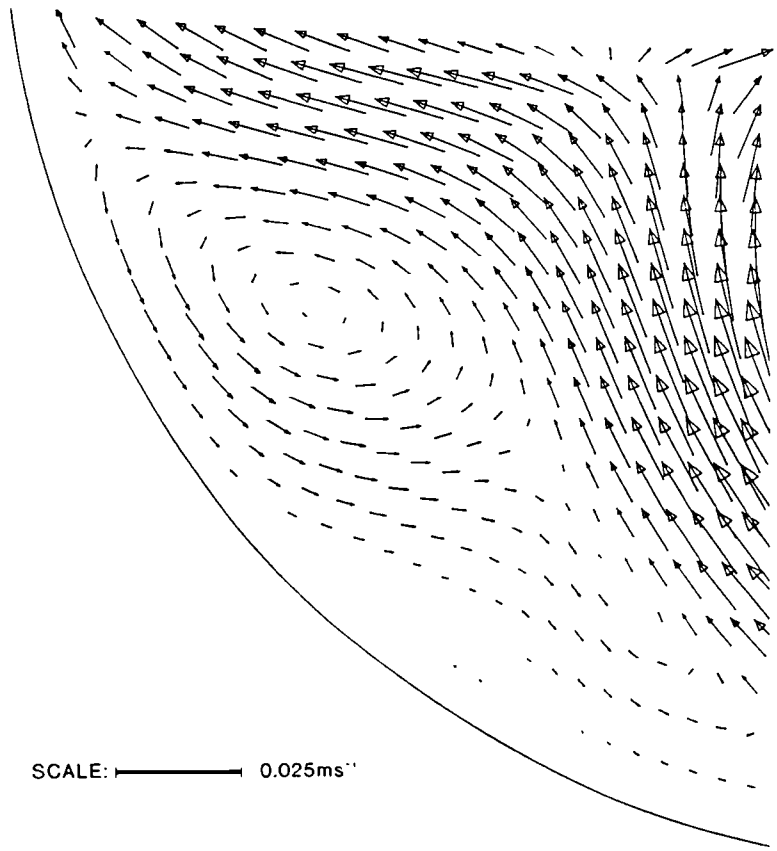


Figure 7.28 Detail of velocity structure in secondary gyre

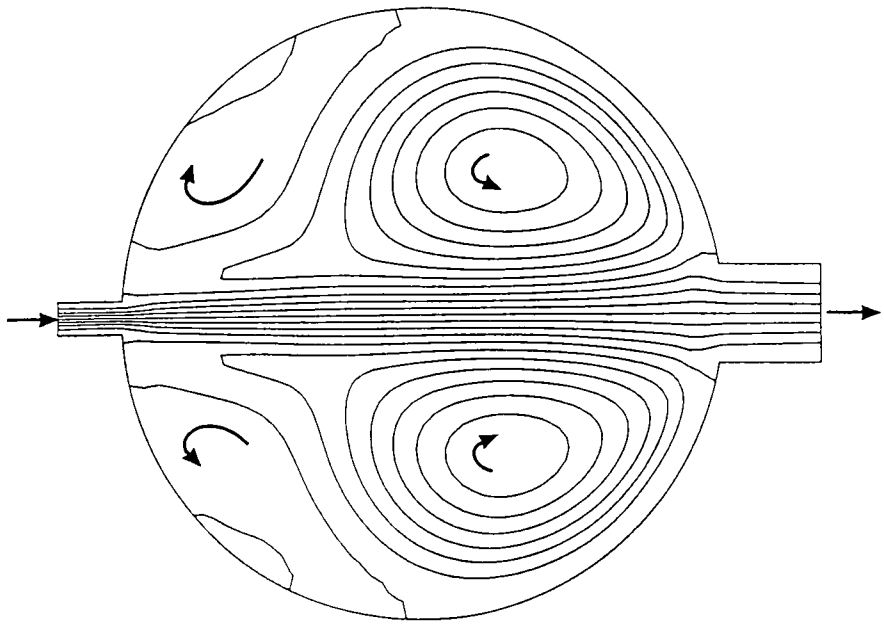


Figure 7.29 Stream function contours at steady-state ($\Delta\psi$ = 0.125 x discharge)

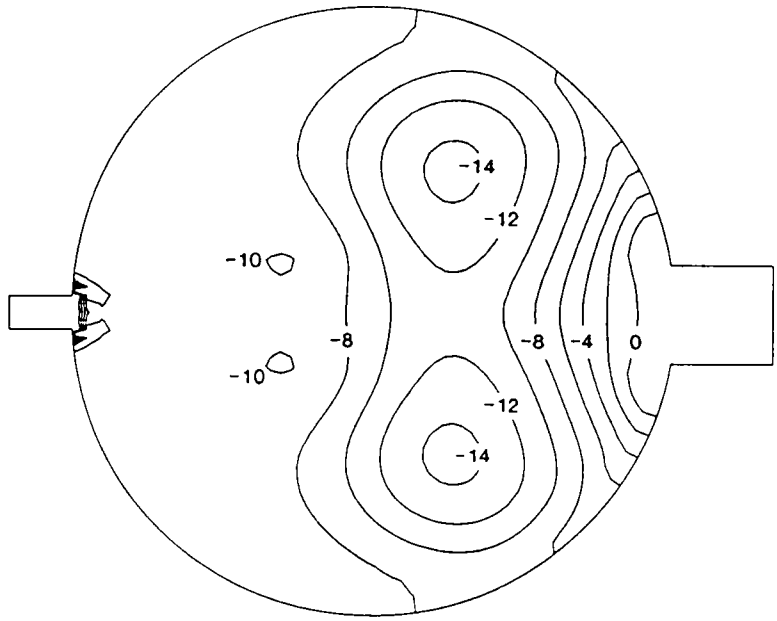


Figure 7.30 Surface elevation contours at steady-state (mm x 10^2)

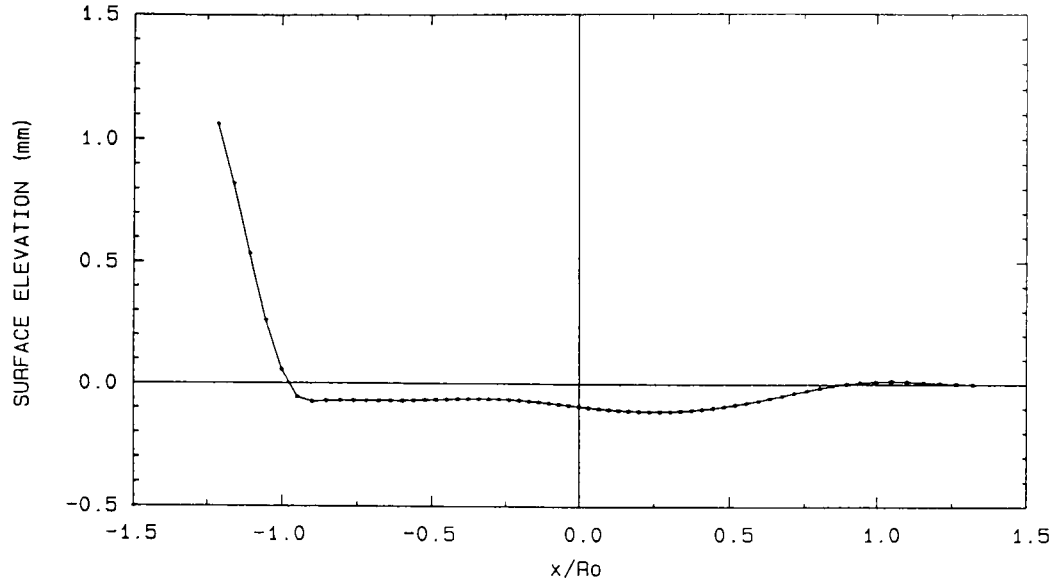


Figure 7.31 Longitudinal surface elevation profile along y = 0

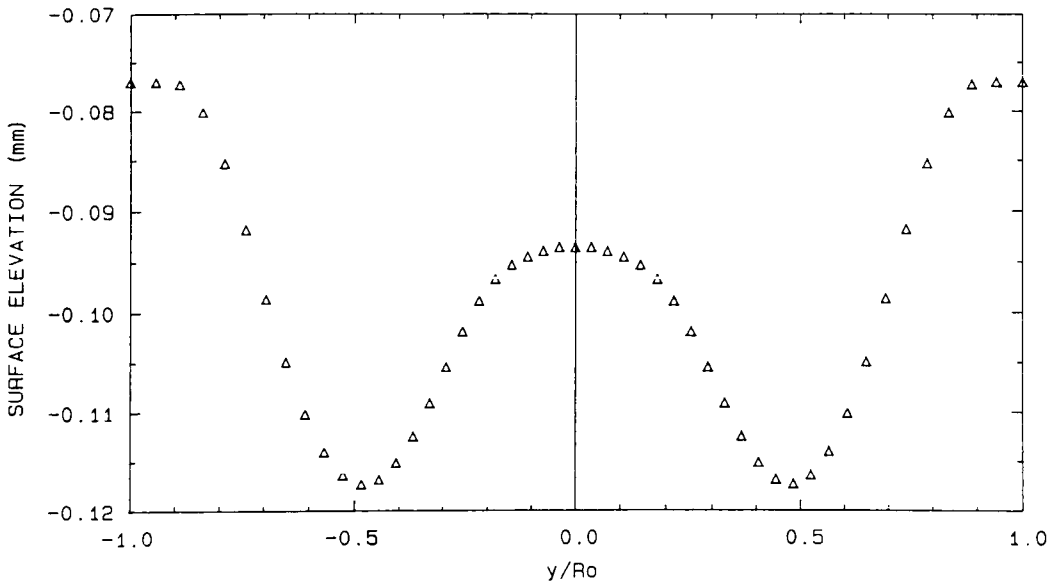
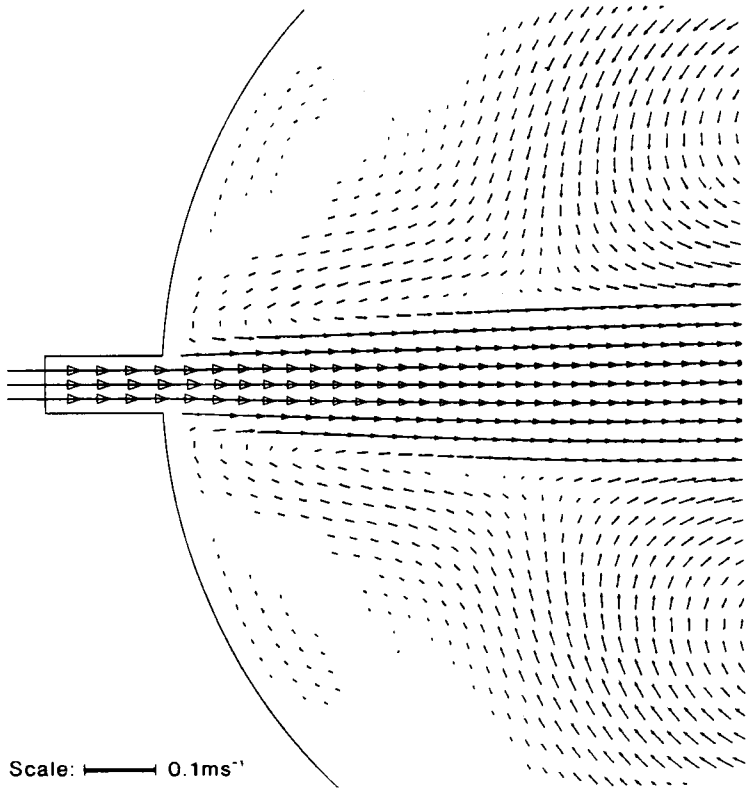
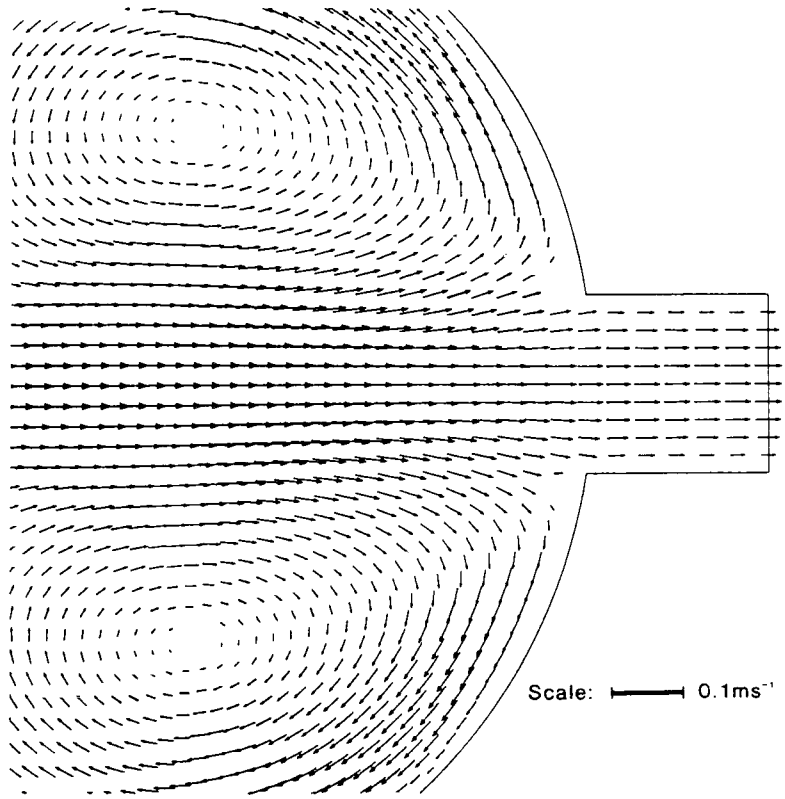


Figure 7.32 Transverse surface elevation profile across mid-section of reservoir

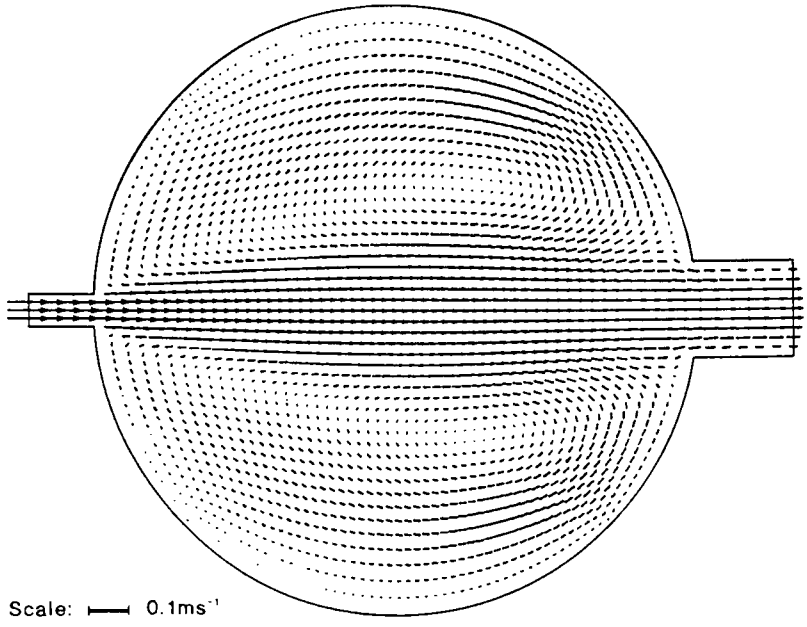


(a) Reservoir inlet

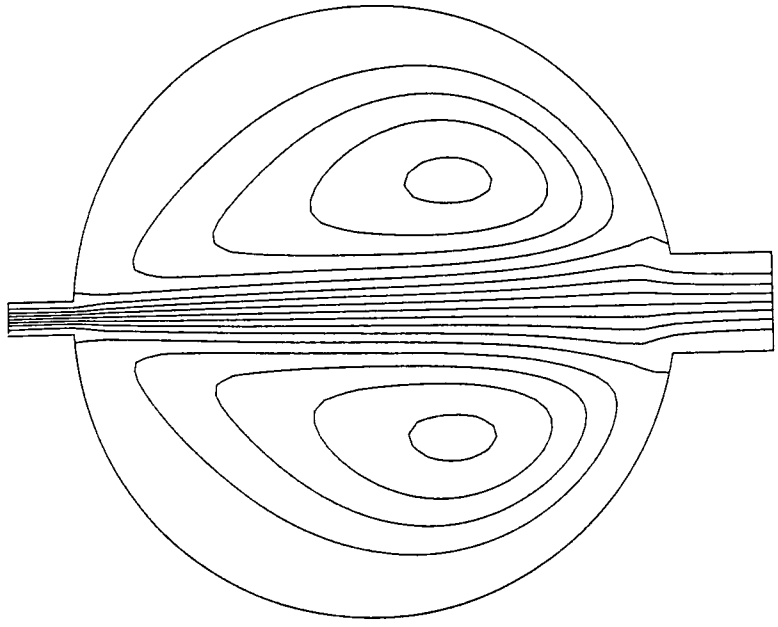


(b) Reservoir outlet

Figure 7.33 Detail of velocity vectors at steady-state

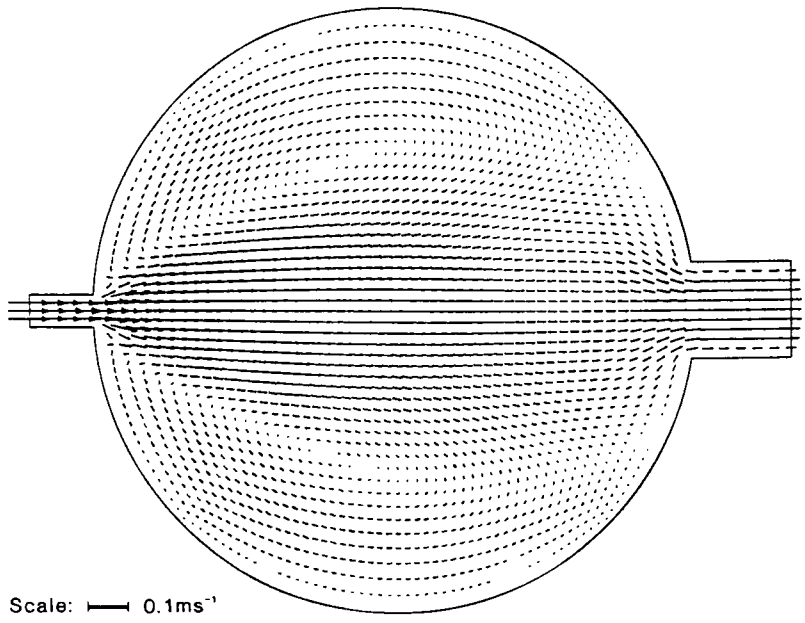


(a) Velocity vectors at steady-state

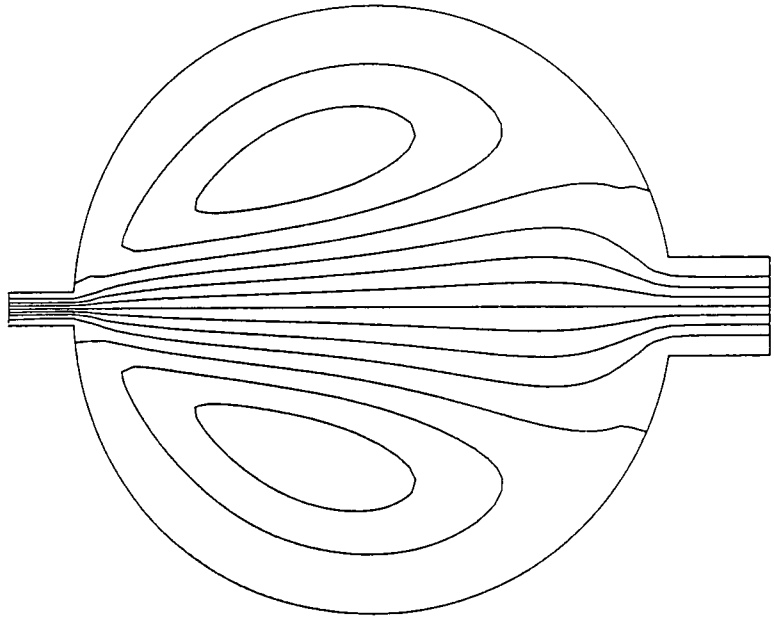


(b) Stream function contours at steady-state
($\Delta\psi$ = 0.125 x discharge)

