1995

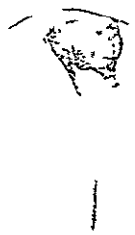
WETTING AND DRYING IN TWO-DIMENSIONAL TIDAL NUMERICAL MODELS

STRIPLING, STUART

WETTING AND DRYING IN TWO DIMENSIONAL TIDAL NUMERICAL MODELS

STUART STRIPLING

Ph.D. 1995



WETTING AND DRYING IN TWO-DIMENSIONAL TIDAL NUMERICAL MODELS

STUART STRIPLING

A thesis submitted to the University of Plymouth
in partial fulfilment for the degree of

DOCTOR OF PHILOSOPHY

Institute of Marine Studies
Faculty of Science

Collaborating Institute: Proudman Oceanographic Laboratory

September 1995

CONTENTS

CONTENTS (continued)

CONTENTS (continued)

CONTENTS (continued)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

LIST OF PLATES

ACKNOWLEDGEMENTS

I would like to sincerely thank Dr. K.J. George for his tremendous undying assistance and enthusiasm for this thesis.

Thanks also to Prof. P.P.G. Dyke for keeping me going, to SERC for providing me with most of the funding for this research and to the Governor of H.M.P. Frieston, North Sea Camp, for allowing us access to the beach.

Thanks to Xiankun and Adrian for their help at the Wash, and to HR Wallingford for the use of their resources in completing this thesis.

Most of all I have to thank my wife for her matter-of-factness where I am concerned, for her love and her friendship through all of this...

AUTHOR'S DECLARATION

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other university award.

This study was financed with the aid of a studentship from the Science and Engineering Research Council, and carried out in collaboration with the Proudman Oceanographic Laboratory.

Publications:

George K.J. & Stripling S. (1995) 'Improving the simulation of wetting and drying in a two-dimensional tidal numerical model', Applied Mathematical Modelling.

Stripling S., Dyer K.R. and Huntley D.A. (1994) 'A review of transport models in the marine environment', Commercial report confidential to Westlakes Research Institute and BNFL, Sellafield.

Signed Stuart Stripling.....
Date 15th February 1996.

WETTING AND DRYING IN TWO-DIMENSIONAL TIDAL NUMERICAL MODELS

STUART STRIPLING

ABSTRACT

The purpose of this research is to adapt and improve existing two-dimensional numerical tidal models so that they can cope with regions where the tide falls and rises to uncover and cover inter-tidal banks without incurring numerical shocks which may be caused by the discretization of time and space.

This thesis presents a review of current practices in the numerical modelling of flooding and drying banks in two dimensions. A two-dimensional depth-averaged numerical model has been written and is presented. It is used as a tool with which to investigate various existing algorithms which represent the physical process of the wetting and drying of intertidal zones.

An alternative method with which to represent the moving boundary has been developed. This method is free from disturbances usually caused by the implementation of a moving boundary in such a numerical scheme.

A 2NM numerical model of the Wash, U.K., is run to provide hind-cast tidal data pertaining to a particular site and period. A field programme is established to provide validation data for the model.

Finally, conclusions from the programme of research are drawn.

INTRODUCTION

When an area of coast becomes exposed as the tide falls, and covered as the tide rises, we can term this phenomenon the drying and wetting of an inter-tidal zone. The representation of this physical process in a numerical model can create disturbances in the solution. These disturbances are essentially the result of an attempt to incorporate, in numerical terms, a boundary which moves in a discrete manner.

Due to the nature of the frictional term in the hydrodynamic equations of motion, a mathematical singularity occurs at zero water depth resulting in the numerically calculated velocity value potentially increasing to infinity. It is therefore essential that this singularity is avoided and an alternative method to represent this very shallow water phenomenon be found, either in the way that the frictional force is represented (i.e. to modify the physics expressed by Newton's Second Law) or by some other physical or numerical means.

Algorithms for drying and wetting used in existing numerical models, whether they be solved implicitly, semi-implicitly or explicitly, appear to have been developed with the elimination of instability as their primary aim rather than the faithful representation of the physical processes in very shallow water (Leendertse, 1970; Flather & Heaps, 1975; Owens, 1984; Falconer, 1985; Cheng *et al.*, 1993). Other workers, Davies (1986), Backhaus (1983), have developed and used three-dimensional models for the analysis of effects such as bottom stress,

wind stress and convective circulations. However, three-dimensional models are very costly to run, and there is much pressure these days to develop two-dimensional models that have a general ability to function even in very shallow seas.

Stripling *et al.* (1994) have established that the incorporation of a moving land/sea boundary is one which commercial companies have spent considerable time addressing. Many companies are now satisfied that their representation of the physical processes involved is adequate; but they also indicate that it could be improved upon if they had sufficient time and inclination for the problem to be addressed further.

Two-dimensional models developed by George at the University of Plymouth over the last nine years incorporate a drying algorithm, but are unsatisfactory and require improvement. In order to attain this improvement, existing drying algorithms have been analyzed using a working two-dimensional model of an idealized sea basin based on those of George & Evans (1991) and Prandle (Proudman Oceanographic Laboratory's General Purpose Model). These algorithms have been critically examined to establish precisely what happens to certain terms in the hydrodynamic equations as the water depth tends to zero. On the basis of this examination, the algorithms have been improved upon and incorporated in the model of an idealized sea basin, leading to the design of a model of a real sea area, The Wash (on the east coast of the U.K.), which has been validated by reference to field measurements taken by the Author, together with Mr. X. Ke[1] and Mr. A. Flavell[1]

It is on the strength of these findings that an improved representation of flooding and drying in two-dimensional numerical tidal models has been developed and presented in this thesis.

[1]Department of Oceanography, University of Southampton.

Chapter 1 provides the reader with a literary review, together with a descriptive view of the modelling of wetting and drying banks.

Chapter 2 describes the development of a two-dimensional depth-averaged numerical model which is used as a tool for investigating existing drying algorithms.

Chapter 3 introduces the reader to a numerical description of the problems encountered when it is necessary to incorporate a moving land/sea boundary. This chapter also gives results from experiments with existing drying algorithms using the numerical model described in Chapter 2 as an experimental basis.

Chapter 4 gives specific detail about the development of a new drying algorithm using the method of sloping facets, and offers results showing that the technique is an improvement upon those methods investigated in Chapter 3.

Chapter 5 introduces a field programme which was set up to provide a database for the validation of a model of a real sea area.

Chapter 6 defines a model of the Wash, U.K. and presents results which are compared to the field observations as described in Chapter 5.

Finally, a summary is presented and conclusions are drawn from the programme of research.

CHAPTER 1

A LITERARY REVIEW

1.1 Introduction

The numerical modelling of tidal propagation in shelf seas was not possible until the late 1960s when computers became large enough to carry out the vast numbers of calculations necessary to execute even a coarse-mesh two-dimensional representation. A classic example of such a model was that by Heaps (1969) in which the tidal propagation of the North-West European Shelf was modelled using finite differences (see Chapter 2, section 2.2.4) applied to a mesh of 20' in latitude by 30' in longitude and containing some 2400 cells. This model was adequate in most places for examining the vertical tide, which varies but slowly in space. Two-dimensional models are based on the depth integrated equations of motion and continuity (see Chapter 2) and as such, provide results in the form of vertical tidal elevations and depth-averaged stream velocities. The results are generally good for the vertical tide in sea areas which are well-mixed. Difficulties, however, arise in areas which become dry as the tide falls and become wet as it rises again.

1.2 State-of-the-Art in Modelling Flooding and Drying Banks

In order to place in a wider perspective the problem of representing the flooding and drying of intertidal flats in a numerical tidal model, attention may be drawn to the weaknesses inherent in some commercially available two-dimensional numerical models.

In a review of state-of-the-art marine transport models, Stripling *et al.* (1994) have assessed the ability of currently available commercial models to predict the fate and dispersion of pollutants in the marine environment. These models are highly modular, in that separate modules are required for the prediction of tidal propagation, sediment transport, pollutant transport, the effect of wind-waves, and the chemistry involved with differing pollutants. To ensure that predictions of the fate and dispersion of pollutants are as accurate as possible, it is first essential to establish a tidal model that represents the tidal physics accurately.

Stripling *et al.* (1994) showed that the current level of technical expertise and presentation was generally excellent. It also indicated areas of development in which knowledge of processes is scant, with the outcome that one can not have full confidence in all of the results obtained.

It is beyond the scope of this thesis and would contravene contractual obligations to provide specific details of the companies reviewed and their modelling procedures. It is possible, however, to draw attention to a particular area of weakness, and therefore potential development, while maintaining complete company anonymity.

The research for the above report has shown that commercial companies have spent considerable time addressing the incorporation of a moving land/sea boundary. One company has found that the introduction of a moving land/sea boundary can have a severely detrimental

effect on tidal solutions both in close proximity and at a distance from intertidal regions. Another company has had to omit wetting and drying processes completely from one of their models, with the operator having to decide on whether the intertidal regions are to be modelled as permanently wet or permanently dry. This is evidently unrealistic.

1.3 Early Work on Wetting and Drying (Reid & Bodine)

As early as the 1960s, it was well recognized that there were numerical problems associated with the accurate representation of the tide through the solution of the depth-averaged hydrodynamic equations. For instance, the advection terms, which are non-linear, were often ignored for the purpose of ease of calculation (Reid & Bodine, 1968). More important in the numerical representation of the drying and subsequent flooding of inter-tidal flats is the mathematical singularity in the frictional term as the water depth approaches zero. Other workers, e.g. Sielecki & Wurtele (1970), evaded this problem by ignoring the friction term completely.

The earliest attempt to attack the problem of this singularity was that by Reid & Bodine (1968), who, although they were not able fully to represent the non-linear terms, tried to treat the singularity numerically. This early scheme involved the application of weir theory to two-dimensional flow in the sea. The conditions necessary for the expected flooding or drying of a grid square are given by Evans (1987) and are quite thorough. Contrary to almost all subsequent work (Leendertse & Gritton, 1971; Flather & Heaps, 1975; Falconer, 1983, 1984 & 1985; Owens, 1984; Stelling *et al.*, 1986; Falconer & Owens, 1987; Flather & Hubbert, 1989) the use of weir theory avoided the incorporation of a critical depth to prevent the

singularity occurring in the frictional term. A predetermined depth was still necessary, however, to indicate that a cell had dried and to induce the setting of the stream velocities to zero. A cell was deemed to dry when the stream velocities fell below a preset value (0.02 ms^{-1}). This scheme was highly artificial, since the natural sea-bed does not consist of cells bounded by barriers which create weir-like flow. However, for Reid and Bodine's study this representation was adequate.

Reid and Bodine (1968) were severely restricted by the processing power of their computer since they had to run their numerical scheme on a GE225 computer which possessed only an 8K memory. In the subsequent few years, computer power increased considerably and more complex numerical techniques could be utilized.

1.4 The Approach of the Civil Engineer

Since shelf-sea modelling became a reality, the need to represent the movement of the land/sea boundary has been one addressed by oceanographers and civil engineers alike.

Essentially, a civil engineer is very practical in his approach to the construction and use of a numerical model to represent the tide. Large sums of money are available to him in his quest to solve a problem; he may have to make decisions that could affect large areas of land and water, and possibly the lives of many people in the vicinity of a proposed project. However, the need to abide by strict time-limits, under the dictum that 'time is money', may lead to an engineer 'cutting corners' at appropriate moments in his appraisal of a situation. For instance, he may decide that it is more important to maintain stability in a numerical scheme than to worry about how he maintains that stability. Thus, accuracy in areas such as

a moving land/sea boundary may not be of prime concern to a civil engineer.

Civil engineers generally have no need to develop numerical models on the oceanic or shelf-sea scale. In fact, a civil engineer is largely concerned with smaller scale modelling such as that pertaining to estuaries, harbours or small bays. His models typically have a mesh-size of tens of metres. This is usually adequate for his contractual obligations where localized conditions are of direct importance.

1.4.1 Leendertse's models

Leendertse (1970) developed a water quality model for well-mixed estuaries and coastal seas. The computational procedures were tested on models of Jamaica Bay, Long Island, New York - a shallow bay containing many tidal flats. This appears to be the first thorough approach to the problem of modelling flow over tidal banks which flood and dry.

The work of Leendertse (1970) was primarily concerned with introducing to his peers, and the public, a quantitative method of assessing the impact of urbanization on the environment, in particular 'the study of technical alternatives in the management of fluid waste discharges in well-mixed estuaries and coastal seas', a subject which is still very much of public concern and is still being addressed by governmental and private sector bodies.

Leendertse's study involved a testing of computational procedure on models of Jamaica Bay, Long Island, New York. A grid size of 500ft (150m) was used. However, a larger grid size of 833.3ft (250m) was used for a series of experiments in which stability was studied. This model was not of the moving land/sea boundary type and, in fact, the land/sea boundary was fixed by the minimum depth being set at 3ft below mean sea-level, highlighting the problems

encountered coping with a moving land/sea boundary as the grid size increases.

In 1970, Leendertse made his first attempt to cope with a moving boundary. As he states, the principle may be simple, but the computational procedures are very complex. His simple principle was that if a cross-section (see figure 1.1, where the water level is shown by ζ, and the cross-sections are shaded) for any particular point decreased to less than a preset value, then that point should be removed from the computational field, the preset value being positive and close to zero. Since finite difference techniques rely upon variables changing gradually in time and space, discrete changes such as those experienced when the land/sea boundary moves, impart discontinuities into the solution. Leendertse overcame this initially by making the check for the location of the new boundary at intervals larger than the time-step, allowing the disturbance generated time to decay before the next check for the boundary location. These disturbances were suppressed by enlarging the bottom friction values artificially as the water became very shallow.

When the water level rose again, grid points were returned to the computational field if an average of the water depths at any of the surrounding grid points which were wet resulted in cross-sections which were greater than the preset value. Thus the average water level all the way around the 'dry' point had to rise above the level left in that cell before the point was considered wet again.

1.4.2 Leendertse & Gritton

In Leendertse & Gritton's report of 1971, the procedure of Leendertse (1970) was modified to include further drying checks and an important alteration to the process of the 'wetting'

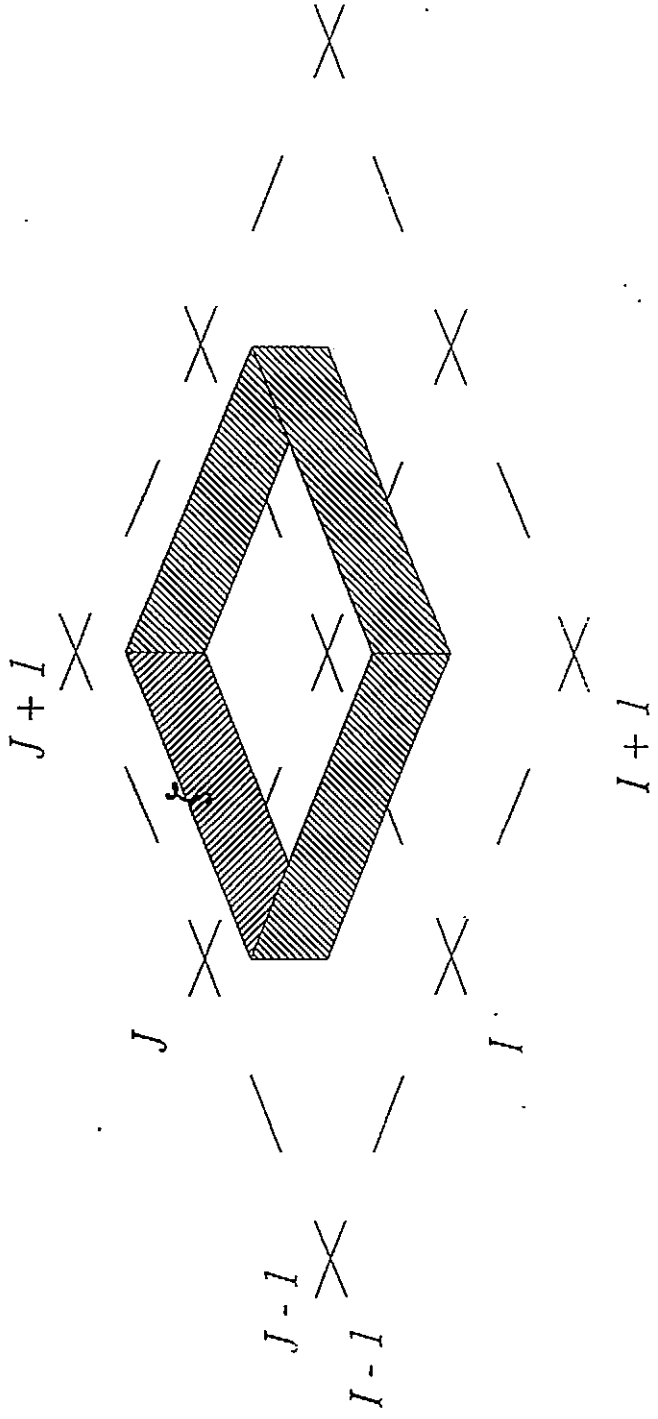


Figure 1.1: Diagrammatic representation of the cross-sections of a cell

of a cell. One of the new additions to the drying check was carried out at every time-step at the time horizon for the calculation of the tidal stream components. This states that, 'if at a particular grid point, the new computed value of the water level has decreased so that a negative volume is obtained, then that grid point is taken out of the computation'. This meant that if the tidal elevation fell below the depth specified at the elevation-point of the grid cell, then the cell was deemed dry. Then, before proceeding to the solution in the next row or column, the velocities and water levels were recalculated for the previous row or column with the point now taken as dry, as well as a recalculation of the row or the column (depending on which direction the calculation is in at the time) on which the recently dried out cell lay. When a cell dried, it was assumed to hold a thin layer of water over it. The depth of this water was set at the previous time-step before a negative volume was obtained.

Another additional check carried out every time-step after the flow calculations was to evaluate all four cross-sections associated with each of the four adjacent grid points. If any of these were negative, the point was noted.

Physically, a negative cross-sectional area is meaningless, but since these cross-sections are concerned with mass-transport, it simply means that transport is worked out as being in the opposite direction to which it should be.

At the end of the calculation of all the velocity components, the grid points marked as having a negative cross-section adjacent to them had all velocity components leading to that point set to zero. A thin layer of water was left over the cell in this case as well.

Leendertse and Gritton found that these two additions had very little effect in adding discontinuities to the tidal solution. This was because a large proportion of cells dried

through the initial check for drying, while only a few would dry as a result of negative volumes or cross-sections, because the preset value was always greater than zero, thus removing a cell before the volume or cross-section had a chance to become negative.

In addition to the initial process for the wetting of a cell *i.e.* the average surrounding wet point depths had to be above the depth of the water left in the dry cell, a dry grid point could become wet only if all of the adjacent transport cross-sections were positive as well. That being the case, then the water level was set to that which was over the cell when it was 'dry'.

The paper by Leendertse & Gritton (1971) does not contain any validation data so the accuracy and performance of the technique remains undisclosed. However, Owens (1984) suggests that the use of this technique is restricted in its ability to model on larger scale grids due to the unrealistic amount of water left in a grid square that is deemed to have dried out. It seems probable that the drying technique developed by Leendertse & Gritton (1971) was not introduced as an accurate representation of the natural phenomena but merely to satisfy the numerical requirements (that is, to ensure stability) of the models.

1.4.3 Falconer's models

By far the closest work to that of J. J. Leendertse in the quest to represent a moving land/sea boundary accurately in a numerical model is that of Prof. R.A. Falconer. Falconer began his interest in numerical modelling in the late 1970s, possibly encouraged by the need at that time to study, in detail, tidal flushing characteristics and circulation in harbours and estuaries in order to provide solutions to water-quality problems. Hydraulic engineers had, traditionally, used physical models to predict water movement in estuaries but, by 1980, computing

knowledge had reached a level where numerical models were just beginning to replace physical models in this field. Thus, numerical techniques had to be developed and improved upon by engineers themselves. Falconer was one engineer who developed a keen interest in these numerical methods. He followed Leendertse and consequently he adopted the ADI (alternating direction implicit) numerical scheme as used by Leendertse.

His early models (Falconer, 1980; 1984; 1985) were used to simulate the propagation of the tide within harbours and estuaries.

However, it was not until 1984 that he published details (Falconer, 1984a) on how he had confronted the problem of a moving boundary. Wessex Water Authority had two proposals for modifying the shape of Holes Bay, a creek in Poole Harbour in Dorset. They had asked Falconer to examine the effects of the two proposals on the tidal characteristics of the bay. Essentially, he used Leendertse's (1970) and Leendertse and Gritton's (1971) procedure. Later, Falconer (1984b), he was required to investigate the velocity fields and temperature distributions (very difficult to do using a physical model) that would be produced by the siting of either a 700 or 350MW capacity power station on the perimeter of Poole Harbour. He stated in this paper that a main problem was 'significant changes' which occur in the plan cross-sectional area as large regions of shallow water were dried out and flooded on each tide'. The procedure adopted in simulating this moving boundary was similar to that used by Leendertse (1970) and Leendertse and Gritton (1971) with some modifications, and the same as that used previously by Falconer (1984a). He reported that the drying regions were 'successfully modelled'. However, lack of field calibration data rendered the numerical predictions as merely a guide.

Falconer persisted with his representation of flooding and drying processes for work carried out in Port Talbot Harbour (Falconer, 1985). He used the same procedure as in Falconer (1984b) and reported that 'the mathematical reproduction of the flooding and drying of the shallow water region at the upper end of the harbour appears to be satisfactorily simulated'. In 1986 (although this work was actually conducted in 1984) Falconer published more work on Holes Bay, Poole Harbour. A slight alteration appeared in this paper in that all four cross-sections of a wet cell were checked for water-level as oppose to just the two cross-sections in the direction of the calculation. This was necessary in order to account for modifications he had made to the finite difference equations.

1.4.4 Owens' model

In 1984, Falconer supervised a research student, P. H. Owens, who developed an improved technique for representing the drying and flooding of tidal flats. He addressed the problem encountered when using Leendertse's method that this method cannot cope with large grid sizes. This is because the water elevation can change considerably with each time-step, thus requiring that the preset limit at which the minimum water-level can be is excessively large. Owens (1984) used a one-dimensional basin with a grid size of 1200m - large in hydrographic engineering terms - as a study base for his improved algorithm. He adopted an approach which was designed to reduce the magnitude of each discontinuity occurring. This was different from Leendertse who tried to reduce the number of discontinuities. To achieve this he allowed cross-sections to 'disappear' one at a time as each became negative, but only deemed the cell dry when the last cross-section became negative.

1.4.5 Other models

Leendertse & Gritton's (1971) technique has, in the main, been adopted by many subsequent authors; Xanthopoulos and Koutitas (1976) used their approach to model the propagation of a flood wave over a flat plain, although not strictly modelling inter-tidal flats, this experiment still required the representation of a moving boundary. Yeh and Chou (1979) were also inspired by Leendertse's (1970) approach in their attempt to introduce a moving land/sea boundary in their prediction of storm surge levels. Their paper compares the results of a fixed boundary model to those obtained using a moving boundary model and concluded that the moving boundary model over-predicted surge levels less than the fixed boundary method. It also stated that, for a very steep bed-slope, there was very little difference between the solutions obtained from a fixed boundary and a moving boundary model. Gunn and Yenigün (1985) claimed to have developed a method which provided solutions in 'good agreement' with observations, over the entire extent of the model field. They used a small grid size of 250m, which required a time-step of less than 12s. Using an estimate of the rate of flood and ebb of the tide over an inter-tidal flat of $1ms^{-1}$, they concluded that for a grid size of 250m, it would be necessary to check for a moving boundary only every 250s. This meant that approximately every twenty time-steps, the position of the boundary was located. This has similarities to Leendertse's (1970) technique in that any discontinuities caused could decay in the time between checks.

Despite their claim that 'good agreement' was achieved throughout the model area, graphical representations in their paper suggest that differences of up to 30cm could occur between computed and observed values.

Stelling *et al.* (1986) demonstrated some of the effects on tidal solutions of cells which are in close proximity to cells which wet and dry. Sensibly, they have stressed that numerical procedures must be carefully considered before calibration is attempted by adjusting physical properties. Although Stelling *et al.* (1986) appear to have compared three different methods of representing the moving land/sea boundary, the procedures adopted in their entirety are almost equivalent to Leendertse and Gritton's (1971) attempt. Therefore, it cannot really be considered as an analysis of three different methods (as is the case with this research). However, this is a valuable study as it addresses solely and directly the problem of how to represent a moving land/sea boundary.

1.5 The Approach of the Oceanographer

For the oceanographer, all space scales are considered from global scale modelling to models of turbulence. He has interest in a wide range of oceanographic modelling problems, particularly in terms of scale, and therefore his approach to finding a solution to a specific problem will essentially be different to that of a civil engineer. Numerical modellers at the Proudman Oceanographic Laboratory in Birkenhead, e.g. Davies, Flather, Prandle, Proctor, Jones, are actively involved in the development of techniques in modelling.

1.5.1 Flather & Heaps

Flather & Heaps (1975) published a paper entitled 'Tidal Computations for Morecambe Bay'. In this paper a finite difference method for solving the shallow water hydrodynamic equations

was introduced, with different conditions defining whether a grid square had become wet or dry. These included a testing of the local water-depth and an examination of the slope of the free surface between adjacent grid squares. The testing of the local water-depth involved comparing the present water-depth in a grid to a predetermined minimum depth. Within the model this prevented the actual depth from falling below the preset minimum depth, terminating the growth of the frictional dissipation before it became too large and before the water depth became so shallow that the physical laws expressed in the model were no longer applicable. The restriction invoked in the model by the introduction of an examination of the slope of the free surface had the effect of preventing artificial grid-scale oscillations which may occur causing a grid square to alternate between being wet and being dry at each time-step calculation.

The results of this model showed clearly that grid squares near to grid squares that were becoming dry were receiving 'shocks' as the squares dried out. These 'shocks' were seen as discontinuities in the tidal profiles that were reproduced. It is in this sense that Flather & Heaps' (1975) paper is important since it clearly demonstrates the problems associated with the mathematical modelling of flooding and drying banks. Their drying procedure has recently been adapted by Flather & Hubbert (1989) in an attempt to reduce the severity of the 'shocks' induced by the discrete movements of the land/sea boundary.

1.5.2 Flather & Hubbert

Flather & Hubbert's (1989) adaption of Flather & Heaps' (1975) drying procedure was based

upon the principle that if, when a grid square dries out, a shock is imparted through the model, then an attempt should be made to reduce the severity of this shock. Flather & Hubbert (1989) applied their new drying procedure to the same area as Flather & Heaps (1975) but to achieve resolution of subgrid-scale features, they digitized the bathymetry in part of Morecambe Bay at ten times the resolution of their basic model. It was in this part of the Bay that they applied their new experimental procedures. They developed an idea using area and breadth factors. These were factors which could change according to the ratio of the wetted area of a grid square to the total area of the grid square, and the ratios of the breadth over which flow occurred to the total breadth of the grid square.

Flather & Hubbert's results showed that having constant area and breadth factors made little difference to the time series obtained from the experiments they did using solely a two-dimensional scheme, and were apparently considerably noisier. However, the experiments in which the wetted area of a grid square was allowed to vary with time substantially reduced the discontinuities by representing the flooding and drying process 'in a physically more realistic manner'. Unfortunately, the practicalities of digitizing bathymetric data to ten times the basic resolution in order to obtain a variable factor is a severe hindrance in the application of this drying procedure.

CHAPTER 2

FORMATION OF A STUDY TOOL

In order that the flooding and drying of intertidal flats may be studied numerically, it is necessary to formulate a numerical model which must be readily adaptable and of good solid foundation. Drying algorithms tend to be rigorous in character, and therefore the source code of the numerical model must be able to cope adequately with whatever algorithms are incorporated within it. It is through long periods of experimentation that a source code becomes the aid that it is required to be. Many problems are encountered during and after the development stages and must be dealt with accordingly; a model simply cannot work if even small errors are left unaltered.

It is important to realise that the problems associated with the incorporation of a moving land/sea boundary do not exist when the numerical model is three-dimensional, since layers within the vertical coordinate representation can be reduced until none remain. The need for reduced computational costs however, requires that accurate tidal predictions are obtained from the more economic two-dimensional depth integrated non-linear hyperbolic differential hydrodynamic equations. These are considered appropriate when modelling the propagation

of the tide in non-stratified, *i.e.* well-mixed, water columns.

A two-dimensional numerical model of tidal propagation is based upon the solution of the hydrodynamic equations in two dimensions.

2.1 The Hydrodynamic Equations in Two Dimensions

The hydrodynamic equations are an expression of the physics of tidal propagation in mathematical notation. They are obtained from Newton's Second Law of motion (1687) which states, crudely, that the rate of change of momentum of a particle of constant mass is proportional to the force acting upon it in the direction of that force. The hydrodynamic equations express Newton's Second Law but are more complex than the crude definition stated above since each 'water particle', or unit mass of water, is subject to three forces, three accelerations and in three dimensions. Equations 2.1, 2.2, and 2.3 below show the easterly, northerly, and upwards components of the hydrodynamic equations (e.g. George, 1994):

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} - 2\omega v\sin\phi = -\frac{1}{\rho}\frac{\partial p}{\partial x} - \frac{1}{\rho}\frac{\partial \tau_{xz}}{\partial z} \qquad (2.1)$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} + 2\omega u\sin\phi = -\frac{1}{\rho}\frac{\partial p}{\partial y} - \frac{1}{\rho}\frac{\partial \tau_{yz}}{\partial z} \qquad (2.2)$$

$$\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} + 2\omega u\cos\phi = -\frac{1}{\rho}\frac{\partial p}{\partial z} - g \qquad (2.3)$$

A B C D E F G

where,

A = Temporal accelerations, ms^{-2}

B = Advective accelerations, ms^{-2}

C = Coriolis accelerations, ms^{-2}

D = Pressure forces / unit mass in the easterly, northerly and upward directions, ms^{-2}

E = Frictional force / unit mass in the easterly direction, ms^{-2}

F = Frictional force / unit mass in the northerly direction, ms^{-2}

G = Gravitational force / unit mass, ms^{-2}

These equations are sometimes referred to as the Navier-Stokes[2] equations.

In the ocean, shallow waters and rivers, the three forces acting on a particle are those of pressure, gravity and friction per unit mass. The three accelerations experienced by a particle are: (*i*) Coriolis acceleration, which is due to the rotation of the Earth, (*ii*) the temporal acceleration, *i.e.* the change of a particle's velocity with time, and (*iii*) the advective acceleration, due to the fact that the particle changes its position in space. Equations 2.1, 2.2, and 2.3 are the equations of motion in three dimensions and express the balance between these accelerations and forces per unit mass.

Large computing power and storage are required to solve these non-linear three-dimensional equations, so two-dimensional versions are often used in preference. Two-dimensional equations of motion are obtained from the three-dimensional equations of motion by integrating them vertically with respect to depth from the bottom to the surface of the water column. When integrated, the unknown velocities u and v in equations 2.1 and 2.2 become U and V where;

$$U = \frac{1}{(\zeta + h)} \int_{-h}^{\zeta} u\,dz \qquad \textit{and} \qquad V = \frac{1}{(\zeta + h)} \int_{-h}^{\zeta} v\,dz$$

[2]Named after L.M.H. Navier (1785-1836) and G.G. Stokes (1819-1903).