Figure 7.34 Jet-forced flow in Falconer's circular geometry
(constant $\tilde{\nu}_t$ = 0.000292 m²/s)

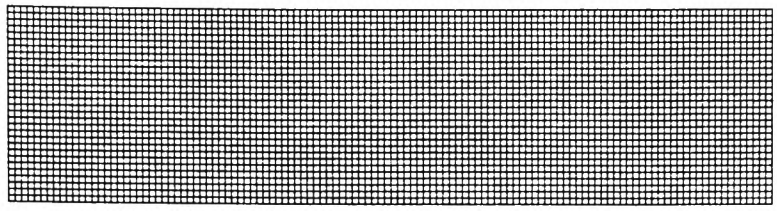


(a) Velocity vectors at steady-state

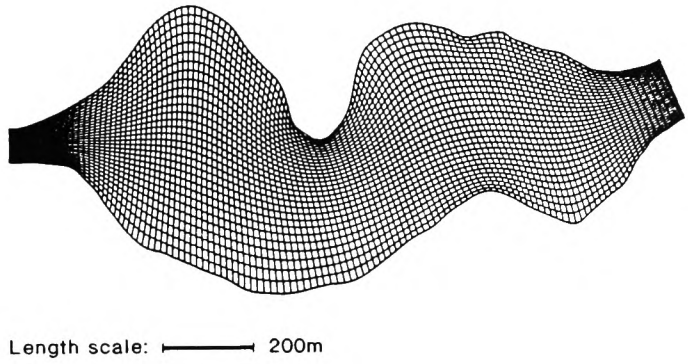


(b) Stream function contours at steady-state
($\Delta\psi$ = 0.125 x discharge)

Figure 7.35 Jet-forced flow in Falconer's circular geometry
(C = 43.8 $m^{\frac{1}{2}}/s$)

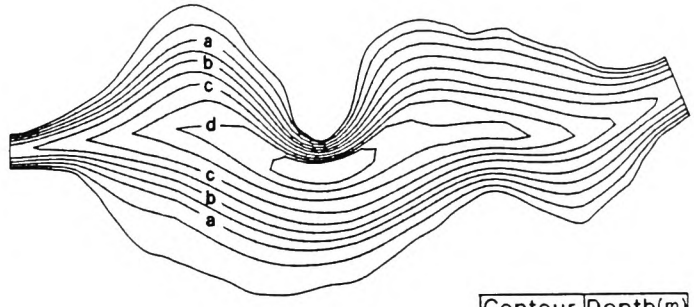


(a) Transformed (ξ, η) plane



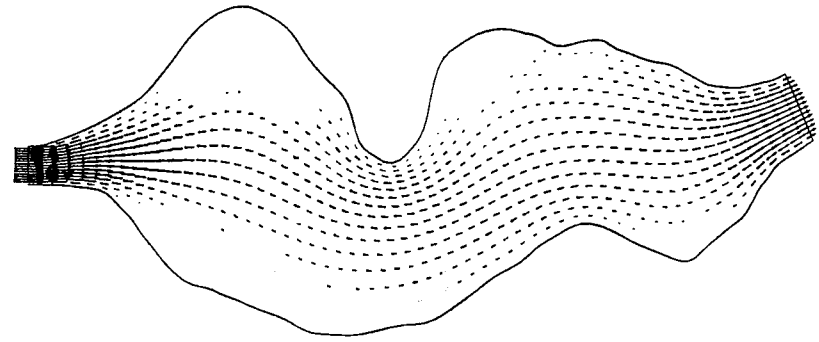
(b) Physical (x,y) plane

Figure 7.36 Boundary-fitted system for arbitrary river geometry

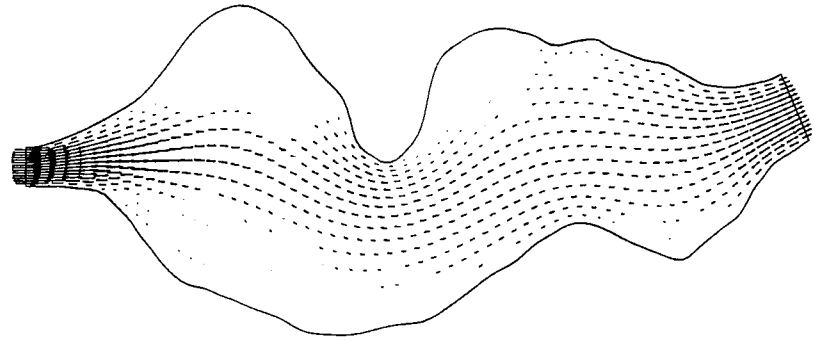


Contour	Depth(m)
a	1.0
b	1.5
c	2.0
d	2.5

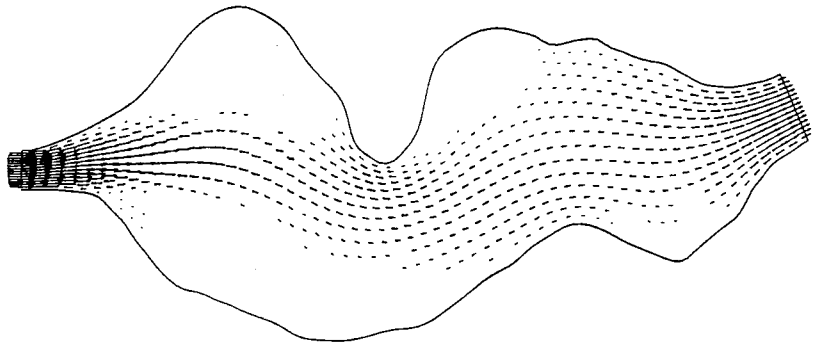
Figure 7.37 Bed topography for arbitrary river geometry (metres below horizontal datum)



Length scale: 200m
Velocity scale: 1.0ms^{-1} (a) $\tilde{\nu}_t = 1.0\ m^2/s$

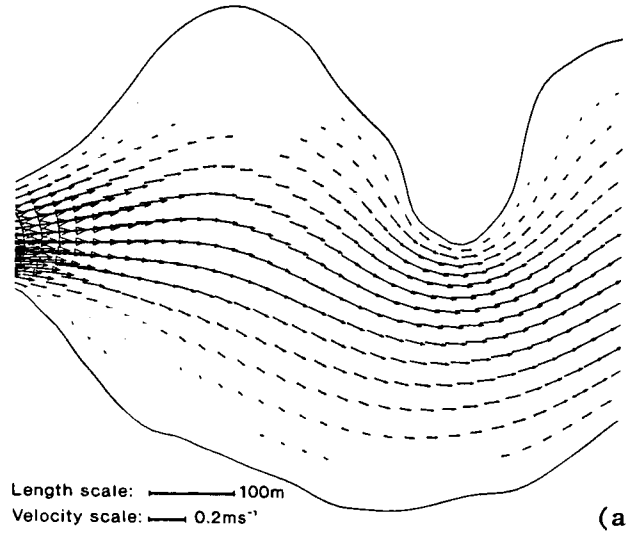


Length scale: 200m
Velocity scale: 1.0ms^{-1} (b) $\tilde{\nu}_t = 0.5\ m^2/s$

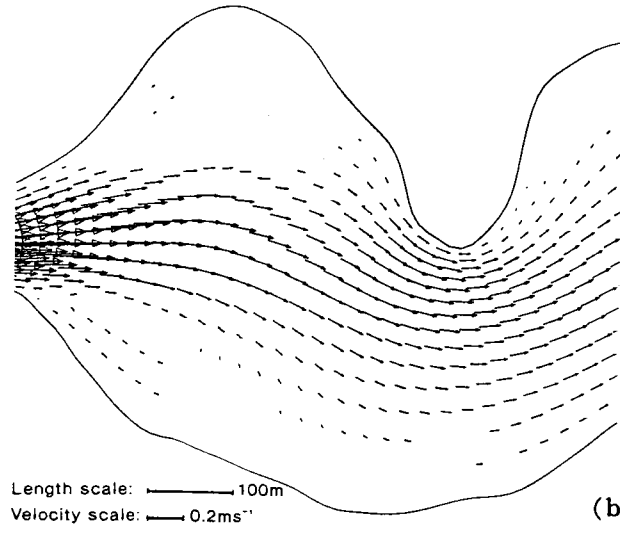


Length scale: 200m
Velocity scale: 1.0ms^{-1} (c) $\tilde{\nu}_t = 0.25\ m^2/s$

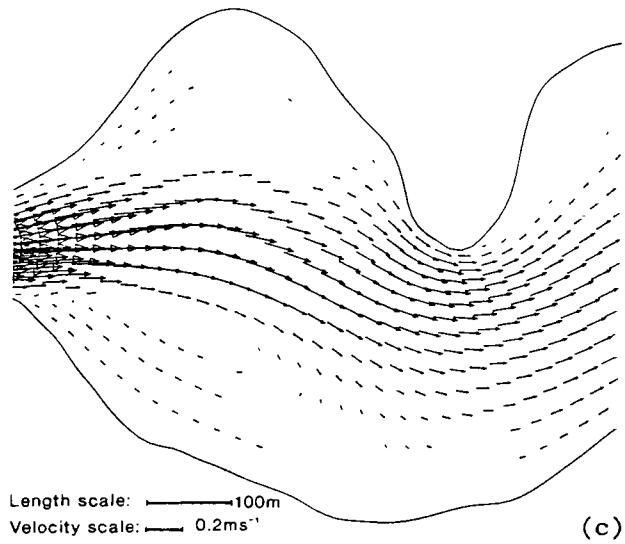
Figure 7.38 Velocity vectors at steady-state: arbitrary river geometry



(a) $\tilde{\nu}_t = 1.0\ m^2/s$



(b) $\tilde{\nu}_t = 0.5\ m^2/s$



(c) $\tilde{\nu}_t = 0.25\ m^2/s$

Figure 7.39 Detail of velocity structure in the inflow jet: arbitrary river geometry

CHAPTER 8
k-ϵ TURBULENCE MODEL

8.1 Introduction

The previous chapter has presented the flow simulations used to validate the boundary-fitted non-orthogonal shallow water equation solver. The hydrodynamic simulations employed zero-equation turbulence models and offered a simple procedure for solving the depth-averaged Reynolds equations. Zero-equation turbulence models have been frequently utilised for modelling large-scale flow phenomena (e.g., Vreugdenhil and Wijbenga, 1982; Stelling, 1983; Wijbenga, 1985a and 1985b; Vemulakonda et al., 1985). Even so, these models neglect the spatial and temporal variations of eddy viscosity and therefore only give crude representations of the actual turbulent conditions. It is therefore desirable to extend the computational scheme by including a higher order turbulence transport model to predict the spatial variations of the eddy viscosity coefficient. As explained in Section 5.5, a reasonable and widely used transport model is the so-called two-equation k-ϵ approach which characterises the state of turbulence via the distributions of turbulent kinetic energy, k, and dissipation rate, ϵ. Consequently, the present study chooses to discretise the depth-averaged k-ϵ equations originally proposed by Rastogi and Rodi (1978) and McGuirk and Rodi (1978). The derivation of the depth-averaged k-ϵ model is described in Section 5.6, whilst the transformation of the governing turbulence transport equations into the computational (ξ,η) coordinate system is detailed in the latter part of Section 5.9.

Early (Cartesian) depth-averaged k-ϵ models (e.g., Rastogi and Rodi, 1978; McGuirk and Rodi, 1978 and 1979; Rodi et al., 1981) were concerned with *near-field* flow problems where the rigid-lid assumption was deemed to provide an acceptable simplification of the governing hydrodynamic equations. As a result, these early models favoured the use of the finite-volume discretisation technique. For the purposes of the boundary-fitted approach, however, the depth-averaged momentum and continuity equations are solved using finite-differences, so that the numerical model can simulate flows with significant variations in free surface. In keeping with this, the non-orthogonal k-ϵ transport equations (5.94a and 5.94b) are also discretised using finite-difference methods.

The computational schemes developed in this chapter are loosely based upon the numerical procedures devised by members of the Department of Civil Engineering, Imperial College, for investigating wave-driven nearshore circulation patterns. The

study of currents in the nearshore environment must account for wave-induced *set-up* and *set-down* which are often significant compared with the total flow depth and, therefore, the numerical schemes used to simulate coastal flow patterns are usually based upon finite-difference approaches. Walker (1987) and Dong (1987) have developed computer programs for simulating wave-induced circulation on gradually sloping beaches, and have both implemented two-equation turbulence models for predicting the spatial variation of eddy viscosity. Consequently, despite the widely differing applications, the Cartesian finite-difference procedures developed by Walker and Dong have much in common with the present simulation of nearly horizontal jet-forced reservoir flow. Walker (1987) implemented an alternating-direction-implicit (A.D.I.) procedure using central differences throughout, whereas Dong (1987) discretised the governing hydrodynamic equations using an explicit finite-difference technique. To prevent the occurrence of non-linear instabilities, Dong had to resort to an angled-derivative-explicit (A.D.E.) scheme proposed by Roberts and Weiss (1966) - (see also Roache, 1972). The present study describes two numerical schemes for solving the $k-\epsilon$ equations in non-orthogonal curvilinear coordinate form; the first modifies Walker's implicit approach, whilst the second uses an explicit discretisation technique. Since the A.D.E. method cannot readily be converted for use with non-orthogonal grids, the present explicit scheme overcomes the destabilising influences of the non-linear advective terms by using the second order upwind and weighted central difference procedures, proposed by Stelling (1983) and Stelling and Willemse (1984).

Numerical experimentation has revealed that both discretisation techniques are unable to model the large spatial variations of k and ϵ in the vicinity of the inlet and outlet stems of the circular reservoir geometries. The destabilising effects were found to originate from the intersections of the parallel sided channel walls and the circular perimeter, but an extensive series of numerical trials failed to resolve the difficulties at the corner nodes. Therefore, in order to demonstrate the $k-\epsilon$ discretisation, the turbulence models are applied to the arbitrary river geometry, described in Section 7.5.

8.2 Numerical scheme

As stated in Section 8.1, two finite-difference techniques for discretising the turbulence transport equations are presented in this chapter. The first modifies Walker's (1987) Cartesian $k-\epsilon$ computational procedure for use with non-orthogonal boundary-fitted coordinate systems, and employs an A.D.I. procedure in conjunction

with central differences for the advective terms. In the second approach, the turbulence variables are updated using an explicit forward-time discretisation with second order upwind and weighted central differencing for the advective components of the k- and ϵ- transport equations.

With reference to Figure 6.2, the turbulence equations are centred at the scalar variable positions of the staggered mesh, and consequently, for the present discretisation purposes, the relationships between the geometric and hydrodynamic indexing systems become

$$I = 2i - 1 \qquad (6.1 \text{ bis})$$

and

$$J = 2j - 1 \quad . \qquad (6.3 \text{ bis})$$

Except for the source and dissipation terms, the depth-averaged k- and ϵ-expressions are very similar, and therefore only the k-equation discretisation is described in detail.

The notation used to define the spatial indexing and time level is identical with that presented in Section 6.6, for the numerical solution of the shallow water equations. Furthermore, in order to improve the clarity of the finite-difference equations, the overbars (denoting that a parameter is not a true depth-averaged quantity) are omitted from all subsequent expressions.

8.2.1 SCHEME 1 : A.D.I. technique using central differences for the advective terms

(a) STEP 1 : ξ-direction release : $t=n\Delta t \rightarrow (n+\frac{1}{2})\Delta t$

All derivatives with respect to ξ are written at the advanced time level, $t = (n+\frac{1}{2})\Delta t$, whereas derivatives involving η are held at $t = n\Delta t$. The non-derivative expressions (e.g., production and dissipation terms) are also evaluated at the advanced time stage. Thus, the individual terms of the transformed k-equation (5.94a) are discretised as:

$$\frac{\partial k}{\partial t} = \frac{k_{i,j}^{n+\frac{1}{2}} - k_{i,j}^{n}}{\frac{1}{2}\Delta t}$$

$$y_\eta \; U \; \frac{\partial k}{\partial \xi} = y_{\eta_{I,J}} \left[\frac{U^{n+\frac{1}{2}}_{i,j} + U^{n+\frac{1}{2}}_{i-1,j}}{2} \right] \left[\frac{k^{n+\frac{1}{2}}_{i+1,j} - k^{n+\frac{1}{2}}_{i-1,j}}{2} \right]$$

$$y_\xi \; U \; \frac{\partial k}{\partial \eta} = y_{\xi_{I,J}} \left[\frac{U^{n}_{i,j} + U^{n}_{i-1,j}}{2} \right] \left[\frac{k^{n}_{i,j+1} - k^{n}_{i,j-1}}{2} \right]$$

$$x_\xi \; V \; \frac{\partial k}{\partial \eta} = x_{\xi_{I,J}} \left[\frac{V^{n}_{i,j} + V^{n}_{i,j-1}}{2} \right] \left[\frac{k^{n}_{i,j+1} - k^{n}_{i,j-1}}{2} \right]$$

$$x_\eta \; V \; \frac{\partial k}{\partial \xi} = x_{\eta_{I,J}} \left[\frac{V^{n+\frac{1}{2}}_{i,j} + V^{n+\frac{1}{2}}_{i,j-1}}{2} \right] \left[\frac{k^{n+\frac{1}{2}}_{i+1,j} - k^{n+\frac{1}{2}}_{i-1,j}}{2} \right]$$

$$y_\eta \; \frac{\partial Kx}{\partial \xi} = y_{\eta_{I,J}} \left[\frac{Kx^{n+\frac{1}{2}}_{i+1,j} - Kx^{n+\frac{1}{2}}_{i-1,j}}{2} \right]$$

$$y_\xi \; \frac{\partial Kx}{\partial \eta} = y_{\xi_{I,J}} \left[\frac{Kx^{n}_{i,j+1} - Kx^{n}_{i,j-1}}{2} \right]$$

$$x_\xi \; \frac{\partial Ky}{\partial \eta} = x_{\xi_{I,J}} \left[\frac{Ky^{n}_{i,j+1} - Ky^{n}_{i,j-1}}{2} \right]$$

and

$$x_\eta \; \frac{\partial Ky}{\partial \xi} = x_{\eta_{I,J}} \left[\frac{Ky^{n+\frac{1}{2}}_{i+1,j} - Ky^{n+\frac{1}{2}}_{i-1,j}}{2} \right] .$$

Letting the Jacobian be denoted by JAC in order to avoid confusion with the spatial index, J, equation (5.94a) is formulated in finite-differences as

$$\frac{k^{n+\frac{1}{2}}_{i,j} - k^{n}_{i,j}}{\frac{1}{2}\Delta t} + \frac{1}{JAC_{I,J}} \left[y_{\eta_{I,J}} UA1 . \left[\frac{k^{n+\frac{1}{2}}_{i+1,j} - k^{n+\frac{1}{2}}_{i-1,j}}{2} \right] - y_{\xi_{I,J}} UA2 . KD \right.$$

$$\left. + x_{\xi_{I,J}} VA2 . KD - x_{\eta_{I,J}} VA1 . \left[\frac{k^{n+\frac{1}{2}}_{i+1,j} - k^{n+\frac{1}{2}}_{i-1,j}}{2} \right] \right]$$

$$= KXY + P^{n+\frac{1}{2}}_{h_{i,j}} + P^{n+\frac{1}{2}}_{kv_{i,j}} - \epsilon^{n+\frac{1}{2}}_{i,j} \qquad (8.1)$$

where

$$UA1 = \left[\frac{U^{n+\frac{1}{2}}_{i,j} + U^{n+\frac{1}{2}}_{i-1,j}}{2} \right]$$

$$UA2 = \left[\frac{U^{n}_{i,j} + U^{n}_{i-1,j}}{2} \right]$$

$$VA1 = \left[\frac{V^{n+\frac{1}{2}}_{i,j} + V^{n+\frac{1}{2}}_{i,j-1}}{2} \right]$$

$$VA2 = \left[\frac{V^{n}_{i,j} + V^{n}_{i,j-1}}{2} \right]$$

$$KD = \left[\frac{k^{n}_{i,j+1} - k^{n}_{i,j-1}}{2} \right]$$

$$KXD1 = y_{\eta_{I,J}} \left[\frac{Kx^{n+\frac{1}{2}}_{i+1,j} - Kx^{n+\frac{1}{2}}_{i-1,j}}{2} \right]$$

$$KXD2 = -y_{\xi_{I,J}} \left[\frac{Kx^{n}_{i,j+1} - Kx^{n}_{i,j-1}}{2} \right]$$

$$KYD1 = x_{\xi_{I,J}} \left[\frac{Ky^{n}_{i,j+1} - Ky^{n}_{i,j-1}}{2} \right]$$

$$KYD2 = -x_{\eta_{I,J}} \left[\frac{Ky^{n+\frac{1}{2}}_{i+1,j} - Ky^{n+\frac{1}{2}}_{i-1,j}}{2} \right]$$

and

$$KXY = \frac{1}{JAC_{I,J}} \left[KXD1 + KXD2 + KYD1 + KYD2 \right] \quad .$$

The discretised k–equation for the ξ–direction release is rearranged to produce the tri–diagonal expression:

$$-a_{i,j}\, k^{n+\frac{1}{2}}_{i-1,j} + b_{i,j}\, k^{n+\frac{1}{2}}_{i,j} - c_{i,j}\, k^{n+\frac{1}{2}}_{i+1,j} = e_{i,j} \qquad (8.2)$$

where

$$a_{i,j} = -\frac{\Delta t}{4}\left[\frac{x_{\eta_{I,J}}}{JAC_{I,J}}.VA1 - \frac{y_{\eta_{I,J}}}{JAC_{I,J}}.UA1\right] \quad ,$$

$$b_{i,j} = 1.0 \quad ,$$

$$c_{i,j} = -\frac{\Delta t}{4}\left[\frac{y_{\eta_{I,J}}}{JAC_{I,J}}.UA1 - \frac{x_{\eta_{I,J}}}{JAC_{I,J}}.VA1\right] = -a_{i,j} \quad ,$$

and

$$e_{i,j} = k^{n}_{i,j} + \frac{\Delta t}{2}\left[\frac{y_{\xi_{I,J}}}{JAC_{I,J}}.UA2.KD - \frac{x_{\xi_{I,J}}}{JAC_{I,J}}.VA2.KD + KXY + P^{n+\frac{1}{2}}_{h_{i,j}} + P^{n+\frac{1}{2}}_{kv_{i,j}} - \epsilon^{n+\frac{1}{2}}_{i,j}\right] \quad .$$

The right-hand side of equation (8.2) contains values of k and ϵ at the advanced time stage, $t=(n+\frac{1}{2})\Delta t$. Furthermore, the coefficients, $a_{i,j}$ and $c_{i,j}$ contain advanced U- and V-velocity components, which are also unknown at the start of the time step. Consequently, in order to improve the time accuracy of the computational scheme, the user has the option of determining the advanced *non-implicit* variables via an iterative procedure; the principles of this technique have already been fully described in Section 6.6 and illustrated in Figure 6.13.

Applying equation (8.2) to each scalar variable position along a given η-line, produces the tri-diagonal matrix system:

$$\begin{bmatrix} b_{2,j} & -c_{2,j} & & & & \\ -a_{3,j} & b_{3,j} & -c_{3,j} & & & \\ & -a_{4,j} & b_{4,j} & -c_{4,j} & & \\ & & \ddots & \ddots & \ddots & \\ & & & -a_{n_i-2,j} & b_{n_i-2,j} & -c_{n_i-2,j} \\ & & & & -a_{n_i-1,j} & b_{n_i-1,j} \end{bmatrix} \begin{Bmatrix} k^{n+\frac{1}{2}}_{2,j} \\ k^{n+\frac{1}{2}}_{3,j} \\ k^{n+\frac{1}{2}}_{4,j} \\ \vdots \\ k^{n+\frac{1}{2}}_{n_i-2,j} \\ k^{n+\frac{1}{2}}_{n_i-1,j} \end{Bmatrix} = \begin{Bmatrix} E_{2,j} \\ E_{3,j} \\ E_{4,j} \\ \vdots \\ E_{n_i-2,j} \\ E_{n_i-1,j} \end{Bmatrix} \qquad (8.3)$$

where $E_{2,j} = e_{2,j} + a_{2,j}.k_{1,j}^{n+\frac{1}{2}}$

$E_{3,j} = e_{3,j}$

$E_{4,j} = e_{4,j}$

$\vdots$

$E_{n_i-2,j} = e_{n_i-2,j}$

and $E_{n_i-1,j} = e_{n_i-1,j} + c_{n_i-1,j}.k_{n_i,j}^{n+\frac{1}{2}}$.

The intermediate turbulent kinetic energy values, $k^{n+\frac{1}{2}}$ are thus obtained by solving equation (8.3) using Gaussian elimination for j = 2, 3, ..., n_j-1. A similar technique is utilised for the evaluation of the viscous dissipation rate, $\epsilon^{n+\frac{1}{2}}$.

The tri-diagonal system in equation (8.3) assumes that the unknown turbulent variables are required between boundary walls at i=1 and i=n_i. However, as explained in Section 6.6.2, the boundary-fitted code allows the outline of the transformed (ξ,η) mesh to be of any shape and consequently the left-hand perimeter subscript is not necessarily i=1. In addition, the A.D.I. algorithms were developed to allow an arbitrary number of implicit line-wise flow segments along each η-line, thereby giving the numerical model the ability to cater for complex grid systems.

(b) STEP 2 : η-direction release : $t=(n+\frac{1}{2})\Delta t \rightarrow t=(n+1)\Delta t$

The η-direction release is discretised in an analogous manner to that shown above for the ξ-direction. In this case, η-derivatives are expressed at the new time level, $t=(n+1)\Delta t$, whereas gradients with respect to ξ and non-derivative expressions (e.g. turbulence production and dissipation terms) are held at the old time increment, $t=(n+\frac{1}{2})\Delta t$. Since the discretisation of the 2nd A.D.I. step is very similar to the scheme presented in Section 8.2.1a, it is unnecessary to describe the algorithms in detail.

8.2.2 SCHEME 2 : Explicit discretisation using 2nd order upwind differences

Unlike the A.D.I. scheme, discussed in the previous section, the explicit

discretisation procedure advances the turbulent kinetic energy, k, and the dissipation rate, ϵ, to the new time level of $t=(n+1)\Delta t$ using a *single* increment of time. The diffusive turbulence transport expressions are discretised in a similar manner to that presented earlier in Section 8.2.1, whereas the advective components are approximated using a combination of second order upwind and weighted central differences to increase the stability of the numerical scheme (cf., Section 6.4). Thus, the individual terms of the transformed k-equation are formulated as

$$\frac{\partial k}{\partial t} = \frac{k^{n+1}_{i,j} - k^{n}_{i,j}}{\Delta t}$$

$$y_\eta \; U \; \frac{\partial k}{\partial \xi} = y_{\eta\, I,J}\left[\frac{U^{n}_{i,j} + U^{n}_{i-1,j}}{2} \right].QKXI$$

.... QKXI is the quadratic upwind difference approximation to $\partial k/\partial \xi$, (see later),

$$y_\xi \; U \; \frac{\partial k}{\partial \eta} = y_{\xi\, I,J}\left[\frac{U^{n}_{i,j} + U^{n}_{i-1,j}}{2} \right].WKETA$$

.... WKETA is the weighted central difference approximation to $\partial k/\partial \eta$,

$$x_\xi \; V \; \frac{\partial k}{\partial \eta} = x_{\xi\, I,J}\left[\frac{V^{n}_{i,j} + V^{n}_{i,j-1}}{2} \right].QKETA$$

.... QKETA is the quadratic upwind difference approximation to $\partial k/\partial \eta$,

$$x_\eta \; V \; \frac{\partial k}{\partial \xi} = x_{\eta\, I,J}\left[\frac{V^{n}_{i,j} + V^{n}_{i,j-1}}{2} \right].WKXI$$

.... WKXI is the weighted central difference approximation to $\partial k/\partial \xi$,

$$y_\eta \; \frac{\partial Kx}{\partial \xi} = y_{\eta\, I,J}\left[\frac{Kx^{n}_{i+1,j} - Kx^{n}_{i-1,j}}{2} \right]$$

$$y_\xi \; \frac{\partial Kx}{\partial \eta} = y_{\xi\, I,J}\left[\frac{Kx^{n}_{i,j+1} - Kx^{n}_{i,j-1}}{2} \right]$$

$$x_\xi \frac{\partial Ky}{\partial \eta} = x_{\xi_{I,J}} \left[\frac{Ky^n_{i,j+1} - Ky^n_{i,j-1}}{2} \right]$$

and

$$x_\eta \frac{\partial Ky}{\partial \xi} = x_{\eta_{I,J}} \left[\frac{Ky^n_{i+1,j} - Ky^n_{i-1,j}}{2} \right] .$$

Hence the depth-averaged k-equation is discretised as

$$k^{n+1}_{i,j} = k^n_{i,j} + \Delta t \left[\frac{1}{JAC_{I,J}} \left[-y_{\eta_{I,J}} UA.QKXI + y_{\xi_{I,J}} UA.WKETA \right.\right.$$

$$\left.\left. - x_{\xi_{I,J}} VA.QKETA + x_{\eta_{I,J}} VA.WKXI \right] + KXY + P^n_{h_{i,j}} + P^n_{kv_{i,j}} - \epsilon^n_{i,j} \right] \qquad (8.4)$$

where

$$UA = \left[\frac{U^n_{i,j} + U^n_{i-1,j}}{2} \right]$$

$$VA = \left[\frac{V^n_{i,j} + V^n_{i,j-1}}{2} \right]$$

$$QKXI = \begin{cases} \frac{3}{2} k^n_{i,j} - 2 k^n_{i-1,j} + \frac{1}{2} k^n_{i-2,j} & \text{if } UA > 0 \\ -\frac{3}{2} k^n_{i,j} + 2 k^n_{i+1,j} - \frac{1}{2} k^n_{i+2,j} & \text{if } UA < 0 \end{cases}$$

... quadratic upwind difference approximation to $\partial k/\partial \xi$,

$$WKETA = \left[\frac{k^n_{i,j+2} - k^n_{i,j-2} + 4 (k^n_{i,j+1} - k^n_{i,j-1})}{12} \right]$$

... weighted central difference approximation to $\partial k/\partial \eta$,

$$QKETA = \begin{cases} \frac{3}{2} k^n_{i,j} - 2 k^n_{i,j-1} + \frac{1}{2} k^n_{i,j-2} & \text{if } VA > 0 \\ -\frac{3}{2} k^n_{i,j} + 2 k^n_{i,j+1} - \frac{1}{2} k^n_{i,j+2} & \text{if } VA < 0 \end{cases}$$

... quadratic upwind difference approximation to $\partial k/\partial \eta$,

$$WKXI = \left[\frac{k^n_{i+2,j} - k^n_{i-2,j} + 4 (k^n_{i+1,j} - k^n_{i-1,j})}{12} \right]$$

... weighted central difference approximation to $\partial k/\partial \xi$,

$$KXD1 = y_{\eta_{I,J}} \left[\frac{Kx^n_{i+1,j} - Kx^n_{i-1,j}}{2} \right]$$

$$KXD2 = -y_{\xi_{I,J}} \left[\frac{Kx^n_{i,j+1} - Kx^n_{i,j-1}}{2} \right]$$

$$KYD1 = x_{\xi_{I,J}} \left[\frac{Ky^n_{i,j+1} - Ky^n_{i,j-1}}{2} \right]$$

$$KYD2 = -x_{\eta_{I,J}} \left[\frac{Ky^n_{i+1,j} - Ky^n_{i-1,j}}{2} \right]$$

and

$$KXY = \frac{1}{JAC_{I,J}} \left[KXD1 + KXD2 + KYD1 + KYD2 \right] .$$

In contrast to the semi-implicit shallow water equation scheme, described in Chapter 6, and the implicit turbulence transport numerical procedures, shown in Section 8.2.1, all terms on the right-hand side of equation (8.4) are evaluated at the old time level, $t=n\Delta t$; i.e., the scheme employs forward time differencing.

The advective finite-difference expressions must be modified when nodes close to flow boundaries are considered. For example, the weighted central difference approximation to $\partial k/\partial \xi$ must be rewritten as a standard central difference operator whenever the expression references variables outside the hydraulic domain, i.e.,

$$WKXI = \frac{\partial k}{\partial \xi} = \left[\frac{k^n_{i+2,j} - k^n_{i-2,j} + 4\,(k^n_{i+1,j} - k^n_{i-1,j})}{12} \right] \qquad (8.5a)$$

is modified to

$$WKXI = \frac{k^n_{i+1,j} - k^n_{i-1,j}}{2} \qquad (8.5b)$$

when $k^n_{i+2,j}$ or $k^n_{i-2,j}$ lies outside the flow region.

In a similar manner, the quadratic upwind approximation to $\partial k/\partial \xi$ also needs to be reformulated whenever the equation for QKXI references a variable outside the flow field. For example, considering the scalar variable nodes adjacent to a 'left-hand/ξ=constant' boundary, then $k_{i-1,j}$ exists but $k_{i-2,j}$ is undefined. Therefore, the upwind expression for QKXI is recast as

$$QKXI = \begin{cases} k^n_{i,j} - k^n_{i-1,j} & \text{if } UA > 0 \\ -\frac{3}{2} k^n_{i,j} + 2 k^n_{i+1,j} - \frac{1}{2} k^n_{i+2,j} & \text{if } UA < 0 \end{cases} \qquad (8.6a)$$

Conversely, at nodes adjacent to 'right-hand/ξ=constant' perimeters, QKXI becomes

$$QKXI = \begin{cases} \frac{3}{2} k^n_{i,j} - 2 k^n_{i-1,j} + \frac{1}{2} k^n_{i-2,j} & \text{if } UA > 0 \\ k^n_{i+1,j} - k^n_{i,j} & \text{if } UA < 0 \end{cases} \qquad (8.6b)$$

Similar modifications are utilised for the η-derivative advective expressions at nodes adjacent to η=constant ('horizontal') boundaries.

8.2.3 Evaluation of the Kx and Ky terms

Both the A.D.I. and the explicit numerical scheme utilise Kx and Ky (and their ϵ-equation equivalents, Ex and Ey) to simplify the discretisation of the diffusive terms in the curvilinear turbulence transport equations. Apart from the different time levels employed by the two discretisation procedures, the finite-difference approximations to equations (5.95a), (5.95b), (5.95c) and (5.95d) are common to

both numerical schemes. With reference to the explicit formulation (at a time level of $t=n\Delta t$), the terms applicable to the k–equation are calculated from the following central difference approximations:

$$Kx^{n}_{i,j} = \frac{{}^{\nu}t^{n}_{i,j}}{\sigma_k} \cdot \frac{1}{JAC_{I,J}} \left[y_{\eta_{I,J}} KXI^{n} - y_{\xi_{I,J}} KETA^{n} \right] \qquad (8.7a)$$

and

$$Ky^{n}_{i,j} = \frac{{}^{\nu}t^{n}_{i,j}}{\sigma_k} \cdot \frac{1}{JAC_{I,J}} \left[x_{\xi_{I,J}} KETA^{n} - x_{\eta_{I,J}} KXI^{n} \right] \qquad (8.7b)$$

where

$$KXI^{n} = \left[\frac{k^{n}_{i+1,j} - k^{n}_{i-1,j}}{2} \right]$$

and

$$KETA^{n} = \left[\frac{k^{n}_{i,j+1} - k^{n}_{i,j-1}}{2} \right] \quad ,$$

whilst the ϵ–equation terms, Ex and Ey are discretised using an analogous procedure.

The k–derivatives (KXI and KETA), presented above, assume that the node is in the interior of the flow field, where central difference approximations can be utilised. At the boundaries of the hydraulic domain, however, the expressions for KXI and KETA must be reformulated as second–order forward/backward differences, in a similar fashion to the velocity derivatives employed in the evaluation of the effective stresses, T_{xx}, T_{xy} and T_{yy} (see Section 6.8). For example, at a scalar variable node, (i_w,j) on a 'left–hand/ξ=constant' wall of the computational plane, the k–derivative in the ξ–direction is written as a quadratic forward difference:

$$\frac{\partial k^{n}}{\partial \xi} = KXI^{n} = -\frac{3}{2} k^{n}_{i_w,j} + 2\, k^{n}_{i_w+1,j} - \frac{1}{2} k^{n}_{i_w+2,j} \qquad (8.8a)$$

whereas, on a 'right–hand/ξ=constant' boundary, a backward difference is utilised:

$$\frac{\partial k^{n}}{\partial \xi} = KXI^{n} = \frac{3}{2} k^{n}_{i_w,j} - 2\, k^{n}_{i_w-1,j} + \frac{1}{2} k^{n}_{i_w-2,j} \quad . \qquad (8.8b)$$

Similar quadratic expressions are implemented for the evaluation of $\partial k/\partial \eta$ along η=constant perimeters.

8.2.4 Discretisation of the turbulence production terms

(a) Horizontal production term, P_h

The generation of turbulent kinetic energy due to the interaction of the Reynolds stresses with the *horizontal* velocity gradients, is formulated in finite-differences as:

$$P_{h_{i,j}}^{n} = {}^{\nu}t_{i,j}^{n} \left[2 \left.\frac{\partial U^n}{\partial x}\right|_{i,j}^{2} + 2 \left.\frac{\partial V^n}{\partial y}\right|_{i,j}^{2} + \left[\left.\frac{\partial U^n}{\partial y}\right|_{i,j} + \left.\frac{\partial V^n}{\partial x}\right|_{i,j} \right]^2 \right] \tag{8.9}$$

where

$$\left.\begin{aligned}
\left.\frac{\partial U^n}{\partial x}\right|_{i,j} &= \frac{1}{JAC_{I,J}} \left[y_{\eta_{I,J}} UXI^n - y_{\xi_{I,J}} UETA^n \right] \\
\left.\frac{\partial V^n}{\partial x}\right|_{i,j} &= \frac{1}{JAC_{I,J}} \left[y_{\eta_{I,J}} VXI^n - y_{\xi_{I,J}} VETA^n \right] \\
\left.\frac{\partial U^n}{\partial y}\right|_{i,j} &= \frac{1}{JAC_{I,J}} \left[x_{\xi_{I,J}} UETA^n - x_{\eta_{I,J}} UXI^n \right] \\
\left.\frac{\partial V^n}{\partial y}\right|_{i,j} &= \frac{1}{JAC_{I,J}} \left[x_{\xi_{I,J}} VETA^n - x_{\eta_{I,J}} VXI^n \right]
\end{aligned}\right\} \tag{8.10}$$

and

$$\left.\begin{aligned}
UXI^n &= U_{i,j}^n - U_{i-1,j}^n \\
VETA^n &= V_{i,j}^n - V_{i,j-1}^n \\
UETA^n &= \left[\frac{U_{i-1,j+1}^n + U_{i,j+1}^n - U_{i,j-1}^n - U_{i-1,j-1}^n}{4} \right] \\
VXI^n &= \left[\frac{V_{i+1,j}^n + V_{i+1,j-1}^n - V_{i-1,j}^n - V_{i-1,j-1}^n}{4} \right]
\end{aligned}\right\} \tag{8.11}$$

The discretisation of the horizontal production term is therefore accomplished by

evaluating the velocity gradients in the computational (ξ,η) plane, transforming these derivatives into the Cartesian system and finally substituting $\partial U/\partial x$, $\partial V/\partial x$, $\partial U/\partial y$ and $\partial V/\partial y$ into the original Cartesian equation for P_h. It should be noted that the velocity derivatives in the curvilinear domain (i.e. equation 8.11) have different discretisation errors since $\partial U/\partial \xi$ and $\partial V/\partial \eta$ are approximated over distances of $\Delta\xi$ and $\Delta\eta$, whilst the other spatial derivatives, $\partial U/\partial \eta$ and $\partial V/\partial \xi$ are evaluated over $2\Delta\eta$ and $2\Delta\xi$, respectively. This is unavoidable due to the staggered cell layout (see Section 6.6.1b).