Since in a two-dimensional model there is no vertical gradient, $\partial w/\partial z = \partial w/\partial t = \partial w/\partial x = \partial w/\partial y = 0$; equation 2.3 thus reduces to the hydrostatic equation and becomes unnecessary. The water column is now considered as one unit from sea surface to sea bed in that the whole of the water column moves in the same direction and with the same velocity within each cell of the model. After integration, equations 2.1 and 2.2 become equations 2.4 and 2.5 which are a two-dimensional representation of motion in an easterly and a northerly direction respectively;

$$\frac{\partial U}{\partial t} + U\frac{\partial U}{\partial x} + V\frac{\partial U}{\partial y} - 2\omega V\sin\phi = -g\frac{\partial \zeta}{\partial x} - KU\frac{\sqrt{U^2+V^2}}{(h+\zeta)} + \varepsilon\left(\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2}\right) \quad (2.4)$$

$$\frac{\partial V}{\partial t} + U\frac{\partial V}{\partial x} + V\frac{\partial V}{\partial y} + 2\omega U\sin\phi = -g\frac{\partial \zeta}{\partial y} - \frac{KV\sqrt{U^2+V^2}}{(h+\zeta)} + \varepsilon\left(\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2}\right) \quad (2.5)$$

where;

U = depth averaged velocity in an easterly direction, ms^{-1}.

V = depth averaged velocity in a northerly direction, ms^{-1}.

ω = period of Earth's rotation, s.

ϕ = latitude, °N.

g = acceleration due to Earth's gravity, ms^{-2}.

ζ = tidal elevation, m.

K = coefficient of drag.

h = bed elevation below a fixed datum, m.

ε = eddy viscosity, m^2s^{-1}.

$\partial t \equiv \Delta t$ = time step, s.

$\partial x, \partial y \equiv \Delta x, \Delta y$ = grid spacing in x and y directions, m.

To ensure that no water enters or leaves the system, a two-dimensional representation of continuity of volume, equation 2.6, is required. In this the density of the water is considered uniform with depth, an assumption based on the premise that the water column is well-mixed

and that sea-water is incompressible. The continuity equation is derived from the volume of water entering and leaving a cuboid (see for example; Proudman, 1953):

$$\frac{\partial \zeta}{\partial t} + \frac{\partial}{\partial x}\{(h+\zeta)U\} + \frac{\partial}{\partial y}\{(h+\zeta)V\} = 0 \qquad (2.6)$$

2.2 Solution of the Hydrodynamic Equations

Several techniques are available for the solution of equations (2.4), (2.5) and (2.6), which form a system of two-dimensional non-linear partial differential equations of the hyperbolic type. These may be the method of characteristics, harmonics, finite elements or finite differences.

2.2.1 The Method of Characteristics

This method relies on the fact that it is a known characteristic of the hydrodynamic equations that they are hyperbolic by nature. Since the characteristics of the equations are known, a solution may be projected forward (in time, perhaps) to where the hyperbolic nature is upheld. The method of characteristics is often used in the modelling of compression waves or situations when a wavefront is progressing through a non-disturbed medium (Jeffery, 1976). It may also be used in the solution of the hydrodynamic equations (Dronkers, 1964) in estuaries.

Characteristics are designed for calculations where discontinuities are prevalent. This would therefore make them seem ideal for the modelling of flooding and drying regions. In fact,

however, this is not the case.

In order to solve two-dimensional problems using the method of characteristics, much time is needed to write the machine coding. Once the coding has been written, the solution is very expensive computationally due to the excessive amounts of interpolation needed to maintain a consistent grid pattern.

Whilst the solution may be precise, and the physical attributes of the equations being solved are followed closely, the extraordinarily complex programming and expensive solution process have meant that tidal modellers have abstained from using the method of characteristics for problems of higher order than one-dimension. In addition, when the point spacing is large (as would be required in this research), truncation errors become significant and convergence of the solution is delayed considerably.

2.2.2 The Harmonic Method

This method of solution makes use of the fact that the dependence of tidal motion on time is known, and that the frequencies at which the tidal motion is forced are known very precisely.

At present it is not possible to include intertidal regions in the solution of the hydrodynamic equations using the method of harmonics.

This is an new branch of tidal modelling that has yet to become established, at least in the U.K., however, there is some British interest developing (e.g. George, 1995).

2.2.3 Finite Elements

Figure 2.1 shows an example of a finite element grid. In order for this type of grid to be

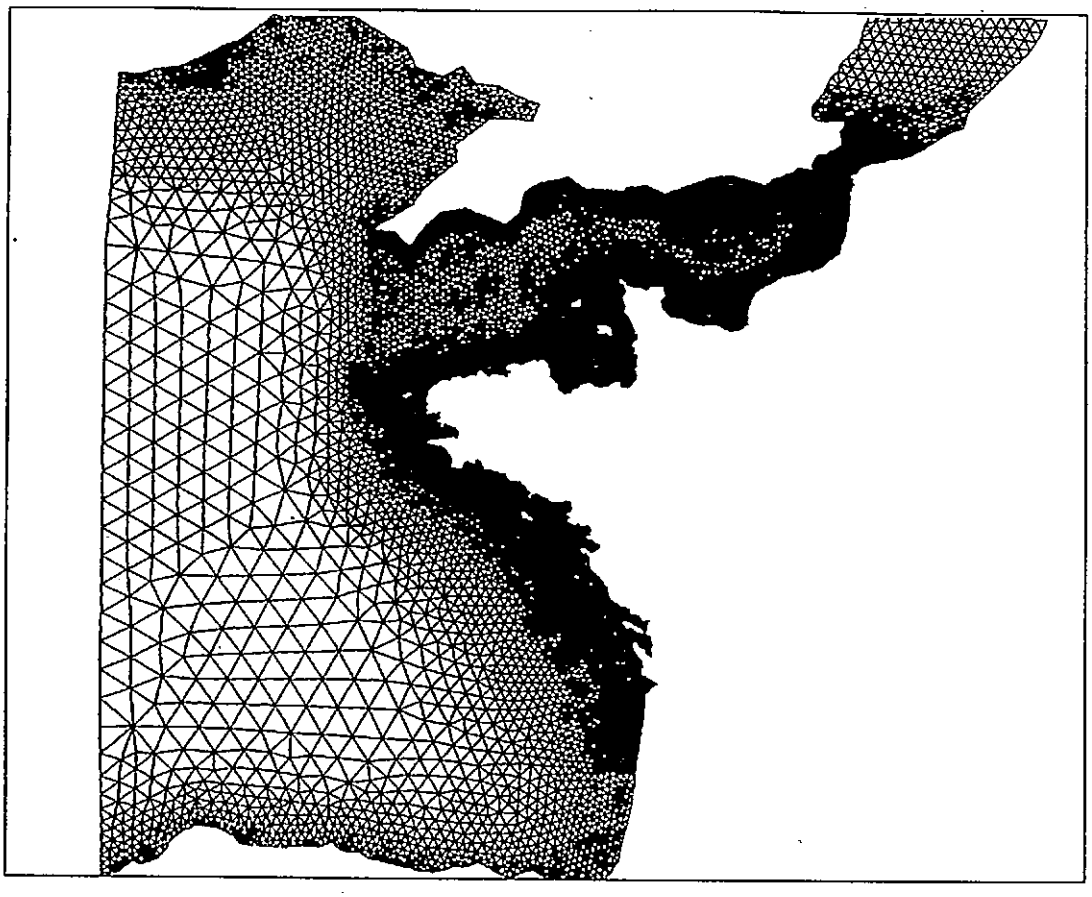


Figure 2.1: A finite element grid showing a wide range of density of cells (HR Wallingford)

established, it is necessary to purchase a standard commercially available finite element grid generator, such as I-DEAS or SIMAIL (Hydraulics Research Ltd., 1993).

The solution of the hydrodynamic equations using a finite element method was, until very recently, computationally expensive. Holz and Witham (1977) present a moving boundary scheme which is analogous to Leendertse (1970) using a fixed-grid finite-element approach. Lynch and Gray (1980) proposed an improved method of incorporating the moving boundary by allowing the boundary of the finite-element grid to deform as the land/sea boundary moved. Recent developments, in the field of finite-elements, have been made (Hervouet, 1991) for use by Electricité de France in the French nuclear industry. The improvements made are dependent upon the use of a vector computer to store the vast numbers of arrays associated with the finite-element scheme. This can decrease the computational time needed by twenty to one hundred times that of a conventional finite-element scheme.

However, finite-element representations are computationally complex and have therefore not been used here.

2.2.4 Finite Differences

It was decided to perform the entire numerical procedure using finite differences on a Richardson grid (Figure 2.2). This grid avoids instabilities by requiring that the unknown terms in equations (2.4), (2.5) and (2.6) are all evaluated at different locations in space.

A finite difference technique involves taking the differences in variables (for instance the U and V stream components) over a finite distance in space, normally a divisor of the grid size, and using these differences to find the unknowns within the equations themselves. For instance, the horizontal eddy diffusion term:

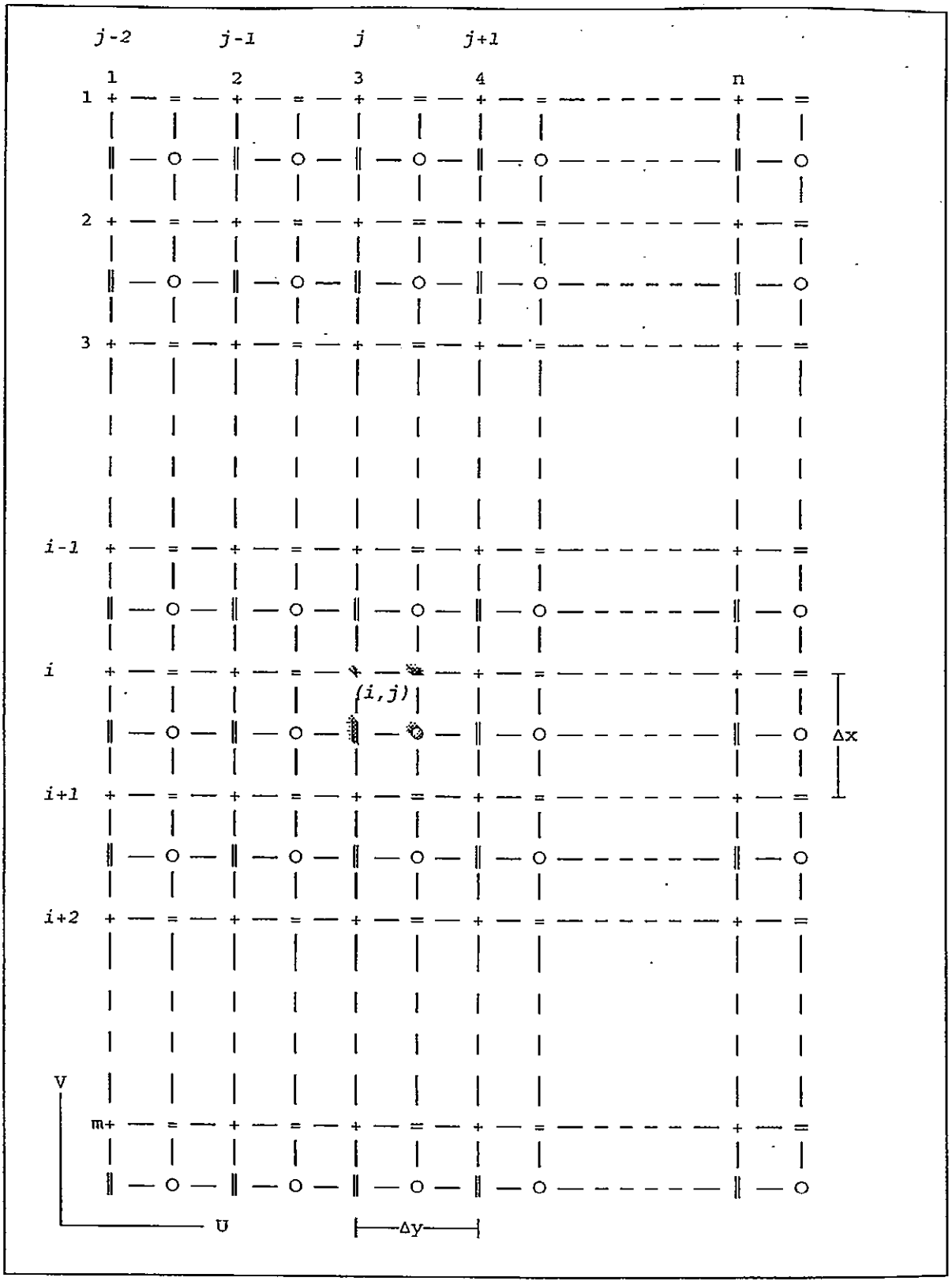


Figure 2.2: Richardson grid used in the model showing grid numbering and labelling ('+': ζ-point, '=': U-point, '‖': V-point, 'O': h-point)

$$\varepsilon\left(\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2}\right) \tag{2.7}$$

when taken a component at a time expressed as differences:

$$\frac{\partial^2 U}{\partial x^2} = \frac{\left(\frac{\partial U}{\partial x}\right)_{(i,j+1)} - \left(\frac{\partial U}{\partial x}\right)_{(i,j-1)}}{\Delta x} = \frac{\frac{U_{(i,j+1)} - U_{(i,j)}}{\Delta x} - \frac{U_{(i,j)} - U_{(i,j-1)}}{\Delta x}}{\Delta x} \tag{2.8}$$

$$\therefore \quad \frac{\partial^2 U}{\partial x^2} = \frac{1}{(\Delta x)^2}\{U_{(i,j+1)} + U_{(i,j-1)} - 2U_{(i,j)}\} \tag{2.9}$$

and;

$$\frac{\partial^2 U}{\partial y^2} = \frac{\left(\frac{\partial U}{\partial y}\right)_{(i-1,j)} - \left(\frac{\partial U}{\partial y}\right)_{(i+1,j)}}{\Delta y} = \frac{\frac{U_{(i-1,j)} - U_{(i,j)}}{\Delta y} - \frac{U_{(i,j)} - U_{(i+1,j)}}{\Delta y}}{\Delta y} \tag{2.10}$$

$$\therefore \quad \frac{\partial^2 U}{\partial y^2} = \frac{1}{(\Delta y)^2}\{U_{(i-1,j)} + U_{(i+1,j)} - 2U_{(i,j)}\} \tag{2.11}$$

would become:

$$H.E.D. = \frac{\varepsilon}{(\Delta x)^2}\{U_{(i,j+1)} + U_{(i,j-1)} - 2U_{(i,j)}\} + \frac{\varepsilon}{(\Delta y)^2}\{U_{(i-1,j)} + U_{(i+1,j)} - 2U_{(i,j)}\} \tag{2.12}$$

A finite difference form of each and every component within the hydrodynamic equations can then be obtained. These expressions are then incorporated as algorithms in a programming language, such as FORTRAN, and solved for 'the unknowns'.

2.2.5 Non-linearity

The non-linearity of the advection of momentum in the hydrodynamic equations is often ignored for the purpose of ease of calculation (e.g. Reid & Bodine, 1968). The equations solved here, however, are fully non-linear and incorporate all the advective terms.

2.2.6 The Courant Number

To achieve stability in an explicitly solved numerical model, a suitable time-step must be chosen. This time-step is dependent upon the maximum depth of a grid cell in the model area and is calculated using the Courant-Friedrichs-Loewy criterion expressed in equation 2.27:

$$C = \Delta t \sqrt{g h_{max}} \left\{ \frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} \right\}^{\frac{1}{2}} \qquad (2.27)$$

where,

C = Courant number (< 1 for stability, see section 2.3.3)
Δt = time-step, s
Δy = meridional grid size, m
Δx = zonal grid size, m
h_{max} = maximum depth of the model area, m.

Since the ideal model presented here is of an area small enough for the meridional grid size not to be significantly affected by the change in latitude, and that we try to define the area

as a series of squares, we can assume that $\Delta x \approx \Delta y$. Equation 2.27 then becomes equation 2.28 below.

$$C = \frac{\Delta t \sqrt{2gh_{max}}}{\Delta y} \tag{2.28}$$

As the grid size changes or the maximum depth of the model changes, the time-step will have to be altered accordingly. For instance, in a morphodynamic model where the bathymetry of the sea-bed is constantly changing, if the maximum depth increases significantly then instabilities could be induced. In this case, a restriction must be imposed on the maximum attainable depth in the model or the time-step must be altered to account for the deeper water.

2.3 Choice of Discretization in Time

Once discretization of the hydrodynamic equations has been established in space, it is then necessary to discretize the solution procedure in time. The interval at which time is advanced is known as a time-step.

In order to maintain stability during the solution of the hydrodynamic equations, it is necessary to stagger the process of establishing the unknowns, i.e. ζ, U and V. There are several methods that may be adopted in order to time-stagger the solution procedure.

2.3.1 Implicit

An implicit process is one in which there are several unknowns in each equation. The equations must, therefore, be solved simultaneously at each whole time-step. One method

adopted is that of Gauss elimination and back substitution (e.g. Smith (1978)). This requires the setting up of large sparse matrices, leading to lengthy computations. Owens (1984) gives the appropriate finite difference equations.

Implicit schemes are unconditionally stable in theory, implying that there is no limit to the length of time-step. In practice, however, large time-steps produce inaccurate solutions and therefore an accuracy constraint based on the Courant number and the speed of propagation of the tidal wave is necessary, thus limiting the time-step.

The popular 'Alternating-Direction-Implicit' scheme (ADI scheme) is so-called because of the implicit manner of solution and the fact that all the streams in one direction are evaluated, after which the streams in the remaining direction are evaluated. In areas of intricate geometry, however, this scheme does not work well.

Although time-steps can be large with an implicit manner of solution, thus giving a shorter overall run-time, the complexity of the programming negates straight-forward implementation of various comparative algorithms.

2.3.2 Semi-Implicit

The semi-implicit scheme has been adopted by various authors (e.g Casulli, 1990). This scheme introduces a time-splitting of the advection terms from the rest of the terms. The advective terms are evaluated explicitly at a set time interval, while the rest of the terms are evaluated implicitly at a time interval less than the interval for the advective terms, hence the term semi-implicit. This avoids the complex programming that is required in the solution of the advective terms in an implicit manner, while still allowing large time-steps. This method of solution has not been chosen here.

2.3.3 Explicit

In explicit schemes, there are fewer unknowns than in implicit schemes and the equations can therefore be solved by rearranging. In a central explicit scheme, the time-step is split into two parts. During the first half time-step, the velocities are evaluated from the elevations given in the previous time-step. In the second half-time step, the new velocity values are used to establish the new tidal elevations.

Explicit schemes are conceptually the simplest of all the solution methods and are also the most straight-forward to programme. The method of solution adopted here is wholly explicit in that subsequent solutions of the unknowns are obtained from solutions from a previous time-step. Hence the solution procedure can be known as an 'explicit forward time-stepping finite-difference numerical model'.

In an explicit model, the Courant number must be less than one for the stability criterion (equation 2.28) to be satisfied *i.e.*

$$1 \geq \frac{\Delta t\sqrt{2gh_{max}}}{\Delta y} \tag{2.29}$$

For the idealized model described in section 2.5, which may have:

$\Delta x = \Delta y = 5000m$,

$h_{max} = 54m$,

then, to satisfy inequality 2.29:

$$1 \geq \frac{\Delta t\sqrt{2\times9.81\times54}}{5000}$$

or,

$$\Delta t \leq \frac{5000}{\sqrt{2\times 9.81\times 54}} \leq 153.61s \qquad (2.30)$$

giving a time-step of less than or equal to 153.61s. In fact, Δt=124.2s is a convenient value to choose, it being one three hundred and sixtieth of a tidal cycle. Therefore, one time-step becomes equivalent to one degree of tidal time; thus not only are there 360° in the cycle but also 360 time-steps.

2.4 Instability

Even experienced modellers that have been practising for perhaps ten years or so are still plagued by instabilities within their numerical schemes. It is through the application of differing techniques that 'weak spots', *i.e.* potential sources of instability, in the source code become apparent, be they simply typing errors that have passed unnoticed for a period of time or more serious anomalies (which are generally harder to locate) such as accidentally taking differences over incorrect positions in space or the incorrect labelling of variables involving the misuse of values in subsequent routines.

The above errors may or may not induce instabilities, as there are many possible reasons why instability should occur. Even programming structure may create the necessary conditions. The actual sources of instabilities are extremely awkward to pinpoint, but even so, for every small error that is eradicated during the search for the cause, whether the cause is found immediately or not, the source code contains less potential for instability. The result is that actual source programs become stronger and more robust.

2.5 A Numerical Model of an Idealized Portion of Shelf Sea

A source code written in FORTRAN, called MODKS.FTN77 (see Appendix 1), was developed by the Author to solve the equations of motion (2.4), (2.5) and continuity (2.6) in two dimensions. It is based on models by George & Evans (1991) and by Prandle (Proudman Oceanographic General Purpose Model).

Specification for a numerical model of an idealized portion of shelf sea has been laid down for the comparison and contrasting of methods, principles and algorithms which represent the physical phenomenon of tidal flooding and ebbing. A theoretical bathymetry has been assumed while tidal conditions for the area have been chosen as roughly that of an area of the North Sea off the Humber coast.

It became apparent that a theoretical model design has a great advantage over a model of a real sea area for research purposes in that it is readily adaptable and is not bound by the need to follow rigid guidelines. It is also not restricted by validation processes, since these are not necessary when only internal comparisons are being made. The results from the model are therefore not a guideline for the assessment of its absolute performance in representing a real situation outside of the model environment; instead, the relative performance of algorithms within the model environment become the judgemental factors. This works on the premise that if, say, algorithm 1 gives better results in relation to those obtained from algorithm 2, when all other factors are equivalent, within a theoretical system, then algorithm 1 will also perform better in a representation of a real system.

2.5.1 Setting Up the Model

The model region itself consists of a square divided into 40 x 40 grid cells, each grid cell

having the nominal dimensions 5km × 5km (although different grid sizes can be chosen). A beach was required (see section 2.5.5) in order that the land/sea boundary should move as the tide progressed through its cycles.

2.5.2 Tidal Input to a Theoretical Model

Tidal input to the idealized model was assumed to be sinusoidal and given by:

$$\zeta(t) = A\cos(\omega t - \mathscr{E}) = \alpha\cos\omega t + a\sin\omega t \tag{2.13}$$

where; $\alpha = A\cos\mathscr{E}$, $a = A\sin\mathscr{E}$, $\mathscr{E}$ = phase.

Since the model region is sufficiently small, then the tidal input may be expressed as a Taylor expansion:

$$\alpha = \alpha_0 + x\frac{\partial\alpha}{\partial x} + y\frac{\partial\alpha}{\partial y} \tag{2.14}$$

$$a = a_0 + x\frac{\partial a}{\partial x} + y\frac{\partial a}{\partial y} \tag{2.15}$$

where α_0 and a_0 are the real and imaginary values at the centre of the model, which may be considered as the origin.

Now, if the real and imaginary components from equation 2.13 are differentiated with respect to the x and y (east-west and north-south) directions, we get equations 2.16, 2.17, 2.18 and 2.19:

$$\frac{\partial\alpha}{\partial x} = \frac{\partial}{\partial x}(A\cos\mathscr{E}) = \cos\mathscr{E}\frac{\partial A}{\partial x} - A\sin\mathscr{E}\frac{\partial\mathscr{E}}{\partial x} \tag{2.16}$$

$$\frac{\partial \alpha}{\partial y} = \frac{\partial}{\partial y}(A\sin\mathscr{E}) = \sin\mathscr{E}\frac{\partial A}{\partial y} + A\cos\mathscr{E}\frac{\partial \mathscr{E}}{\partial y} \tag{2.17}$$

$$\frac{\partial a}{\partial x} = \frac{\partial}{\partial x}(A\sin\mathscr{E}) = \sin\mathscr{E}\frac{\partial A}{\partial x} + A\cos\mathscr{E}\frac{\partial \mathscr{E}}{\partial x} \tag{2.18}$$

$$\frac{\partial a}{\partial y} = \frac{\partial}{\partial y}(A\cos\mathscr{E}) = \cos\mathscr{E}\frac{\partial A}{\partial y} - A\sin\mathscr{E}\frac{\partial \mathscr{E}}{\partial y} \tag{2.19}$$

The model area was taken to be a square of side 200km, aligned north-south, with $A_0 = 2m$ and $\mathscr{E}_0 = 180°$, these particular values being close to values from P.O.L.s' Tidal Atlas Of The British Isles for an area of the North Sea off the Humber coast for species M_2. There were several reasons for this particular choice. The first was that the maximum depth of the theoretical model was close to that of the sea-bed off the Humber coast. Secondly, this region had an area of similar size to that of the theoretical model which had almost straight co-phase lines for the M_2 tide, with the range varying (approximately) only zonally, and the phase only meridionally, thus allowing simplification of the calculation of the tidal input.

If we assume that, in the theoretical model, the range varies only meridionally, and the phase varies only zonally then, with $\mathscr{E} = 180°$, equation 2.16 becomes equation 2.20, and 2.17 becomes 2.21. Since there is no zonal variation of A, 2.18 becomes 2.22. Equation 2.19 becomes 2.23;

$$\frac{\partial \alpha}{\partial x} = -\frac{\partial A}{\partial x} = 0 \tag{2.20}$$

$$\frac{\partial \alpha}{\partial y} = -A\frac{\partial \mathscr{E}}{\partial y} = -2\times 0.005 = 10mm/km \tag{2.21}$$

$$\frac{\partial a}{\partial x} = 0 \tag{2.22}$$

$$\frac{\partial a}{\partial y} = -\frac{\partial A}{\partial y} = 10mm/km \tag{2.23}$$

However, when the phase is at 180°, the imaginary component of the tidal elevation is equal to zero. Therefore, the change in tidal elevation is represented by Equation (2.21). This value is based on the assumption that the figure 0.005rad/km is equivalent to 0.3°/km, the phase difference obtained from the Tidal Atlas Of The British Isles.

2.5.3 Boundary Conditions

When starting off the numerical model, there is deemed to be no tide anywhere *i.e.* $\zeta = U = V = 0$ everywhere. A tidal regime has to be specified at the sea boundaries in order that a tide may propagate throughout the model area. For a 'wind-up' period, the amplitude of the tide on the boundaries is gradually and linearly increased, which ensures that momentum within the model builds up gradually.

Since there are no restrictions caused by validation procedures, any tide may be chosen for any theoretical basin whether large volumes of water are required to move or smaller volumes. The only restriction is that of maintaining stability. In short, a tidal regime can be chosen to suit whichever phenomenon is being studied. However, it is perhaps better practice to attempt as close a match as possible to real conditions. Figure 2.3 shows a surface plan of the model area including the phase difference along the north and south boundaries and the amplitude differences along the east and west boundaries, there being no phase difference

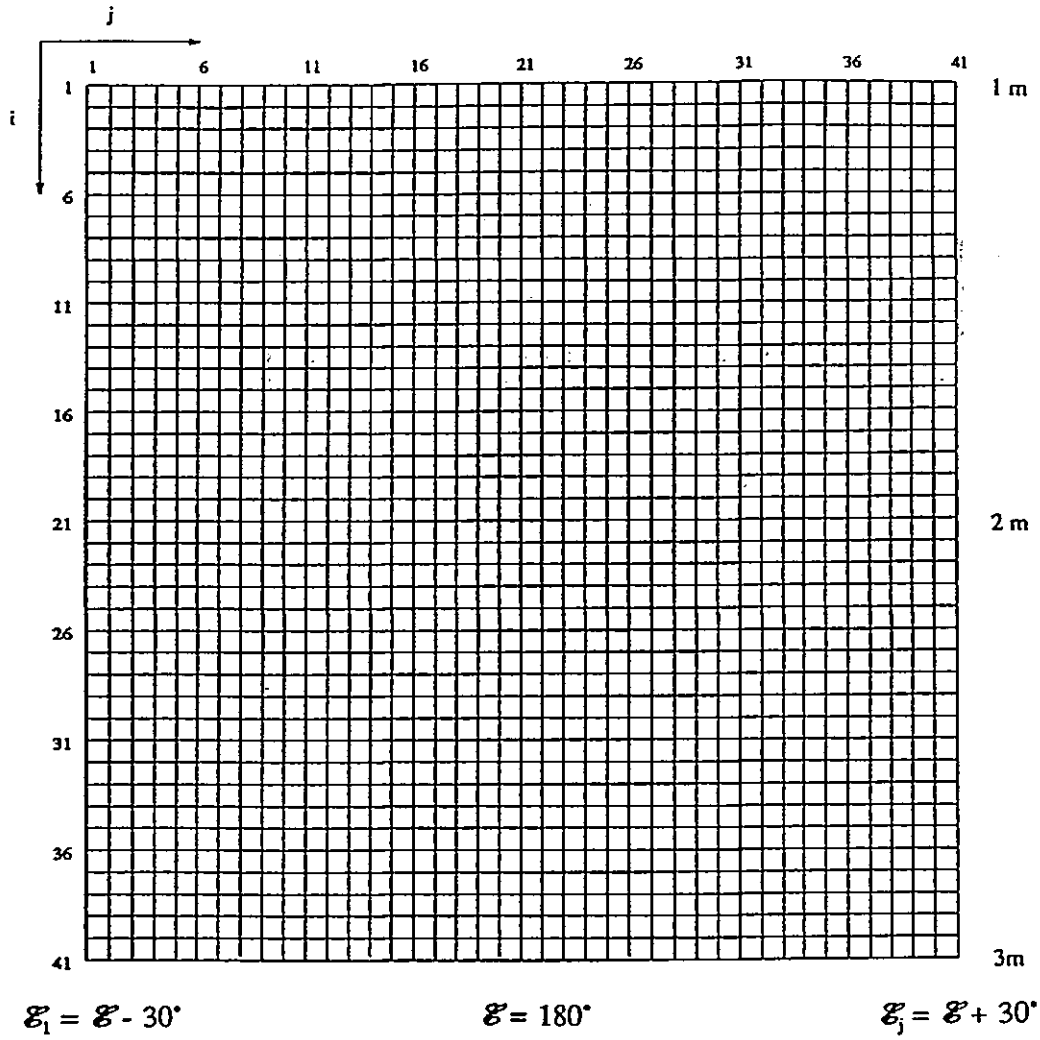


Figure 2.3: Surface plan of model area showing amplitude and phase differences

from north to south and no amplitude difference from east to west.

2.5.4 Bathymetric Design

The process of developing a numerical model from scratch involves the correction of many mistakes and much experimentation. Many of the methods used during the development of a model tend to cope only with the temporary requirements at any given stage in its development. Steps which carry out specific functions are added and subtracted throughout the course of the procedure as different problems and new ideas arise.

One fine example of such a process comes about from the need to establish a bathymetry over which the tide can propagate. Before any flooding and drying of intertidal flats can be incorporated, *i.e.* the introduction of a moving land/sea boundary, the source code must perform in a stable manner throughout a run.

In order that stability could be established, the model was first tested using a completely flat sea-bed with each cell having a depth of 50m, thus avoiding the numerical need of coping with a moving boundary. This gave the sea basin the appearance of a large box. However, instabilities frequently arose leading to an addressing of the problem of defining the bathymetry in a theoretical manner whilst retaining a measure of closeness to the real world.