(b) Vertical production term, P_{kv}

The source term, P_{kv}, originating from the non-uniformity of the velocity profile in the vertical plane, is formulated in finite-differences as:

$$P^n_{kv_{i,j}} = c_k \frac{(U_*{}^n_{i,j})^3}{D^n_{i,j}} \qquad (8.12)$$

where

$$U_*{}^n_{i,j} = \left[c_f \left[\left[\frac{U^n_{i,j} + U^n_{i-1,j}}{2} \right]^2 + \left[\frac{V^n_{i,j} + V^n_{i,j-1}}{2} \right]^2 \right] \right]^{\frac{1}{2}} . \qquad (8.13)$$

8.2.5 Evaluation of the eddy viscosity

The depth-averaged $k-\epsilon$ Kolmogorov-Prandtl equation (5.58) is discretised at the scalar variable nodes of the staggered mesh:

$$\nu_t{}^n_{i,j} = c_\mu \frac{\left[k^n_{i,j} \right]^2}{\epsilon^n_{i,j}} \qquad . \qquad (8.14)$$

8.2.6 Boundary conditions

The usual method of calculating the values of k and ϵ along the solid perimeter boundaries is to use the so-called wall function technique, recommended by Launder and Spalding (1974). In the near wall region, beyond the viscous sublayer, k and ϵ

can be calculated from the following expressions (see Rastogi and Rodi, 1978; Rodi, 1984a):

$$k_w = \frac{U_*^2}{\sqrt{c_\mu}} \quad ; \quad \epsilon_w = \frac{U_*^3}{\kappa . y_w} \tag{8.15}$$

where U_* is the friction velocity in the near wall region,
y_w is the normal distance from the side wall,
and κ is the von Kármán constant (= 0.42).

The expressions shown above are usually employed in conjunction with the following universal law of the wall:

$$\frac{U_{res}}{U_*} = \frac{1}{\kappa} \ln (Ey^+) \tag{8.16}$$

where

U_{res} is the resultant velocity parallel to the wall,
$y^+ = yU_*/\nu$,
and E is a parameter accounting for the wall roughness
($\approx$ 9 for hydraulically smooth walls).

Equation (8.16) is only applicable for $30 < y_w^+ < 100$, where y_w^+ is the non-dimensionalised distance to the wall, and therefore the numerical grid must have sufficient resolution to ensure that the boundary nodes lie within the stipulated zone.

Bernard (1989b) implemented a boundary-fitted non-orthogonal Navier-Stokes equation solver and chose an alternative method of computing the friction velocity at the boundary wall by defining an empirical relationship of the form:

$$U_* = \left[\frac{1}{2} C_d \left[U_b^2 + V_b^2 \right] \right]^{\frac{1}{2}} \tag{8.17}$$

where C_d is a side wall drag coefficient,
U_b is the x-velocity component,
and V_b is the y-velocity component adjacent to the side wall.

However, Bernard implied that the grid resolution employed to model real-life flow domains was generally too coarse for the wall function method to be effective. He proposed that it is usually sufficient to determine the boundary values of k and ϵ

using Neumann (zero normal derivative) conditions, and consequently, this is the procedure adopted in the present study. The decision to utilise zero normal derivatives for the turbulence variables ties in with the Cartesian computational schemes developed by Walker (1987) and Dong (1987). Setting $\partial k/\partial n$ to zero is theoretically acceptable in low Reynolds number flow regimes, but the specification of $\partial \epsilon/\partial n = 0$ cannot be justified from physical principles (Younis, 1990). Nevertheless, the results presented by both Walker and Dong indicate that the approximate boundary conditions are at least capable of producing acceptable flow simulations in the nearshore environment. Following Bernard (1989b), Neumann derivative conditions are also stipulated at the inlet and outlet boundaries, thereby allowing the distributions of k and ϵ at the inflow and outflow to develop as part of the numerical simulation.

The directional derivative of a scalar quantity, f, normal to a line of constant ξ can be expressed as (see Thompson et al., 1977b or Johnson and Thompson, 1978):

$$\frac{\partial f}{\partial n(\xi)} = \frac{1}{J\sqrt{\alpha}} (\alpha f_\xi - \beta f_\eta) \quad , \tag{8.18a}$$

whereas the directional derivative normal to a line of constant η is shown to be:

$$\frac{\partial f}{\partial n(\eta)} = \frac{1}{J\sqrt{\gamma}} (\gamma f_\eta - \beta f_\xi) \quad , \tag{8.18b}$$

where

$$\alpha = x_\eta^2 + y_\eta^2$$

$$\beta = x_\xi x_\eta + y_\xi y_\eta$$

$$\gamma = x_\xi^2 + y_\xi^2$$

and J = Jacobian $= x_\xi y_\eta - x_\eta y_\xi$ as before.

Consider the scalar-variable boundary node (i_w,j) on a 'left-hand/ξ=constant' perimeter, as illustrated in Figure 8.1. The Neumann derivative constraint for the turbulent kinetic energy at (i_w,j) is expressed as

$$\left.\frac{\partial k}{\partial \xi}\right|_{i_w,j} = \frac{\beta_{I,J}}{\alpha_{I,J}} \cdot \left.\frac{\partial k}{\partial \eta}\right|_{i_w,j} \quad . \tag{8.19}$$

Using a central difference approximation for $\partial k/\partial \eta$ and a quadratic forward difference operator for the ξ-derivative leads to the following discretised boundary condition:

$$- \frac{3}{2} k_{i_w,j} + 2 k_{i_w+1,j} - \frac{1}{2} k_{i_w+2,j} = \frac{\beta_{I,J}}{\alpha_{I,J}} \left[\frac{k_{i_w,j+1} - k_{i_w,j-1}}{2} \right]. \quad (8.20)$$

The present computational schemes use an explicit iterative technique to calculate the values of the turbulence variables along the boundaries, i.e., equation (8.20) is rearranged to give

$$k_{i_w,j} = \frac{2}{3} \left[2 k_{i_w+1,j} - \frac{1}{2} k_{i_w+2,j} - \frac{\beta_{I,J}}{\alpha_{I,J}} \left[\frac{k_{i_w,j+1} - k_{i_w,j-1}}{2} \right] \right] . \quad (8.21)$$

Conversely, along 'right-hand/ξ=constant' walls, the zero normal k-derivative condition is discretised as:

$$k_{i_w,j} = \frac{2}{3} \left[2 k_{i_w-1,j} - \frac{1}{2} k_{i_w-2,j} + \frac{\beta_{I,J}}{\alpha_{I,J}} \left[\frac{k_{i_w,j+1} - k_{i_w,j-1}}{2} \right] \right] . \quad (8.22)$$

Similar finite-difference expressions are utilised for the Neumann boundary conditions along perimeters formed by lines of constant η.

Equations (8.21), (8.22) and their η=constant counterparts are solved by repeatedly sweeping the perimeter of the boundary-fitted mesh. The convergence of the boundary turbulence variables during the early stages of a numerical simulation usually requires at least ten complete cyclical iterations per time step. Fortunately, the computational costs of setting $\partial k/\partial n=0$ and $\partial \epsilon/\partial n=0$ are small in comparison with the computer time used to evaluate the turbulence variables at the interior mesh nodes.

8.2.7 Corner nodes

The explicit finite-difference boundary equations described in the previous section cannot be utilised at the special nodes of the circular reservoir geometries (illustrated schematically in Figure 6.9) on account of the breakdown in the validity

of the transformation derivatives at the 'concave' corner points and the ambiguity in deciding the direction of the normal derivative at the 'convex' corner positions. Consequently, the computational procedures adopt the same interpolative/extrapolative methods used to determine the surface elevations at the corner points. For example, at the intersection between the circular perimeter and the inlet/outlet stems (Figure 6.9a), the turbulent kinetic energy is found from a two-point interpolation in the flow direction, i.e.

$$k_{i_w, j_w} = \frac{k_{i_w-1, j_w} + k_{i_w+1, j_w}}{2} \quad . \tag{8.23}$$

At the 'concave' singularity position, shown in Figure 6.9b, the turbulence variables are determined by extrapolating along the ξ=constant and η=constant perimeter walls:

$$\left.\begin{aligned} k_1 &= 2\, k_{i_w, j_w-1} - k_{i_w, j_w-2} \\ k_2 &= 2\, k_{i_w+1, j_w} - k_{i_w+2, j_w} \end{aligned}\right\} \tag{8.24}$$

and $$k_{i_w, j_w} = \frac{k_1 + k_2}{2} \quad .$$

Analogous interpolation techniques are utilised for the evaluation of the viscous dissipation rate, ϵ.

8.3 Results

The computational schemes, proposed in Section 8.2, were tested initially using the 61 x 61 node (30 x 30 flow-variable cell) symmetrical circular reservoir boundary-fitted system, shown in Figure 7.9. To minimise the computational costs of the preliminary studies, the k-ϵ discretisation procedures were implemented using the steady-state velocity and surface elevation fields determined from the zero-equation turbulence simulations, described in Chapter 7. This allowed the number of unknown variables per flow cell to be reduced from five (U, V, ζ, k and ϵ) to two (k and ϵ). The initial k-ϵ computations were thus based upon the steady-state velocity and surface elevation fields of the constant (ν_t = 0.000784 m^2/s) eddy viscosity flow prediction, shown in Figure 7.10.

To prevent the occurrence of zero eddy viscosity at the start of the flow

computation, the initial turbulence variables were chosen as $k = \epsilon = 5 \times 10^{-4}$ - (corresponding to an eddy viscosity of 4.5×10^{-5} m^2/s). The temporal filter coefficients were initially set to the values recommended by Butler (1978a), i.e., (a, b, c) = (0.6, 0.3, 0.1). Numerical experimentation revealed that both discretisation techniques, proposed in Section 8.2, suffered from severe destabilising effects at the intersection points between the parallel sided flow channels and the circular perimeter. It was found that the discontinuous boundary conditions at the convex corner nodes created severe numerical difficulties in the vicinity of the inlet stem. Both computational schemes failed to reach steady-state; instead, spurious contours of k and ϵ began to grow from the nodes downstream of the convex corner points. The onset of instability, for the implicit finite-difference procedure, is demonstrated in Figure 8.2 which illustrates the distribution of k, the turbulent kinetic energy, 6.5 s after the start of the $k-\epsilon$ simulation. Similar contour patterns were observed in the ϵ-field.

Extensive numerical tests were conducted in an attempt to overcome the instabilities. These included:

(a) Increasing the initial values of k and ϵ to 5×10^{-3}.
(b) Increasing the damping of the temporal filter; i.e., the filter coefficients were changed to (a, b, c) = (0.2, 0.5, 0.3).
(c) Reducing the time step from 0.05 s to 0.01 s.
(d) Changing the interpolation technique for determining the values of k and ϵ at the 'convex' corner nodes (i.e., interpolate in the η-direction rather than in the ξ-direction).
(e) Increasing the resolution of the mesh by using the 121 x 121 node (60 x 60 flow-cell) circular reservoir coordinate system, depicted in Figure 7.20.
(f) Using the parabolic ramp function, presented in equation (7.6), to build up the velocity distribution over a period of time.

None of the above modifications were able to control the numerical instabilities, thus underlining the problems of specifying k and ϵ in the region of wall discontinuities. As a consequence, no further simulation of the turbulence transport equations in circular reservoir geometries was attempted; additional experimental information is required (outside the scope of this thesis) in order to provide a basis for a better theoretical model of the discontinuous boundary conditions.

In the second application, the depth-averaged curvilinear $k-\epsilon$ model was applied to the arbitrary river geometry, described in Section 7.5. As before, the turbulence

transport algorithms were separated from the shallow water equation solver, in order to reduce the computational costs of the investigation. The turbulence predictions were based upon the ν_t = 0.25 m^2/s flow patterns, illustrated in Figures 7.38c and 7.39c. A time step of 0.5 s was utilised, the temporal filter coefficients were set to (a, b, c) = (0.6, 0.3, 0.1) and the initial conditions were specified as k = ϵ = 1.0 x 10^{-5}. The central differenced implicit numerical scheme again suffered from an explosive growth in grid-scale oscillations after approximately 1200 time steps. However, the upwind differenced explicit discretisation procedure (SCHEME 2) converged to steady-state after about 3600 s of simulation time, thereby confirming the stabilising influence of the quadratic upwind difference terms.

Figures 8.3 and 8.4 present the steady-state contours of eddy viscosity, computed using SCHEME 2. The rapid throughflow jet is characterised by a large amount of turbulence, whereas the recirculation zones either side of the jet correspond to regions of low eddy viscosity. An investigation of the magnitude of ν_t at the inlet, revealed that the flow in the arbitrary river geometry is dominated by the bed generated turbulence created in the high velocity inflow region. For example, the depth-averaged k-ϵ model predicts an eddy viscosity of approximately 4.5 x 10^{-3} m^2/s at the centre of the inflow boundary; this can be compared against the bed generated turbulence formula presented by Rastogi and Rodi (1978):

$$\nu_{t_b} = 0.0765\ U_* D \qquad (8.25)$$

where D is the flow depth,
and U_* is the friction (or shear) velocity.

The arbitrary river flow simulations assumed a constant Chezy roughness coefficient of 45 $m^{\frac{1}{2}}/s$ across the entire hydraulic domain and, consequently, the bed friction factor was specified as

$$c_f = \frac{g}{C^2} = \frac{9.81}{45^2} = 4.84 \times 10^{-3} \quad . \qquad (8.26)$$

At the centre of the inlet jet, U $\approx$ 0.5 m/s and therefore the friction velocity was approximately

$$U_* = (4.84 \times 10^{-3} \times 0.5^2)^{\frac{1}{2}} = 0.035 \text{ m/s} \quad . \qquad (8.27)$$

The flow depth at the centre of the inlet jet was 1.6 m, and thus the bed generated eddy viscosity can be estimated (using equation 8.25) as

$$\nu_{t_b} = 0.0765 \times 0.035 \times 1.6 \approx 4.3 \times 10^{-3} \ m^2/s \quad . \qquad (8.28)$$

Since the depth-averaged k-ϵ model predicts a value of 4.5 x 10^{-3} m^2/s in the centre of the inflow jet, it can be concluded that the turbulence in the vicinity of the inlet is almost entirely created by bed roughness. This can also be verified by analysing the magnitudes of the various source terms in the depth-averaged k-ϵ transport equations. In the relatively slow moving flow regions, downstream of the inlet, the level of turbulence was found to be significantly greater than the distribution computed using equation (8.25). This can be attributed to the fact that the high levels of turbulence created in the inflow zone are transported by the flow to other parts of the hydraulic domain. The simulation therefore demonstrates the advantages of using a turbulence transport model, by illustrating that the generation and dissipation of turbulence are often likely to be associated with different regions.

Although the 'wiggles' in the eddy viscosity contours of Figures 8.3 and 8.4 are very similar to the grid-scale oscillations which occur in the early stages of non-linear instability, they are caused by an entirely different numerical process. Walker (1987) and Dong (1987) found that their finite-difference turbulence discretisations were unable to prevent the values of k and/or ϵ from becoming negative in certain areas of the flow. Consequently, their finite-difference schemes employed a base level for the turbulence variables whenever k or ϵ became negative. The same system is employed in the present curvilinear discretisation; the base level in this case being defined as k = ϵ = 1.0 x 10^{-5}. Whenever this mechanism is utilised, however, the eddy viscosity contours around the node are likely to become distorted. Fortunately, only a few grid positions in the outer edges of the throughflow jet are affected by the problem and, therefore, the eddy viscosity contours in the centre of the flow field are not seriously altered.

The final flow simulation, using the arbitrary river geometry, attempted to solve the depth-averaged k-ϵ transport expressions and the shallow water equations as the flow developed (i.e. five unknowns per flow cell). Severe grid-scale *surface elevation* oscillations, indicative of non-linear instabilities in the shallow water equation solver, completely overwhelmed the flow calculation after approximately 225 s of simulation time. In hindsight, this could have been predicted from a consideration of the cell Reynolds numbers in the inflow jet. The distribution of eddy viscosity for the combined (U, V, ζ, k and ϵ) simulation can be assumed to be similar to the turbulence predictions, shown in Figures 8.3 and 8.4. The cell Reynolds number at the centre of the inflow boundary, is therefore estimated as

$$R_c = \frac{U.\Delta x}{\nu_t} \approx \frac{0.5 \times 20}{4.5 \times 10^{-3}} = 2222 \quad . \qquad (8.29)$$

Referring back to Section 7.5, it can be seen that the curvilinear shallow water equation solver is only able to prevent the occurrence of non-linear instabilities when the cell Reynolds number is less than approximately 40, and consequently the unbounded growth in surface elevation oscillations is unavoidable in the present simulation.

The instabilities caused by the low values of eddy viscosity coefficient effectively eliminate the possibility of using the curvilinear shallow water equation solver (proposed in Chapter 6) in conjunction with the k-ϵ model. Instead, further investigations should be conducted into alternative numerical schemes for the solution of the shallow water equations, with the aim of increasing the stability of the (U, V, ζ) discretisation. As discussed in Section 7.6, more advanced schemes for the solution of the depth-averaged momentum and continuity equations involve the use of conservative hydrodynamic expressions, contravariant (or covariant) velocity components and the use of a conservative discretisation technique, such as the finite-volume method. However, none of these possibilities has yet been implemented in the boundary-fitted shallow water equation computer code.

A review of recent computational methods for simulating the shallow water equations has revealed some interesting points which are particularly relevant to the present dilemma concerning the estimation of the eddy viscosity coefficient. It was found that there are large disparities in the magnitude of ν_t employed by different researchers for apparently similar flow problems. For example, Falconer's more recent numerical scheme (Falconer and Owens, 1987) calculates the spatial variations of eddy viscosity from the following bed generated turbulence formula:

$$\nu_{t_b} = \frac{1.154\ g\ |V|\ D}{C^2} \qquad (8.30)$$

where $|V|$ is the magnitude of the velocity,

g is the acceleration due to gravity,

D is the flow depth,

and C is the Chezy roughness coefficient.

This expression assumes a one-seventh power law velocity distribution in the vertical plane, and produces a value of eddy viscosity in the same order of magnitude as

that predicted by the depth-averaged k-ϵ model. However, other researchers (e.g., Vreugdenhil and Wijbenga, 1982; Stelling, 1983; Wijbenga, 1985a and 1985b) have chosen to model flow phenomena in rivers and estuaries using large values of eddy viscosity, i.e. $\nu_t > 1.0$ m^2/s. It can therefore be concluded that the estimation of eddy viscosity (in models without a k-ϵ discretisation) is often more dependent upon the numerical solution technique than the physical momentum diffusion processes which are being simulated. For example, Falconer and Owens used a first order upwind difference technique (which introduces a significant amount of artificial diffusion) and were able to utilise low values of eddy viscosity. Stelling (1983) and Wijbenga (1985a, 1985b), on the other hand, utilised second order upwind differencing and found that large values of ν_t were necessary to prevent non-linear instabilities. The use of first order upwind differences to simulate flows where low values of eddy viscosity prevail cannot be justified, however, because of the increased level of artificial diffusion created by such schemes.

The results have shown that the curvilinear shallow water equation solver may be grossly unstable when it is used in conjunction with the depth-averaged k-ϵ model. There is, nevertheless, sufficient evidence in the literature to suggest that the non-orthogonal (U, V, ζ) scheme, even without a higher order turbulence closure, represents a considerable improvement on standard Cartesian finite-difference solutions of the shallow water equations.

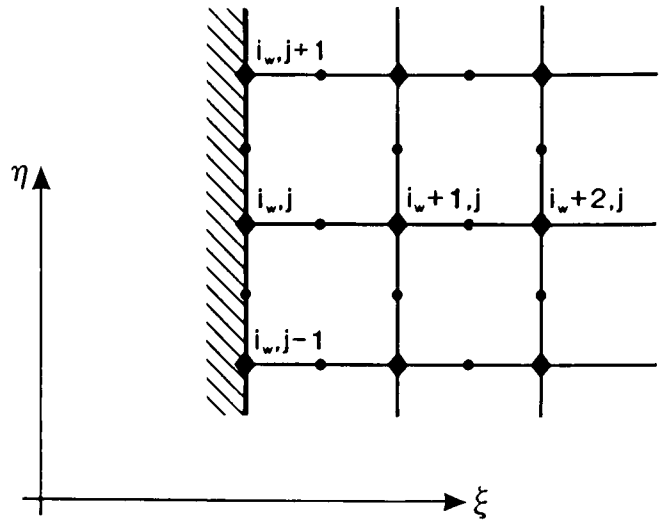
8.4 Conclusions and recommendations

This chapter has described two numerical schemes for discretising the depth-averaged k-ϵ turbulence transport equations on non-orthogonal boundary-fitted grids. Both models utilise finite-difference techniques and are based upon the staggered mesh arrangement proposed in Chapter 6. The first scheme modifies Walker's (1987) Cartesian k-ϵ computational procedures and employs an A.D.I. method in conjunction with central differences for the advective terms. However, the results show that the implicit scheme suffers from severe non-linear instabilities and is unable to reach steady-state for the cases considered here. This demonstrates that the use of an implicit discretisation technique by itself may not prevent the occurrence of destabilising effects.

The second approach utilises an explicit forward-time discretisation with second order upwind and weighted central differences for the non-linear advective terms. Although the scheme is still extremely unstable when modelling the circular reservoir

geometries of Chapter 7, the approach is able to predict the distributions of eddy viscosity in the arbitrary river geometry, detailed in Section 7.5. The extra stability of SCHEME 2 demonstrates that the advective terms of the $k-\epsilon$ equations should be discretised using upwind difference techniques. However, the eddy viscosities predicted by the $k-\epsilon$ approach are too small to enable the depth-averaged (U, V, ζ) simulation to be solved in conjunction with the turbulence transport model as the flow develops. As a result, a more stable shallow water equation solver needs to be developed before the $k-\epsilon$ model can predict the mass and momentum transport in general large-scale prototypical flow regimes.

The explicit finite-difference $k-\epsilon$ scheme is unable to prevent the turbulence variables from becoming negative at certain node positions, thereby demonstrating that the computational approach is not fully conservative. As a consequence, it is recommended that future turbulence studies are conducted using a non-orthogonal *finite-volume* $k-\epsilon$ discretisation (e.g., Rhie and Chow, 1982); this should conserve the turbulence quantities in a more exact manner.



KEY

- velocity positions
- ◆ scalar-variable positions

Figure 8.1 Section of transformed plane adjacent to a 'left-hand/ξ-constant' boundary

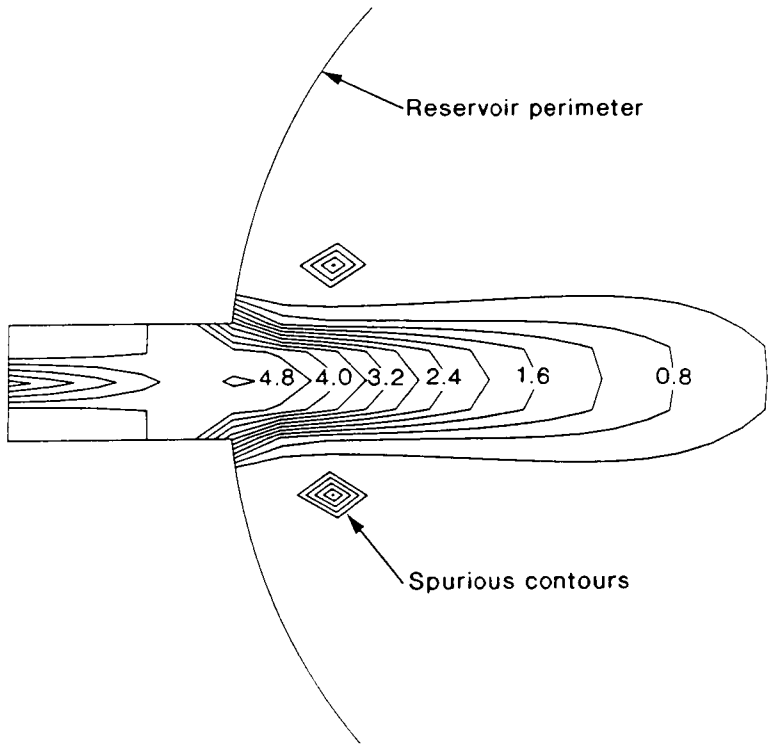


Figure 8.2 Contours of k in vicinity of circular reservoir inlet prior to corner point instability (m^2s^{-2} x 10^5)

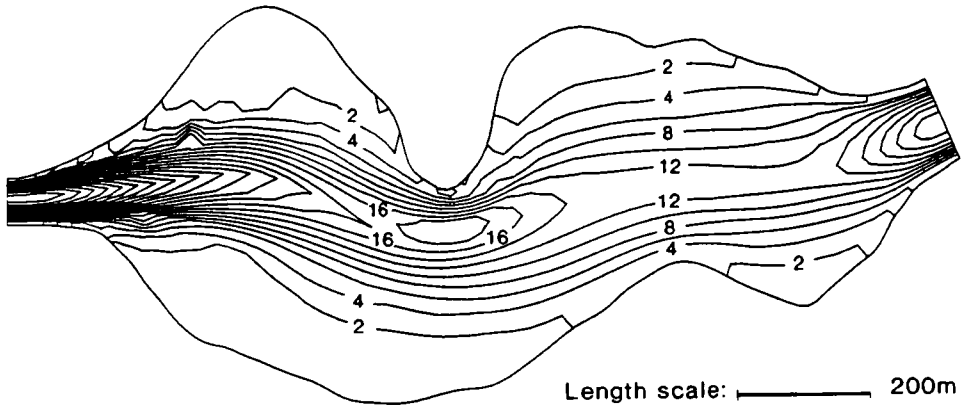


Figure 8.3 Eddy viscosity contours at steady-state: arbitrary river geometry ($m^2/s \times 10^4$)

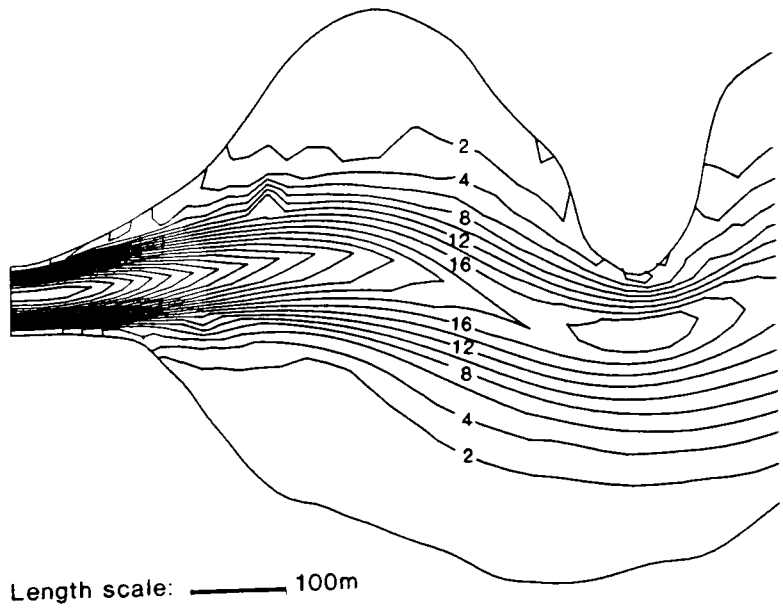


Figure 8.4 Detail of eddy viscosity contours in vicinity of inlet: arbitrary river geometry ($m^2/s \times 10^4$)

CHAPTER 9
CONCLUSIONS AND RECOMMENDATIONS

9.1 Introduction

The numerical studies described in this thesis are concerned with the prediction of momentum-driven recirculating flows in shallow bodies of water. Three separate computational procedures have been investigated for predicting mass and momentum transfer in nearly-horizontal flow regimes. These comprise a Biot-Savart discrete vortex model, a stream function/vorticity-transport (ψ,ω) finite-difference simulation and a boundary-fitted systems solution of the shallow water equations in non-orthogonal curvilinear coordinate form. For intercomparison and validation purposes, all three procedures were used to simulate jet-forced flow in a circular reservoir with vertical side walls. Benchmark data of radial flow into circular cylinders have been provided by the analytical solutions of Dennis (1974) and Mills (1977), and the numerical and experimental results presented by Falconer (1976, 1980).

9.2 Discrete vortex simulation

The discrete vortex model provided a qualitative method for determining the circulation patterns in cylindrical domains and was capable of predicting many of the important features of radial jet-forced reservoir flow. These included the alignment of the throughflow jet so that it passed more or less directly between the inlet and outlet, and the creation of two counter-rotating zones of recirculation either side of the main flow. Nevertheless, the inability of the approach to produce quantitative results, the empiricism in the choice of position for the introduction of the nascent vortices, the incorrect velocity distributions at the inlet/outlet openings and the inevitable tendency to chaos resulting in failure to achieve steady-state indicate that the method (as presented in Chapter 2) is not yet sufficiently refined to be of practical use in reservoir flow studies. Furthermore, the use of a 'direct summation' or Biot-Savart approach is computationally expensive, since the number of arithmetic operations per time step is approximately proportional to N^2 where N is the number of vortices. It is therefore recommended that future discrete vortex reservoir circulation studies should employ the vortex-in-cell (V.I.C.) method. This approach is described briefly in Section 2.1, and should be more suitable because of the reduced costs of increasing the number of elemental vortices used to *discretise* the vorticity field. Moreover, the V.I.C. technique is ideally suited to parallel computing

applications, whereby an array of n transputers may be used to undertake computations in n cells simultaneously. The procedure may be *farmed* so that as calculations for one cell are completed, the relevant transputer begins work on the next available cell. Furthermore, it is anticipated that the V.I.C. approach would be well suited to the boundary-fitted concept, proposed by Thompson et al. (1974 onwards). This would provide complete generality in the shape of the reservoir's outer perimeter.

9.3 Stream function/vorticity-transport (ψ,ω) simulation

The second numerical investigation utilised a stream function/vorticity-transport finite-difference procedure for solving the two-dimensional Navier-Stokes equations. Free surface and bed friction effects were not applicable (as for the discrete vortex model); instead, the simulation considered low Reynolds number laminar flows in an infinitely deep circular cylinder with a single inlet and a single outlet. The stream function/vorticity-transport equations were applied to a polar coordinate finite-difference mesh which was distorted in the radial direction to increase the grid resolution near the no-slip circular boundary walls. A central difference alternating-direction-implicit (A.D.I.) discretisation technique was employed to solve the transformed governing hydrodynamic equations. The flow predictions were in almost exact agreement with the stream function contours presented by Dennis (1974) using a series truncation method, and with the ψ-contours obtained by Mills (1977) from an iterative integral solution technique.

Although the semi-analytical approaches proposed by both Dennis and Mills were unable to model recirculation beyond an inlet Reynolds number of 10, the finite-difference scheme presented in Chapter 3 was able to simulate flows up to a Reynolds number of 200. Whilst this still appears to be insufficient for practical engineering applications, it must be remembered that the *effective* Reynolds number of real-life turbulent flows often turns out to be quite small if based upon the horizontal eddy viscosity coefficient, $\tilde{\nu}_t$, rather than the kinematic viscosity, ν (Officier et al., 1986). Consequently, the low Reynolds number investigations, presented in Chapter 3, have practical significance. Moreover, the predictions provided a valuable set of benchmark flow fields which were subsequently used to validate the boundary-fitted non-orthogonal shallow water equation solver.

The stream function/vorticity-transport simulations were also particularly useful in illustrating the changes which occur in momentum-driven recirculating flows as the

Reynolds number is increased. For the circular reservoir geometries considered here, the flow patterns did not exhibit any recirculation below an inlet Reynolds number of 2.5. Between Re_I=2.5 and Re_I=50, two gyres of increasing strength were formed either side of the throughflow, with the reattachment points gradually moving towards the outlet, as the Reynolds number was increased. By Re_I=100, two pairs of counter-rotating gyres were formed at steady-state, instead of the single pair observed in the lower Re_I simulations. The formation of the secondary pair of eddies was accompanied by a slight decrease in the circulation strength of the primary gyres. Another important physical phenomenon predicted by the stream function/vorticity-transport model was the onset of unsteadiness at an inlet Reynolds number of approximately 200. The shear layers either side of the throughflow jet began to roll up into vortices which were initially advected towards the outlet in a symmetrical manner. Discrepancies in the positions of the vortices soon became accentuated, causing the throughflow stream to waver. Instabilities in the numerical scheme occurred as the first vortex reached the outlet and disrupted the boundary layers in the outflow stem. This prevented a detailed investigation of the limit cycle properties of the oscillating throughflow jet.

9.4 Boundary-fitted shallow water equation solver

The inflexibility in specifying the shape of the outer perimeter of the flow domain in both the discrete vortex and stream function/vorticity-transport models led to the decision to adopt a boundary-fitted procedure for the remaining numerical investigations. It was judged that the flow predictions would also benefit from the use of a primitive variable shallow water equation formulation. This would allow the numerical scheme to simulate free surface effects, thereby permitting the model to simulate the hydrodynamic conditions in situations with significant variations in water surface elevation, i.e., long river reaches, tidal harbours or in service reservoirs which are subjected to diurnal water level changes caused by variations in demand. Furthermore, the primitive variable (U, V, ζ) approach has the advantage over stream function/vorticity-transport methods in that the model can easily be extended to cater for flows around internal obstacles (such as islands).

The first step in the application of a boundary-fitted system involves the generation of an appropriate curvilinear grid. An essential aim of the boundary-fitted concept is to produce a one-to-one (i.e., non-overlapping) curvilinear mesh system which has coordinate lines coincident with the perimeter, no matter how irregular the shape of the region. This eliminates many of the problems

associated with the interpolative/extrapolative techniques used in conventional Cartesian finite-difference hydrodynamic schemes when solving partial differential equations on irregularly shaped flow domains. Following Thompson et al.'s methodology (1974), the creation of the physical mesh was accomplished by the numerical solution of a pair of quasi-linear Poisson equations, using a successive-over-relaxation finite-difference procedure.

The geometrical data sets defining the outer perimeter of the circular reservoirs were specified analytically, but the computational scheme also allowed the boundary coordinate data to be prepared using a digitising tablet. This latter technique enabled scale drawings of irregularly shaped flow domains to be converted accurately into digitised perimeter coordinates suitable as cubic-spline data for the grid generation code. Although the computational costs of mesh generation were small in comparison to the costs of solving the governing hydrodynamic equations, it is recommended that a more efficient method should be developed for the solution of the elliptic mesh generation equations. The multigrid approach (Häuser et al., 1986a) may offer a considerable increase in the computational efficiency of the grid generation scheme. Even so, the author considers that the main area of development work should be directed towards the implementation of a multigrid technique for the solution of the shallow water equations. The present computational scheme for solving the transformed curvilinear depth-averaged Reynolds equations uses a *single grid* approach and requires approximately 320 minutes of C.P.U. time on a PRIME 9955 computer for the flow computations to reach steady-state with a 30 x 30 flow-variable circular reservoir geometry. On the other hand, the generation of the grid only requires about 15 minutes of C.P.U. time. Therefore, it is in the computationally expensive area of flow simulation where the use of a multigrid technique would be most beneficial.

In the second stage of the boundary-fitted procedure, the governing Cartesian hydrodynamic expressions were converted for use with non-orthogonal curvilinear coordinate systems and were then discretised on a staggered mesh using a semi-implicit alternating-direction finite-difference scheme. The computational algorithms proposed to solve the transformed shallow water equations were comprehensively validated against analytical wind-induced surface elevation effects in circular and elliptical wedge-shaped basins, open channel flow in deliberately distorted non-orthogonal meshes and by comparison with alternative numerical schemes for the prediction of jet-forced flow in circular reservoirs (i.e., the two-dimensional stream function/vorticity-transport model and the Cartesian shallow water equation solver presented by Falconer, 1976 and 1980). In particular, the excellent agreement

between the jet-forced flow computations of the primitive variable boundary-fitted technique and the stream function/vorticity-transport scheme indicates that the second order upwind differencing, in the (U, V, ζ) approach, introduces very little additional artificial diffusion above that due to the A.D.I. technique itself. Furthermore, the combination of the reservoir's circular perimeter and the parallel sided flow channels illustrates that the boundary-fitted methodology is able to circumvent the requirement to employ finite-element methods whenever conditions in irregular shaped flow domains are modelled. This is also graphically demonstrated by the use of the boundary-fitted finite-difference hydrodynamic scheme for predicting velocities in a highly contorted flow geometry.

Most of the validation tests have been concerned with the prediction of steady-state jet-forced flow in circular reservoirs with vertical side walls. This example could be of interest to water resource engineers who wish to avoid problems arising from areas of stagnation by optimising the positions of reservoir inlets and outlets so as to mix the flow as much as possible. However, the ability of the computational scheme to model free surface phenomena allows the non-orthogonal shallow water equation solver to predict the depth-averaged flow fields in many other hydraulic applications, for example, in river channels and tidal harbours. The ability of the model to simulate flows around irregular curved boundaries represents a considerable improvement over more conventional Cartesian finite-difference shallow water equation solvers.

9.5 k-ϵ Model

Chapter 8 has presented two finite-difference schemes for discretising the depth-averaged k-ϵ transport equations on curvilinear boundary-fitted grids. The results demonstrate that the implementation of an alternating-direction-implicit discretisation technique with central differences for the advective terms cannot prevent the occurrence of non-linear grid scale oscillations in the k- and ϵ- fields. An explicit forward time finite-difference scheme, using second order upwind and weighted central differences for the advective components, proved to be a more stable numerical approach. However, it was discovered that the explicit k-ϵ scheme was unable to prevent the turbulence variables from becoming negative at certain node positions, thereby demonstrating that the discretisation procedure was not fully conservative. Consequently, it is recommended that future turbulence studies are modelled using a non-orthogonal finite-volume scheme. It was also discovered that the k-ϵ expressions tended to predict values of eddy viscosity which were too small

to be handled by the shallow water equation solver. This latter finding has major implications since it forces the present boundary-fitted (U, V, ζ) scheme to model turbulence using either the assumption of a constant eddy viscosity or the depth-averaged algebraic relationship presented in equation (5.43). It is therefore recommended that future research be directed towards a non-orthogonal shallow water equation solver which is stable at higher cell Reynolds numbers; this would enable lower values of eddy viscosity to be implemented.

9.6 Overall recommendations

The proposed non-orthogonal scheme, detailed in Chapter 6, utilises the *advective* form of the momentum expressions in conjunction with the non-conservative geometrical derivative relationships presented by Thompson et al. (1974), and therefore, represents the most basic form of non-orthogonal boundary-fitted approach. A number of alternative procedures for solving the depth-averaged Reynolds equations on curvilinear meshes have already been described in Chapters 5 and 6. These include:

(a) Using the conservative momentum and geometric derivative equations.
(b) The use of contravariant or covariant velocities as the primitive variables, in place of Cartesian components.
(c) Employing a better solution technique such as the multigrid method or the strongly-implicit-procedure.
(d) The use of collocated meshes instead of a staggered variable arrangement.

Extensive numerical experimentation should therefore be conducted to test the feasibility of each of these options. As stated in Section 7.6, the large number of possible combinations for the implementation of a boundary-fitted finite-difference hydrodynamic scheme suggests that it will be some time before a *definitive* solution procedure is available for solving the shallow water equations on non-orthogonal meshes. The author's personal view is that a staggered multigrid shallow water equation solver would considerably reduce the computational costs of the (U, V, ζ) scheme. This, in turn, could allow increased mesh resolutions to be used, thereby reducing the spatial discretisation errors of the flow solutions.