A short analysis can demonstrate the theory involved when a flat sea-bed is used as the bathymetric specification in an idealized model:

If equation 2.6 is expanded to show the various terms:

$$\frac{\partial \zeta}{\partial t} = -U\left(\frac{\partial \zeta}{\partial x}+\frac{\partial h}{\partial x}\right)-(\zeta+h)\frac{\partial U}{\partial x}-V\left(\frac{\partial \zeta}{\partial y}+\frac{\partial h}{\partial y}\right)-(\zeta+h)\frac{\partial V}{\partial y} \qquad (2.24)$$

and if spatial derivatives of ζ and h are separated equation 2.24 becomes equation 2.25:

$$\frac{\partial \zeta}{\partial t} = -\left(U\frac{\partial \zeta}{\partial x} + V\frac{\partial \zeta}{\partial y}\right) - \left(U\frac{\partial h}{\partial x} + V\frac{\partial h}{\partial y}\right) - (\zeta + h)\left(\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y}\right) \qquad (2.25)$$

then typical sizes of the terms from equation 2.25 in a shelf sea might be:

	INSHORE	OFFSHORE
U, V	1 ms^{-1}	1 ms^{-1}
ζ+h	10 m	100 m
$\partial\zeta/\partial x$, $\partial\zeta/\partial y$	10 $mmkm^{-1} = 10^{-5}$	10 $mmkm^{-1} = 10^{-5}$
$\partial h/\partial x$, $\partial h/\partial y$	10 $mkm^{-1} = 10^{-2}$	1 $mkm^{-1} = 10^{-3}$
$\partial U/\partial x$, $\partial V/\partial y$	1 $ms^{-1}km^{-1} = 10^{-3}s^{-1}$	1 $ms^{-1}10km^{-1} = 10^{-4}s^{-1}$
giving:		
$-\left(U\frac{\partial \zeta}{\partial x} + V\frac{\partial \zeta}{\partial y}\right)$	10^{-5} ms^{-1}	10^{-5} ms^{-1}
$-\left(U\frac{\partial h}{\partial x} + V\frac{\partial h}{\partial y}\right)$	10^{-2} ms^{-1}	10^{-3} ms^{-1}
$-(\zeta + h)\left(\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y}\right)$	10^{-2} ms^{-1}	10^{-2} ms^{-1}
$\frac{\partial \zeta}{\partial t}$	$<10^{-3}$ ms^{-1}	$<10^{-3}$ ms^{-1}

If the terms in the equation of continuity of mass, equation 2.6, are scaled (see above), the analysis shows that under normal circumstances, the spatial variation of depth below M.S.L. is typically 2 dex greater than the spatial variation of the tidal elevation. *i.e.* $\partial h/\partial x \approx \partial h/\partial y \gg \partial\zeta/\partial x \approx \partial\zeta/\partial y$. In the case of the flat bottomed idealized model however, $\partial h/\partial x = \partial h/\partial y = 0$ which leads to equation 2.6 being dominated by the terms containing $\partial\zeta/\partial x$ and $\partial\zeta/\partial y$ whereas in normal circumstances these terms are negligible. This appeared to provoke instability and so a random variation in sea-bed depth was imposed.

2.5.5 Introducing a Moving Land/Sea Boundary

Once the model had remained stable for several tidal cycles it became necessary to introduce a moving land/sea boundary. This was achieved by adding a Gaussian expression, equation 2.26, as a loop to MODKS.FTN77, producing a Gaussian shoal. This gave a representation of an island in the middle of a basin with coastline all the way around on which the land/sea boundary would move.

$$h_{(i,j)} = -50\times\left(1 - e^{(-R^2/100)}\right) - 4 \tag{2.26}$$

where, $R^2 = (i-21)^2 + (j-21)^2$, i and j being the model coordinates.

However, before the bathymetry was altered to introduce a 'sudden' moving boundary, the top of the Gaussian shoal was set at a depth of 4m below mean sea-level for five tidal cycles, accounting for the *4* at the end of equation 2.26. At this point the shoal was set to slowly rise to mean sea level, exposing the shoal as the tide fell to ensure drying. The shoal would then stay at mean sea level for a further six tidal cycles, when the model reached a steady state. When confidence had increased in the programming, the shoal was input as an island that did not rise and was set with the 'peak' at mean sea-level. This became the 'test ground' for three existing drying algorithms. Figure 2.4 shows a three-dimensional view of the Gaussian bathymetry.

2.5.5.1 Addition of a Drying Algorithm

Now that a moving boundary had been introduced, it became necessary to include a drying algorithm. Since the Author was most familiar with that of Flather & Heaps' (1975) method,

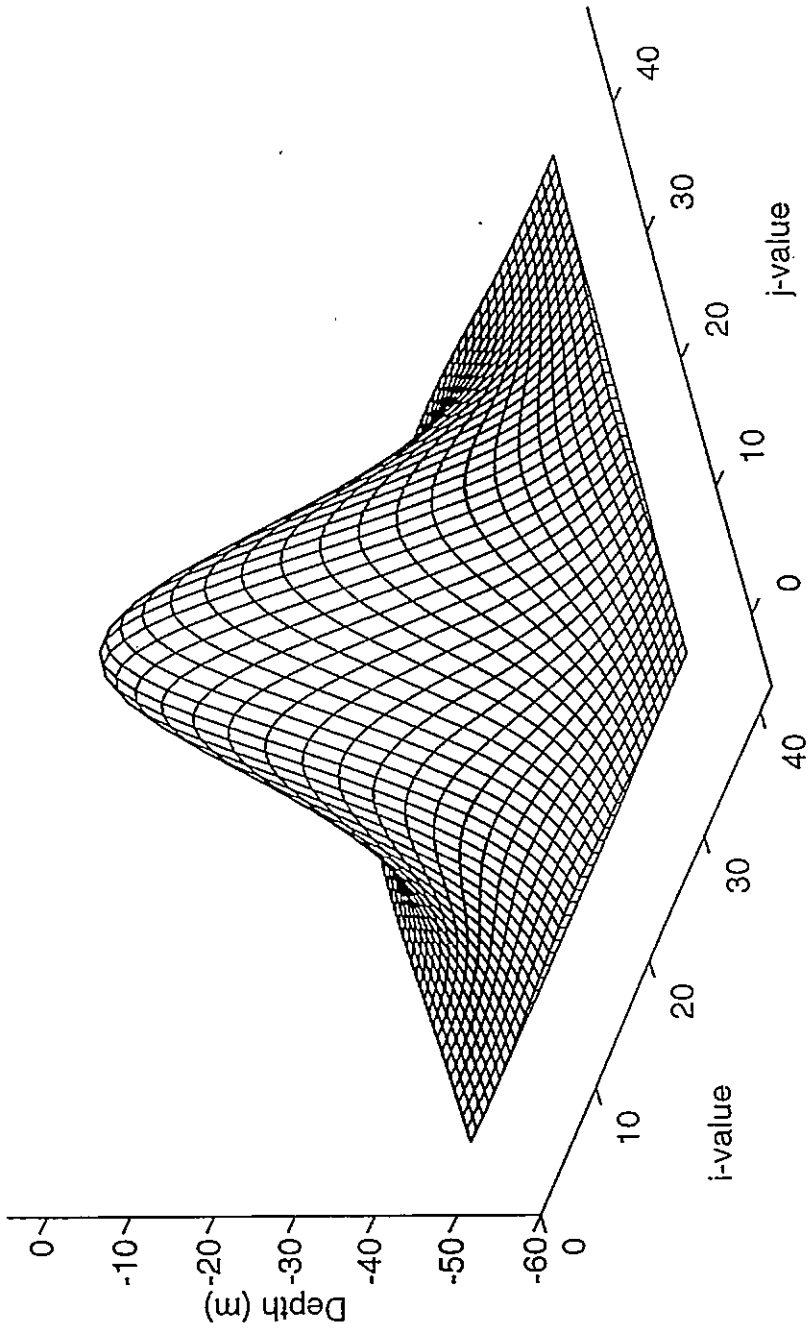


Figure 2.4: Gaussian shoal bathymetry used to test algorithms

it being the one adopted by Dr. George at the University of Plymouth after his initial work incorporating Reid and Bodine's (1968) algorithm, this became the first drying algorithm incorporated into the source code (see Chapter 3). The strength and stability of the numerical scheme was built solidly around this algorithm before the incorporation of any other was considered.

2.5.6 Horizontal Eddy Diffusion

Horizontal eddy diffusion can be quite a perceptual nightmare, it being a rather indefinable quantity. However, possibly the best way to understand the actual functioning of this phenomenon is to picture a swimming pool with a wave machine working. At certain points in the swimming pool, waves will meet and peak, possibly resulting in clapotis (a splash or a spout of water into the air). In modelling terms, this may be considered as a precursor to instability. Although in Public Baths this may be a desirable effect, in a numerical model it is most undesirable. But, supposing the above effect were to be prevented, given that all energy input must remain at the same frequencies, then the most assured method would be to restrict the movement of the water. This could be achieved by covering the surface with a material that has just the correct properties to prevent the splashing from occurring - or a heavier material that could even begin to suppress the formation of waves. This is the behaviour of the horizontal eddy diffusion term. A particularly high value makes the sea system behave as though it has a material spread over the surface that has a strong damping effect. A lower value may be considered to have a lesser damping effect. Obviously, it is not desirable to achieve stability simply by applying a greater and greater eddy diffusion value as this would provide misleading results through over-restriction of water movement.

Ideally, eddy diffusion should be as light as possible in order that the water column has as much freedom of movement as possible.

In MODKS.FTN77, horizontal eddy diffusion was made depth-dependent, increasing with increase in depth. Once stability was achieved with a flat-bottomed sea-bed, a sensitivity analysis was conducted to establish a minimum value of the horizontal eddy diffusion required to ensure stability. This turned out to be ten times the depth.

2.5.7 Coriolis Acceleration

The Coriolis acceleration has been included in the model since the scale of the water-body considered here makes it significant. The spinning of the Earth has an effect on accelerating particles, of any size, and deflects the path of the particle to the right in the Northern Hemisphere. Calculation of the 'Coriolis effect' can be simplified in numerical tidal models for small sea areas, over which the latitude changes only slightly, by adopting the '*f-plane approximation*' (e.g. Gill, 1982) which allows $2\omega\sin\phi$ to remain constant over the small change in latitude. For models which cover large sea areas, however, it is apparent that this component would vary significantly over the large change in latitude and therefore must be included as a varying quantity (e.g. George, 1993).

2.6 Data Output

Output files consisted of consecutive or periodic values of all possible variables within the hydrodynamic equations.

A subroutine was written into MODKS.FTN77 to produce an interim output over a 'window' of the model area which would provide a time-sequenced view of an area specifically chosen

to observe the progression of the tidal elevation and current component values. This was of particular value in locating the cause/source of an instability as it developed.

Other components of the hydrodynamic equations were logged throughout a cycle so as to provide time-series which would show how each component varied for a particular cell and its surrounding cells. This provided the basis for a qualitative analysis of the effects of a drying and wetting cell on its surrounding cells and the physical processes occurring within the chosen cell.

2.6.1 Representation of Output Data

Graphical illustration of output data was achieved through software written by the Author in FORTRAN using calls to graphics subroutines in the GINO-F package. This software provided graphical representation of up to five of the time-series obtained, and can be seen in Appendix 2. This proved to be invaluable for the initial qualitative assessment of different drying algorithm performances since the effects of discontinuities and errors in programming could be observed. It was an aim to produce smooth tidal curves, so, for instance, it could be assumed that the more effective the drying algorithm used, the smoother the tidal curves should appear.

Abnormally long or short drying periods could also be spotted by comparison of the tidal curve obtained to a basic tidal curve for a specific point. A basic tidal curve, in this case, was one obtained analytically assuming that there was no shoal present. This provided an approximation of the time when the tidal height should, ignoring shallow water distortion, pass the sea-bed elevation of a dry cell, thus indicating that it should become wet at around that time.

Tidal stream information can also be represented as 'arrows', showing their direction and magnitude (set and rate). Commercially available software, such as MATLAB®, allows the production of 'arrow' pictures from the large data matrices output from MODKS.FTN77 without the need to write extensive plotting software.

CHAPTER 3

EXISTING 'DRYING' ALGORITHMS

Since there have been numerous attempts at the incorporation of a moving land/sea boundary in two-dimensional numerical tidal models, the Author has chosen to analyze the underlying principles of the apparent two differing approaches to the solution of the problem.

In this chapter, the minutiae of the existing wetting and drying algorithms are critically examined. In addition, subsequent improvements to each traditional method have been analyzed.

3.1 Leendertse & Gritton

Leendertse & Gritton (1971) adopted a method whereby three tests were applied at each grid point, in order to locate grid cells which were becoming dry, and to remove them from the computational field, thereby inducing a moving land/sea boundary. Firstly, a volumetric flow was calculated through the walls of each grid square and if this cross-sectional flow became negative (that is, in the opposite sense to which it should be) then the grid square was assumed to have become dry and removed from the computation whilst the current values

were set to zero. Or, with the volume given by (see figure 3.1):

$$VOL_{i,j}^{t} = [\zeta_{i,j}^{t} + \frac{1}{4}(h_{v_1} + h_{v_2} + h_{u_1} + h_{u_2})].\Delta s^2$$

where s = grid spacing, if $VOL_{i,j}^{t}$ was negative, then the grid square was assumed dry and the water depth set to the value at the previous computation, $\zeta_{i,j}^{t-\frac{1}{2}}$. In addition, the previous row or column of calculations were then recalculated with the grid square taken as being dry.

Secondly, the depths of each of the four cross-sections (see figure 1.1) surrounding each grid square were calculated at the end of each half time step, i.e. $H_{c_1}^{t}$, $H_{c_2}^{t}$, $H_{c_3}^{t}$, $H_{c_4}^{t}$ (see figure 3.2) where;

$$H_{c_1}^{t} = h_{v_1} + \frac{1}{2}(\zeta_{i,j}^{t} + \zeta_{i-1,j}^{t})$$

Again, if the results were numerically negative, the grid square was deemed to have become dry.

The third and most rigorous test of whether a grid square had dried or not was that if any of the four cross-sections decreased to less than a critical value, then that grid point was removed from the computation and the water level was set at its last value and maintained at that level until the grid square flooded. The critical value, ζ_{cr}, was determined by the depths of the surrounding grid squares such that:

$$\zeta_{cr_{i,j}} = h_{pr} - h_{\min_{i,j}}$$

where h_{pr} is a height preset above the highest surrounding mean depth value and $h_{\min_{i,j}}$ is the minimum value of h_{v_1}, h_{v_2}, h_{u_1} and h_{u_2}.

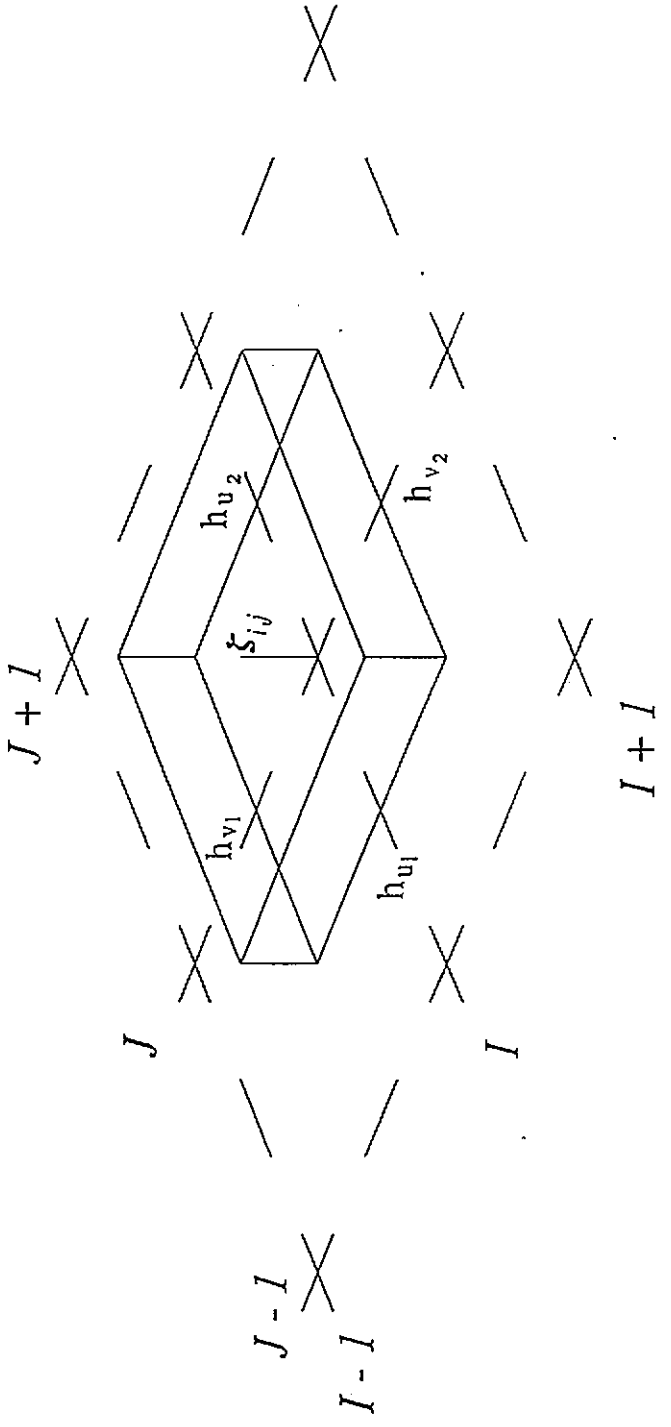


Figure 3.1: Diagram to show depth related variables used to evaluate the volume of a cell

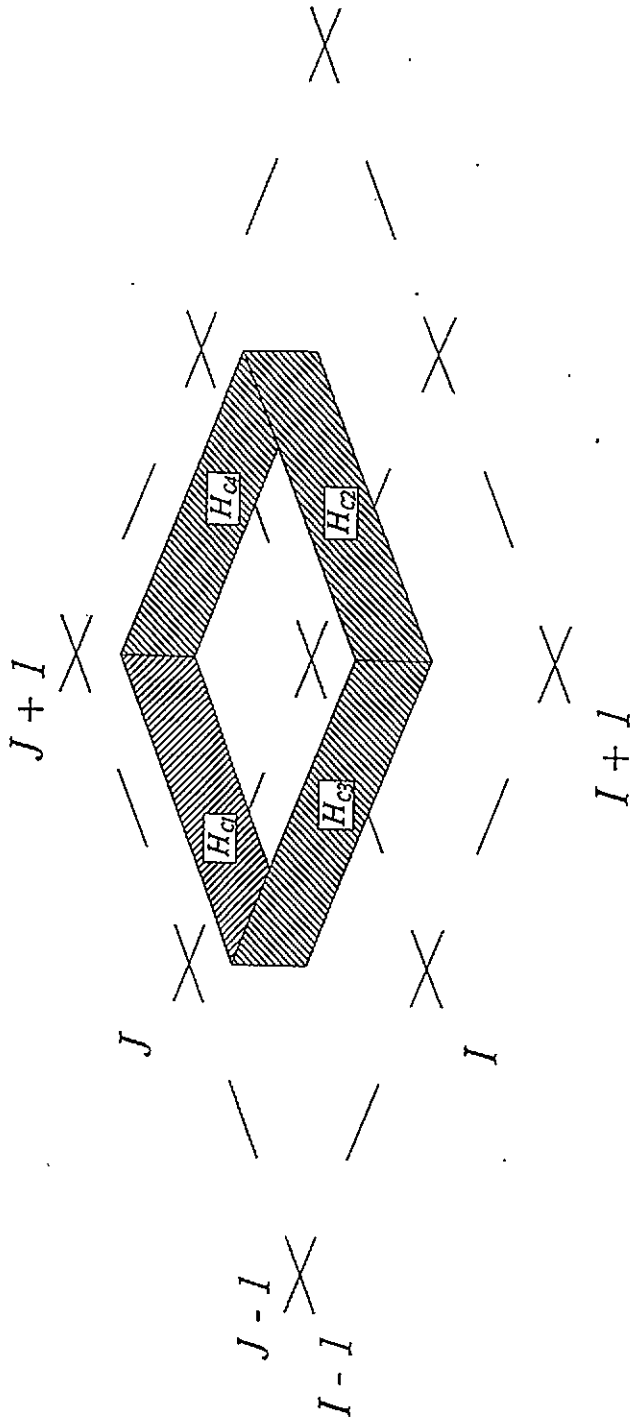


Figure 3.2: Diagram to show cross-sections used for determining the moving boundary

Leendertse and Gritton (1971) found that discrete changes in the land boundary as grid squares began to dry out, and also to flood, created numerical disturbances. To overcome this they allowed computational time for the disturbances to decay by making this third and final test, since it was by this test that most grid squares became dry, at intervals greater than the time-step.

The test for flooding was whether or not the average water depth of the surrounding grid squares was greater than the depth of water left in the grid square by the drying procedure:

$$\zeta_{i,j}^{t-\frac{1}{2}} < \frac{1}{4}(h_{v_1} + h_{v_2} + h_{u_1} + h_{u_2})$$

Again, transport cross-sections were computed with a positive cross-section resulting in the flooding of that square. This check was also carried out at intervals greater than the time-step to allow the computational noise generated by the moving boundary time to decay.

This method has been incorporated in the study tool described in Chapter 2.

3.2 Falconer & Owens

In 1984 Falconer applied his technique to a comparison of the hydraulic features and flushing characteristics of Holes Bay in Dorset, to test two proposed new outlines of the bay. In this bay, large areas of marshland are dried out and flooded throughout the tidal cycle, creating a moving land boundary. Falconer (1984, 1986) overcame the associated problems using Leendertse & Gritton's (1971) drying technique with two basic differences. Firstly, the first two checks were combined and carried out at a different time in the tidal solution and only two cross-sections in the direction of the calculation were considered rather than all four cross-sections for each grid square. However, in the 1986 paper all four cross-sections were

considered in the finite difference equations. Secondly, an average water elevation was used in calculating each cross-section rather than a particular elevation in a wet grid square (see figure 3.3):

$$H_{c_1}^t = h_{v_1} + \frac{1}{4}(\zeta_{i-1,j-1}^t + \zeta_{i-1,j+1}^t + \zeta_{i,j-1}^t + \zeta_{i,j+1}^t)$$

Falconer reported that although this increased the computational cost, it improved the accuracy of the scheme, with the possibility of numerical instabilities being generated reduced.

However, Falconer discovered that when this procedure was applied to a model of the Humber Estuary, unstable solutions were produced despite the claim that implicit solutions are unconditionally stable. In order to refine the procedure, a method was developed which would allow for large grid size and time-step models (Owens, 1984; Falconer & Owens, 1987). An analysis of the method was made using an idealized basin. This illustrated the discontinuities produced by the sudden setting to zero of streams in grid squares which still retained significant volumes of water. Owens (1984) redefined the way that the volume element of a grid square was evaluated:

$$VOL_{i,j}^t = (\zeta_{i,j}^t + h_{\max_{i,j}}).\Delta s^2$$

where, $h_{\max_{i,j}}$ = maximum value of h_{v_1}, h_{v_2}, h_{u_1} and h_{u_2}.

This assumed that the bed within a grid square was level with the value of the lowest surrounding mean depth. Falconer & Owens (1987) modified the method by introducing two new tests. The first was to check each cross-section at the end of every half time-step, if a cross-section was found to be negative then it was equated to zero along with the perpendicular velocity component. Therefore, if the deepest cross-section of a grid square

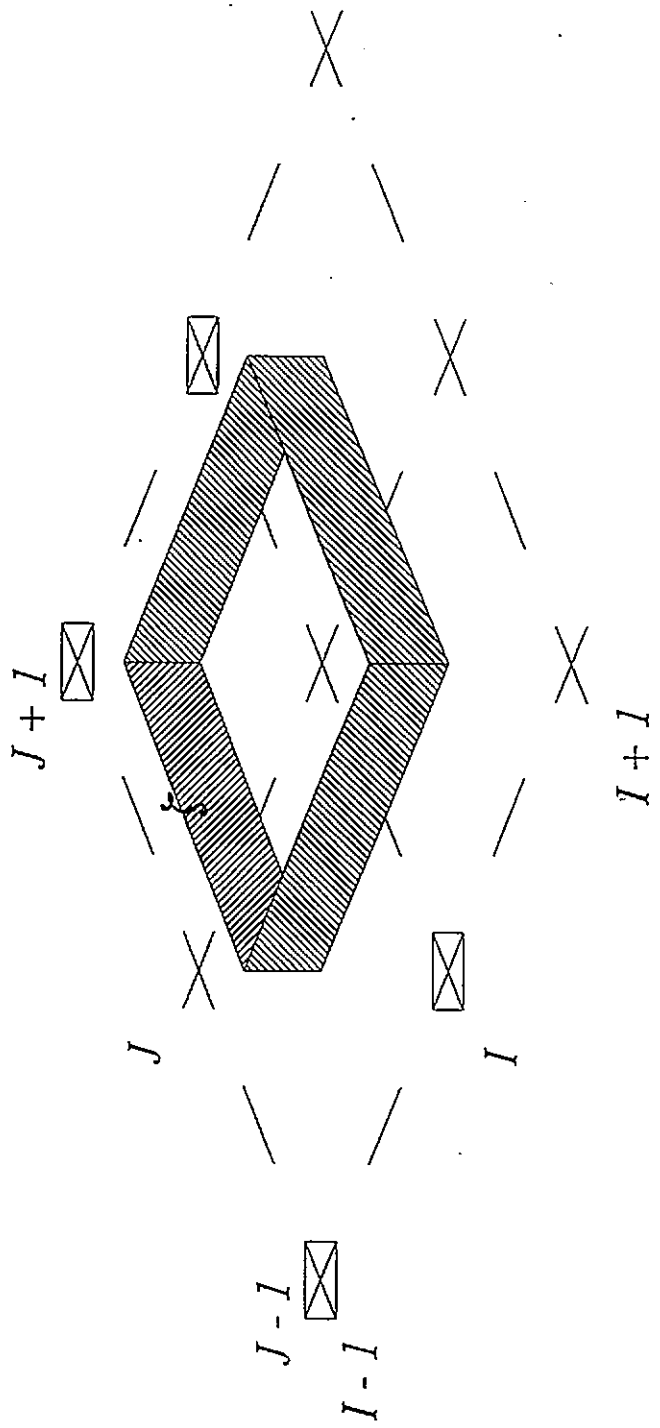


Figure 3.3: Diagram showing the location of elevation points used to establish the elevation at a stream point

became negative then the stream velocities across all four cross-sections were zero and the grid square had become dry. This had the effect of smoothing the reduction of velocity components to zero by removing the grid squares from the calculation in a more evenly spaced manner during the solution of the hydrodynamic equations. Also, the test for a dry square becoming wet was carried out at every time-step rather than every few time-steps (Leendertse 1970; Leendertse & Gritton 1971; Falconer 1984, 1985, 1986). The second test involved setting a critical level at a predetermined depth above the lowest surrounding depth. This defined a minimum cross-sectional area through which water could flow when a grid square flooded again. With this method however, the predetermined depth is most effective when set at different levels depending on the tidal range. For instance, Falconer found the best results for the tidal range in his study of the Humber Estuary were obtained using a predetermined depth of 0.2m.

This method has been incorporated in the study tool described in Chapter 2.

3.3 Flather & Heaps

Independently of the work initiated by Leendertse (1970), Flather and Heaps (1975) proposed a new finite difference method for tidal computations in shallow water.

The test-bed for the development of their new techniques was Morecambe Bay. This was considered to be suitable due to the large areas of sandbanks exposed at low tide. Their approach was less complex than that of Leendertse and Gritton (1971). The specific detail of whether or not a cell had become wet or dry was:

If (i) $H_{i,j}^{t+\Delta t} > 0$ and $H_{i,j+1}^{t+\Delta t} > 0$ so that $H_{u_{i,j}}^{t+\Delta t} > 0$,

or (ii) $H_{i,j}^{t+\Delta t} > 0$ and $H_{i,j+1}^{t+\Delta t} \leq 0$ and $H_{u_{i,j}}^{t+\Delta t} > 0$ and $\zeta_{i,j}^{t+\Delta t} - \zeta_{i,j+1}^{t+\Delta t} > \varepsilon$,

or (iii) $H_{i,j}^{t+\Delta t} \leq 0$ and $H_{i,j+1}^{t+\Delta t} > 0$ and $H_{u_{i,j}}^{t+\Delta t} > 0$ and $\zeta_{i,j+1}^{t+\Delta t} - \zeta_{i,j}^{t+\Delta t} > \varepsilon$,

where H_u is the total water depth at the east-west velocity point and ε is a critical elevation difference, then there was still considered to be flow through the grid square (see figure 3.4 for a description of the variables involved).

As with other drying procedures, if a grid square was deemed to have become dry, then the streams were set to zero; if it had become wet, then new values of the tidal streams were calculated from the solution of the equations of motion.

Flather and Heaps' 1975 paper presents time series which show the distortion of the tide as it propagates through shallow water as well as discontinuities induced by the presence of cells which wet and dry.

This method has been tested using the study tool described in Chapter 2.

3.4 Flather & Hubbert

Flather and Hubbert (1989) presented a refined version of Flather and Heaps' (1975) drying technique.

This refined method was developed through the comparison of various schemes. In their first scheme the depth was defined at the centre of each coarse grid square but in the high resolution area the depth was taken to be:

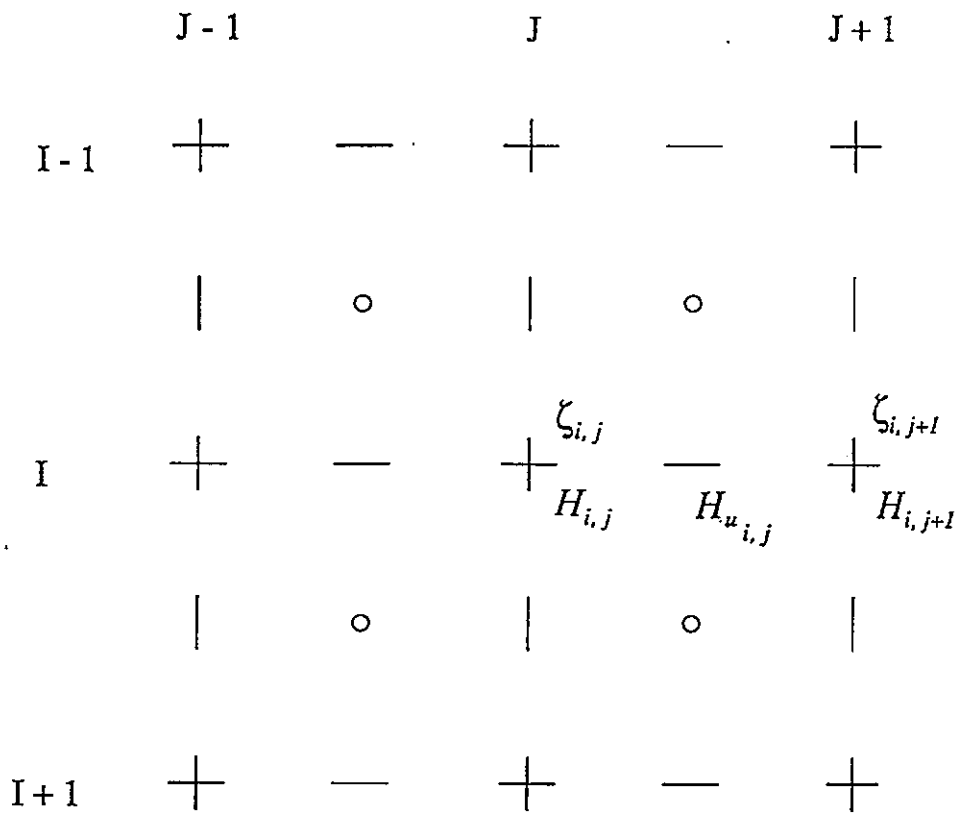


Figure 3.4: Diagram showing the positions in space of the variables which define the moving boundary

$$h = \frac{1}{N}.\sum(h_j \mid h_j > h_{land}) \qquad if \;\; 50 \le N \le 100$$

$$h = h_{land} \qquad if \;\; 0 \le N < 50$$

where N is the number of high resolution grid points within a grid square which are not permanent land points. This provided a mean depth of the 'wet' part of a grid square. It also 'smoothed out' a deep channel, so Flather & Hubbert artificially increased the depth along a route corresponding to the topographical position of the channel thus artificially creating a deep water channel. This made the channel unrealistically wide, so Flather & Hubbert decided to represent the deep channel in a one-dimensional format by modifying the two-dimensional scheme in this area. By including additional factors, in terms of area and breadth constants, Flather & Hubbert were able to make the one-dimensional and two-dimensional representations of the equation of continuity equivalent. To achieve this, the continuity equation, equation 2.6, was re-written thus:

$$A\frac{\partial \zeta}{\partial t} + \frac{\partial}{\partial x}\{(h+\zeta)BU\} + \frac{\partial}{\partial y}\{(h+\zeta)CV\} = 0 \qquad (3.1)$$

If $A = B = C = 1$, then equation (3.1) is identical to equation (2.6). Written in finite-differences, equation (3.1) becomes equation (3.2):

$$A(\zeta_{i,j}^{t+\Delta t} - \zeta_{i,j}^{t}) + \Delta t\left(\frac{H_{u_{i,j}}^{t} B_{i,j} U_{i,j}^{t} - H_{u_{i,j-1}}^{t} B_{i,j-1} U_{i,j-1}^{t}}{\Delta x}\right)$$

$$+ \Delta t\left(\frac{H_{v_{i-1,j}}^{t} C_{i-1,j} V_{i-1,j}^{t} - H_{v_{i,j}}^{t} C_{i,j} V_{i,j}^{t}}{\Delta y}\right) = 0 \qquad (3.2)$$

where;

A	=	area factor
	=	wetted area of grid box/total area of grid box
B	=	breadth factor for *U*-point of cell
	=	breadth factor over which flow occurs/total breadth of cell
C	=	breadth factor for *V*-point of cell
H_u	=	total water depth at *U*-point
H_v	=	total water depth at *V*-point

Thus, for example, if there is no north-south flow (i.e. $V = 0$) then with arbitrary values of the area and breadth factors, the two-dimensional equation approximates a one-dimensional flow in the east-west direction.

Changing from one-dimensional to two-dimensional flow as the banks of the channel is breached is represented by an appropriate drying procedure which either sets *V* to zero if the surrounding banks are dry or computes its value from the momentum equation.

The drying test used was that by Flather & Heaps (1975) (see section 3.3), where the tests were carried out after each *U* and *V* calculation to determine whether the associated grid point was 'wet' or 'dry'.

A final scheme allowed the area and breadth factors to be included but used as variables so that they could change as the water level changed. This allowed the water level in a cell to decrease progressively until the cell dried completely, thus reducing the shock that a sudden change from a cell being entirely 'wet' to entirely 'dry' induces in the system.

This was attained by combining the breadth factors with the total water depths at the stream points of a cell. This meant replacing BH_u and CH_v in equation (3.2) with the redefined factors B' and C'. These now represent the transport across the cross-sectional area of each side of the cell. The new areal and cross-sectional factors were established from the high

resolution bathymetry and defined at fixed surface elevation levels every 0.1m in the range -3.5m to 3.5m. If the surface water level was between any of the 0.1m increments, the values of A', B' and C' were calculated by linear interpolation.

With the variable areal and cross-sectional factors, equation (3.2) now becomes equation (3.3) expressed in finite-differences:

$$A\left(\zeta_{i,j}^{\Delta t+t} - \zeta_{i,j}^{t}\right) + \Delta t\left(\frac{B'^{t}_{i,j}U^{t}_{i,j} - B'^{t}_{i,j-1}U_{i,j-1}}{\Delta x}\right)$$

$$+ \Delta t\left(\frac{C'^{t}_{i-1,j}V^{t}_{i-1,j} - C'^{t}_{i,j}V^{t}_{i,j}}{\Delta y}\right) = 0 \qquad (3.3)$$

Publication of results for this method show that there are still considerable anomalies present in the solution. Figure 3.5 shows the wetting part of the tidal cycle for the same cell under two different experimental conditions. It can be seen that there is a considerable measure of disturbance caused by the wetting of this cell. The result of a cell wetting on nearby cells has not been published. In addition, R.A. Flather has indicated, through personal communication, that although the present level of moving land/sea boundary computations at P.O.L. is quite refined, the results are still unsatisfactory. It is a problem, however, that he considers would require too much time for further development.

Due to the impracticalities of digitizing the bathymetry to obtain an area of high resolution and the significant increase in computational costs (particularly high when large areas wet and dry), the Author has not tested this numerical scheme.

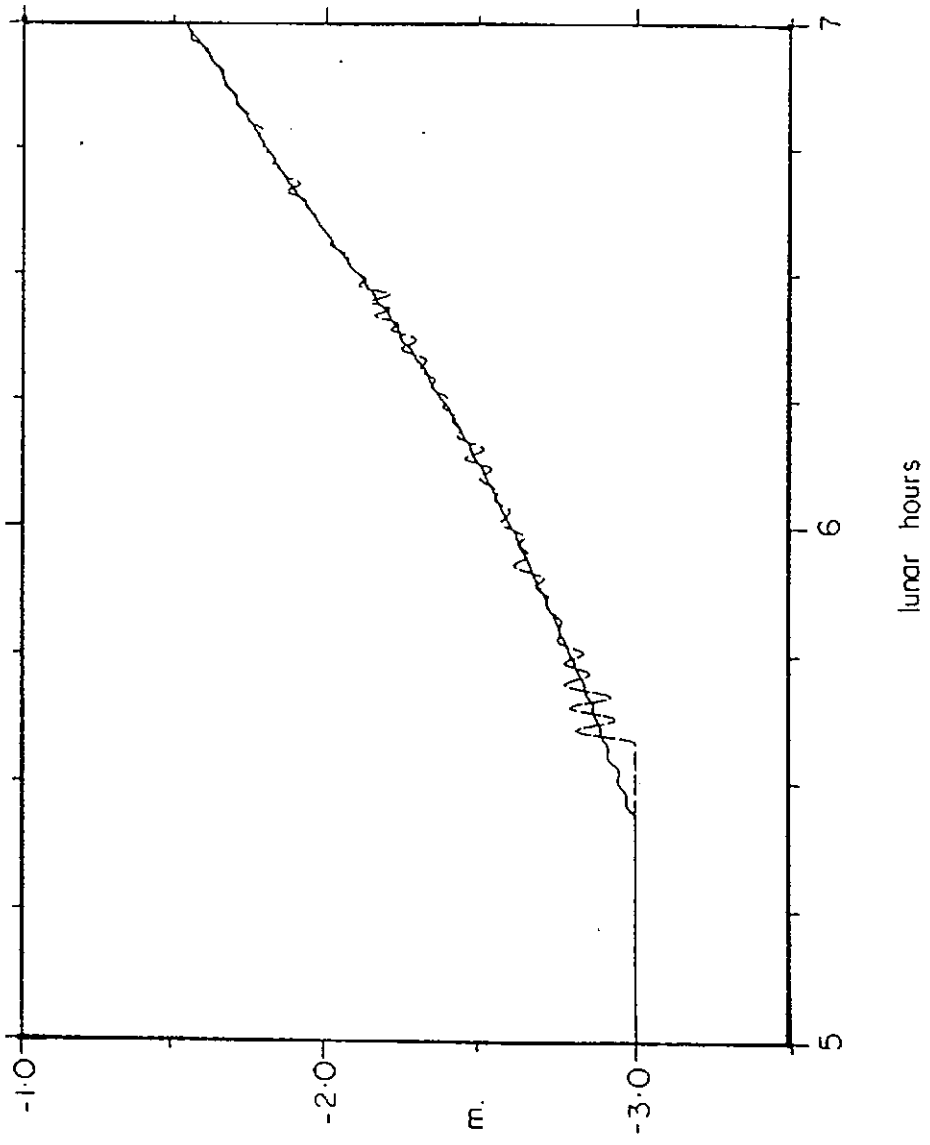


Figure 3.5: To show high frequency oscillations for two different experimental runs (after Flather & Hubbert, 1989)

3.5 Comparison of Results

The results obtained from MODKS.FTN77 have been presented in graphical format, showing time series of; the elevation of a cell chosen to be a test cell, the elevation of a cell adjacent to the test cell in order to demonstrate the effect that nearby cells can have on each other, and various components from the hydrodynamic equations in two dimensions which pertain specifically to the test cell already mentioned.

3.5.1 Leendertse's (1970) Drying Algorithm

Figure 3.6 shows two tidal elevation profiles and the easterly component of the depth-averaged velocity. Qualitative analysis shows that there are quite powerful 'shocks' introduced into the system, causing the elevation profiles to kink quite considerably at , for example, about 1750°. These kinks are caused by a nearby cell becoming wet; slowing the rate of rise in the already wet cells as the momentum balance adjusts to the sudden appearance of a large area of sea. This slowing of the rate of rise becomes too exaggerated and the rate of rise is then accelerated, which is also too severe and eventually another decrease in the rate of rise occurs. This time, however, the rate of decrease in the rate of rise is less severe than the initial decrease, indicating the damping of an oscillatory process which is trying to attain equilibrium.

The effects of this disruption in the elevation profiles can be seen until just after high water when equilibrium is attained for a short while.

The velocity profile shown in figure 3.6 is the u-component of the depth averaged velocity (*mutatis mutandis*, the v-component behaves similarly). It can be clearly seen in figure 3.6 that there is uncertainty in the wetting process of this particular algorithm as the velocity

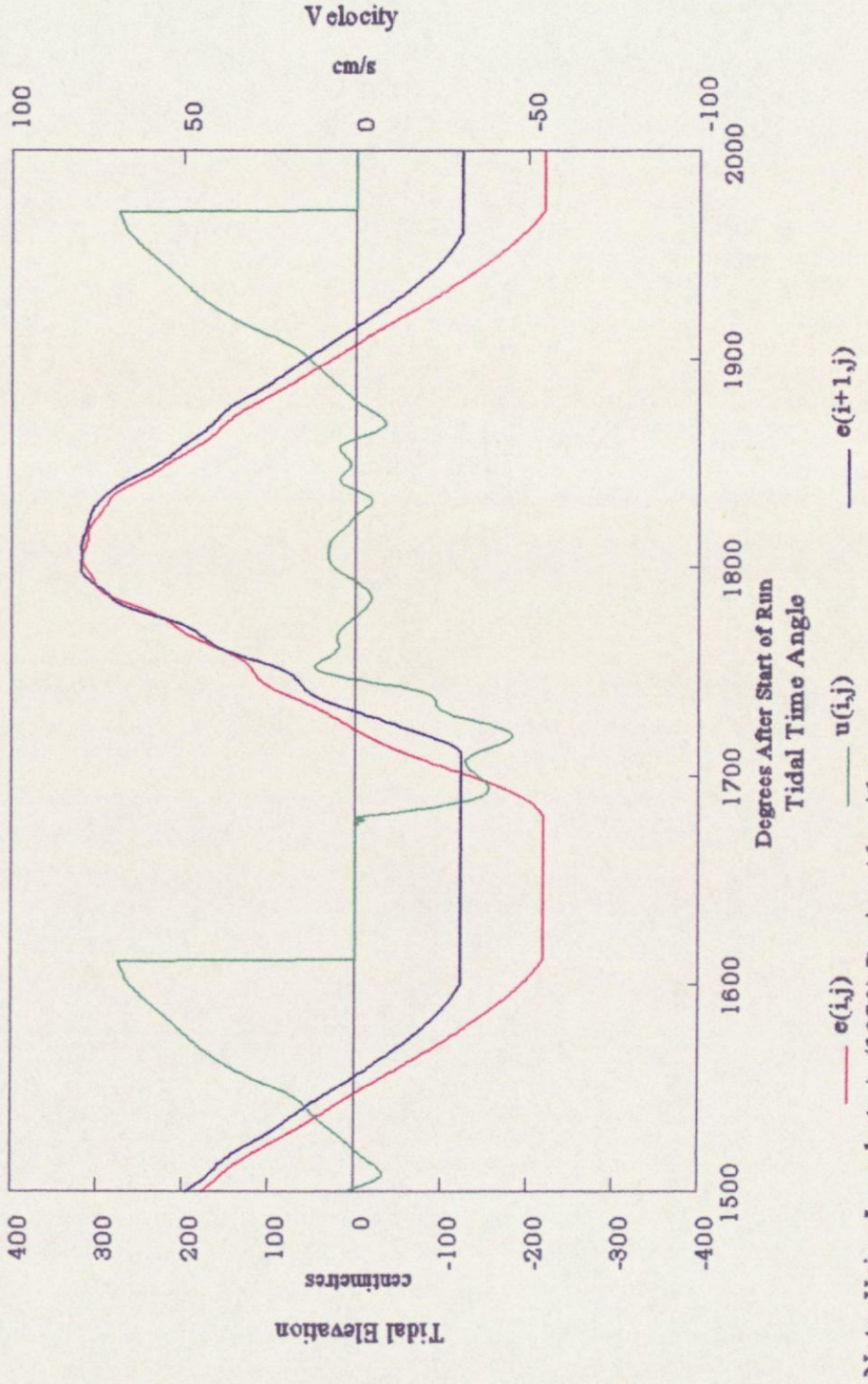


Figure 3.6: Elevation and easterly velocity component of a test cell with elevation of an adjacent cell

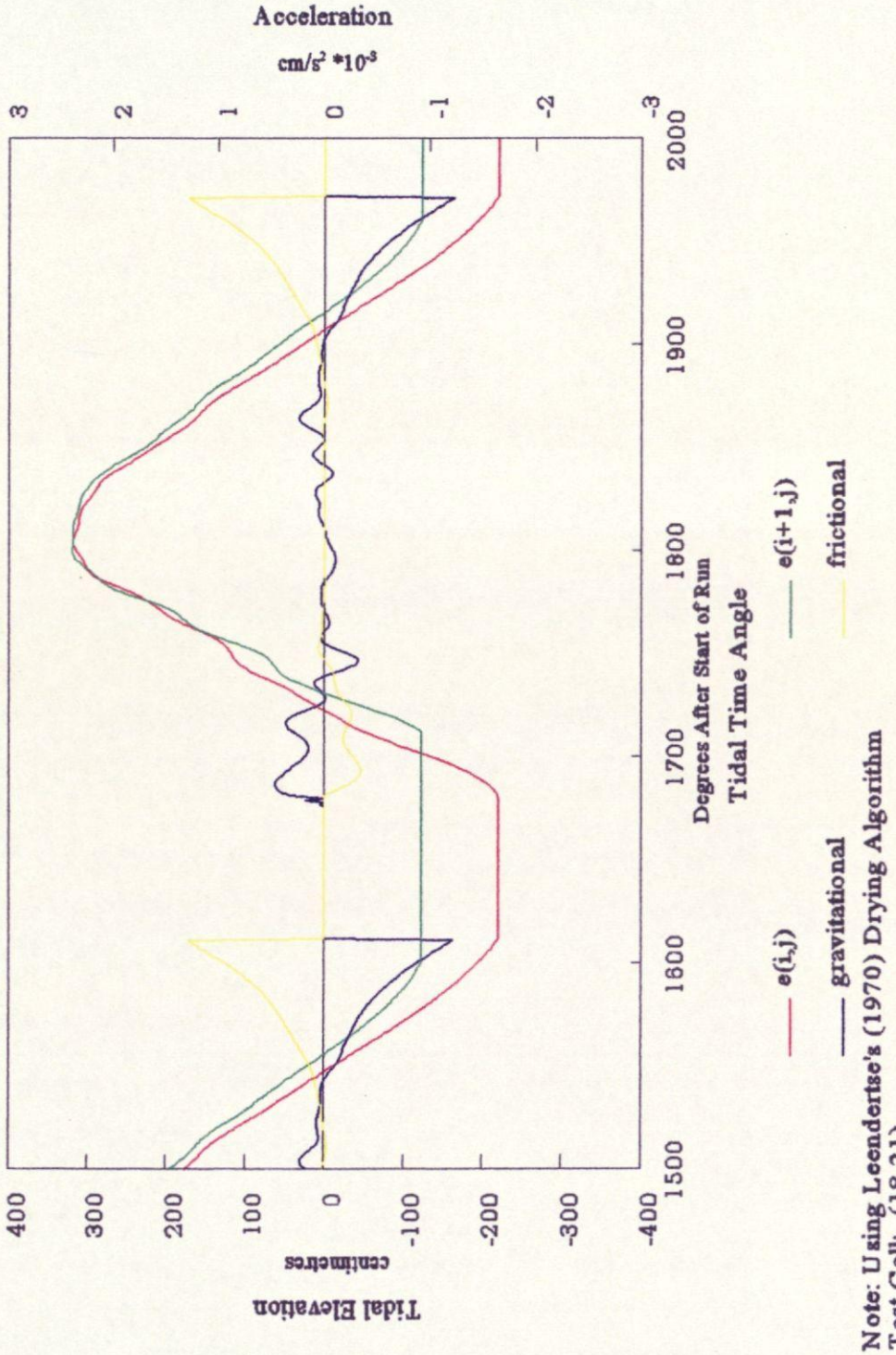


Figure 3.7: Elevation and accelerations due to both gravity and friction of a test cell with the elevation of an adjacent cell

Figure 3.8: Initial wetting period

Note: Using Leendertse (1970) Algorithm
Test Cell: (18,21)

alternates between a real value and zero for several time-steps after the initial wetting. This is also reflected in the profiles of the gravitational and frictional terms (in the easterly direction only) shown with the tidal elevations in figure 3.7 at around 1680°.

Figure 3.7 also shows that the gravitational and frictional terms tend to balance each other out as the water becomes shallow.

The drying process appears to operate satisfactorily, however, one must bear in mind that once a cell has been 'switched off', it takes no further part in the computational process until it becomes wet again. The actual effects of a cell drying on its surroundings are therefore far less pronounced than those disturbances caused by a cell becoming wet and its subsequent behaviour.

The initial wetting period can be seen in more detail in figure 3.8. This figure demonstrates the alarming effect that the wetting of a cell using Leendertse's (1970) method can have. We can see here that there is a pronounced disparity between the rates of rise of the two adjacent cells and the effect that this has on the shape of the profile after the initial wetting. This figure also enlarges the disturbance in the velocity field. This disturbance shows the uncertainty of wetting in this method.

3.5.2 Owens' (1984) Drying Algorithm

Time series that were produced with the incorporation of Owens' (1984) drying algorithm can be seen in figures 3.9 and 3.10. The elevation profiles show distinct short period and longer period oscillations about a smooth profile. In order that an equilibrium state is obtained, the model has to be run for a longer period of time than is necessary for the other algorithms used.

There are ripples apparent in the solution shortly after high water at about 2200°. At this time, there are no nearby cells which are becoming wet or dry. It seems plausible that these small anomalies in the profile may be caused by energy balance problems, possibly created by the symmetry of the Gaussian bathymetry. The purely mathematical nature of this shape may have repercussions in the solution, but it is also likely that an island of this shape may have the ability to create shorter period trapped internal waves which propagate around the island. The shorter period oscillations seen in the elevation profiles may well be crude attempts of the model to represent this phenomenon. This explanation is unlikely to be the case though, since the resolution of the model is probably insufficiently fine to represent this type of feature.

Alternatively, a more likely explanation for these short period oscillations may be that they are, in fact, an indication of trains of energy advecting around the model, perhaps with some form of total (or partial) internal reflection from the boundaries, with their source being a shock induced by the moving of the land-sea boundary. If this is the case then the conservation of momentum seems to be adhered to quite rigorously in the model.

The short period oscillations are also apparent in the velocity profile shown in figure 3.5. As with Leendertse's (1970) method, there are quite large discontinuities apparent caused by the wetting of nearby cells. These discontinuities cause the rise of the tide to be modelled as a tendency towards a smooth curve rather than a smooth curve. In other words, the profile is attempting to become smooth with the apparent effect of the discontinuities being dampened towards high water.

The drying of the cells appear to be quite well represented, with only a slightly marked decrease in the rate of fall of the tide in the shallowest cell as it approaches dryness at around 2320°.

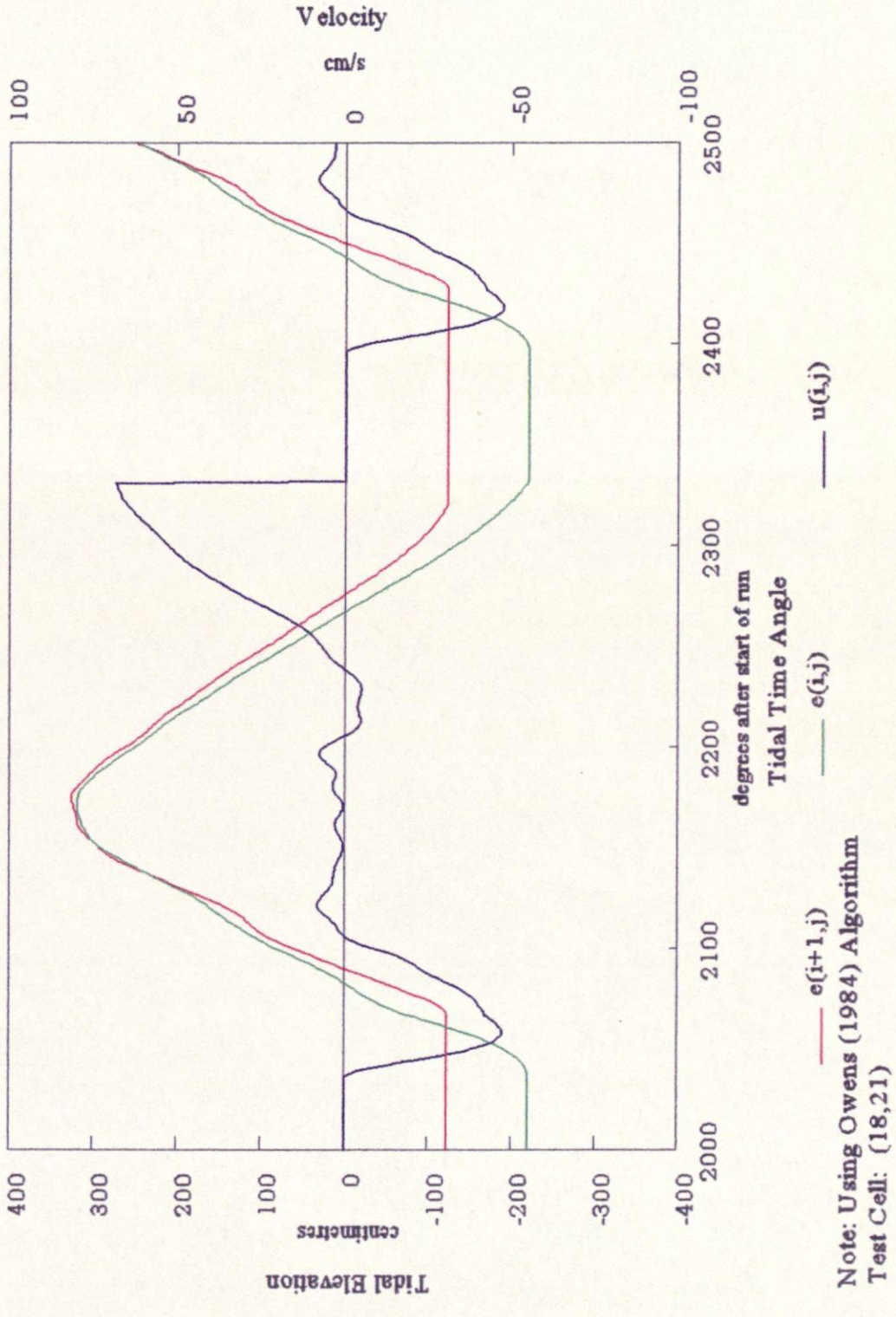


Figure 3.9: Elevation and easterly velocity component of a test cell with elevation of an adjacent cell

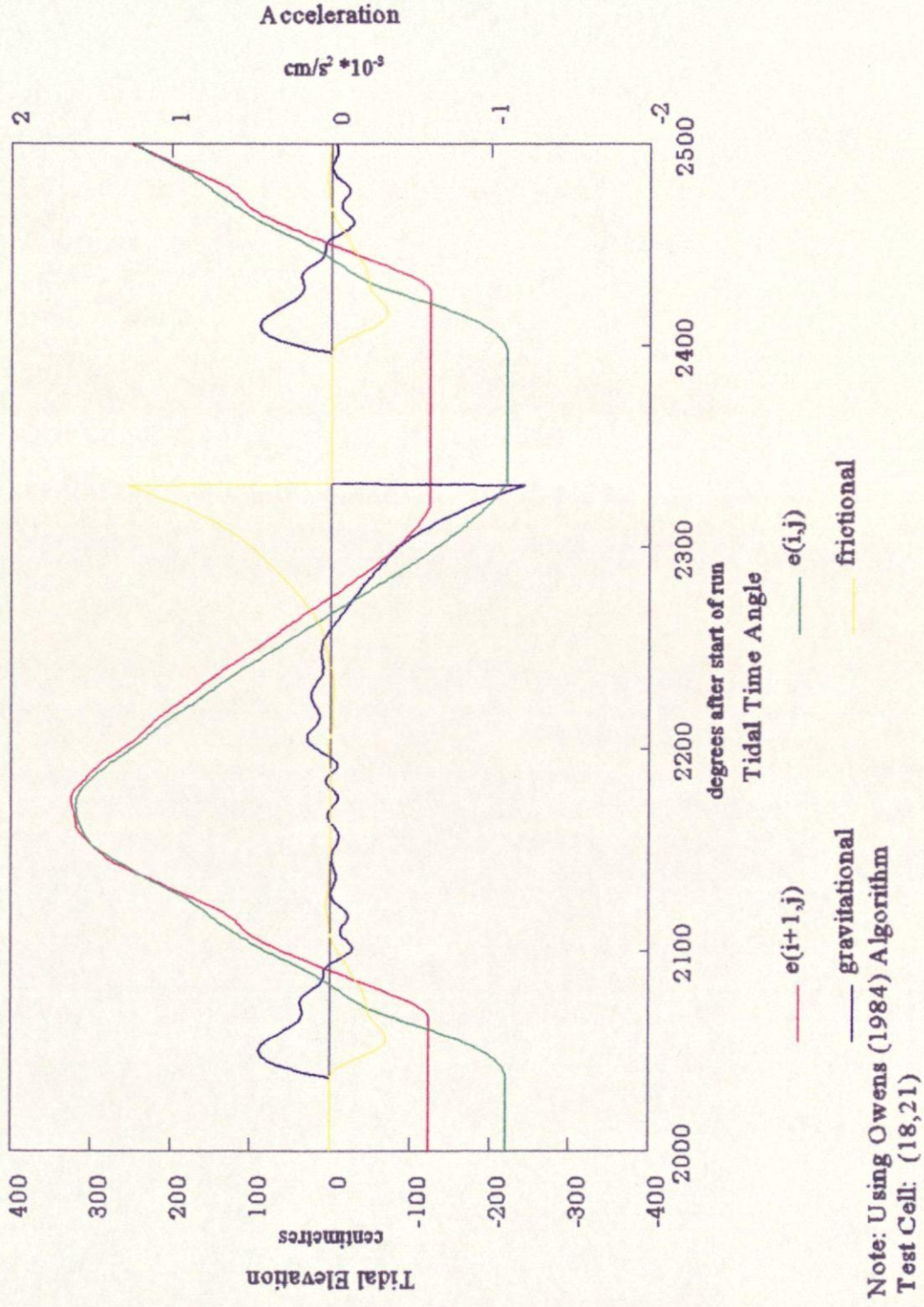


Figure 3.10: Elevation and accelerations due to both gravity and friction of a test cell with the elevation of an adjacent cell

Figure 3.11: Initial wetting period

Note: Using Owens (1984) Algorithm

The initial wetting period can be seen much more clearly in figure 3.11. Again, we can see the differences in the rise rates of an already wet cell and a cell which has just become wet, along with the subsequent oscillation caused by the attempt of the momentum balance to gain a hold.

The gravitational and frictional terms again show a balance as the water depth decreases.

3.5.3 Flather & Heaps (1975) Drying Algorithm

Time series obtained through the implementation of Flather & Heaps (1975) drying algorithm can be seen in figures 3.12 to 3.14. Figure 3.12 shows some elevation profiles of adjacent cells together with the easterly velocity component. The elevation profiles can be seen to be suffering from a similar effect previously described, where there are long frequency oscillations in the profiles caused by the unrealistic wetting rates.

The velocity profile in figure 3.12 also demonstrates this effect. From about 2090° to 2130° there are some quite severe falls in the strength of the predicted tidal stream, which would not occur in nature. These anomalies are a reflection of the momentum balance trying to gain equilibrium between the two different water levels in the adjacent cells. In contrast, the rate of the tidal stream during the falling tide is free from such numerical disturbances.