The boundary-fitted shallow water equation solver described in Chapters 6, 7 and 8 is only in the early stages of development and requires additional work before it could be of general use in the strategic management of large-scale flow situations.

In order for the numerical scheme to have a wider applicability, the boundary-fitted technique should be capable of modelling hydraulic regimes where the cross-sectional profile of the flow depends upon the surface elevation (e.g., in estuarial areas). Conventional techniques for simulating changes in wetted plan-form area using flooding/drying cells (e.g., Leendertse and Gritton, 1971; Falconer and Owens, 1987) often produce troublesome oscillations in water surface elevations at mesh points near the perimeter of the flow field. Boundary-fitted coordinate systems, however, offer the possibility of utilising a time-dependent mesh, which continuously adapts to the changing shape of the wetted perimeter. This would completely eliminate the need to use flooding/drying methods and may provide a more elegant solution technique for modelling hydraulic conditions in tidal regimes. The development of an adaptive time-dependent boundary-fitted solution procedure would entail rewriting the governing hydrodynamic expressions in terms of a moving coordinate frame, instead of the static frame used in the current investigation. Furthermore, a time-dependent boundary which is controlled by the surface elevation field would require the mesh generation equations to be solved at every time step of the flow development. It is therefore imperative that a fast grid generation technique is implemented before attempting an adaptive boundary-fitted shallow water equation discretisation.

It is also recommended that the curvilinear systems model should be extended to include a depth-averaged species equation for water quality/pollutant transport simulations. In addition, multi-layered boundary-fitted hydrodynamic modelling could be developed for sediment transport studies. It is felt that non-orthogonal curvilinear discretisation techniques could also be of considerable value in coastal engineering applications where the ability to simulate wave-induced flow conditions around curved headlands and bays would be an asset. In this latter case, wave-current interaction would have to be considered via the development of additional boundary-fitted computer routines to solve the wave dispersion, wave ray and energy equations. Finally, research should also be conducted into the vectorisation of the shallow water equation code for use on computers which have parallel processing capabilities.

The numerical investigations, presented in this thesis, indicate that the boundary-fitted shallow water equation approach offers a flexible computational method for predicting the hydrodynamic conditions in awkwardly shaped flow regimes. It is envisaged that the technique could therefore provide water resource engineers with a powerful alternative to finite-element or patched/nested finite-difference methods. However, field data is required in order to assess fully the merits and drawbacks of boundary-fitted approaches for solving the shallow water equations.

Moreover, there is a need for a coherent series of benchmark tests to be designed in order to judge the applicability of shallow water equation codes since this would enable such programs to be employed with confidence and help avoid use in inappropriate situations.

REFERENCES

Abbott, M.B., Damsgaard, A. and Rodenhuis, G.S. (1973) "System 21, 'Jupiter' (A design system for two-dimensional nearly-horizontal flows)", J. of Hydraulic Research, Vol. 11(1), pp. 1-28.

Abernathy, F.H. and Kronauer, R.E. (1962) "The formation of vortex streets", J. Fluid Mech., Vol. 13, pp. 1-20.

Ali, K.H.M. (1983) "The investigation of reservoir circulation", Modelling of Environmental Flow Systems, Winter Annual Meeting of the ASME, Boston, Massachusetts, Nov. 1983, pp. 25-34.

Ali, K.H.M. (1985) "Reservoir circulation caused by jets, wind and by the earth's rotation", Int. Symp. on Modelling Environmental Flows, Albuquerque, New Mexico, June 1985, pp. 71-80.

Ali, K.H.M. and Pateman, D. (1981) "Prediction of the circulation in reservoirs", Proc. Inst. Civ. Engrs., Part 2, Vol. 71, pp. 427-461.

Ali, K.H.M., Hedges, T.S. and Whittington, R.B. (1978) "A scale model investigation of the circulation in reservoirs", Proc. Inst. Civ. Engrs., Part 2, Vol. 65, pp. 129-161.

Anderson, D.A. (1983) "Adaptive mesh schemes based on grid speeds", AIAA Paper 83-1931, Danvers, Mass.

Audunson, T., Mathisen, J.P., Naeser, H. and McClimans, T. (1975) "Comparison between physical and mathematical modelling of a tidal fjord system in northern Norway", Symp. on Modelling Techniques, 2nd Annual Symp. of the Waterways, Harbours and Coastal Engineering Division of the ASCE, pp. 1291-1310.

Aziz, K. and Hellums, J.D. (1967) "Numerical solution of the three dimensional equations of motion for laminar natural convection", The Physics of Fluids, Vol. 10, No. 2, pp. 314-324.

Baker, A.J. (1985) "Finite Element Computational Fluid Mechanics", McGraw-Hill.

Baker, G.R. (1979) "The 'Cloud in Cell' technique applied to the roll up of vortex sheets", J. Comp. Physics, Vol. 31, pp. 76-95.

Ball, D.J., Penoyre, R.B.S. and O'Connor, B.A. (1982) "Numerical modelling of the hydrodynamics of complex civil engineering structures", Int. Conf. on the Hydraulic Modelling of Civil Engineering Structures, Coventry, England, Paper A2, pp. 13-32.

Basco, D.R. (1983) "Surfzone currents", Coastal Engineering, Vol. 7, pp. 331-355.

Bauer, S.W. and Schmidt, K.D. (1983) "Irregular-grid finite-difference simulation of Lake Geneva surge", J. of Hydraulic Engineering, Proc. ASCE, Vol. 109, No. 10, pp. 1285-1297.

Benque, J.P., Cunge, J.A., Feuillet, J., Hauguel, A. and Holly, F.M. (1982) "New method for tidal current computation", J. Waterway, Port, Coastal and Ocean Div., Proc. ASCE, Vol. 108, No. WW3, pp. 396-417.

Bernard, R.S. (1989a) "Explicit numerical algorithm for modelling incompressible approach flow", Tech. Report REMR-HY-5, U.S. Army Engineer Waterways Experiment Station Hydraulics Laboratory, Vicksburg, Mississippi, U.S.A.

Bernard, R.S. (1989b) "STREMR: numerical model for depth-averaged incompressible flow", Draft Report - Sept. 1989, U.S. Army Engineer Waterways Experiment Station Hydraulics Laboratory, Vicksburg, Mississippi, U.S.A.

Bernard, R.S. (1989c) Personal communication with Dr. R.S. Bernard, Hydraulics Laboratory, Mathematical Modelling Group, US Army Engineer Waterways Experiment Station, Vicksburg, Mississippi, U.S.A.

Boericke, R.R. and Hall, D.W. (1974) "Hydraulics and thermal dispersion in an irregular estuary", J. Hydraulics Div., Proc. ASCE, Vol. 100, No. HY1, pp. 85-102.

Borthwick, A.G.L. (1986a) "Comparison between two finite-difference schemes for computing the flow around a cylinder", Int. J. for Numerical Methods in Fluids, Vol. 6, No. 5, pp. 275-290.

Borthwick, A.G.L. (1986b) "Orbital flow past a cylinder: a numerical approach", Int. J. for Numerical Methods in Fluids, Vol. 6, No. 10, pp. 677-713.

Borthwick, A.G.L., Chaplin, J.R. and Ali, K.H.M. (1988) "Discrete vortex model of jet-forced flow in a circular reservoir", J. of Hydraulic Engineering (ASCE), Vol. 114, No. 3, pp. 283-298.

Bradshaw, P., Cebeci, T. and Whitelaw, J.H. (1981) "Engineering calculation methods for turbulent flow", Academic Press.

Briley, W.R. (1971) "A numerical study of laminar separation bubbles using the Navier-Stokes equations", J. Fluid Mech., Vol. 47, Part 4, pp. 713-736.

Butler, H.L. (1978a) "Numerical simulation of tidal hydrodynamics, Great Egg Harbour and Corson Inlets, New Jersey", Technical Report H-78-11, U.S. Army Engineer Waterways Experiment Station Hydraulics Laboratory, Vicksburg, Mississippi, U.S.A.

Butler, H.L. (1978b) "Coastal flood simulation in stretched coordinates", Vol. 1, Proc. 16th Coastal Eng. Conf. (ASCE), Hamburg, Germany, pp. 1030-1048.

Byrd, P.F. and Friedman, M.D. (1954) "Handbook of elliptic integrals for engineers and physicists", Springer.

Chaplin, J.R. (1973) "Computer model of vortex shedding from a cylinder", J. Hydraulics Div., Proc. ASCE, Vol. 99, No. HY1, pp. 155-165.

Chiang, W.-L. and Lee J.-J. (1982) "Simulation of large-scale circulation in harbours", J. Waterway, Port, Coastal and Ocean Div., Proc. ASCE, Vol. 108, No. WW1, pp. 17-31.

Chorin, A.J. (1973) "Numerical study of slightly viscous flow", J. Fluid Mech., Vol. 57, Part 4, pp. 785-796.

Chorin, A.J. and Bernard, P.S. (1973) "Discretisation of a vortex sheet, with an example of roll-up", J. Comp. Physics, Vol. 13, pp. 423-429.

Clements, R.R. and Maull, D.J. (1975) "The representation of sheets of vorticity by discrete vortices", Prog. Aerospace Sci., Vol. 16, No. 2, pp. 129-146.

Codell, R.B. (1975) "Two-dimensional model of flow toward intakes", Vol. 2, Symp. on Modelling Techniques (ASCE), San Francisco, Sept. 1975, pp. 426-438.

Coleman, R.M. (1982) "Generation of boundary-fitted coordinate systems using segmented computational regions", in Numerical Grid Generation, J.F. Thompson (editor), Proc. of a Symp. on the Numerical Generation of Curvilinear Coordinate Systems and their Use in the Numerical Solution of Partial Differential Equations, held at Nashville, Tennesse, U.S.A., pp. 633-651.

Conte, S.D. and de Boor, C. (1980) "Elementary numerical analysis : an algorithmic approach", 3rd Ed., McGraw-Hill.

Cooley, P. and Harris, S.L. (1954) "The prevention of stratification in reservoirs", J. Inst. Water Engrs., Vol. 8, No. 7, pp. 517-537.

Crowley, W.P. (1968) "Numerical advection experiments", Monthly Weather Review, Vol. 96, No. 1, pp. 1-11.

Dennis, S.C.R. (1974) "Application of the series truncation method to two-dimensional internal flows", Proc. 4th Int. Conf. on Numerical Methods in Fluid Dynamics, Univ. of Colorado, Springer, pp. 146-151.

Dong, P. (1987) "The computation of wave-induced circulations with wave-current interaction and refined turbulence modelling", Ph.D. Thesis, Department of Civil Engineering, Imperial College, London.

Douglas, J. (1955) "On the numerical integration of $\partial^2 u/\partial x^2 + \partial^2 u/\partial y^2 = \partial u/\partial t$ by implicit methods", J. of the Soc. of Industrial and Applied Math., Vol. 3, No. 1, pp. 42-65.

Elder, J.W. (1959) "The dispersion of marked fluid in turbulent shear flow", J. Fluid Mech., Vol. 5, pp. 544-560.

Falconer, R.A. (1976) "Mathematical modelling of jet-forced circulation in reservoirs and harbours", Ph.D. Thesis, Imperial College, London.

Falconer, R.A. (1980) "Numerical modelling of tidal circulation in harbours", J. Waterway, Port, Coastal and Ocean Div., Proc. ASCE, Vol. 106, WW1, pp. 31-48.

Falconer, R.A. (1984a) "A mathematical model study of the flushing characteristics of a shallow tidal bay", Proc. Inst. Civ. Engrs., Part 2, Vol. 77, pp. 311-332.

Falconer, R.A. (1984b) "Model study of jet-induced circulation and mixing in a reservoir", J. Inst. of Water Engrs. and Scientists, Vol. 38, No. 3, pp. 231-245.

Falconer, R.A. and Owens, P.H. (1987) "Numerical simulation of flooding and drying in a depth-averaged tidal flow model", Proc. Inst. Civ. Engrs., Part 2, Vol. 83, pp. 161-180.

Flokstra, C. (1977) "The closure problem for depth-averaged two-dimensional flow", Proc. 17th Congress of the IAHR, Baden-Baden, paper A106, pp. 247-256. [reprinted as - Publication No. 150, Delft Hydraulics Laboratory, Delft, The Netherlands.]

Flokstra, C. (1981) "Aspects of modelling horizontal momentum transfer in shallow flow", Report No. R1150, Delft Hydraulics Laboratory, Delft, The Netherlands.

Gerrard, J.H. (1967) "Numerical computation of the magnitude and frequency of the lift on a circular cylinder", Phil. Trans. Roy. Soc., (A)261, pp. 137-162.

Gilding, B.H. (1988) "A numerical grid generation technique", Computers & Fluids, Vol. 16, No. 1, pp. 47-58.

Gosman, A.D., Khalil, E.E. and Whitelaw, J.H. (1979) "The calculation of two-dimensional turbulent recirculating flows", in Turbulent Shear Flows I, Springer-Verlag, Berlin.

Gray, A.V. (1985) "Removal of phosphate at sewage treatment works and implications on phosphate loadings into Lough Neagh", J. Inst. Water Engrs. and Scientists, Vol. 39, No. 2, pp. 137-154.

Hamilton, J. (1979) "Finite-difference storm surge prediction", Deutsche Hydrographische Zeitschrift, Jahrgang 32, Heft 6 (Sonderdruck), pp. 267-278.

Hansen, W. (1956) "Theorie zur Errechnung des Wasserstandes und der Strömungen in Randmeeren nebst Anwendungen", Tellus, Vol. 8, No. 3, pp. 287-300.

Hansen, W. (1962) "Hydrodynamical methods applied to oceanographic problems", Proc. of the Symp. on Mathematical-Hydrodynamical Methods of Physical Oceanography, Institut für Meereskunde Der Universität Hamburg.

Häuser, J., Paap, H.G., Eppel, D. and Mueller, A. (1985) "Solution of shallow water equations for complex flow domains via boundary-fitted coordinates", Int. J. for Numerical Methods in Fluids, Vol. 5, pp. 727-744.

Häuser, J., Paap, H.G., Eppel, D. and Sengupta, S. (1986a) "Boundary conformed coordinate systems for selected two-dimensional fluid flow problems. Part 1: Generation of BFG's", Int. J. for Numerical Methods in Fluids, Vol. 6, pp. 507-527.

Häuser, J., Paap, H.G., Eppel, D. and Sengupta, S. (1986b) "Boundary conformed coordinate systems for selected two-dimensional fluid flow problems. Part 2: Application of the BFG method", Int. J. for Numerical Methods in Fluids, Vol. 6, pp. 529-539.

Heaps, N.S. (1969) "A two-dimensional numerical sea model", Phil. Trans. Roy. Soc., (A)265, pp. 93-137.

Heaps, N.S. (1977) "Development of storm-surge models at Bidston", Report No. 53, Institute of Oceanographic Sciences, Bidston.

Hinze, J.O., (1959) "Turbulence", McGraw-Hill Book Company, New York.

Hirt, C.W. (1968) "Heuristic stability theory for finite-difference equations", J. of Comp. Physics, Vol. 2, pp. 339-355.

Hodgins, S. (1977) "An improved computational method for the shallow water wave equations based the Leendertse (1967) finite-difference scheme", Report STF60A-77058, River and Harbour Laboratory, Norwegian Institute of Technology, Trondheim, Norway.

Houston, J.R. and Butler, H.L. (1979) "A numerical model for tsunami inundation", Technical Report HL-79-2, U.S. Army Engineer Waterways Experiment Station Hydraulics Laboratory, Vicksburg, Mississippi, U.S.A.

Johnson, B.H. (1980) "VAHM - a vertically averaged hydrodynamic model using boundary-fitted coordinates", Misc. Paper HL-80-3, U.S. Army Engineer Waterways Experiment Station Hydraulics Laboratory, Vicksburg, Mississippi, U.S.A.

Johnson, B.H. (1982) "Numerical modelling of estuarine hydrodynamics on a boundary-fitted coordinate system", in Numerical Grid Generation, J.F. Thompson (editor), Proc. of a Symp. on the Numerical Generation of Curvilinear Coordinate Systems and their Use in the Numerical Solution of Partial Differential Equations, held at Nashville, Tennesse, U.S.A., pp 409-436.

Johnson, B.H. and Thompson, J.F. (1978) "A discussion of boundary-fitted coordinate systems and their applicability to the numerical modelling of hydraulic problems", Misc. Paper H-78-9, U.S. Army Engineer Waterways Experiment Station Hydraulics Laboratory, Vicksburg, Mississippi, U.S.A.

Johnson, B.H. and Thompson, J.F. (1986) "Discussion of a depth-dependent adaptive grid generator for use in computational hydraulics", in Numerical Grid Generation in Computational Fluid Dynamics, J. Häuser and C. Taylor (Editors), Proc. of Int. Conf. at Landshut, West Germany, July 1986, pp. 629-640.

Johnson, B.H., Stein, A.B. and Thompson, J.F. (1982) "Modelling of flow and conservative substance transport on a boundary-fitted coordinate system", Vol. 2, Proc. of the IAHR Symp. on Refined Modelling of Flows, Paris, pp. 509-518.

Johnson, B.H., Thompson, J.F. and Baker, A.J. (1984) "A discussion of adaptive grids and their applicability in numerical hydrodynamic modelling", Misc. Paper HL-84-4, U.S. Army Engineer Waterways Experiment Station Hydraulics Laboratory, Vicksburg, Mississippi, U.S.A.

Jones, W.P. and Launder, B.E. (1972) "The prediction of laminarization with a two-equation model of turbulence", Int. J. Heat and Mass Transfer, Vol. 15, pp. 301-314.

Joynes, S.A. (1989) "Horizontal diffusion of a buoyant pollutant in coastal waters", Ph.D. Thesis, Dept. of Civil Engineering, University of Salford, U.K.

Kawaguti, M. (1953) "Numerical solution of the Navier-Stokes equations for the flow around a circular cylinder at Reynolds number 40", J. Phys. Soc. of Japan, Vol. 8, No. 6, pp. 747-757.

Kuipers, J. and Vreugdenhil, C.B. (1973) "Calculations of two-dimensional horizontal flow", Research Report S163, Part 1, Delft Hydraulics Laboratory, Delft, The Netherlands.

Laird, A.D.K. (1971) "Eddy formation behind circular cylinders", J. Hydraulics Div., Proc. ASCE, Vol. 97, No. HY6, pp. 763-775

Laufer, J. (1951) "Investigation of turbulent flow in a two-dimensional channel", National Advisory Committee for Aeronautics, Rep. No. 1053.

Launder, B.E. (1984) "Second-moment closure: methodology and practice" in Turbulence models and their applications, Collection de la Direction des Etudes et Recherches d'Electricite de France, Eyrolles.

Launder, B.E. and Spalding, D.B. (1972) "Lectures in: Mathematical models of turbulence", Academic Press.

Launder, B.E. and Spalding, D.B. (1974) "The numerical computation of turbulent flows", Computer Methods in Applied Mechanics and Engineering, Vol. 3, pp. 269-289.

Lean, G.H. and Weare, T.J. (1979) "Modelling two-dimensional circulating flow", J. Hydraulics Div., Proc. ASCE, Vol. 105, No. HY1, pp. 17-26.

Leendertse, J.J. (1967) "Aspects of a computational model for long-period water-wave propagation", The Rand Corporation, RM-5294-PR.

Leendertse, J.J. (1970) "A water-quality simulation model for well-mixed estuaries and coastal seas : Vol. 1, Principles of computation", The Rand Corporation, RM-6230-RC.

Leendertse, J.J. and Gritton, E.C. (1971) "A water-quality simulation model for well mixed estuaries and coastal seas : Vol. 2, Computation procedures", The Rand Corporation, R-708-NYC.

Lin, C.L., Pepper, D.W. and Lee, S.C. (1976) "Numerical methods for separated flow solutions around a circular cylinder", AIAA Journal, Vol. 14, No. 7, pp. 900-907.