There are more examples of this phenomenon in figure 3.13. This figure shows the balance between the slope term (gravitational) and the friction term. The slope term is as it says, in that it is dependent on the surface slope between adjacent cells. During the period on the rising tide where there is a discrepancy between the rates of rise of the tide in nearby cells, *i.e.* the slope between them is uncertain, the slope term demonstrates dramatic oscillations which appear to dampen by high water. The frictional acceleration is very small when the

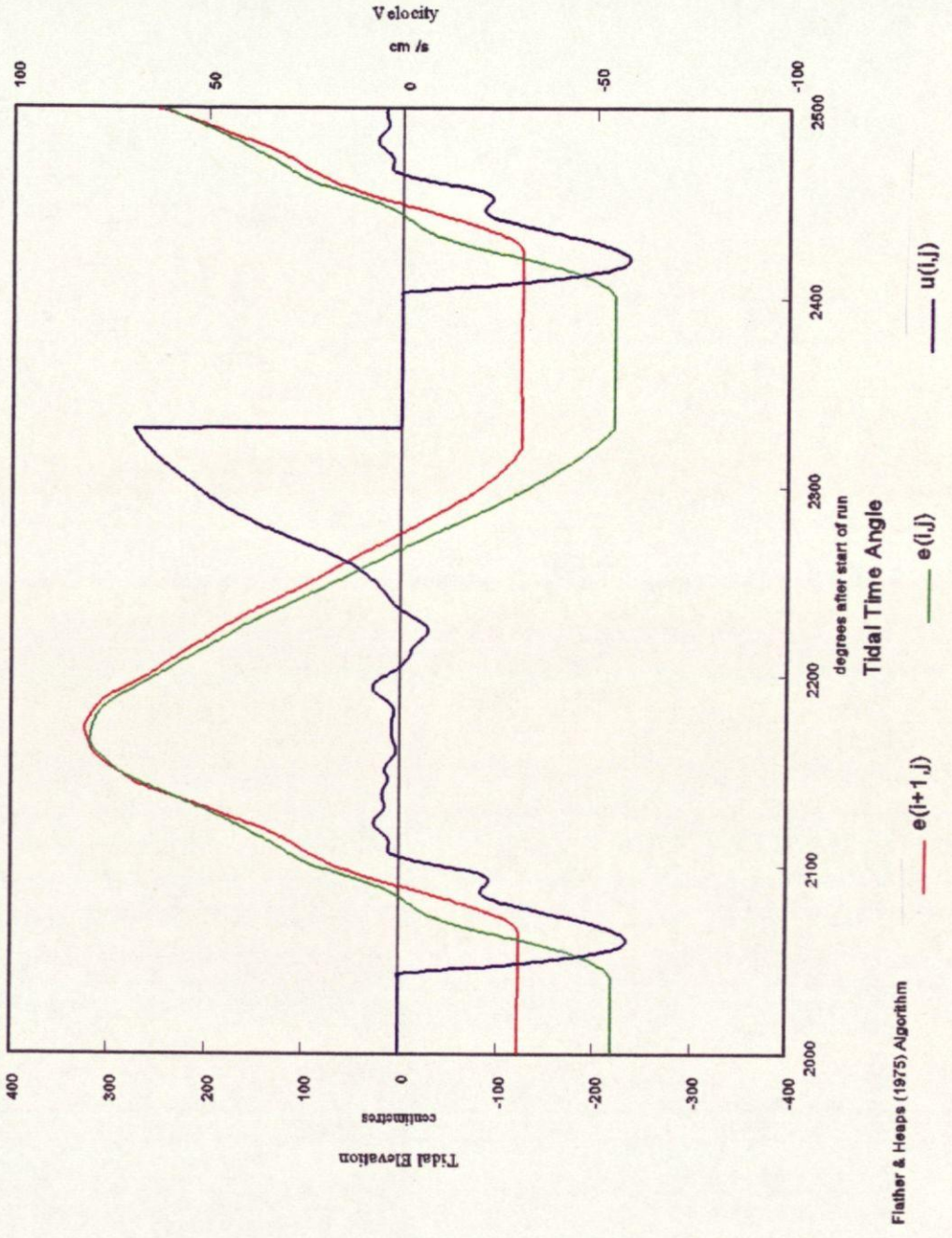


Figure 3.12: Elevation and easterly velocity component of a test cell with elevation of an adjacent cell

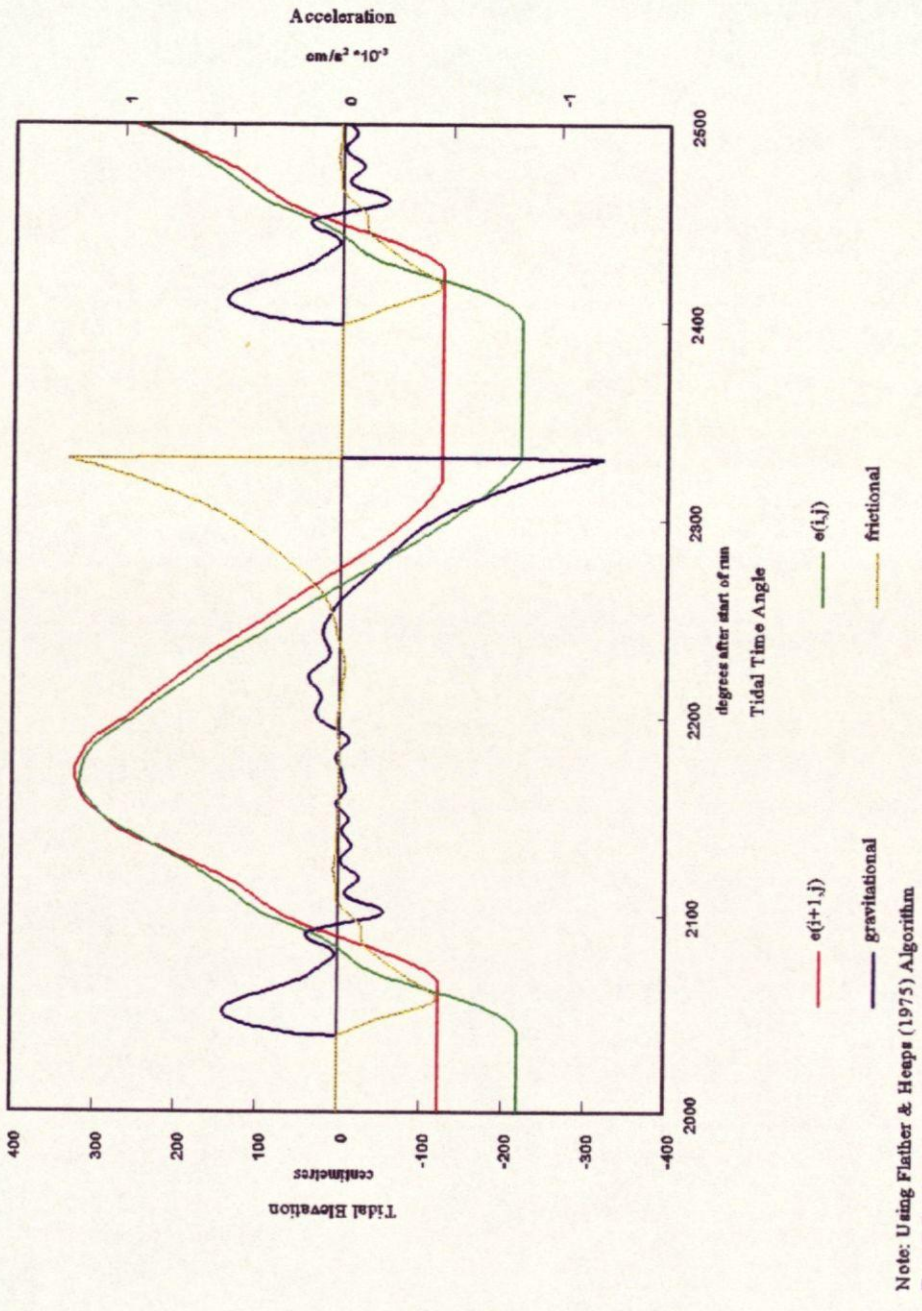


Figure 3.13: Elevation and accelerations due to both gravity and friction of a test cell with the elevation of an adjacent cell

Figure 3.14: Initial wetting period

Note: Using Flather & Heaps (1974) Algorithm
Test Cell: (18,21)

water is deeper, and the relative effect of the uncertainty in rates of rise is therefore not as marked in the profile of the frictional acceleration. When the water is shallower, however, the frictional and gravitational terms tend to balance each other.

Figure 3.14 shows an enlargement of the initial wetting period for the method presented by Flather & Heaps (1975). In this figure we can just make out that at about 2030°, cell e(1+1,j) dries completely after having become slightly wetted about 50° of tidal time earlier. This is purely a numerical artefact. We can also see the wide differential between the rates of rise of the adjacent cells at the point of cell e(i+1,j) becoming wet.

3.5.4 Rate of Rise of Flooding Cells

Table 1 shows quantitatively the difference between the rates of rise of the tide between the test cell and a nearby cell (as shown in figures 3.6 to 3.14) the instant that the test cell becomes wet. The nearby cell may be considered to be representative of the environmental tidal conditions as it is already completely wet. As can be seen, there is an order of magnitude difference in the rise rates between the test cell as it becomes wet and the environmental rate of rise of the tide in the nearby wet cell. Ideally, the rates of rise of the tide in the wetting cell and its environment should be almost identical. Included in the table is information on the improved algorithm as presented in Chapter 4.

Chapter 4 discusses the problems demonstrated by the above drying algorithms, together with explanations of the causes.

Drying Algorithm	Environmental Rate of Rise of Tide, ms^{-1}	Rate of Rise of Tide in Wetting Cell, ms^{-1}
Leendertse (1970)	$5.11*10^{-4}$	$4.11*10^{-5}$
Flather & Heaps (1975)	$5.65*10^{-4}$	$9.66*10^{-5}$
Owens (1984)	$5.75*10^{-4}$	$5.79*10^{-5}$
George & Stripling (1994)	$4.35*10^{-4}$	$3.91*10^{-4}$

Table 1: To show the difference between the rate of rise of tide in a wetting cell and the environmental rate of rise

CHAPTER 4

THE NEW DRYING ALGORITHM

As has been shown in Chapter 1, much work has been done in attempts to eradicate the inherent errors which occur when representing the moving land/sea boundary during solution of the hydrodynamic equations in two dimensions. A method has been developed whereby the moving boundary can be modelled with a minimum of numerical disturbance while computational time allowed for the calculations has not been significantly increased.

This improvement in moving boundary representation is applicable to finite difference methods in particular and, although the physical representation has been modified, the relevant physical laws expressed by Newton have remained unaltered.

4.1 The Richardson Grid

For the purposes of wetting and drying, a cell in the Richardson grid (figure 2.2) is defined as being centred about the elevation point (figure 4.1), rather than consisting of an elevation point, two stream points, and a depth point (as shown in figure 2.2). Such a cell is deemed to be bordered by four 'gates' leading to the four adjacent cells. This provides a

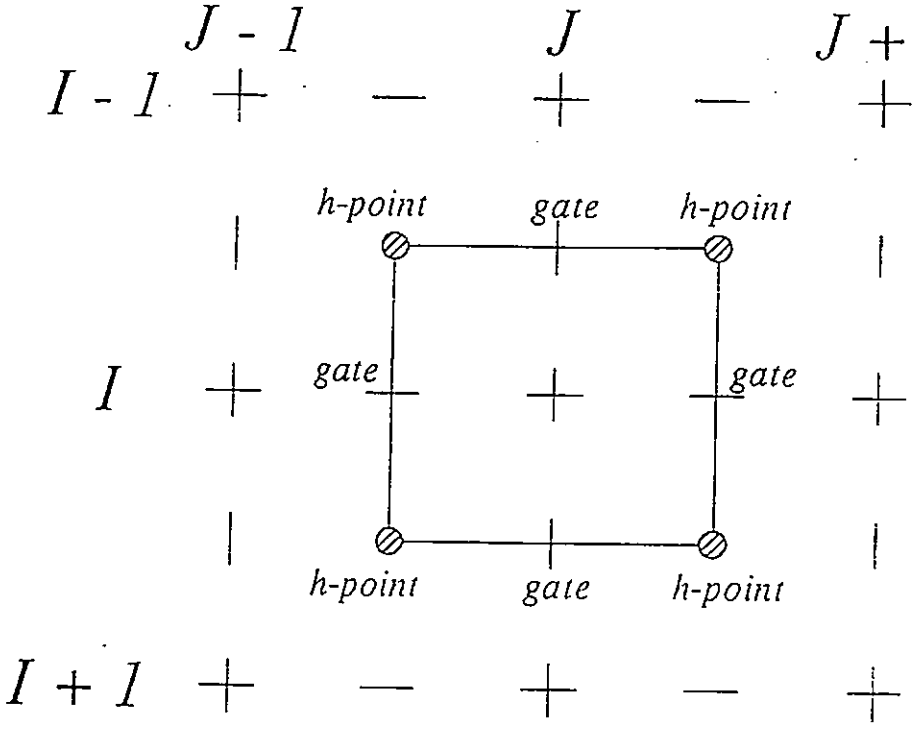


Figure 4.1: Definition of a cell in the Richardson grid for the purposes of wetting and drying

computational grid with an elevation point at the centre, four stream points at each side and four depth points, one at each of the vertices of the cell.

Typically, cells are deemed dry if the water elevation at the elevation point is less than the elevation of the sea bed at the elevation point and wet if the water elevation is above the elevation of the sea bed at this point. Gates are switched on and off according to techniques already discussed in Chapter 3.

Since the depth of the cell as defined in figure 2.2 is specified not at the 'ζ-point' but at the '*h*-point', then the depth of the sea bed (referenced to a fixed datum) at the 'ζ-point' of a potential drying cell, defined as in figure 4.1, must be evaluated from the depths at the surrounding '*h*-points'.

4.2 Difficulties in Wetting and Drying

4.2.1 Cells Which Dry too Slowly

Early test runs of the existing algorithms analyzed showed that cells which were drying had a layer of water left in them which amounted to a few decimetres. This water was gradually draining out of the grid square but not fast enough to keep up with the decrease in water level in the surrounding wet grid squares. This represented a state whereby a 'dry' grid square was 'high but not dry'. For this water to keep up with the surrounding environmental fall rate, the remaining water had to leave the drying grid square at grossly exaggerated speeds of up to 20 ms^{-1}, thus affecting the streams in nearby wet grid squares through the continuity equation.

In order to avoid having grid squares which do not dry fast enough, the real depth of water is replaced by a computational depth if it drops below a critical depth, H_c. The usual reason

given for the necessity of a 'critical depth' is to avoid the frictional term becoming too large and inducing instability. It is, however, possible for the critical depth to be very small (e.g. 0.01m) before instability due to this cause arises. It is therefore suggested that the introduction of a critical depth will be more effective in preventing the slow drainage of drying grid squares than in preventing instabilities occurring. It is true, however, that some value of a critical depth is needed to prevent a singularity occurring in the friction term.

The value required to prevent grid squares suffering from slow drainage can be estimated from theory. This theory begins with the assumption that in very shallow water, the motion is dominated by friction and gravity. This can be shown by scaling. The equations of motion therefore may be reduced to:

$$-g\frac{\partial \zeta}{\partial x} - \frac{KU\sqrt{U^2+V^2}}{\zeta + h} \approx 0$$

$$-g\frac{\partial \zeta}{\partial y} - \frac{KV\sqrt{U^2+V^2}}{\zeta + h} \approx 0$$

If we consider a cell with only one adjacent wet cell, (cell *b*, figure 4.2) then the equations may be reduced to one dimension:

$$-g\frac{\partial \zeta}{\partial r} - \frac{KR|R|}{\zeta + h} \approx 0$$

where R = rate of stream.

Ignoring signs, and putting S = $\partial\zeta/\partial$r = slope of the water surface, we have:

$$gSH \approx KR^2 \qquad (4.1)$$

where H = ζ + h.

If we consider a cell of side 'r'; the rate of discharge of water from it, Q, is given by:

$$Q = RHr$$

and also by:

$$Q = \phi r^2$$

where ϕ is the rate of fall.

Equating these two expressions gives:

$$\phi = \frac{RH}{r} \tag{4.2}$$

Substitution of 4.1 in 4.2 gives:

$$\phi = \frac{H}{r}\sqrt{\frac{gSH}{K}} = \frac{1}{r}\sqrt{\frac{gS}{K}}H^{3/2} \tag{4.3}$$

In order for cells not to be subjected to slow drainage, the rate of fall as given by very shallow water theory in equation 4.3 must be the same as that experienced in the adjacent deeper cell due to the general (environmental) rate of fall of the tide. Rearranging equation 4.3 gives:

$$H = \left(\frac{Kr^2\phi^2}{gS}\right)^{1/3} \tag{4.4}$$

where S is taken to be the environmental rate of fall.

Typical values of these parameters in the idealized model may be:

$\phi = 3\text{x}10^{-4}$ ms^{-1}, S = 10 mm/km = $1\text{x}10^{-5}$, r = 5000m,

g = 10 ms^{-1} and K = 0.0025 giving:

$$H_c = \left\{\frac{0.0025(5000\times3\times10^{-4})^2}{10\times1\times10^{-5}}\right\} = 3.8m$$

This is surprisingly deep, and since equation 4.1 is less likely to hold at such a depth, it must

be regarded as an estimate only. It would appear that operating the idealized model with a critical depth of 3.8m would eliminate the difficulties associated with drying and wetting. As this is clearly not an option, an alternative must be sought.

4.2.2 Cells Which Wet too Slowly

As a dry cell begins to wet, the environmental rate of rise of the tide is larger than the rate of rise that can be withstood by the wetting cell while retaining sensible flooding velocities. To show this, consider a cell which is beach and is about to flood (figure 4.2):

The wet cell, cell *s*, has a head of water waiting to wet the dry cell, cell'*b*'. In this instance, cell *b* is totally dry. Now, consider what will happen on the first wetting time-step:

The cross sectional area = $(\zeta_s - \zeta_b)r$,

where; ζ_s = elevation in cell *s*,

ζ_b = elevation in cell *b*,

r = length of side of cell.

Therefore, the volume of water entering cell *b* on the first wetting time-step is given by:

$$Vol = (\zeta_s - \zeta_b) r U \Delta t$$

This volume of water is instantly distributed over the whole area of cell *b* (r^2). Therefore, the depth of water,D, in cell *b* immediately after the first wetting time-step is given by:

$$D = \frac{(\zeta_s - \zeta_b) U \Delta t}{r}$$

which gives a rate of rise of tide in cell *b*, R_b, of:

$$R_b = \frac{(\zeta_s - \zeta_b) U}{r} \qquad (4.5)$$

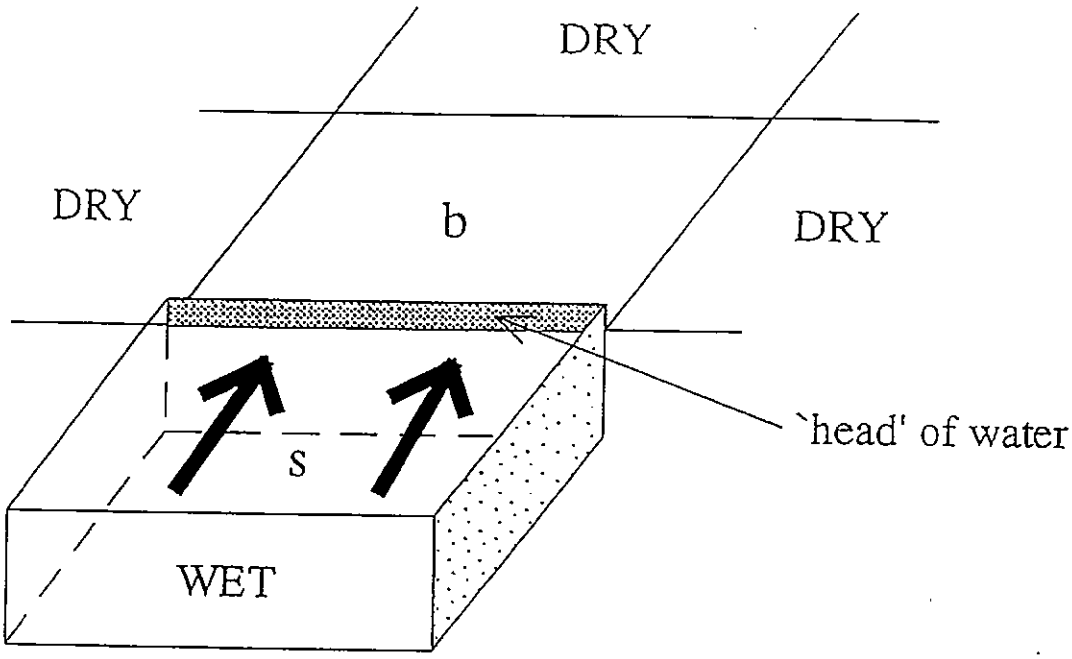


Figure 4.2: A dry cell, b, about to become wet through cell s

For a 10m tidal range, the rate of rise of tide in cell s, $R_s \approx 0.7 \times 10^{-3} ms^{-1}$. With a time-step of 124.2s, the difference in sea-surface elevations between the wet and the wetting cell is given by:

$$\zeta_s - \zeta_b = R_s \Delta t = 0.7 \times 10^{-3} \times 124.2 = 0.09m$$

With the rate of rise of the tide in the wetting cell given by equation 4.5, assuming a remarkably fast flow rate of 2 ms^{-1};

$$R_b = \left(\frac{0.09 \times 2.00}{5000} \right) = 3.6 \times 10^{-5} ms^{-1}$$

Thus there is a large difference in the rate of rise of the tide in the wet cell and the wetting cell.

For a model which has a much smaller grid size, e.g. a fine-scale model of The Wash:

r = 740m, Δt = 62.1s.

Here, $\zeta_s - \zeta_b = 0.7 \times 10^{-3} \times 62.1 = 0.043$m

Again, assuming a reasonably fast flow of 2ms^{-1}:

$$R_b = \left(\frac{0.043 \times 2.000}{740} \right) = 1.2 \times 10^{-4} ms^{-1}$$

This shows that even for a finite difference grid of a considerably smaller size, the problem of wetting too slowly is still significant.

We can determine at what rate the water needs to enter the wetting cell for there to be no difference in the rate of rise of the tide therein, *i.e.* $R_s = R_b$:

$$R_b = \left(\frac{(\zeta_s - \zeta_b) U}{r} \right) = \frac{R_s \Delta t U}{r}$$

Therefore, for $R_s = R_b$:

$$\frac{\Delta t U}{r} = 1 \quad \text{or;} \quad U = \frac{r}{\Delta t}$$

For r = 5000m and Δt = 124.2s:

$$U = \frac{5000.0}{124.2} = 40.3 ms^{-1}$$

Clearly, this is unrealistic.

Eventually, the differences in rates of rise of tide between wetting and wet grid squares will be overcome as the pressure difference induces 'normal' flow.

There are several ways of solving this problem. These are either artificially to slow down the rate at which the tidal wave rises (the 'Canute method'), or to increase the rate of rise in the flooding cell whilst retaining a sensible flow rate.

As the first option would represent an unnatural phenomenon, it can not really be considered. The second approach involves the addition of 'layers' of water in the flooding cell at discrete intervals to keep the sea-surface elevation in equilibrium with the seaward rise in elevation. However, the addition of small volumes of water in such a process again introduces discontinuities in the solution of the Navier-Stokes equations.

4.3 The Method of Sloping Facets

An alternative solution to the introduction of a critical depth is to allow the cell in question, defined by the Richardson grid, to remain operational for as long as possible without disobeying the physical laws.

Traditionally, in the use of finite difference methods, the bathymetry of the sea bed is schematized through the representation of each grid square as having a constant depth over its entire area. When the depth of each grid square is defined in this manner, the sea bed loses its gently sloping nature and becomes artificially represented by a series of steps or plateaux. Thus, computationally, grid squares appear to have vertical walls leading to adjacent grid squares (figure 4.3).

It may be noted here that when the bathymetry of a sea area is shown graphically as a three-dimensional image (e.g. figure 4.4), what is being seen is <u>not</u> what is being used computationally. However, it does seem to be a closer representation of nature than a series of plateaux joined together at vertical walls.

A method has been developed whereby the bathymetry <u>is</u> approximated by a series of 'sloping facets' (figure 4.5). As with the stepped sea-bed representation, these facets are also centred on the 'ζ-point'.

This creates a situation where each gate can allow water to flow through it even when the water level has dropped below the bed elevation at the 'ζ-point'. In fact, water may flow in or out of a cell when as little as 1% of the sloping facet has water in it. Obviously, this will be in the deepest corner of the cell.

4.3.1 Introducing Areal Factors

The idea of areal factors is not a new one. Flather and Hubbert (1989) have also applied a method of areal factors. However, their technique remained rigid to the finite difference scheme and consequently only considered a higher resolution of grid squares within a coarser scale grid square.

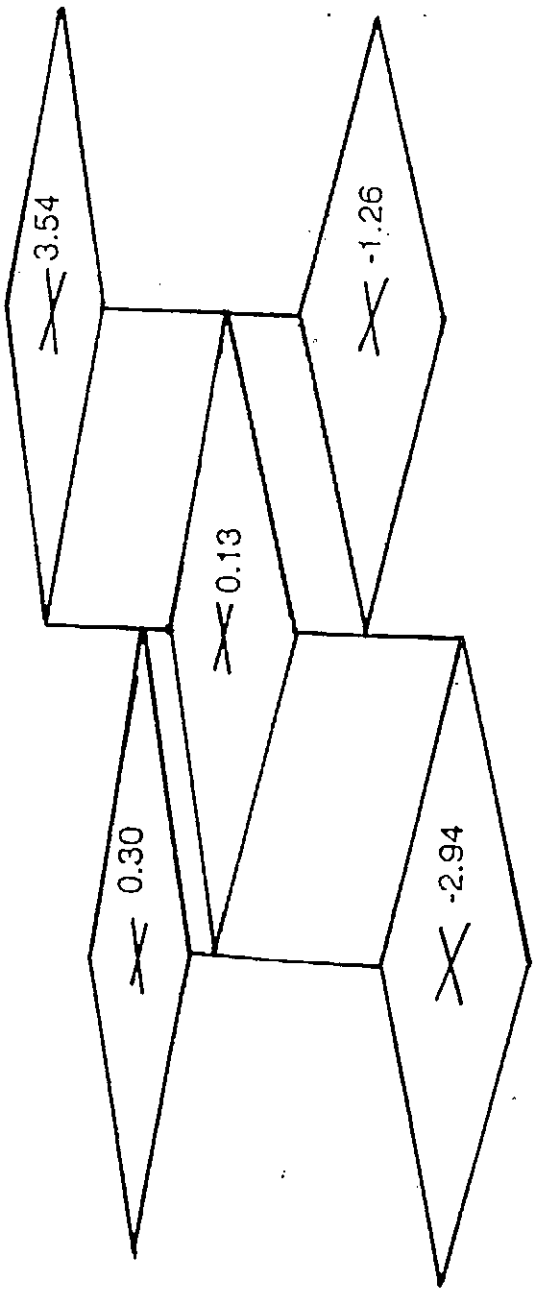


Figure 4.3: A stepped bathymetry

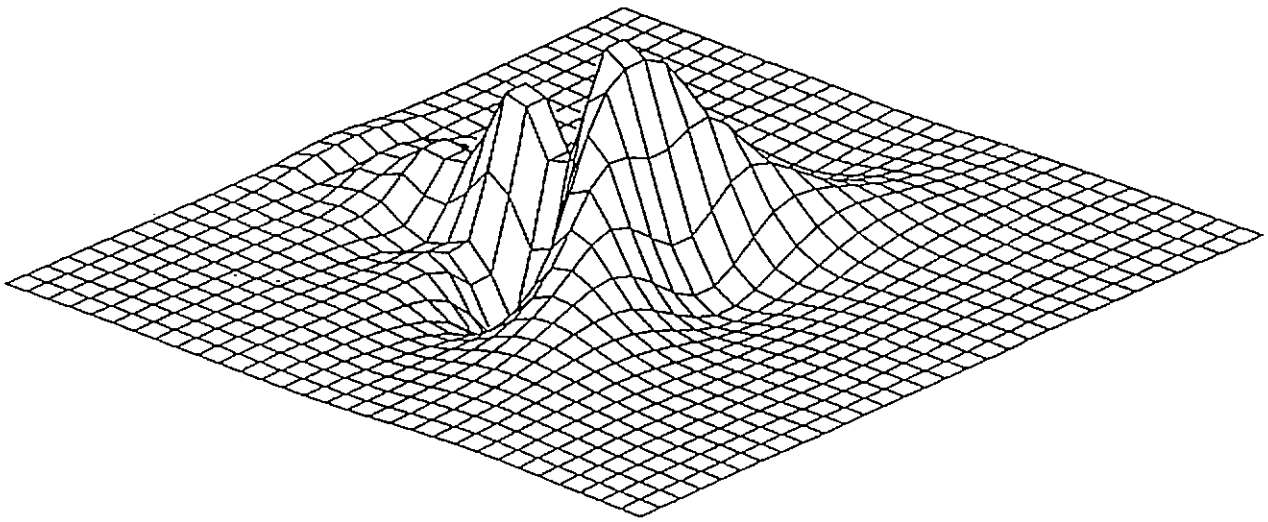


Figure 4.4: An example of an isometric projection (HR Wallingford)

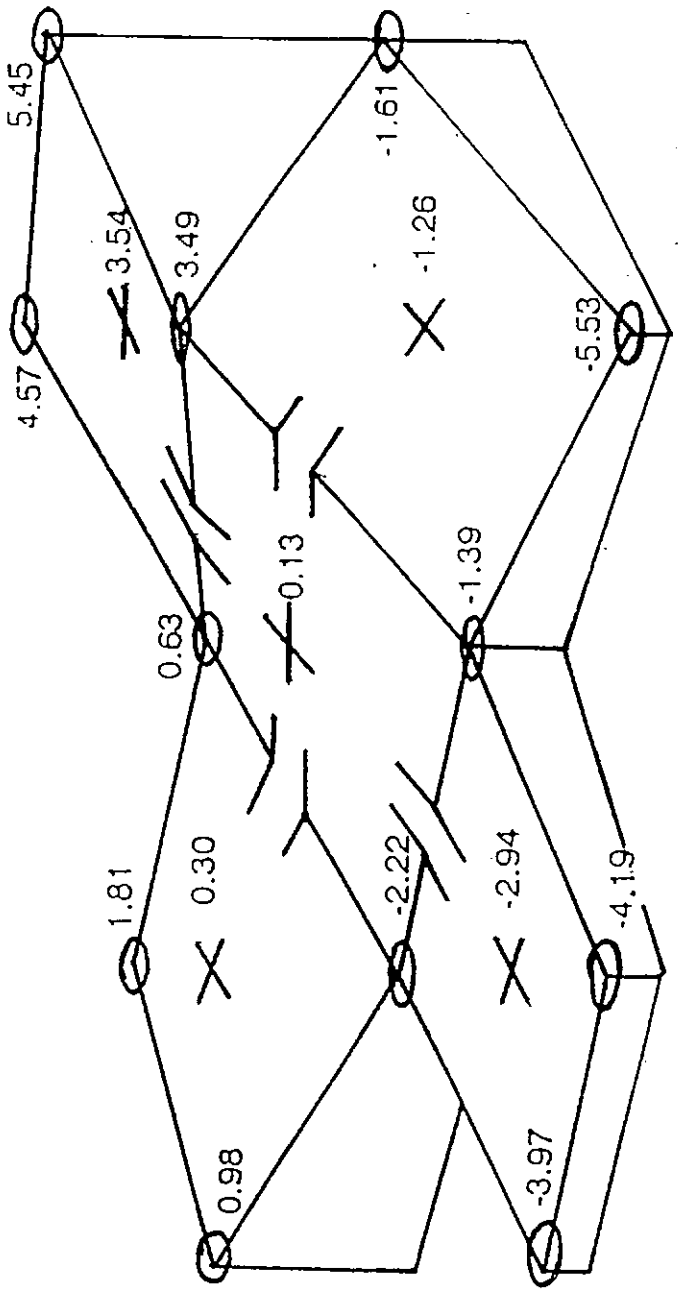


Figure 4.5: The stepped bathymetry from fig. 4.3 shown as sloping facets

Through the improved application of areal factors, there is absolutely no need for extra bathymetric digitization and higher computational costs. It works by the areal factor being defined as a triangular portion of the whole grid square. Consequently, a cell may gradually become wet or dry by the change of area of a triangle or two triangles, depending on the orientation of the slope and the water level within that sloping cell.