Loc, T. P. (1980) "Numerical analysis of unsteady secondary vortices generated by an impulsively started circular cylinder", J. Fluid Mech., Vol. 100, Part 1, pp. 111-128.

Markatos, N.C. (1986) "The mathematical modelling of turbulent flows", Applied Math. Modelling, Vol. 10, pp. 190-220.

McClimans, T.A. and Gjerp, S.A. (1978) "Numerical study of distortion in a Froude model", Vol. 3, Proc. 16th Coastal Eng. Conf. (ASCE), Hamburg, Germany, pp. 2887-2904.

McGuirk, J.J. and Rodi, W. (1978) "A depth-averaged mathematical model for the near field of side discharges into open-channel flow", J. Fluid Mech., Vol. 86, Part 4, pp. 761-781.

McGuirk, J.J. and Rodi, W. (1979) "Calculation of unsteady mass exchange between a main stream and a dead water zone", Proc. 18th IAHR Congress, Cagliari, paper B.A.4, pp. 25-32.

Miller, T.F. and Schmidt, F.W. (1988) "Use of a pressure-weighted interpolation method for the solution of the incompressible Navier-Stokes equations on a non-staggered grid system", Numerical Heat Transfer, Vol. 14, pp. 213-233.

Mills, R.D. (1977) "Computing internal viscous flow problems for the circle by integral methods", J. Fluid Mech., Vol. 79, Part 3, pp. 609-624.

Milne-Thomson, L.M. (1968) "Theoretical Hydrodynamics", 5th Ed., Macmillan Press, New York.

Nakahashi, K. and Deiwert, G.S. (1984) "A practical adaptive-grid method for complex fluid-flow problems", Report No. TM85989, National Aeronautics and Space Administration, Ames Research Centre, California.

Neumann, G. and Pierson, W.J. (1966) "Principles of physical oceanography", Prentice-Hall Inc., N.J.

Officier, M.J. and Wiersma, A.K. (1986) "Experience with numerical grid generation techniques and their application in flow problems at the Delft Hydaulics Laboratory", in Numerical Grid Generation in Computational Fluid Dynamics, J. Häuser and C. Taylor (Editors), Proc. of Int. Conf. at Landshut, West Germany, July 1986, pp. 641-652.

Officier, M.J., Vreugdenhil, C.B. and Wind, H.G. (1986) "Applications in hydraulics of numerical solutions of the Navier-Stokes equations", in Recent Advances in Numerical Methods in Fluids, Vol. 5, C. Taylor, J.A. Johnson and W.R. Smith (Editors), Pineridge Press, Swansea, U.K, pp. 115-147.

Ogink, H.J.M. (1985) "The effective viscosity coefficient in 2-D depth-averaged flow models", Proc. 21st IAHR Cong., Melbourne, Australia, Aug. 1985, pp. 474-479.

Otnes, R.K. and Enochson, L. (1972) "Digital time series analysis", John Wiley and Sons, New York.

Paris, J. and Whitaker, S. (1965) "Confined wakes: a numerical solution of the Navier-Stokes equations", A.I.Ch.E. Journal, Vol. 11, No. 6, pp. 1033-1041.

Peaceman, D.W. and Rachford, H.H. (1955) "The numerical solution of parabolic and elliptic differential equations", J. of the Soc. of Industrial and Applied Math., Vol. 3, No. 1, pp. 28-41.

Perić, M. (1985) "A finite volume method for the prediction of three-dimensional fluid flow in complex ducts", Ph.D. Thesis, University of London.

Perić, M., Kessler, R. and Scheuerer, G. (1988) "Comparison of finite-volume numerical methods with staggered and collocated grids", Computers and Fluids, Vol. 16, No. 4, pp. 389-403.

Prandle, D. and Crookshank, N.L. (1974) "Numerical model of St.Lawrence river estuary", J. Hydraulics Div., Proc. ASCE, Vol. 100, No. HY4, pp. 517-529.

Prandtl, L. (1925) "Uber die ausgebildete Turbulenz", Z.A.M.M., 5, p. 136.

Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T. (1986) "Numerical recipes : the art of scientific computing", Cambridge University Press.

Raghunath, R., Sengupta, S. and Häuser, J. (1987) "A study of the motion in rotating containers using a boundary-fitted coordinate system", Int. J. for Numerical Methods in Fluids, Vol. 7, pp. 453-464.

Rai, M.M. and Anderson, D.A. (1981) "Grid evolution in time asymptotic problems", J. of Comp. Physics, Vol. 43, pp. 327-344.

Ramanathan, S. and Kumar, R. (1988) "Comparison of boundary-fitted coordinates with finite-element approach for solution of conduction problems", Numerical Heat Transfer, Vol. 14, pp. 187-211.

Rastogi, A.K. and Rodi, W. (1978) "Predictions of heat and mass transfer in open channels", J. Hydraulics Div., Proc. ASCE, Vol. 104, No. HY3, pp. 397-420.

Raudkivi, A.J. and Callander, R.A. (1975) "Advanced Fluid Mechanics", Edward Arnold Ltd., London.

Rayleigh, Lord (1893) "On the flow of viscous liquids, especially in two dimensions", Phil. Mag., S5, Vol. 36, No. 221, pp. 354-372.

Reid, R.O. and Bodine, B.R. (1968) "Numerical model for storm surges in Galveston Bay", J. Waterways and Harbours Div., Proc. ASCE, Vol. 94, No. WW1, pp. 33-57.

Rhie, C.M. and Chow, W.L. (1982) "A numerical study of the turbulent flow past an isolated aerofoil with trailing edge separation", AIAA/ASME 3rd Joint Thermophysics, Fluids, Plasma and Heat Transfer Conference, St. Louis, Missouri, June 1982.

Richtmeyer, R.D. and Morton, E.W. (1967) "Difference methods for initial value problems", 2nd Ed., Interscience Publishers, New York.

Roache, P.J. (1972) "Computational fluid dynamics", Hermosa Publishers, Albuquerque, N.M.

Roberts, K.V. and Weiss, N.O. (1966) "Convective difference schemes", Mathematics of Computation, Vol. 20, pp. 272-299.

Robinson, S.J. (1979) "Hydraulic modelling of circulation in reservoirs", M.Phil. Thesis, Imperial College, London.

Rodi, W. (1984a) "Turbulence models and their application in hydraulics - a state of the art review", 2nd Ed., I.A.H.R.

Rodi, W. (1984b) "Examples of turbulence-model applications", in Turbulence models and their applications, Collection de la Direction des Etudes et Recherches d'Electricite de France, Eyrolles.

Rodi, W., Pavlovič R.N. and Srivatsa, S.K. (1981) "Prediction of flow and pollutant spreading in rivers", Symp. on the predictive ability of surface water flow transport models for inland and coastal waters, Academic Press Inc.

Rosenhead, L. (1931) "The formation of vortices from a surface discontinuity", Proc. Roy. Soc., (A)134, pp. 170-192.

Sarpkaya, T. (1968) "An analytical study of separated flow about circular cylinders", Trans. ASME, J. Basic Engineering, Vol. 90, pp. 511-520.

Sarpkaya, T. (1989) "Computational methods with vortices - The 1988 Freeman Scholar Lecture", J. Fluids Engineering, Vol. 111, pp. 5-52.

Sarpkaya, T. and Schoaff, R.L. (1979) "Inviscid model of two-dimensional vortex shedding by a circular cylinder", AIAA Journal, Vol. 17, No. 11, pp. 1193-1200.

Schlichting, H. (1968) "Boundary-layer theory", McGraw Hill Book Company, New York.

Shapiro, R. (1970) "Smoothing, filtering and boundary effects", Reviews of Geophysics and Space Physics, Vol. 8, No. 2, pp. 359-387.

Shapiro, R. (1971) "The use of linear filtering as a parameterisation of atmospheric diffusion", J. Atmospheric Sciences, Vol. 28, pp. 523-531.

Shapiro, R. (1975) "Linear filtering", Mathematics of Computation, Vol. 29, No. 132, pp. 1094-1097.

Sheng, Y.P. (1986) "Numerical modelling of coastal and estuarine processes using boundary-fitted grids", Vol. 3, Proc. Third Int. Symp. on River Sedimentation, Jackson, U.S.A., pp. 1426-1442.

Sheng, Y.P. and Hirsh, J.E. (1984) "Numerical solution of shallow water equations in boundary fitted grids", Technical Memo. 84-15, Aeronautical Research Associates of Princeton, Inc., Princeton, New Jersey.

Shore Protection Manual (1977) 3rd Ed., U.S. Army Coastal Engineering Research Centre.

da Silva Lima, S.S.L. (1981) "Wave-induced nearshore currents", Ph.D. Thesis, Dept. of Civil Engineering, University of Liverpool, U.K.

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

Smith, I.M. and Griffiths, D.V. (1988) "Programming the finite element method", 2nd Ed., John Wiley & Sons.

Smith, P.A. and Stansby, P.K. (1987) "Generalised discrete vortex method for cylinders without sharp edges", AIAA Journal, Vol. 25, No. 2, pp. 199-200.

Sobey, R.J. (1973) "Flow patterns in lakes and reservoirs", Ph.D. Thesis, Imperial College, London.

Sobey, R.J. and Savage, S.B. (1974) "Jet-forced circulation in water-supply reservoirs", J. Hydraulics Div., Proc. ASCE, Vol. 100, No. HY12, pp. 1809-1828.

Son, J.S. and Hanratty, T.J. (1969) "Numerical solution for the flow around a cylinder at Reynolds numbers of 40, 200 and 500", J. Fluid Mech., Vol. 35, Part 2, pp. 369-386.

Stansby, P.K. (1977) "An inviscid model of vortex shedding from a circular cylinder in steady and oscillatory far flows", Proc. Inst. Civ. Engrs., Part 2, Vol. 63, pp. 865-880.

Stansby, P.K. and Dixon, A.G. (1983) "Simulation of flows around cylinders by a Lagrangian vortex scheme", Applied Ocean Research, Vol. 5, No. 3, pp. 167-178.

Stelling, G.S. (1983) "On the construction of computational methods for shallow water flow problems", Ph.D. Thesis, Delft University of Technology, Delft, The Netherlands.

Stelling, G.S. and Willemse, J.B.T.M. (1984) "Remarks about a computational method for shallow water equations that works in practice", Colloquium Topics in Applied Numerical Analysis, J.G. Verwer (Editor), C.W.I. Syllabus No. 5, pp. 337-362, Centrum voor Wiskunde en Informatica, Amsterdam, The Netherlands.

Stone, H.L. (1968) "Iterative solution of implicit approximations of multidimensional partial differential equations", SIAM J. of Numerical Analysis, Vol. 5, No. 3, pp. 530-558.

Taylor, G. (1954) "The dispersion of matter in turbulent flow through a pipe", Proc. Royal Soc., A, 223, pp. 446-468.

Thacker, W.C. (1977) "Irregular-grid finite-difference techniques: simulations of oscillations in shallow circular basins", J. Physical Oceanography, Vol. 7, No. 2, pp. 284-292.

Thacker, W.C. (1979) "Irregular-grid finite-difference techniques for storm surge calculations for curving coastlines", in Marine Forecasting, J.C.J. Nihoul (Editor), Elsevier Oceanographic Series, Vol. 25, Elsevier Scientific Publishing Company, Amsterdam, Holland, pp. 261-283.

Thames, F.C. (1975) "Numerical solution of the incompressible Navier-Stokes equations about arbitrary two-dimensional bodies", Ph.D. Dissertation, Mississippi State University, U.S.A.

Thames, F.C., Thompson, J.F., Mastin, C.W. and Walker, R.L. (1977) "Numerical solutions for viscous and potential flow about arbitrary two-dimensional bodies using body-fitted coordinate systems", J. of Comp. Physics, Vol. 24, pp. 245-273.

Thom, A. (1933) "The flow past circular cylinders at low speeds", Proc. Roy. Soc., A., Vol. 141, pp. 651-669.

Thoman, D.C. (1966) "Numerical solutions of time dependent two-dimensional flow of a viscous incompressible fluid over stationary and rotating cylinders", Ph.D. Thesis, Dept. of Mech. Eng., University of Notre Dame, Indiana.

Thoman, D.C. and Szewczyk, A.A. (1969) "Time-dependent viscous flow over a circular cylinder", Physics of Fluids Supplement II, pp. 76-86.

Thompson, J.F. (1982a) "Numerical grid generation", Proc. of a Symp. on the Numerical Generation of Curvilinear Coordinate Systems and their Use in the Numerical Solution of Partial Differential Equations, held at Nashville, Tennesse, U.S.A.

Thompson, J.F. (1982b) "Elliptic grid generation", in Numerical Grid Generation, J.F. Thompson (editor), Proc. of a Symp. on the Numerical Generation of Curvilinear Coordinate Systems and their Use in the Numerical Solution of Partial Differential Equations, held at Nashville, Tennesse, U.S.A., pp. 79-105.

Thompson, J.F. (1983) "A boundary-fitted coordinate code for general two-dimensional regions with obstacles and boundary intrusions", Technical Report E-83-8, U.S. Army Engineer Waterways Experiment Station Hydraulics Laboratory, Vicksburg, Mississippi, U.S.A.

Thompson, J.F. (1984) "Grid generation techniques in computational fluid dynamics", AIAA Journal, Vol. 22, No. 11, pp. 1505-1523.

Thompson, J.F. (1987) "A composite grid generation code for general 3-D regions", AIAA 25th Aerospace Sciences Meeting, Reno, Nevada.

Thompson, J.F. and Johnson, B.H. (1985) "Development of an adaptive boundary-fitted coordinate code for use in coastal and estuarine areas", Misc. Paper HL-85-5, U.S. Army Engineer Waterways Experiment Station Hydraulics Laboratory, Vicksburg, Mississippi, U.S.A.

Thompson, J.F., Thames, F.C. and Mastin, C.W. (1974) "Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies", J. of Comp. Physics, Vol. 15, pp. 299-319.

Thompson, J.F., Thames, F.C., Mastin, C.W. and Shanks, S.P. (1975) "Use of numerically generated body-fitted coordinate systems for solution of the Navier-Stokes equations", Proc. of AIAA 2nd Computational Fluid Dynamics Conf., Hartford, Conn., pp. 68-80.

Thompson, J.F., Thames, F.C. and Mastin, C.W. (1977a) "TOMCAT - a code for numerical generation of boundary-fitted curvilinear coordinate systems on fields containing any number of arbitrary two-dimensional bodies", J. of Comp. Physics, Vol. 24, pp. 274-302.

Thompson, J.F., Thames, F.C. and Mastin, C.W. (1977b) "Boundary-fitted curvilinear coordinate systems for solution of partial differential equations on fields containing any number of arbitrary two-dimensional bodies", NASA Report No. CR-2729, National Aeronautics and Space Administration, Washington, D.C.

Thompson, J.F., Warsi, Z.U.A. and Mastin, C.W. (1982) "Boundary-fitted coordinate systems for numerical solution of partial differential equations - a review", J. of Comp. Physics, Vol. 47, pp. 1-108.

Thompson, J.F., Warsi, Z.U.A. and Mastin, C.W. (1985) "Numerical grid generation : foundations and applications", North-Holland Press.

Van Dorn, W.G. (1953) "Wind stress on an artificial pond", J. Marine Research, Vol. 12, Part 3, pp. 249-276.

Vemulakonda, S.R., Swain, A., Houston, J.R., Farrar, P.D., Chou, L.W. and Ebersole, B.A. (1985) "Coastal and inlet processes numerical modelling system for Oregon Inlet, North Carolina", Technical Report CERC-85-6, U.S. Army Engineer Waterways Experiment Station Coastal Engineering Research Center, Vicksburg, Mississippi, U.S.A.

Verboom, G.K. (1983) "Curvilinear coordinates in WAQUA : a preparatory study", Internal Report, Delft Hydraulics Laboratory, Note S 584-II.

Vreugdenhil, C.B. (1973) "Secondary-flow computations", Proc. 15th Congress of the IAHR, Istanbul, paper B20, pp. 153-160. [reprinted as - Publication No. 114, Nov. 1973, Delft Hydraulics Laboratory, Delft, The Netherlands.]

Vreugdenhil, C.B. and Wijbenga, J.H.A. (1982) "Computation of flow patterns in rivers", J. Hydraulics Div., Proc. ASCE, Vol. 108, No. HY11, pp. 1296-1310.

Walker, D.J. (1987) "Nearshore hydrodynamics and the behaviour of groynes on sandy beaches", Ph.D. Thesis, Department of Civil Engineering, Imperial College, London.

Wanstrath, J.J. (1977) "Nearshore numerical storm surge and tidal simulation", Technical Report H-77-17, U.S. Army Engineer Waterways Experiment Station, Vicksburg, Mississippi, U.S.A.

Watson, A. (1989) "Comparison of algal penetration through rapid-gravity filter beds", J. Inst. of Water and Environmental Management, Vol. 3, No. 5, pp. 443-450.

Weare, T.J. (1976a) "Finite element or finite difference methods for the two-dimensional shallow water equations?", Computer Methods in Applied Mechanics and Engineering, Vol. 7, pp. 351-357.

Weare, T.J. (1976b) "Instability in tidal flow computational schemes", J. Hydraulics Div., Proc. ASCE, Vol. 102, No. HY5, pp. 569-580.

Weare, T.J. (1979) "Errors arising from irregular boundaries in ADI solutions of the shallow-water equations", Int. J. for Numer. Methods in Engineering, Vol. 14, pp. 921-931.

White, C.M., Harris, S.L. and Cooley, P. (1955) "The hydraulic aspects of stagnation in reservoirs", Hawksley Hydraulic Laboratory, Imperial College, Unpublished Report to Metropolitan Water Board.

Wijbenga, J.H.A. (1985a) "Determination of flow patterns in rivers with curvilinear coordinates", Proc. 21st IAHR Cong., Melbourne, Australia, Aug. 1985. [reprinted as - Publication No. 352, Oct. 1985, Delft Hydraulics Laboratory, Delft, The Netherlands.]

Wijbenga, J.H.A. (1985b) "Steady depth-averaged flow calculations on curvilinear grids", 2nd Int. Conf. on the Hydraulics of Floods and Flood Control, Cambridge, England, Sept. 1985, pp. 373-387

Willemse, J.B.T.M., Stelling, G.S. and Verboom, G.K. (1985) "Solving the shallow water equations with an orthogonal coordinate transformation", Presented at the Int. Symp. on Computational Fluid Dynamics, Tokyo, Sept. 1985. [reprinted as Delft Hydraulics Communication No. 356, Jan. 86, Delft Hydraulics Laboratory, The Netherlands.]

Wind, H.G. and Vreugdenhil, C.B. (1986) "Rip-current generation near structures", J. Fluid Mech., Vol. 171, pp. 459-476.

Woods, L.C. (1954) "A note on the numerical solution of fourth order differential equations", Aeronautical Quarterly, Vol. 5, Part 3, pp. 176-184.

Yin, F. and Chen, S.-H. (1982) "Tidal computation on Taiwan Strait", J. Waterway, Port, Coastal and Ocean Division, Proc. ASCE, Vol. 108, No. WW4, pp. 539-553.

Young, S.N., Clough, W.T., Thomas, A.J. and Siddall, R. (1988) "Changes in plant community at Foxcote Reservoir following use of ferric sulphate to control nutrient levels", J. Inst. of Water and Environmental Management, Vol. 2, No. 1, pp. 5-12.

Younis, B.A. (1990) Personal communication with Dr. B.A. Younis, City University, London.

Zienkiewicz, O.C. (1977) "The finite element method", 3rd Ed., McGraw-Hill, London.

APPENDIX A

NO-SLIP WALL VORTICITY CONDITIONS : NAVIER-STOKES SIMULATION

Circular walls:

Consider a node, (i,M) on the circular perimeter of the reservoir; the radial and tangential velocity components are both zero and therefore,

$$\left.\frac{\partial \psi}{\partial \theta}\right|_{i,M} = 0 \quad \text{and} \quad \left.\frac{\partial \psi}{\partial \alpha_r}\right|_{i,M} = 0 \quad . \tag{A1}$$

Furthermore,

$$\left.\frac{\partial^2 \psi}{\partial \theta^2}\right|_{i,M} = 0 \quad . \tag{A2}$$

Substituting the above expressions into the transformed stream function equation (3.11), leads to

$$\left.\frac{1}{\mu_{r_M}^2}\frac{\partial^2 \psi}{\partial \alpha_r^2}\right|_{i,M} = -\omega_{i,M} \quad . \tag{A3}$$

Applying a Taylor series expansion of ψ about the point, (i,M) in the α_r-direction gives

$$\psi_{i,M-1} = \psi_{i,M} - \left.\frac{\partial \psi}{\partial \alpha_r}\right|_{i,M} \Delta\alpha_r + \frac{1}{2}\left.\frac{\partial^2 \psi}{\partial \alpha_r^2}\right|_{i,M} \Delta\alpha_r^2 - \frac{1}{6}\left.\frac{\partial^3 \psi}{\partial \alpha_r^3}\right|_{i,M} \Delta\alpha_r^3 + O(\Delta\alpha_r^4) \quad , \tag{A4}$$

which may be rewritten, after substitution as

$$\psi_{i,M-1} = \psi_{i,M} - \frac{1}{2}\mu_{r_M}^2 \omega_{i,M} \Delta\alpha_r^2 - \frac{1}{6}\left.\frac{\partial^3 \psi}{\partial \alpha_r^3}\right|_{i,M} \Delta\alpha_r^3 + O(\Delta\alpha_r^4) \quad . \tag{A5}$$

Rearranging equation (A5) and ignoring terms above $\Delta\alpha_r^2$, yields the first order accurate vorticity condition:

$$\omega_{i,M} = \frac{2(\psi_{i,M} - \psi_{i,M-1})}{\mu_{r_M}^2 \Delta\alpha_r^2} + O(\Delta\alpha_r) \quad . \tag{A6}$$

For increased spatial accuracy, Woods' second order method (1954) is employed. This is obtained by first differentiating the stream function equation with respect to α_r:

$$- \frac{\partial \omega}{\partial \alpha_r} = \frac{1}{\mu_r^2} \frac{\partial^3 \psi}{\partial \alpha_r^3} + \frac{\partial (1/\mu_r^2)}{\partial \alpha_r} \frac{\partial^2 \psi}{\partial \alpha_r^2} + \frac{1}{\mu_r} \left[\frac{1}{r} + \frac{\partial (1/\mu_r)}{\partial \alpha_r} \right] \frac{\partial^2 \psi}{\partial \alpha_r^2}$$

$$+ \frac{\partial}{\partial \alpha_r} \left[\frac{1}{\mu_r} \left[\frac{1}{r} + \frac{\partial (1/\mu_r)}{\partial \alpha_r} \right] \right] \frac{\partial \psi}{\partial \alpha_r} + \frac{1}{r^2} \frac{\partial^3 \psi}{\partial \alpha_r \partial \theta^2} + \frac{\partial (1/r^2)}{\partial \alpha_r} \frac{\partial^2 \psi}{\partial \theta^2} \quad . \qquad \text{(A7)}$$

Using the fact that the order of partial differentiation is irrelevant:

$$\left. \frac{\partial^3 \psi}{\partial \alpha_r \partial \theta^2} \right|_{i,M} = \left. \frac{\partial^3 \psi}{\partial \theta^2 \partial \alpha_r} \right|_{i,M} = 0 \quad . \qquad \text{(A8)}$$

Applying (A7) to the boundary node at (i,M) and using the expressions shown in (A1), (A2) and (A8) gives

$$\left. - \frac{\partial \omega}{\partial \alpha_r} \right|_{i,M} = \frac{1}{\mu_{r_M}^2} \left. \frac{\partial^3 \psi}{\partial \alpha_r^3} \right|_{i,M} + K_1 \left. \frac{\partial^2 \psi}{\partial \alpha_r^2} \right|_{i,M} \qquad \text{(A9)}$$

where

$$K_1 = \frac{3}{\mu_{r_M}} \left. \frac{\partial (1/\mu_r)}{\partial \alpha_r} \right|_M + \frac{1}{\mu_{r_M} r_M} \quad .$$

Substituting equation (A3) into (A9) leads to

$$\left. - \frac{\partial \omega}{\partial \alpha_r} \right|_{i,M} = \frac{1}{\mu_{r_M}^2} \left. \frac{\partial^3 \psi}{\partial \alpha_r^3} \right|_{i,M} + K_2 \, \omega_{i,M} \qquad \text{(A10)}$$

where

$$K_2 = - K_1 \, \mu_{r_M}^2 = - 3 \, \mu_{r_M} \left. \frac{\partial (1/\mu_r)}{\partial \alpha_r} \right|_M - \frac{\mu_{r_M}}{r_M} \quad .$$

The second stage of the derivation of Woods' boundary condition involves estimating $\partial^3 \psi / \partial \alpha_r^3$ at the boundary, using (A5):

$$\left. \frac{\partial^3 \psi}{\partial \alpha_r^3} \right|_{i,M} = \frac{6(\psi_{i,M} - \psi_{i,M-1})}{\Delta \alpha_r^3} - \frac{3}{\Delta \alpha_r} \mu_{r_M}^2 \, \omega_{i,M} + O(\Delta \alpha_r) \quad . \qquad \text{(A11)}$$

The final stage uses a Taylor expansion about the node, (i,M) for the vorticity, viz.

$$\omega_{i,M-1} = \omega_{i,M} - \left. \frac{\partial \omega}{\partial \alpha_r} \right|_{i,M} \Delta \alpha_r + O(\Delta \alpha_r^2) \quad . \qquad \text{(A12)}$$

Thus, substituting equations (A10) and (A11) into the above Taylor series produces

$$\omega_{i,M-1} = \omega_{i,M} + \left[\frac{1}{\mu_{r_M}^2}\left[\frac{6(\psi_{i,M} - \psi_{i,M-1})}{\Delta\alpha_r{}^3} - \frac{3}{\Delta\alpha_r} \mu_{r_M}^2 \omega_{i,M} + O(\Delta\alpha_r) \right] + K_2 \omega_{i,M} \right] \Delta\alpha_r + O(\Delta\alpha_r{}^2) \quad , \qquad (A13)$$

which may be rearranged to give the Woods' vorticity condition:

$$\omega_{i,M} = \frac{1}{(2 - K_2\Delta\alpha_r)} \left[\frac{6(\psi_{i,M} - \psi_{i,M-1})}{\Delta\alpha_r{}^2 \mu_{r_M}^2} - \omega_{i,M-1} \right] + O(\Delta\alpha_r{}^2) \quad . \qquad (A14)$$

Radial walls (inlet and outlet stems):

Consider a typical node, (NA,j) along the lower inlet radial wall; again, both the radial and tangential velocity components are zero at the perimeter and therefore

$$\left.\frac{\partial\psi}{\partial\theta}\right|_{NA,j} = 0 \quad , \quad \left.\frac{\partial\psi}{\partial\alpha_r}\right|_{NA,j} = 0 \quad \text{and} \quad \left.\frac{\partial^2\psi}{\partial\alpha_r{}^2}\right|_{NA,j} = 0 \quad . \qquad (A15)$$

At the boundary, the stream function equation may thus be simplified to

$$\frac{1}{r_j{}^2} \left.\frac{\partial^2\psi}{\partial\theta^2}\right|_{NA,j} = - \omega_{NA,j} \quad . \qquad (A16)$$

Applying a Taylor expansion of ψ about the point, (NA,j) and using the convention that θ increases in an anti-clockwise direction, leads to the equation,

$$\psi_{NA-1,j} = \psi_{NA,j} - \left.\frac{\partial\psi}{\partial\theta}\right|_{NA,j} \Delta\theta + \frac{1}{2} \left.\frac{\partial^2\psi}{\partial\theta^2}\right|_{NA,j} \Delta\theta^2 - \frac{1}{6} \left.\frac{\partial^3\psi}{\partial\theta^3}\right|_{NA,j} \Delta\theta^3 + O(\Delta\theta^4) \qquad (A17)$$

which may be recast as

$$\psi_{NA-1,j} = \psi_{NA,j} - \frac{1}{2} r_j{}^2 \omega_{NA,j} \Delta\theta^2 - \frac{1}{6} \left.\frac{\partial^3\psi}{\partial\theta^3}\right|_{NA,j} \Delta\theta^3 + O(\Delta\theta^4) \quad . \qquad (A18)$$

Ignoring terms higher than $\Delta\theta^2$, results in the first order accurate vorticity boundary equation:

$$\omega_{NA,j} = \frac{2(\psi_{NA,j} - \psi_{NA-1,j})}{r_j{}^2\Delta\theta^2} + O(\Delta\theta) \quad . \qquad (A19)$$

The derivation of the second order Woods' vorticity condition follows an analogous procedure to that described for the circular wall. In this case, the stream function equation is differentiated with respect to θ:

$$- \frac{\partial \omega}{\partial \theta} = \frac{1}{\mu_r{}^2} \frac{\partial^3 \psi}{\partial \theta \partial \alpha_r{}^2} + \frac{1}{\mu_r} \left[\frac{1}{r} + \frac{\partial (1/\mu_r)}{\partial \alpha_r} \right] \frac{\partial^2 \psi}{\partial \theta \partial \alpha_r} + \frac{1}{r^2} \frac{\partial^3 \psi}{\partial \theta^3} \quad . \tag{A20}$$

Along the radial wall,

$$\left. \frac{\partial^2 \psi}{\partial \theta \partial \alpha_r} \right|_{NA,j} = \left. \frac{\partial^2 \psi}{\partial \alpha_r \partial \theta} \right|_{NA,j} = 0 \quad \text{and} \quad \left. \frac{\partial^3 \psi}{\partial \theta \partial \alpha_r{}^2} \right|_{NA,j} = \left. \frac{\partial^3 \psi}{\partial \alpha_r{}^2 \partial \theta} \right|_{NA,j} = 0 \quad . \tag{A21}$$

Therefore, applying equation (A20) to the boundary node at (NA,j) gives

$$- \left. \frac{\partial \omega}{\partial \theta} \right|_{NA,j} = \frac{1}{r_j{}^2} \left. \frac{\partial^3 \psi}{\partial \theta^3} \right|_{NA,j} \quad . \tag{A22}$$

Equation (A18) is rearranged as

$$\left. \frac{\partial^3 \psi}{\partial \theta^3} \right|_{NA,j} = \frac{6(\psi_{NA,j} - \psi_{NA-1,j})}{\Delta \theta^3} - \frac{3 r_j{}^2}{\Delta \theta} \omega_{NA,j} + O(\Delta \theta) \quad . \tag{A23}$$

Finally, a second Taylor series is utilised to describe the vorticity variation near the wall:

$$\omega_{NA-1,j} = \omega_{NA,j} - \left. \frac{\partial \omega}{\partial \theta} \right|_{NA,j} \Delta \theta + O(\Delta \theta^2) \quad . \tag{A24}$$

Substituting equations (A22) and (A23) into the above vorticity expansion gives

$$\omega_{NA-1,j} = \omega_{NA,j} + \left[\frac{1}{r_j{}^2} \left[\frac{6(\psi_{NA,j} - \psi_{NA-1,j})}{\Delta \theta^3} - \frac{3 r_j{}^2}{\Delta \theta} \omega_{NA,j} + O(\Delta \theta) \right] \right] \Delta \theta + O(\Delta \theta^2) \tag{A25}$$

which may be rearranged to give the radial wall Woods' vorticity condition:

$$\omega_{NA,j} = \frac{3(\psi_{NA,j} - \psi_{NA-1,j})}{r_j{}^2 \Delta \theta^2} - \frac{1}{2} \omega_{NA-1,j} + O(\Delta \theta^2) \quad . \tag{A26}$$

APPENDIX B

TREATMENT OF CENTRAL POINT : NAVIER-STOKES SIMULATION

At the central point of the finite-difference mesh, the polar stream function and vorticity-transport formulae need to be replaced by the Cartesian expressions:

$$\frac{\partial^2\psi}{\partial x^2} + \frac{\partial^2\psi}{\partial y^2} = -\omega \qquad \text{(B1)}$$

and

$$\frac{\partial\omega}{\partial t} = -\frac{\partial(u\omega)}{\partial x} - \frac{\partial(v\omega)}{\partial y} + \nu\left[\frac{\partial^2\omega}{\partial x^2} + \frac{\partial^2\omega}{\partial y^2}\right] \quad . \qquad \text{(B2)}$$

In order to discretise the above equations, the radial lines of the polar mesh must possess right-angled symmetry, i.e.

$$N/4 = \text{INT}\,(N/4) \quad , \qquad \text{(B3)}$$

where N is the number of radial mesh lines.

Although the formulae presented in (B1) and (B2) could simply be applied to the *primary* Cartesian coordinates (x-axis parallel to the $\theta = 0$ line), a better estimate of ψ and ω at the origin can be obtained by utilising all the available information from the nodes on the j = 1 line. This is accomplished using an extension of the technique suggested by Smith (1985). Consider a set of Cartesian coordinate axes, (x',y'), at an angle of $\theta = +i\Delta\theta$ from the primary (x,y) coordinates. The transformation to the (x',y') coordinate system merely introduces primes into equations (B1) and (B2), viz.

$$\frac{\partial^2\psi}{\partial x'^2} + \frac{\partial^2\psi}{\partial y'^2} = -\omega \qquad \text{(B4)}$$

and

$$\frac{\partial\omega}{\partial t} = -\frac{\partial(u'\omega)}{\partial x'} - \frac{\partial(v'\omega)}{\partial y'} + \nu\left[\frac{\partial^2\omega}{\partial x'^2} + \frac{\partial^2\omega}{\partial y'^2}\right] \qquad \text{(B5)}$$

where u' and v' are the velocity components parallel to the x'- and y'-axes, respectively.

Thus, the finite-difference approximation to the stream function equation for this particular coordinate frame is

$$\left[\frac{\psi_{i,1} - 2\psi_0 + \psi_{i+N/2,1}}{\delta r^2}\right] + \left[\frac{\psi_{i+N/4,1} - 2\psi_0 + \psi_{i+3N/4,1}}{\delta r^2}\right] = -\omega_0 \qquad \text{(B6)}$$

where ψ_0 = stream function at the central point,

ω_0 = vorticity at the central point,

and δr = length of the first grid increment in the radial direction of the physical mesh, i.e.,

$$\delta r = a\,\Delta\alpha_r^4 + b\,\Delta\alpha_r^3 + c\,\Delta\alpha_r^2 + d\,\Delta\alpha_r \quad . \qquad \text{(B7)}$$

Equation (B6) may be rearranged to give the stream function at the central node as

$$\psi_0 = \frac{\psi_{i,1} + \psi_{i+N/4,1} + \psi_{i+N/2,1} + \psi_{i+3N/4,1}}{4} + \frac{\omega_0 \delta r^2}{4} \quad . \qquad \text{(B8)}$$

On account of the symmetry of equation (B6), the mean value of ψ_0 can be found by averaging (B8) over the first quadrant ($0 \leq \theta \leq \pi/2$). Thus the best estimate of the stream function at the origin is

$$\psi_0 = \frac{1}{N/4} \sum_{i=0}^{N/4-1} \left[\frac{\psi_{i,1} + \psi_{i+N/4,1} + \psi_{i+N/2,1} + \psi_{i+3N/4,1}}{4} + \frac{\omega_0 \delta r^2}{4}\right] \qquad \text{(B9)}$$

which can be simplified to give

$$\psi_0 = \frac{1}{N} \sum_{i=0}^{N-1} \psi_{i,1} + \frac{\omega_0 \delta r^2}{4} \quad . \qquad \text{(B10)}$$

Equation (B10) is used after each sweep of the S.O.R. stream function determination, described in Section 3.4.2.

The vorticity-transport equation is treated in a complementary manner. Applying the Chain Rule to the advective terms of (B5) gives the advection vorticity-transport formulation:

$$\frac{\partial \omega}{\partial t} = -\left[u' \frac{\partial \omega}{\partial x'} + \omega \frac{\partial u'}{\partial x'}\right] - \left[v' \frac{\partial \omega}{\partial y'} + \omega \frac{\partial v'}{\partial y'}\right] + \nu \left[\frac{\partial^2 \omega}{\partial x'^2} + \frac{\partial^2 \omega}{\partial y'^2}\right]. \qquad \text{(B11)}$$

Considering the same coordinate frame as before (x'-axis at an angle of $\theta = +i\Delta\theta$ to the x-axis), then the diffusive term in the vorticity-transport equation is simply

$$\nu \left[\frac{\omega_{i,1} + \omega_{i+N/4,1} + \omega_{i+N/2,1} + \omega_{i+3N/4,1} - 4\,\omega_0}{\delta r^2}\right] \quad . \qquad \text{(B12)}$$

The advective terms in (B11) are discretised using the fact that

$$u' = \frac{\partial \psi}{\partial y'} \qquad \text{and} \qquad v' = - \frac{\partial \psi}{\partial x'} \quad , \tag{B13}$$

and by relating the 'local' Cartesian velocity components, u' and v' to the radial velocity, V_r:

$$\left.\begin{aligned} & u'_{i,1} = V_{r_{i,1}} \quad ; \quad u'_{i+N/2,1} = - V_{r_{i+N/2,1}} \\ \text{and} \\ & v'_{i+N/4,1} = V_{r_{i+N/4,1}} \quad ; \quad v'_{i+3N/4,1} = - V_{r_{i+3N/4,1}} \end{aligned}\right\} \tag{B14}$$

for $0 \leq i \leq N/4 - 1$.

Thus,

$$\left.\begin{aligned} & u'_{i,0} = \left[\frac{\psi_{i+N/4,1} - \psi_{i+3N/4,1}}{2\delta r} \right] \quad , \\ & \left.\frac{\partial \omega}{\partial x'}\right|_{i,0} = \left[\frac{\omega_{i,1} - \omega_{i+N/2,1}}{2\delta r} \right] \\ \text{and} \quad & \left.\frac{\partial u'}{\partial x'}\right|_{i,0} = \left[\frac{V_{r_{i,1}} + V_{r_{i+N/2,1}}}{2\delta r} \right] \quad . \end{aligned}\right\} \tag{B15}$$

Similar expressions to those shown above, can be obtained for the y-direction advective components. Hence, the vorticity-transport equation applicable to the (x',y') coordinate axis is

$$\left.\frac{\partial \omega}{\partial t}\right|_{i,0} = - \frac{1}{4\delta r^2}(\psi_{i+N/4,1} - \psi_{i+3N/4,1})(\omega_{i,1} - \omega_{i+N/2,1})$$

$$- \frac{\omega_0}{2\delta r}(V_{r_{i,1}} + V_{r_{i+N/2,1}}) - \frac{1}{4\delta r^2}(\psi_{i+N/2,1} - \psi_{i,1})(\omega_{i+N/4,1} - \omega_{i+3N/4,1})$$

$$- \frac{\omega_0}{2\delta r}(V_{r_{i+N/4,1}} + V_{r_{i+3N/4,1}})$$

$$+ \frac{\nu}{\delta r^2}(\omega_{i,1} + \omega_{i+N/4,1} + \omega_{i+N/2,1} + \omega_{i+3N/4,1} - 4\,\omega_0)$$

for $0 \leq i \leq N/4 - 1$. (B16)

The mean value of $\partial\omega/\partial t$ can be found by averaging this equation over just one

quadrant (due to the implicit symmetry). Thus, the updated central point vorticity is obtained from the following explicit forward difference expression:

$$\omega_0^{n+1} = \omega_0^n + \overline{\frac{\partial \omega_0}{\partial t}} \, \Delta t \qquad \text{(B17)}$$

where

$$\overline{\frac{\partial \omega_0}{\partial t}} = \sum_{i=0}^{N/4-1} \left. \frac{\partial \omega}{\partial t} \right|_{i,0} .$$

By implementing equation (B17) before the A.D.I. vorticity determination for the standard interior nodes (Section 3.4.1), the advanced vorticity at the origin can be utilised during the α_r-direction release.