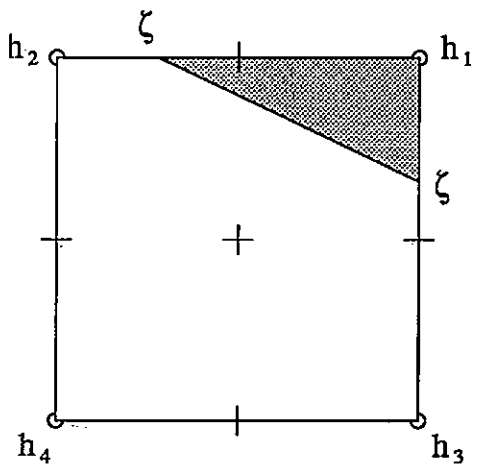
There are five cases possible, each requiring a different method of evaluating the areal factor. Let the heights above a fixed datu of the sea bed at the corners of a cell defined on the Richardson grid be h_1, h_2, h_3 and h_4, where $h_1 < h_2 < h_3 < h_4$ (h measured positive above a fixed datum). Let ζ be the elevation of the sea surface within the cell:

Case 1

$\zeta < h_1$ Areal Factor = 0 (*i.e.* cell is completely dry)

Case 2

$h_1 < \zeta < h_2$ Areal Factor = shaded area $= \frac{1}{2}\left(\frac{\zeta - h_1}{h_2 - h_1}\right)\left(\frac{\zeta - h_1}{h_3 - h_1}\right)$

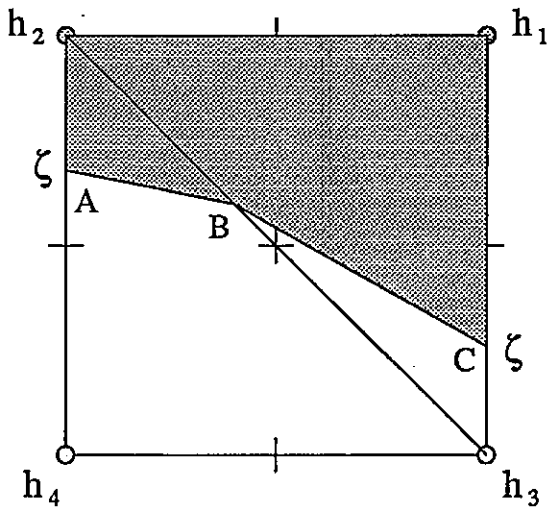


Case 3

$h_2 < \zeta < h_3$

The areal factor is established using a local coordinate system and Heron's formula is used to find the areas of the triangles whose vertices are indicated below.

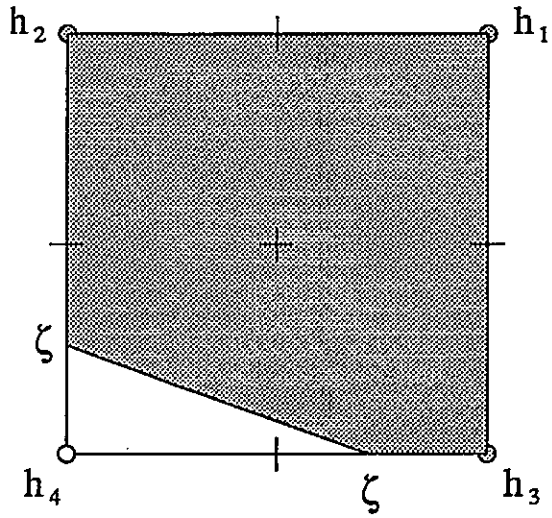
Areal factor = (area$\Delta h_1h_2h_3$ - areaΔBCh_3) + areaΔh_2AB



Case 4

$h_3 < \zeta < h_4$

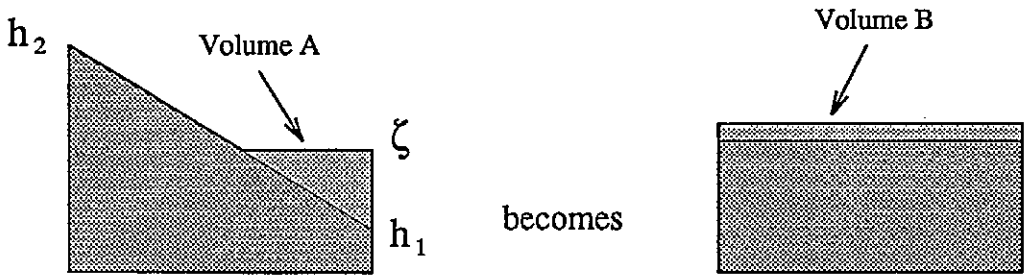
Areal factor = shaded area $= 1 - \frac{1}{2}\left(\frac{h_4 - \zeta}{h_4 - h_3}\right)\left(\frac{h_4 - \zeta}{h_4 - h_2}\right)$



Case 5

$h_4 < \zeta$ Areal factor = 1

In order that the cell is computationally active, even when the tidal elevation is below the depth at the elevation point of the cell, a breadth factor is applied. This factor has the effect of redistributing the volume of water in a partially wet sloping cell along the whole edge of a computational cell which has horizontal edges, causing the stream point along the edge to be considered wet. Thus, volume A is equal to volume B below:



There are three possibilities which may arise here, firstly, if $\zeta < h_1$ then the breadth factor is zero, secondly, if $\zeta > h_2$ then the breadth factor is one. In the third case, when $h_1 < \zeta < h_2$, the breadth factor is given by:

$$breadth\ factor = \frac{\zeta - h_1}{h_2 - h_1}$$

Figure 4.6: Elevation and easterly velocity component of a test cell with the elevation of an adjacent cell

Figure 4.7: Elevation in a cell with accelerations due to both gravity and friction

Figure 4.8: Initial wetting period

Note: Using George & Stripling (1994) Algorithm

4.4 Comparison With Other Drying Methods

The idealized model has been used as a tool to compare the method of sloping facets with the algorithms tested in Chapter 3.

The results can be seen in Figures 4.6, 4.7 and 4.8. This comparison shows a marked improvement on the existing algorithms, with numerical shocks caused by the wetting of a nearby cell having been completely eliminated using the new method of sloping facets. The velocity profile (figure 4.6) shows little or no adverse effects due to surrounding cells. The initial uncertainty of whether or not a cell has become wet or not as demonstrated in figure 3.8 (Leendertse ;1970) does not apply with this method.

The oscillations in the gravitational profile that were so prevalent in figures 3.7, 3.10 and 3.13 during the rise of the tide have been almost completely eliminated, see figure 4.7.

The initial wetting period shown in detail in figure 4.8 illustrates how the rate of rise of the tide in the cell and its environment are almost identical, even at the precise moment of wetting. Indeed, as table 1 shows (p. 72), the rate of rise of tide in the wetting cell at the instant it becomes wet is almost identical to the environmental rate of rise that is represented by the rate of rise of the tide in a fully wetted adjacent cell. This appears to have eradicated the associated resultant oscillations in the solution entirely.

4.5 A Rogue Situation

During the course of experimentation, a situation was found in which the method of sloping facets ceased to work. As has been previously stated, on a finite difference grid the depths of the sea-bed are specified at the h-points, however, the Richardson grid requires that a cell is centred on a ζ-point. This leads to the necessity of having to establish the depth of the sea-

bed at the ζ-point by averaging the depths specified at the four surrounding h-points.

It was thought that if the depths of the sea-bed were initially specified at the ζ-point, as well as at the h-points, then the extra computational time which arises from the process of averaging would be avoided. This is readily achieved when the model is idealized and the extra depths merely need to be calculated with a simple algorithm. However, this led to a 'pyramidal' cell (figure 4.9) appearing at the peak of the Gaussian shoal. In this instance, the method of sloping facets does not work.

In reality though, this situation would not arise as depths are extracted from charts and other sources, not created through short algorithms. Even so, the author recommends that if this method is to be implemented anywhere that the tradition of specifying the depths of the sea-bed at the h-points should continue.

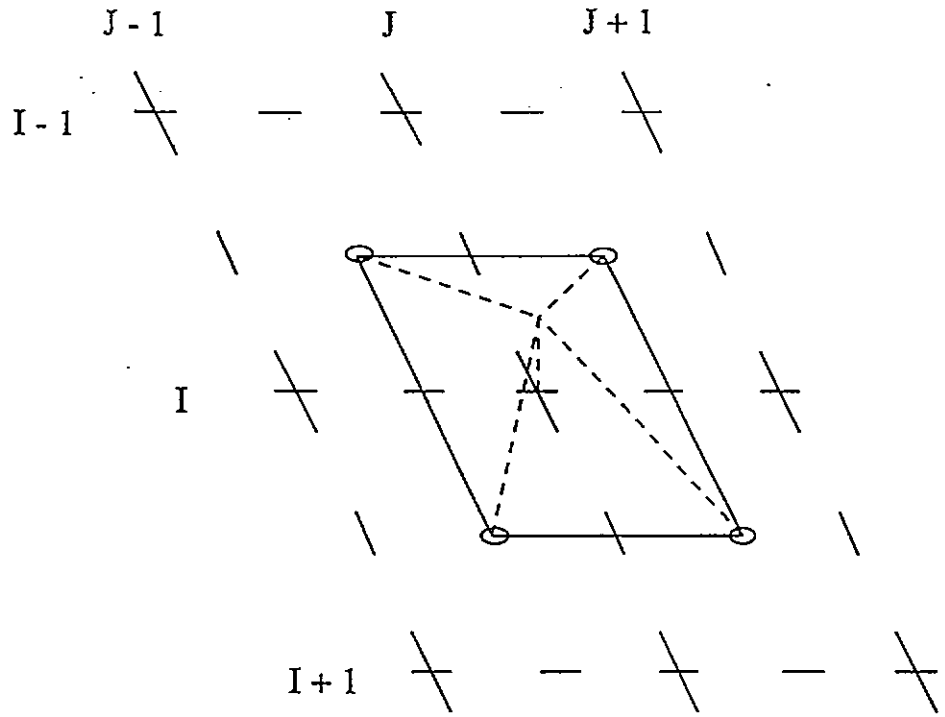


Figure 4.9: A pyramidal cell

CHAPTER 5

'THE WASH' - FIELD MEASUREMENTS

As a final test in determining the performance of the new drying algorithm it is necessary to apply the technique in a real situation. It was decided that a suitable test-bed for the algorithm would be The Wash, East Anglia.

5.1 The Wash

The Wash is a large embayment on the east coast of England, latitude 53°00'N, longitude 00°20'E (figure 5.1). It faces approximately northeast and is about 20km wide by 30km in length. At the entrance to the embayment there are deep channels (e.g. the Lynn Deeps) with maximum depths of about 40m. Over most of the area, however, depths are generally less than 10m. Mean tidal ranges are 6.5m during spring tides and 3.5m during neap tides.

Before land reclamation began, the Wash was a much larger area of sea and marsh (e.g. Harris, 1953). During the 17th century, a Dutch engineer called Vermuyden (1567 - 1641) initiated a remarkable feat of engineering which involved the draining of large parts of marshland surrounding the Wash, and its reclamation from the sea. This area is known as the Fennes. Since this reclamation began, extensive deposits of sediment have formed broad

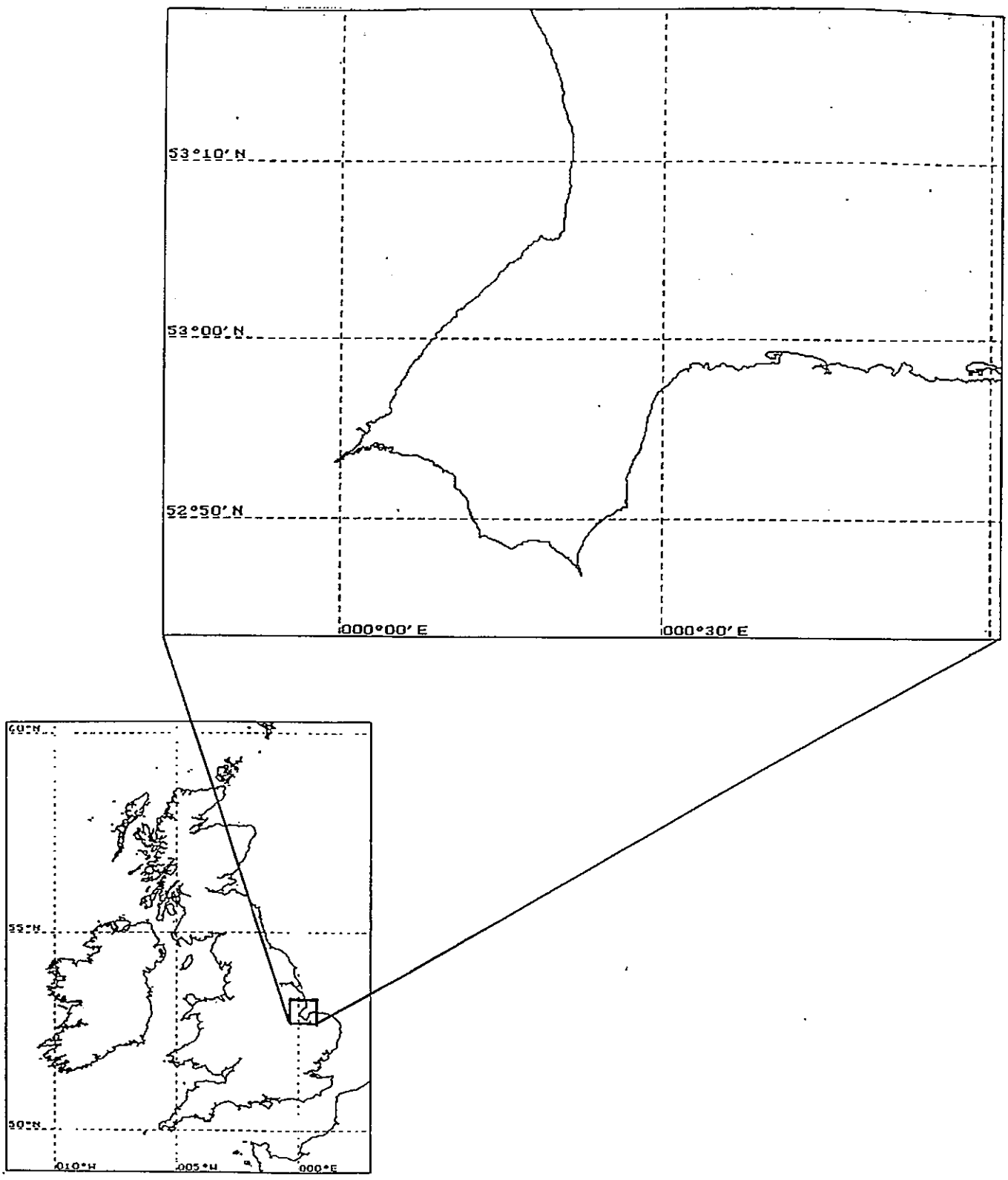


Figure 5.1: Location map of the Wash, U.K.

bands of intertidal regions along the northern and western regions of the Wash. Offshore, there are also large intertidal banks. Figure 5.2 shows an aerial photograph of the Wash. The sandbanks can be clearly seen, as well as the possible presence of large amounts of suspended sediment. It is probable that the majority of the silt in the Wash stems from the erosion of parts of the coast north up to the Humber estuary.

There are three main rivers draining in to the Wash; the Welland, Nene and Ouse. These drain a large portion of central eastern England (1.3×10^6 hectares (= 13000km^2), (Evans and Collins (1975)) between the Trent drainage basin to the north and the Thames to the south. Shipping channels for the inland ports of Spalding, Wisbech and Kings Lynn on these rivers can be seen in figure 5.2. There are substantial difficulties in keeping these channels open to shipping due to the vast quantities of sediment that are deposited in them. Retaining walls have had to be constructed to maintain the buoyed shipping lanes at constant positions. Regular dredging helps to keep them open.

Further inland, long sections of these rivers are raised high above the surrounding land (some surrounding areas can be more than 3m below mean sea level) and it is often possible to walk alongside the river bank and be below the level of the river bed. This gives some idea of the effect of land reclamation in the Fenland area.

Due to the fact that the majority of the surrounding countryside is very close to the mean springs high water level, the Wash is surrounded by a man-made sea-wall. Large sections of this sea-wall were built by convicts in the 1940s and 1950s. Apparently, very few returned for another go. Landward of the sea-wall runs a dyke system which holds any water which has seeped through during periods of high water. Periodically, automatic and manual pumping stations return the water to the sea.

Figure 5.2: Aerial photograph of the Wash showing survey line

5.2 The Study Area

A visit to the Wash was made in January 1992. This was intended as a reconnaissance exercise to assess a likely situation for measurement. After observing possible positions, a meeting with the harbourmaster at the port of Kings Lynn was arranged. The author was informed that during the previous year several people had drowned on the mudflats and advice was given that a survey should only be attempted by a minimum of two people, with distress packs, lifejackets and V.H.F. radio. In addition, an inflatable boat with a powerful engine, if not two, would be required.

It was therefore apparent that to set up a survey programme would require extensive planning and considerable funding. Fortunately, it came to the Author's attention that the Department of Oceanography at the University of Southampton had an ongoing programme of field measurements in the Wash, under the direction of Professor M.B. Collins. It was arranged that the Author would accompany their research team during the next period of field work. A week of field measurements were conducted by the Author in association with two of Professor M.B. Collins' research students (Mr. X. Ke and Mr. A. Flavell). This period of observation included the measurement of suspended sediment concentrations, water depth and current velocity detail for six different sites during nine tidal cycles.

The observations were taken along line 'AB' shown in figure 5.2 on the Frieston Shore. Access to the site was made through H.M.P Frieston (North Sea Camp) with the permission of the prison Governor (this particular site provided an undisturbed working environment. It also provided access to the shore by road and was situated away from the R.A.F bombing range).

Figure 5.3 shows a cross sectional representation of Frieston shore along the survey line.

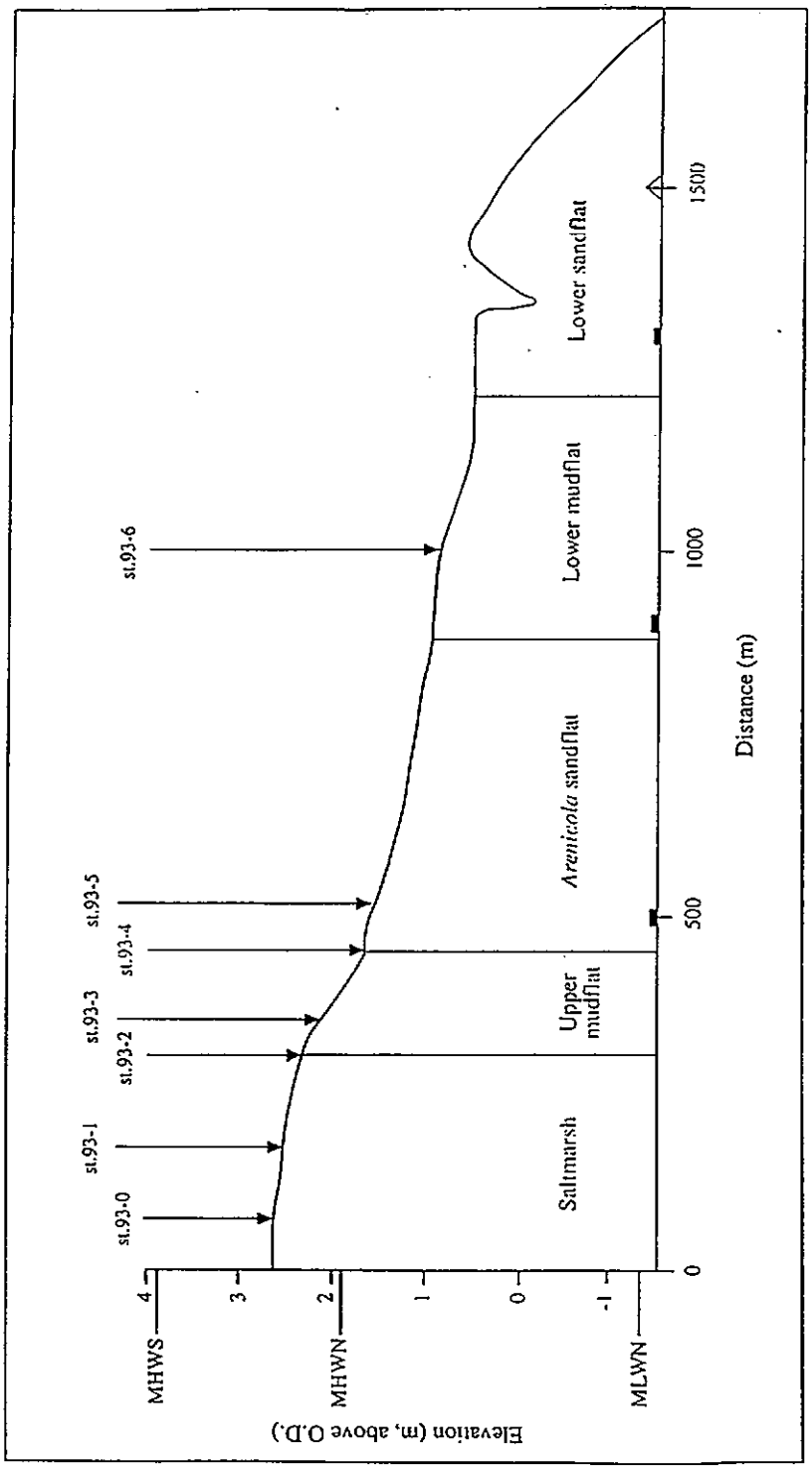


Figure 5.3: Locations of hydrological survey stations, Frieston Shore, 1993 (courtesy of Mr. X. Ke)

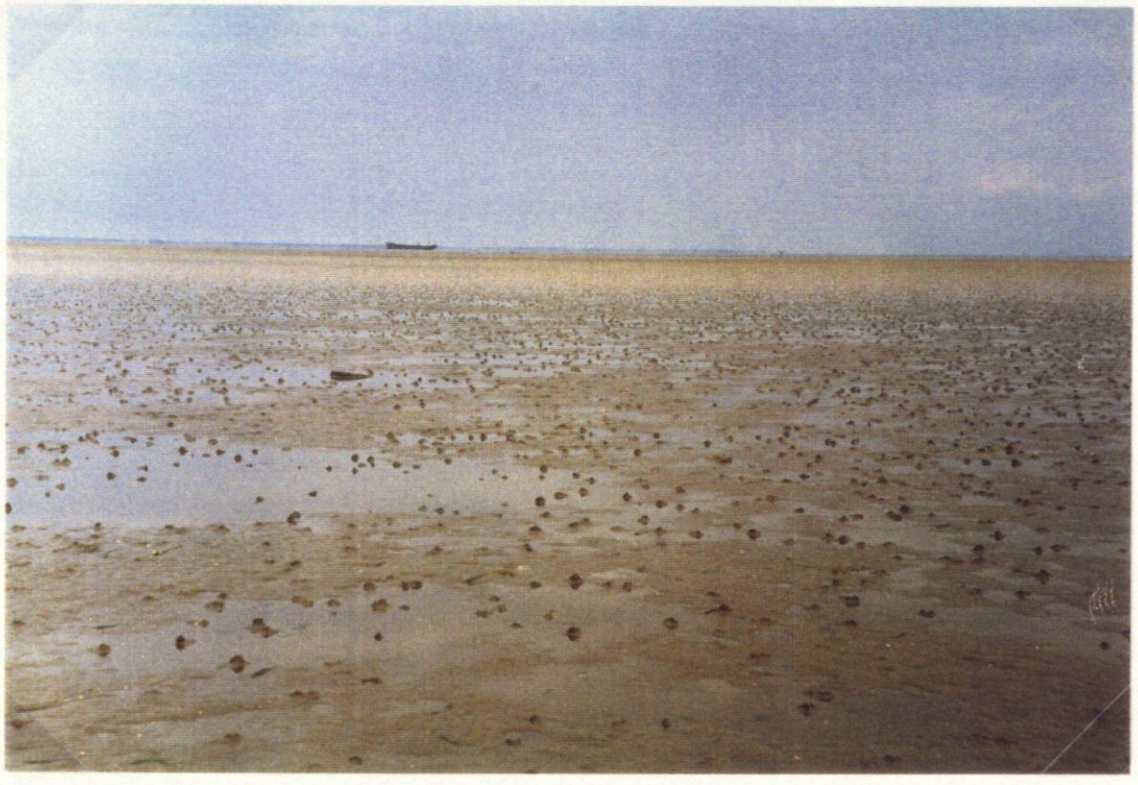


Plate 1: Seawards from station 6

Plate 2: Current meter rig

It also shows the positions of the six survey stations, and their approximate elevations relative to Ordnance Datum Newlyn. Plate 1 shows the expanse of inter-tidal flat from station 6, seaward.

5.3 Aims of the Field Work

Since the critical times during the tidal cycle, with repect to the accurate representation of flooding and drying, are the periods of wetting and immediately after, and the short period approaching drying, *i.e* when the water is extremely shallow, it was required that detailed measurements should be taken during these times. These measurements would then be used for validation purposes in a hind-casting model of the Wash.

5.4 Methods of Measurement

At present, there are no methods known to the Author which allow the measurement of tidal currents in extremely shallow water to any great accuracy, even the electromagnetic method can give spurious results in extremely shallow water. Therefore, the existing method of determining currents from the rotation of impellers was used and eventually adapted (see the tilting method below).

Two Braystoke direct-reading current meter rigs (see plate 2) were employed at each station of the survey site. The five impellers on each rig were set at optimum heights in order to obtain velocity readings throughout the water column. These current meters were connected to a Valeport multi-channel pulse counter (see plate 3) by thirty metres of cable.

During the initial wetting period and the period shortly before drying, the smaller of the two rigs, mounted with microimpellers (1.5cm in diameter), was held by hand and tilted so that

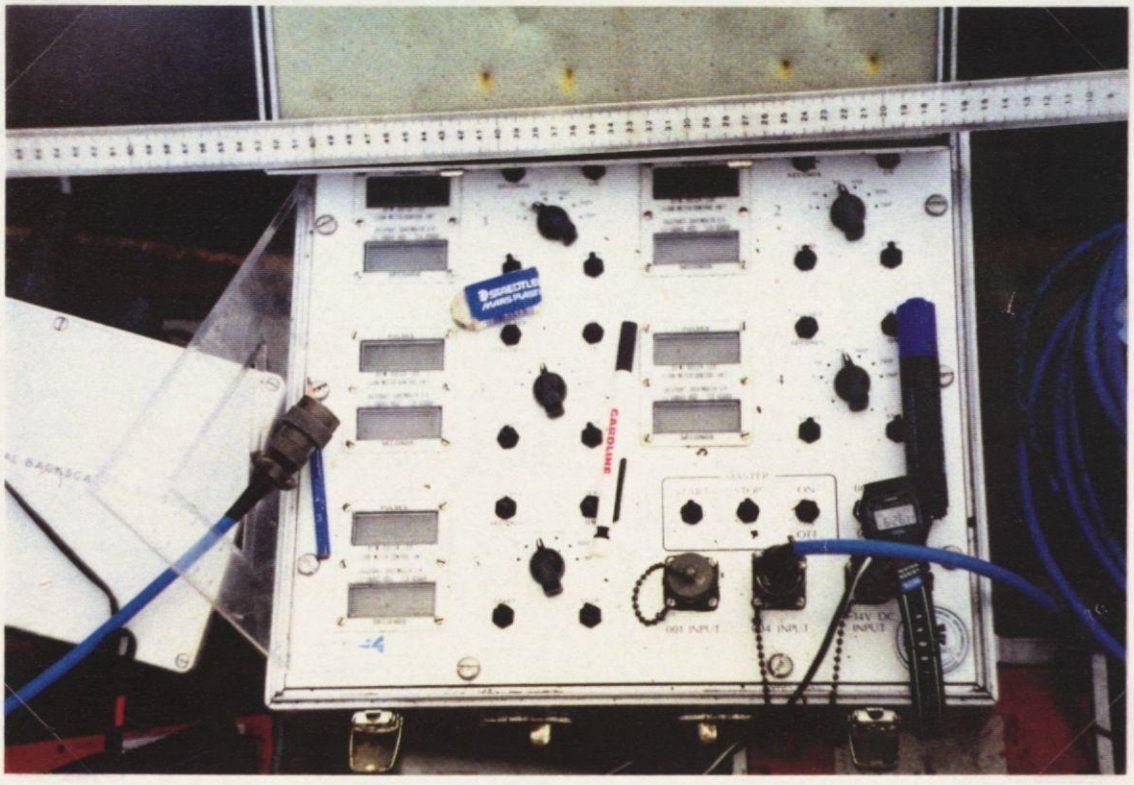


Plate3: Valeport multi-channel pulse-meter

Plate 4: Near high water showing benefit of bow and stern anchors

the lower impellers could be positioned as close to the sea-bed as possible. The reliability of this method is not known as there was a tendency for small sediment particles to jam the impeller and prevent rotation. If this occurred, the rig was removed from the water, freed, and returned as quickly as possible. Generally, though, measurements were continuous.

During periods when the water was extremely shallow, readings were taken every thirty seconds if possible, however, once the water depth had reached about 0.2m, readings were taken at greater intervals.

The inflatable boat was anchored fore-and-aft at some distance from the deployment site in order that it remained in a fixed position away from the current meter rigs. Thus avoiding over-riding the rigs at high water (see plate 4). Anchor lines were lengthened and shortened as the tide rose and fell, thus preventing the anchors dragging or keeping the slack out of them. Measurements could not be taken throughout consecutive tidal cycles as alternate tides were required to float the boat so that it could be towed to the next station.

Out of the nine tidal cycles measured, five provided thorough time series measurements in extremely shallow water. This was due to the differing requirements of the survey team and the rota system adopted to allow periods of rest. The Author was therefore not present on every tide measured and time series obtained during his absence were not entirely of the required detail.

5.5 Period of Observation

The survey took place from the 19th June 1993 to the 24th June 1993. This was during spring tides, so that the maximum amount of water was entering and leaving the Wash.

The weather during this period was exceptionally calm, leading to the almost complete

absence of wind waves. Only on one occasion was there rough weather of any kind. This turned out to be extremely valuable as interference by wind/wave induced currents with the tidal signal was kept to an absolute minimum.

Table 2 shows a summary of the survey programme, including a brief description of the sea-state and weather conditions.

Date	Time (BST)	Station	H.W. (m)	Survey Contents (H,R,S,SPM,WS)	Environmental Conditions
19/06/93	03:45-07:15	93-5	6.7	H,R,SPM	Breeze, high cloud, sunny, waves ↓20cm.
19/06/93	16:15-19:15	93-4	6.7	H,R,S,SPM	Light breeze, sunny, calm sea.
20/06/93	04:30-07:45	93-3	6.9	H,R,SPM,WS	W/NW breeze, clear sky, waves ↓5cm.
20/06/93	17:20-19:50	93-2	7.0	H,R,SPM,WS	NW breeze, high cloud, no waves.
21/06/93	05:15-08:00	93-1	7.0	H,R,SPM,WS	Fine, calm sea.
21/06/93	17:25-21:25	93-5	7.0	H,R,SPM,WS	SE breeze, bright, calm sea.
22/06/93	17:45-23:00	93-6	7.1	H,R,S,SPM,WS	SE breeze, drizzle, wind increasing, waves ↑50cm, choppy.
23/06/93	19:00-22:45	93-3	7.1	H,R,S,SPM,WS	Light breeze, overcast, slight swell.
24/06/93	07:25-11:00	93-0	7.4	H,R,SPM,WS	Calm, no breeze, no waves.

Where; H = Water Depth, R = Current Rate, S = Current Set, SPM = Suspended Particulate Matter, WS = Water Sample.

Table 2: Summary of Survey Programme.

Appendix 3 shows the entire data set of field measurements.

CHAPTER 6

'THE WASH' - A NUMERICAL MODEL

Typically, model development has proceeded by comparing new model output with earlier test results, as presented in Chapter 3, or by studying the sensitivity of model results to changes in model input parameters (e.g. section 2.5.6). Whilst refinements in modelling and parameterization are intuitively sensible extensions to existing procedures and are expected to result in better predictions, there is, generally, disappointingly little direct evidence from field observations that this is the case. Field researchers, struggling with limited resources and frequently attempting to identify particular processes from amongst the complexity of other processes which occur in nature, have generally not used models to inform and assist with the planning of field work, with the result that many of their results are not directly relevant to model testing. The result of these two separate approaches is too often modellers who are uncritical of their model results and see no need for field testing, and field researchers who distrust models because of their experience of the real environment.

It is quite apparent that in order for model predictions to become more accurate and field workers to begin to accept the valuable contribution of numerical modelling to the understanding of natural phenomena, that the two must be drawn closer together. However, due to the lack of resources available for modellers to carry out their own field measurements and the field researchers reluctance to place trust in modelling techniques, further

understanding of marine processes is unnecessarily hindered.

It is to further this end that the Author has carried out a programme of field observations with which to validate a numerical model; to show that it is possible for field work and numerical modelling to combine together to achieve greater understanding.

6.1 Model Design

There are several differences between the layout of an idealized model and a model of a real area. Whereas in an idealized model it is possible to avoid time-consuming construction of hydrographic data-files, in a model of a real area it is necessary to spend considerable time creating the required input files.

Two models have been designed here; a coarse grid model (2NM) and a fine grid model (0.4NM). Due to processing restrictions, only the coarse grid model has been run, with the intention of obtaining adequate computational power to run the finer grid (0.4NM) model in the future

6.1.1 Bathymetric Data File

Whereas for the idealized models the bathymetry could be specified using simple loops within the source code, when a model of a real sea area is desired, the shape of the sea-bed must be arrived at in a more complicated and time-consuming manner.

In order to provide a bathymetric data-file for a numerical model, two steps must be taken; firstly to gather the necessary data and secondly to interpolate these raw data in order to be represented on the structured grid. Data can be obtained by digitizing bathymetric charts, taking bathymetry from a larger scale model which covers the same region, and using

available measurements. Care must be taken when using data from several different sources to ensure that differing reference points for the measurement of depth values are accounted for.

The quality of certain data may be suspect depending on how they were obtained or how recent they are. This may cause problems when it comes to interpolation of bathymetric data as good quality data may easily become contaminated by the presence of poorer quality data. The combination of such data should be carefully considered before the decision is made to include or discard any material.

The best bathymetry is not always obtained by assigning specific depth measurements to the exact geographical position in the model area. Because two-dimensional numerical models calculate averaged stream velocities and water depths, it follows that averaged bathymetric features are more important than bathymetric features at discrete positions.

It is possible for a bathymetric file to be constructed in one of two ways. One method may be termed the 'gridding method', the other the 'sounding method' (George, 1993).

The gridding method requires that the average depth of a grid square be obtained directly from a grid drawn on the chart and a method of proportion assigned to account for any soundings and isobaths that may be present in the superimposed grid square (see George (1993) for specific detail on averaging methods).

The sounding method requires that depth soundings and their locations in space be compiled in a file and interpolation methods be used to establish a schematization of the sea-bed bathymetry over the model area. This has the advantage that one database can be used for many different models over a region which has had its depths digitized. However, isobaths, which provide much needed information in regions of sparse soundings, are difficult to incorporate in the interpolation process, whereas the gridding method readily allows isobaths

to be taken into consideration during the method of averaging. Due to the fact that the soundings found on an Admiralty Chart are referred to chart datum while tidal elevations in numerical models are computed relative to mean sea level, it is necessary to adjust each depth value taken from the chart in order that they should also be relative to mean sea level.

Since this numerical model of the Wash is 'study specific', *i.e* it is not deliberately intended to be part of a larger database, the gridding method has been used to create a bathymetric database. Admiralty Charts Nº's 108 and 1200 have been used to provide the bathymetric information. The bathymetric file for the 0.4NM model may be seen in Appendix 4, while figure 6.1 shows an isometric presentation produced from this data set. The bathymetric input for the 2NM model was obtained by averaging the bathymetry in 5x5 squares of cells from the 0.4NM model, which correspond to the same area as one cell from the 2NM model.

6.1.2 Masking File

For the model of the Wash the precise locations of land and sea are defined by a masking file (see George, 1993). This has a similar format to that of the bathymetric file and can be superimposed on the bathymetric file to produce a graphical image of the area being modelled. This is invaluable during the development of such files as areas of inconsistency between bathymetric and masking files can be readily observed and altered.

6.1.3 Model Area

Figure 6.2 is a graphical representation of the whole of the 2NM model of the Wash. This figure shows the masking file and bathymetry (in decimetres) superimposed, as suggested above. In addition, the coastline and the open sea boundary of the model can be seen.

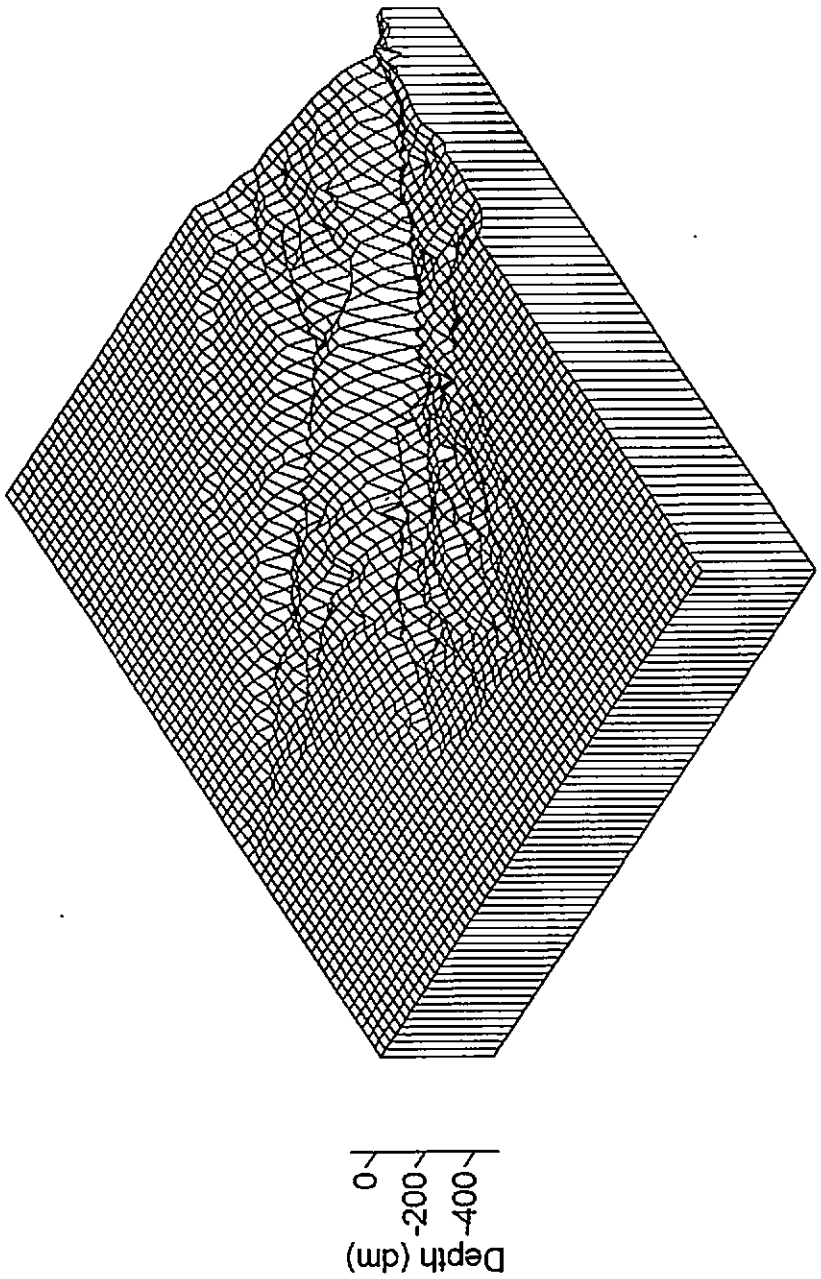


Figure 6.1: Isometric projection of the Wash

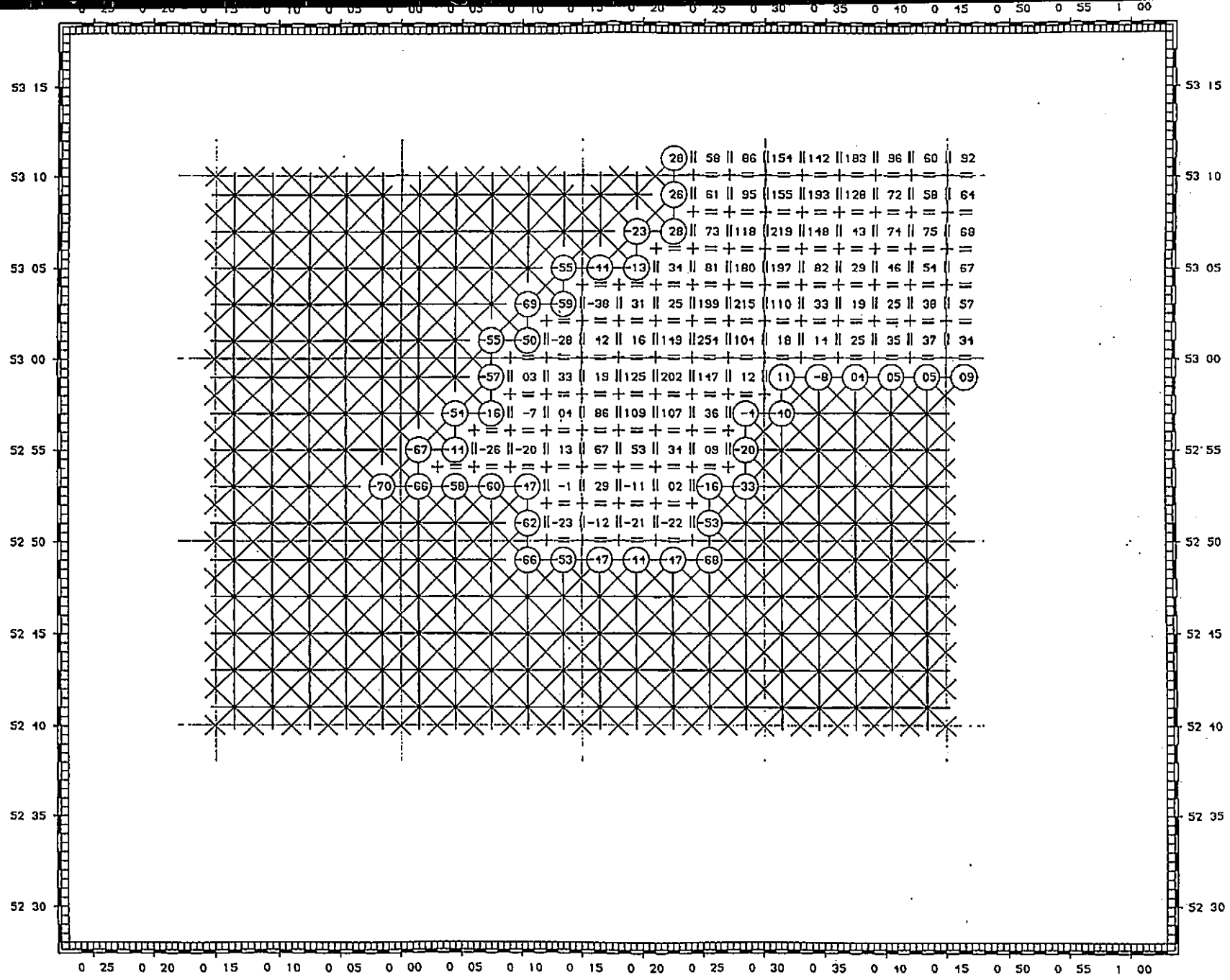


Figure 6.2: 2NM model of the Wash

6.1.4 Open Boundary Conditions

Tidal input for the idealized model has been described in Chapter 2, section 2.5.2. This is inapplicable to the model of the Wash, and therefore boundary conditions must be defined in some other manner.

There are five basic forms of boundary conditions, which may be applied in different situations, in different combinations. These are water levels, tidal velocities, water fluxes, Riemann (weakly reflective) boundaries, or gravitational effects. The exact form of the boundaries depends on the phenomena being studied. The form should, ideally, be chosen to give the best driving force to the modelled phenomena. For instance, models of water in an estuary should be driven by the specification of water levels at the entrance to the estuary. Fluxes, stream velocities, and Riemann boundaries can equally be applied in order to obtain the same internal solution. However, since velocities and fluxes are only weakly connected to water levels, then in the estuarine case used as an example above, the control over the final solution would be less than if the tidal input were in the form of water levels.

If there is more than one open boundary, different forms of boundary conditions should be applied at each boundary. For instance, at the open ends of a straight channel, the application of velocities at both boundaries would lead to continuity problems if the fluxes were not compatible. In this instance, it would probably be best to specify at one end water levels and at the other end velocities. To model a large basin (of the order of kilometres), the prescription of water level forcing at the boundaries is generally adequate.

In practice, the availability of certain types of boundary condition data determines the selection of type made. Most open sea models can be driven only by water level input at the boundaries since these may well be the only quantities known at the sea boundaries.

Table 3: Data provided by P.O.L. for production of boundary conditions

M2 AMPLITUDE (CM) - ELEVATION

0.0000	0.0000	0.0000	228.9792	219.2066	207.4745
0.0000	0.0000	252.1376	239.5120	226.2165	213.0491
0.0000	270.6418	262.7488	254.8744	0.0000	0.0000
0.0000	0.0000	269.3411	266.4209	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

M2 PHASE (DEGS) - ELEVATION

0.0000	0.0000	0.0000	168.0250	168.6870	170.1100
0.0000	0.0000	180.9720	178.1630	175.3540	174.4910
0.0000	189.2860	187.8050	187.3220	0.0000	0.0000
0.0000	0.0000	190.4240	190.8710	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

When water level boundaries are applied on their own it is the gradient of the sea-surface level which drives the system and therefore care must be taken with the data as small errors in these values can be compensated within the model only by large responses of the internal forces of the model. This could, for instance, induce higher velocities in order to compensate for the exaggerated pressure gradients on the boundary. The influence of this kind of error is not restricted to the area of data error determined by grid sizes, but by the physical area of influence according to the error in water level specification.

In order to minimize the effects of possible errors in boundary specifications, it is normal practice to specify the boundaries as far away geographically as possible from the area of interest.

The information needed to provide boundary conditions for the model of the Wash has been obtained from a larger scale model (POL Continental Shelf model: CS3), figure 6.3. These data can be seen in table 3, where the values of elevation and phase pertaining to the M_2 tidal constituent are given in the form of a grid which provides coverage of the area of the Wash model. Table 3 can be considered as a crude co-tidal chart. From the values provided by POL, a detailed co-tidal chart was constructed (figure 6.4). The crosses on figure 6.4 represent the positions of the data points provided by P.O.L.. From this chart, the phase and amplitude of the M_2 tidal constituent were extracted at each ζ-point on the 0.4NM model boundary (see figure 6.5). The boundary values pertaining to the coarser 2NM model were taken at every fifth point from the 0.4NM model boundary.

In addition to the M_2 constituent, the amplitude and phase of the S_2 and N_2 constituents were also used as input. These were extracted from co-tidal charts published by Howarth (1990) and are based on observation rather than on the results of modelling. Appendix 5 shows the boundary values.

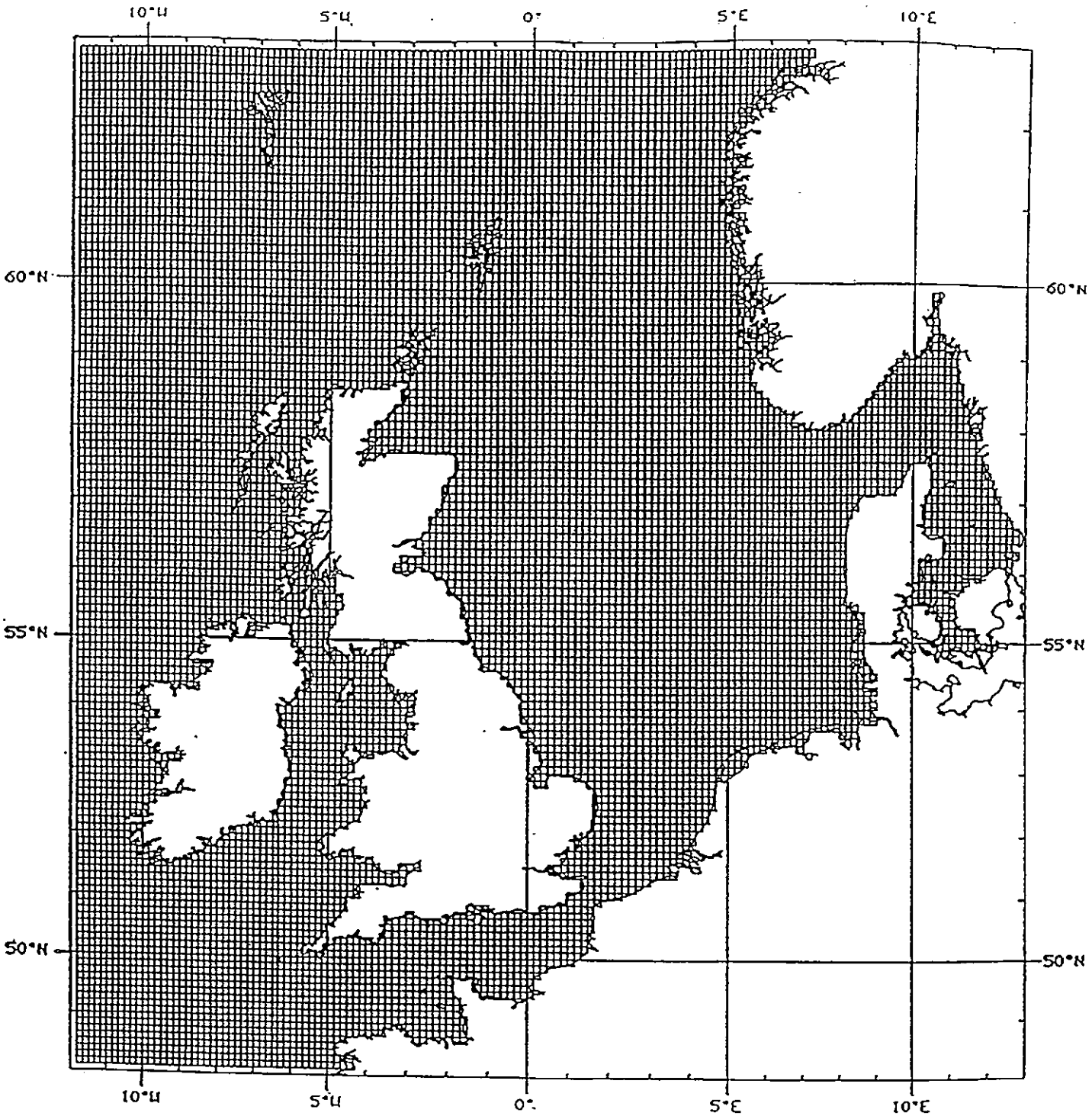


Figure 6.3: Continental Shelf model: CS3 (P.O.L.)

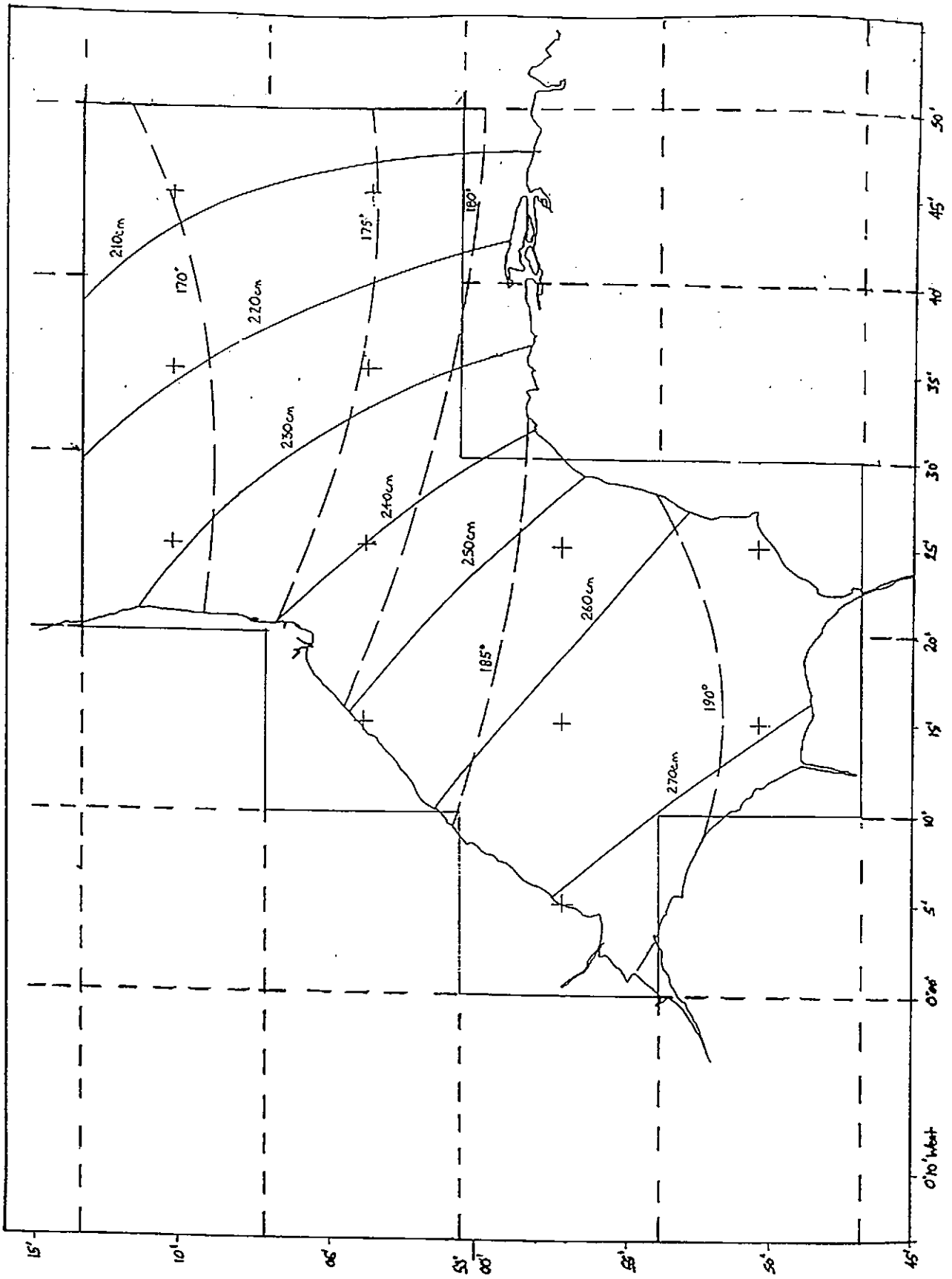


Figure 6.4: M_2 co-tidal chart of the Wash, U.K.

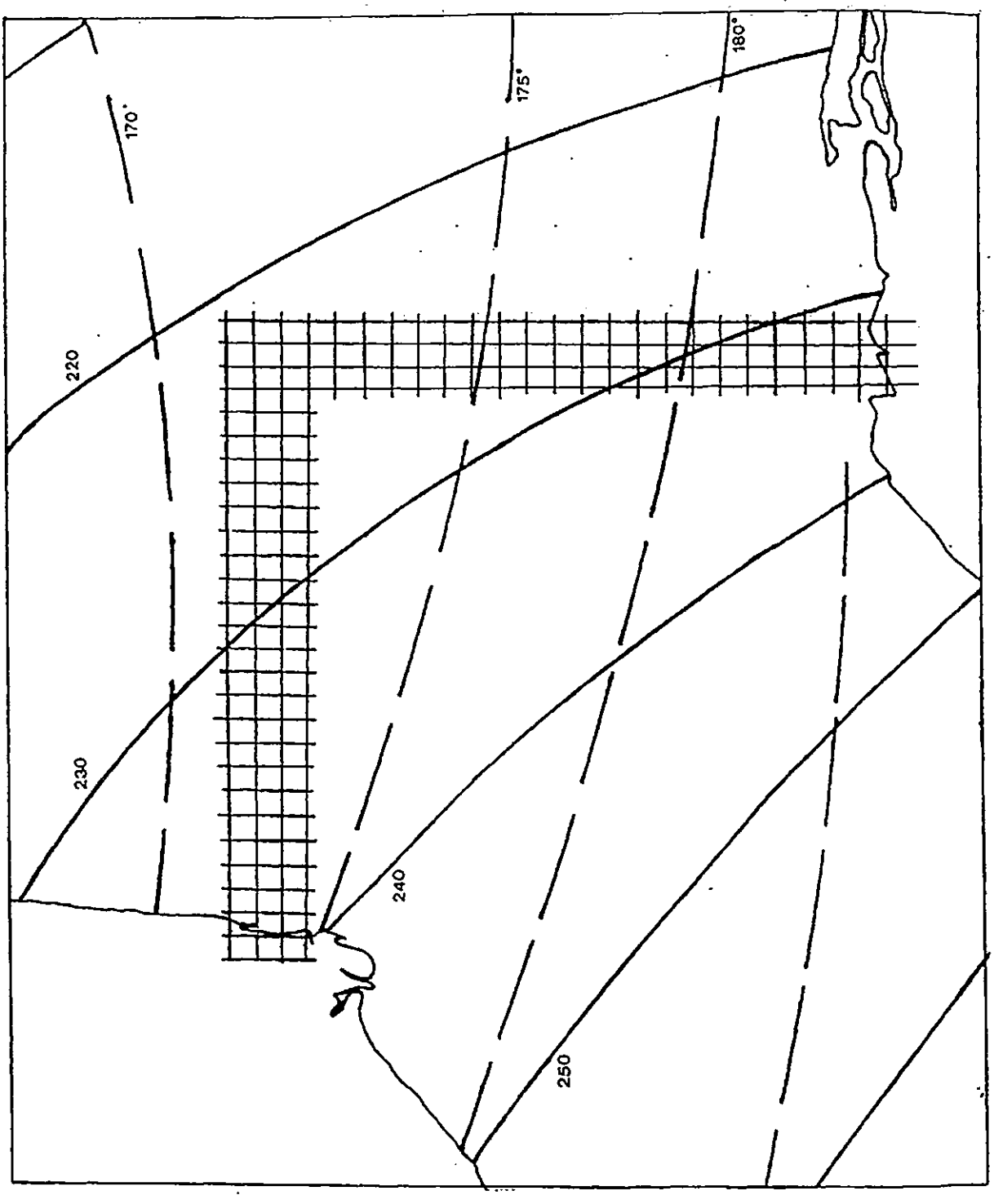


Figure 6.5: Obtaining boundary values from the co-tidal chart

6.1.5 Model Time-Step

By rearranging the inequality equation 2.29, the time-step for the 2NM model of the Wash can be evaluated from:

$$\Delta t \leq \frac{3700}{\sqrt{2 \times 9.81 \times 40}} \leq 132.1s$$

Since one three hundred and sixtieth of the tidal cycle is 124.2s, and it fulfils the stability requirement above, it has been chosen as the time-step for the model of the Wash.

6.2 Results

In this comparison exercise, results output from a cell in the 2NM model are compared to the measurements taken along the transect 52° 57' 17''N, 00° 06' 33''E to 52° 57' 34''N, 00° 05' 40''E. The time, date and the tidal conditions of the model output correspond to the field measurements. The exact position in space of the nearest model point, however, is 52° 56' 00''N, 00° 06' 00''E; over a mile away from the field survey transect (see figure 6.6).

Results from the 2NM model of the Wash can be seen in Appendix 6. Particular sets of results are also reproduced here in figures 6.7 to 6.12. These sets have been chosen due to the completeness of the corresponding field data sets.

Elevations of the survey stations were taken from figure 5.3, and converted to mean sea level. Meteorological and environmental effects (such as river discharge) on the tide cannot, at present, be represented in the model, and therefore, perturbations in the tide caused by wind set-up/set-down, wave/current interaction, or freshwater influences are deemed to be inherent errors of the model, and will be dealt with no further in the discussion of the results, although it is recognized that there is some contribution in the measurements from these phenomena.

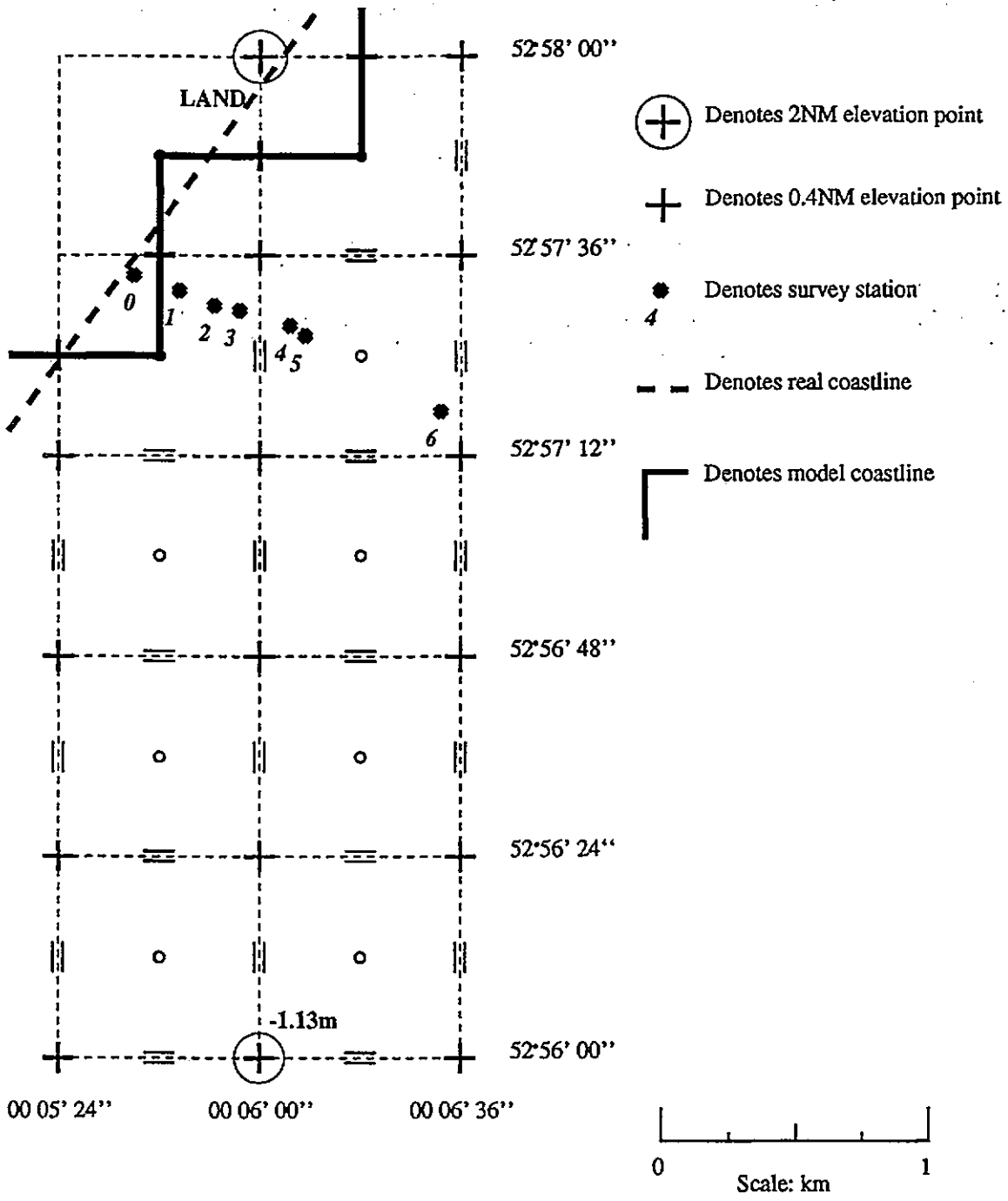


Figure 6.6: A portion of 0.4NM/2NM model area showing positions of survey stations and the coastline

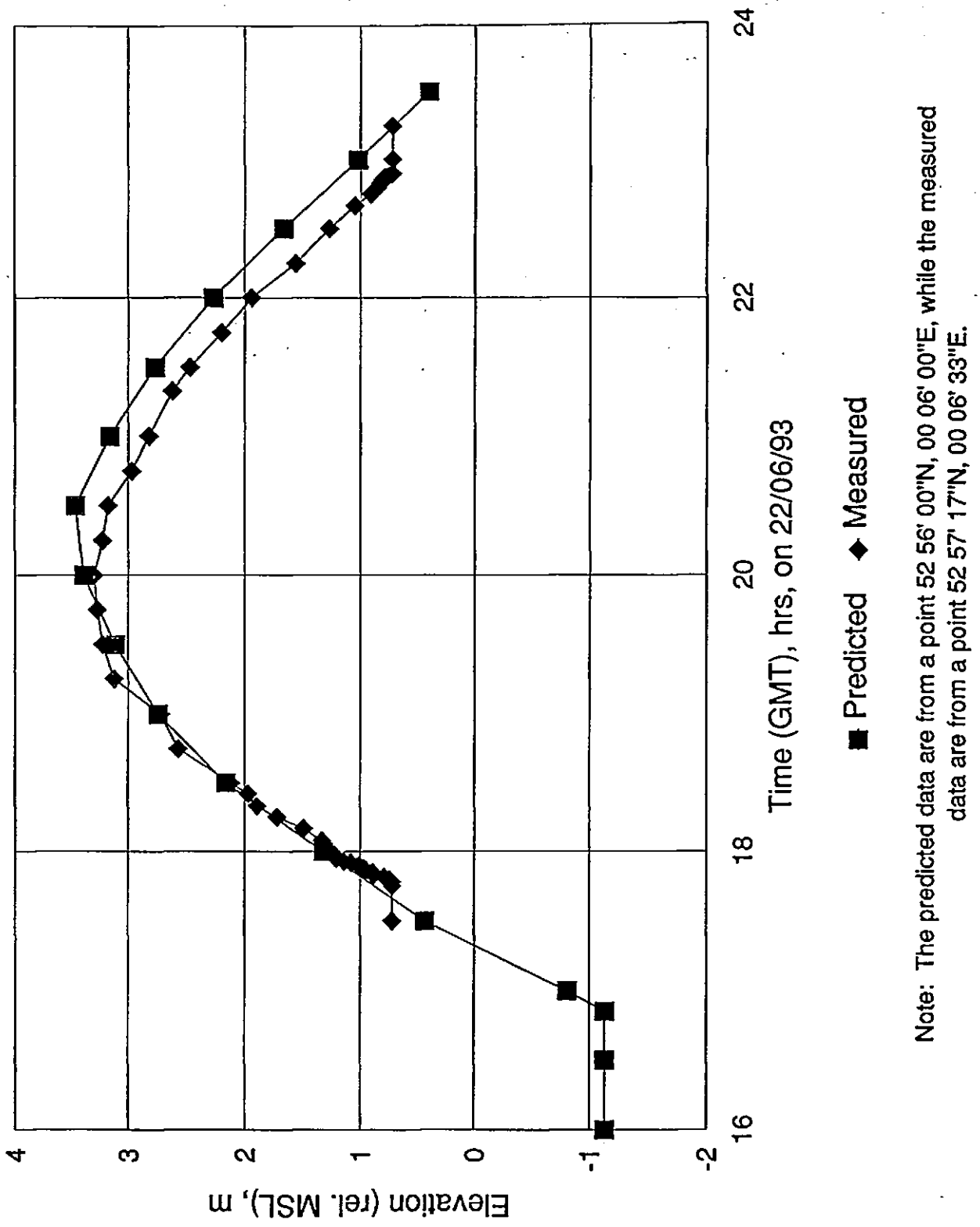


Figure 6.7: Comparison of measured and predicted elevations for Station 6

Note: The predicted data are from a point 52 56' 00"N, 00 06' 00"E, while the measured data are from a point 52 57' 17"N, 00 06' 33"E.

Figure 6.8: Comparison of measured and predicted stream rate and set for Station 6

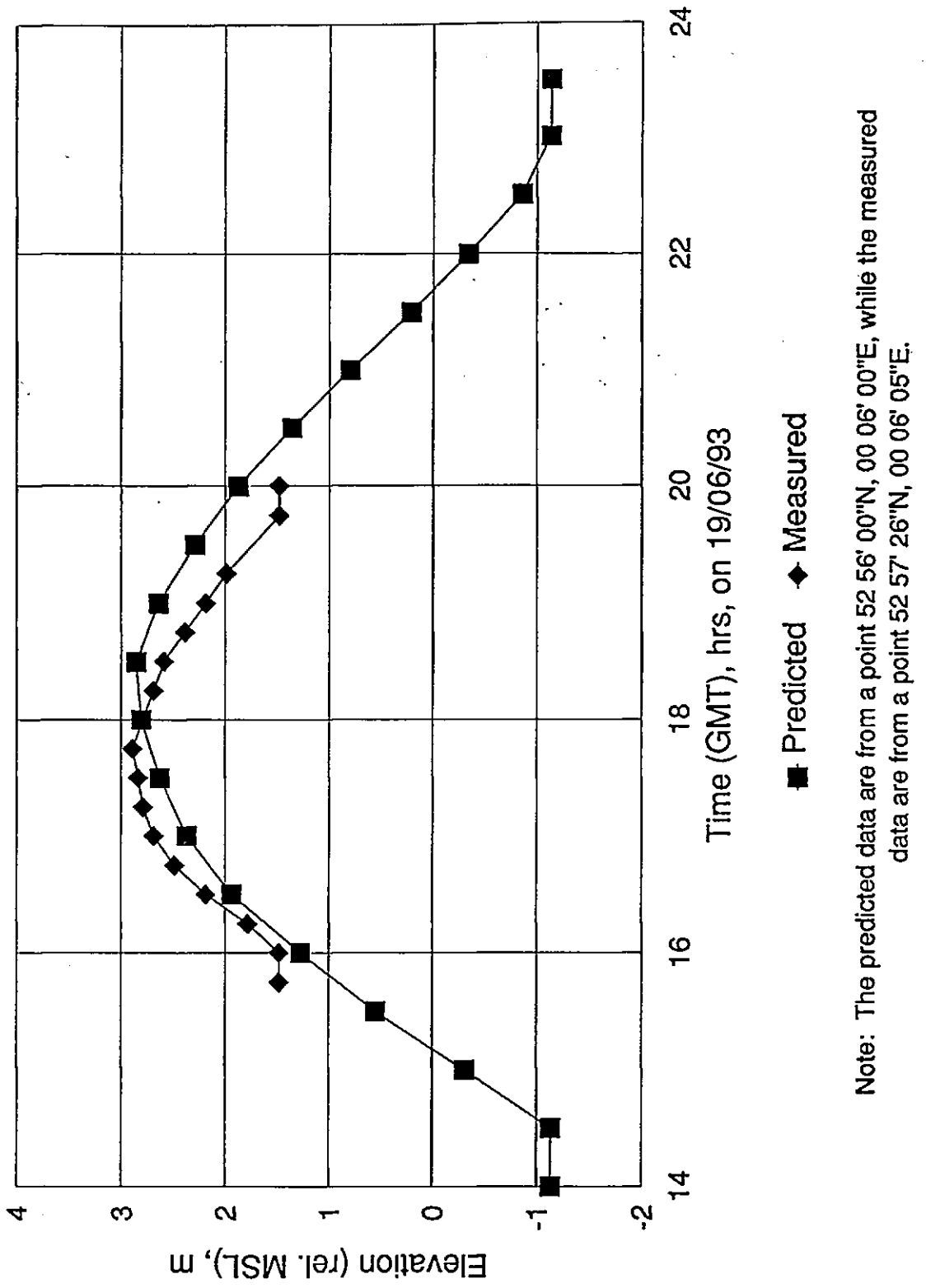


Figure 6.9: Comparison of measured and predicted elevations for Station 4

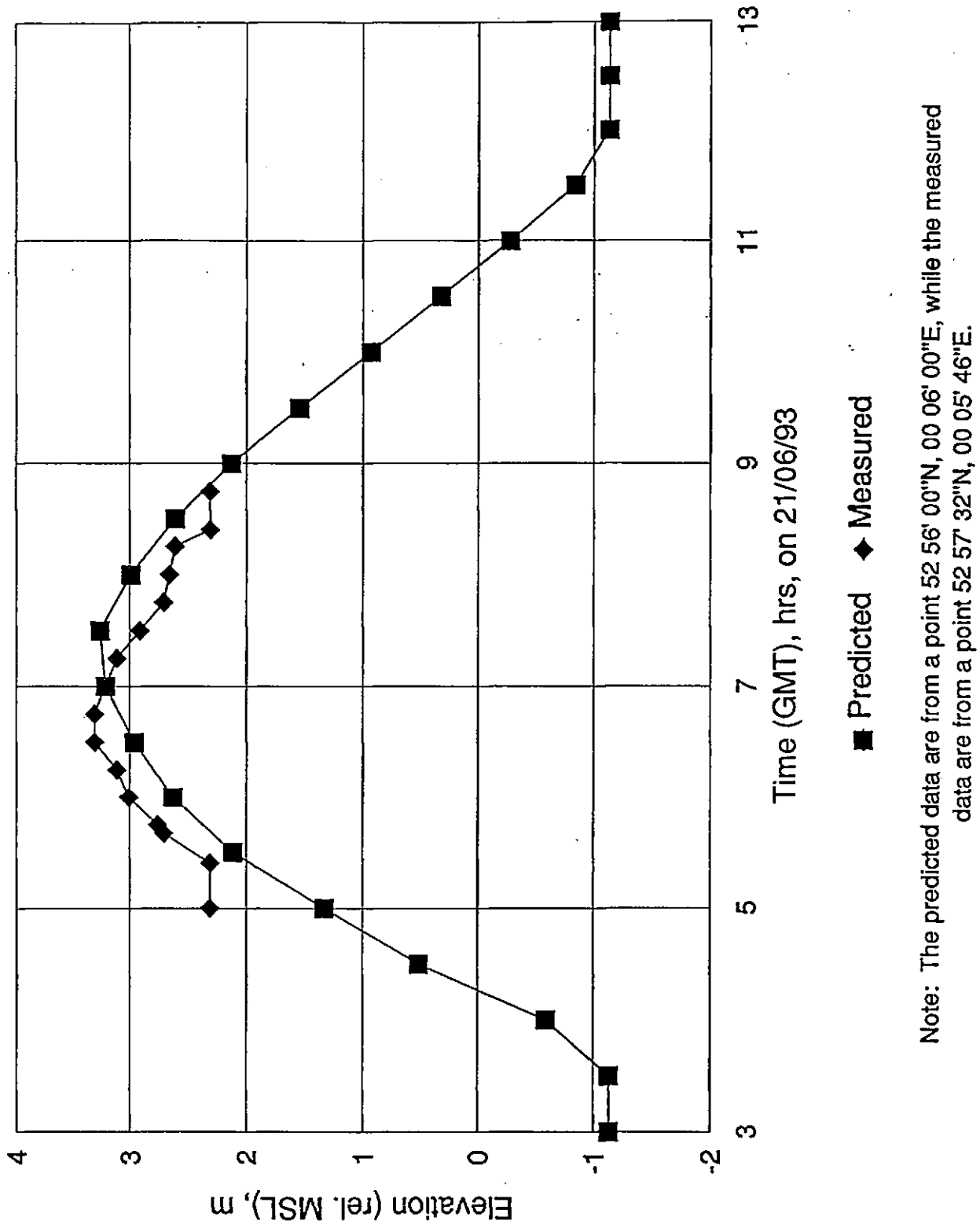


Figure 6.10: Comparison of measured and predicted elevations for Station 1

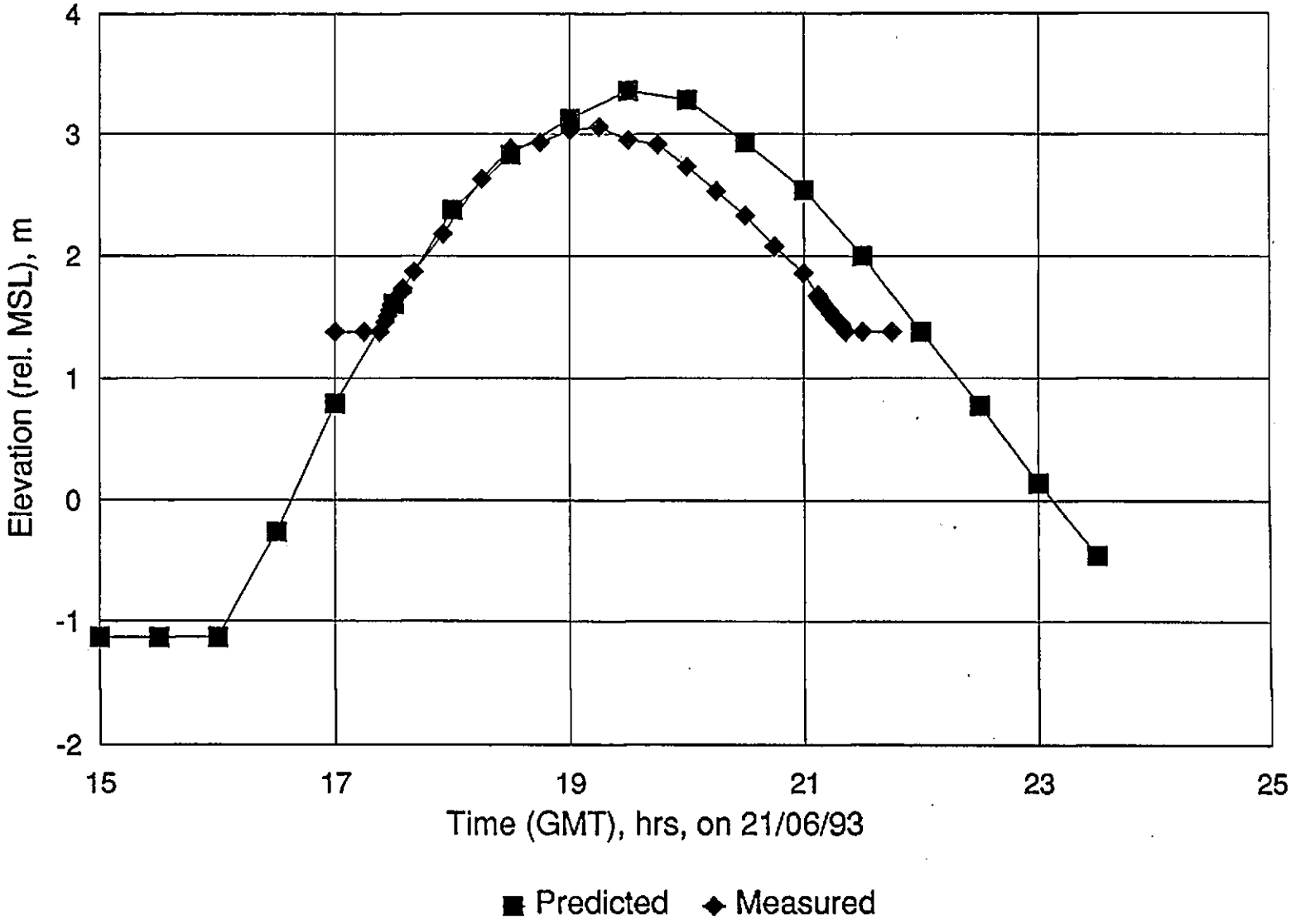


Figure 6.11: Comparison of measured and predicted elevations for Station 5

Note: The predicted data are from a point 52 56' 00"N, 00 06' 00"E, while the measured data are from a point 52 57' 25"N, 00 06' 08"E.

Figure 6.12: Comparison of measured elevation and stream rate with predicted elevation and stream rate for Station 3

Note: The predicted data are from a point 52 56' 00"N, 00 06' 00"E, while the measured data are from a point 52 57' 29"N, 00 05' 57"E.

6.2.1 Station 6

The nearest survey station to the model output point was Station 6 at 52° 57' 17''N, 00° 06' 33''E. Figure 6.7 shows the measured elevation profile and the elevation profile at the nearest 2NM model point. The model cell wets suddenly and smoothly, with no sign of spurious oscillations. The field measurements, too, show a sudden wetting. Unfortunately, measurements on this particular survey were hampered by waves and poor light. As a result, there are apparent oscillations in the measured profile.

Good agreement is shown in the general shape of the tidal curves, with the model reproducing the faster rate of rise of the tide than the rate of fall of the tide demonstrated by the field measurements. However, there is an obvious discrepancy between the two curves; the time series from the model cell implies that the survey station became wet approximately half an hour after it actually did.

The time of high water being predicted half an hour late is likely to be partly due to the inaccurate representation of bed friction, which may have allowed too much water to propagate inshore, thus delaying the time of high water. It may also be a feature of the coarseness of the model or perhaps due to the difference in bed-levels as explained below.

In the field, water-levels could be determined relative to the level of the sea-bed, but absolute water-levels were difficult to determine, because the survey stations were well over a kilometre from the nearest bench-mark. Predicted tide heights were determined for the secondary port of Tabs Head (close to Frieston and less than 2km from the survey stations) using the simplified harmonic method (N.P. 159A). Comparisons of the estimated absolute levels with these predictions suggested that the former were up to 1m in error. In the light of this discrepancy, the bed-levels were each raised by 0.8m to produce the curves shown in

figures 6.7 to 6.12.

The remaining discrepancy in the actual value of high water could be due to several other points. One is the inadequate specification of boundary input data. For the three tidal components which were used, errors could be ±0.1m in total, while an estimate of the effects of all the other harmonics which were not used (there were no quarter diurnal or species 0 input) could be ±0.3m. Another, the inadequate representation of bed friction (see section 6.2.3), may be responsible for an error of ±0.2m.

Figure 6.7 shows the tidal stream and set predictions with the tidal stream measurements and estimates of the actual stream set. It is difficult to assess the validity of comparisons between the measured and predicted values of these two variables in this instance, since stream measurements close to the shore are affected by currents from short period wave action and local topographic features. Longshore currents may also contribute to the signal, particularly close to the surf zone. This station, however, was too far seaward of the surf zone for longshore currents to cause too much of a problem, although the effect of possible resultant rip currents is not known. Perhaps the most problematic cause of variance is the topography and characteristics of the bed, which may vary considerably, even on a scale of metres. The fact that the model cells are 3700x3700m indicates that local features will not be delineated by the model.

In spite of all the potential for inaccurate predictions, figure 6.8 shows that the time series produced are very good when compared to the measurements. With regards to the stream rate, the initial peak velocity predicted by the model is ~30% higher than the initial peak velocity that was measured. Apart from errors in measurement, this is probably due to the larger volume of water being propagated inshore by the model than the measurements indicate (as shown by the higher predicted level at high tide). The model had to cope with this 'extra'

volume of water through continuity, thus moving more water in the same amount of time, resulting in a higher flux. In addition, it must be remembered that the model output point is over a mile away from the survey station.

The shape of the stream rate profile is very encouraging, however, with peak velocities predicted on wetting and on drying, by both the model and the measurements, with lower velocities predicted at around slack water. The velocities on drying are predicted to be smaller than the velocities on wetting, which agrees with the measurements, obviously due to the vast mass of water seaward 'holding back' the receding water-line. The character of the flood stream rate is predicted very well in that it shows a sudden rise to a peak velocity from dry as the point becomes wet. Following this, the model predicted a steady decrease in the velocity magnitude at a rate close to that shown by the measurements. The stream set is shown to be reasonably well reproduced; however, the measurements cannot be relied upon as they were estimates of the direction of movement, since no apparatus was deployed on site for the purpose of measuring this attribute.

6.2.2 Further Survey Stations

Figure 6.9 to 6.12 show the elevations measured at four other stations along the transect that yielded comprehensive data sets and were sea points according to the model. Again, the rates of wetting and drying are predicted well; however, the times and heights of high water are consistently poorly reproduced. The retardation of the time of high-water is more than likely due to two factors. The first is minimal, and is due to the predictions being made at such a distance from the survey transect. The main factor is probably the representation of bed friction in this, or any, numerical model where discretization in space is necessary.

6.2.3 Bed Friction

The fact that an area of sea is separated into discrete cells, which themselves may cover a substantial area of sea, leads to the necessity of making assumptions or generalisations. One generalisation made here is that the roughness of the sea bed was considered to be constant over the whole area of the numerical model, *i.e.* the drag coefficient applied was not variable. In fact, no attempt was made here to tune the model through the drag coefficient.

When tuning a tidal numerical model, adjusting the drag coefficient parameter is a good place to begin, since friction is of paramount importance in tidal propagation (Svendsen & Jonsson, 1980), especially in shallow water as shown, for instance, in figure 3.7. The Author decided here though, that simply to change the drag coefficient to another number in order to obtain a closer fit to the measurements was inappropriate.

The concept of having a drag coefficient (or bed roughness) that is variable over the entire model area is not a new concept (e.g. Wallace, 1994). Varying the bed roughness is possible, at present, only from cell to cell, or area to area. When we are dealing with large cells, such as those in the 2NM model of the Wash presented here, the bed roughness can alter considerably from one part of the cell to another, so varying the bed roughness from cell to cell solves only part of the problem. This is based on the presumption that the bed roughness over the entire area (or at least the area of interest) of the model is known. In addition, bed roughness is certainly not linear with horizontal distance or depth, but dependent upon bed-forms and material, so discrete measurements or calculations are again only of limited value and very costly. The expense is further increased when we consider that, even if such detailed measurements were to become available to modellers, they would be site specific, and consequently, applicable only to the area where the measurements were taken.

CHAPTER 7

SUMMARY AND CONCLUSIONS

SUMMARY

Three existing drying algorithms were incorporated into a numerical scheme and analyzed. Their performance was found to be unsatisfactory so a new algorithm was developed. The results have shown an improvement upon general modelling techniques in wetting and drying models.

A week of field work was conducted alongside the numerical development. This has proved that provided there are sufficient research funds, it is possible for numerics and real life observations to work in unison towards a common end.

Field measurement techniques were adapted to provide the best measurements possible of the initial wetting and the late drying periods on a tidal flat in the Wash, U.K.. A 2NM numerical tidal model of the Wash was designed and validated by these field measurements. On the whole, the model performed very well, with the biggest problems being; precisely where the measurements were output from (*i.e.* the model output point was over a mile from the survey transect), the lack of a detailed survey of the bed levels at the survey stations, and the uncertainty in the representation of bed friction in large cells.

CONCLUSIONS

1/ With regards to wetting and drying in tidal numerical models, there is one problem which appears to cause more disruption of the solutions than any other. This is the fact that cells are not able to wet quickly enough when the tidal elevation surrounding them has risen sufficiently. For cells which are large, e.g. 5km, the current methods for wetting a cell require that a 25km^2 area of sea are required to suddenly and instantaneously become wholly wet and active within the numerical scheme. There is simply not the volume of water available for this to happen, neither can water physically travel fast enough to reach the opposite computational edge of the cell.

2/ Disturbances which are generated by unnaturally imposed requirements (e.g. the sudden wetting of a cell) have a considerable effect on cell dynamics and surrounding solutions. If a cell is required to wet suddenly, there will naturally be an imbalance in the conservation of momentum equation.

3/ A new numerical technique has been devised which allows a cell to wet and dry gradually. In principle this is a very straightforward idea. However, the computational process can be quite complex. This method, the 'method of sloping facets', was used to eliminate discontinuities inherent in a 'large cell' wetting and drying process as the tide rises and falls. The testing of this method has proved that sloping facets provide solutions which are not adversely affected by the wetting and drying of cells. It was concluded that the models incorporating this new technique resembled nature in that discontinuities related to cell size inherent with discretization demonstrated in previous work were removed.

4/ It is suggested that a finer scale model of the Wash be run in order to resolve the field observation profile better, thus providing model output at a position in space closer to where

these field measurements have been taken. It may also be interesting to compare output from coarse and fine models of the same area.

5/ An area of further work which would require considerable attention is to attempt to find a more realistic manner of expressing the effect of frictional drag on the water column, and for the new knowledge to be integrated into numerical models. This suggests that it will be necessary for field workers to work more closely with numerical modellers, in order to combine their experience and understanding.

6/ There has been some light shed in this thesis on the relationship between frictional and gravitational accelerations in very shallow water, *i.e.* that they balance each other. This has interesting connotations, as the resulting hydrodynamic equations become, in this case, ordinary differential equations which can be solved through an iterative process, with no need for time-stepping in the solution technique. This suggests that, potentially, hydrodynamic approximations for cases tens of years into the future could be found very quickly. This would naturally be tempered though, perhaps when applied to a practical situation such as the need for long-term morphodynamic modelling, in which the speed of computation would be dependant upon the time-step required to maintain stability of the morphodynamic scheme. Nonetheless, it could be a valuable area for future research.

REFERENCES

Backhaus J.O. (1983) A semi-implicit scheme for the shallow water equations for the application of shelf sea modelling. *Continental Shelf Research*, 2, 243-245.

Casulli V. (1990) Semi-implicit finite difference methods for the two-dimensional shallow water equations, *Journal of Computational Physics*, 86, pp56-74.

Cheng R.T. and Casulli V. (1992) 'Tidal residual, intertidal mudflat (TRIM) model using a semi-implicit Eulerian-Lagrangian method', *U.S. Geol. Surv. Open File Rep.*, 96-62

Davies A.M. (1986) Numerical modelling of marine systems. *Numerical Modelling Application to Marine Systems*, ed. J. Noye, Published by North Holland.

Davies A.M. (1991) Modelling currents in highly sheared surface and bed boundary layers. *Continental Shelf Research*, December.

Dronkers J.J. (1964) *Tidal Computations in Rivers and Coastal Waters.* Published by North Holland, Amsterdam.

Evans E.M. (1987) 'Tidal stream energy', Unpublished Ph.D. thesis, Plymouth Polytechnic.

Evans J.J. & Pugh D.T. (1982) Analysing clipped sea-level records for harmonic tidal constituents. *International Hydrographic Review*, Monaco, LIX (2), July.

Evans G. and Collins M.B. (1975) 'The transportation and deposition of suspended sediment over the intertidal flats of the Wash', in: 'Hails J. and Carr A. (1975) *Nearshore Sediment Dynamics and Sedimentation*, John Wiley & Sons, pp. 273-306, Ch. 11.

Falconer R.A. (1980) Numerical modeling of tidal circulation in harbors. *J. of the Waterway, Port, Coastal & Ocean Division, Proc. of the Am. Soc. of Civ. Engnrs.*, Vol. 106, No. WW1

Feb..

Falconer R.A. (1984a) A mathematical model study of the flushing characteristics of a shallow tidal bay. *Proc. Instn. Civ. Engrs.*, Part 2, 77, Sept. pp 311-332.

Falconer R.A. (1984b) Temperature Distributions in Tidal Flow Field. *J. of Environmental Engineering*, Vol.110, No. 6, Dec..

Falconer R.A. (1985) Residual Currents in Port Talbot Harbour: a mathematical model study. *Proc. Instn. Civ. Engrs.*, Part 2, 79, Mar., 33-55.

Falconer R.A. (1986) Water quality simulation study of a natural harbor. *Port, Coastal, and Ocean Engineering Div. Am. Soc. Civ. Engnrs.*, Vol 112, Jan. No.1, 15-34.

Falconer R.A. & Owens P.H. (1987) Numerical simulation of flooding and drying in a depth-averaged tidal flow model. *Proc. Instn. Civ. Engrs.*, Part 2, 83, Mar., 161-180.

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

Flather R.A. & Hubbert K.P. (1989) Tide and Surge Models for Shallow Water - Morecambe Bay Revisited. *Modeling Marine Systems*, CRC Publications, ed. A. M. Davies (1991).

George K.J. (1993) 'Hydrographic data bases for tidal numerical models', *International Hydrographic Review*, LXX(2), September 1993.

George K.J. (1994) *Tides for Marine Studies*. Institute of Marine Studies, University of Plymouth

George (1995) 'A split-mesh harmonic tidal numerical model using finite differences', submitted to *Journal of Marine Systems*

George K.J. & Evans E.M. (1991) Using tidal stream data to fine-tune a 2-D tidal numerical model. *Tidal Hydrodynamics*, Chapter 12. Ed. B.B. Parker, J. Wiley & Sons Inc..

George K.J. and Stripling S. (1995) 'Improving the simulation of drying and wetting in a two-dimensional tidal numerical model', *Applied Mathematical Modelling*, Vol 19 pp. 2-6.

Gill A.E. (1982) *Atmosphere-Ocean Dynamics*, Academic Press.

Gunn D.J. and Yenigün O. (1985)'Modelling of tidal motion in shoaling waters: The estuary of Milfird Haven', *Estuarine, Coastal and Shelf Science*. 21, pp. 337-356

Heaps N.S. (1969) 'A two-dimensional numerical sea model', *Phil. Trans. R. Soc. Lond.* A265, 93-137.

Hervouet J.-M. (1991) 'TELEMAC, a fully vectorized finite element software for shallow water equations', *EDF-LNH*, Report No. E4303R.

Howarth M.J. (1990) 'Atlas of tidal elevations and currents around the British Isles' H.M.S.O., London.

HR Wallingford (1993) TELEMAC-2D - Background Information. October.

HR Wallingford (1993) 'PISCES A morphodynamic Coastal Area Model, First Annual Report'. Report No. SR 337.

Jeffery A. (1976) *Research Notes in Mathematics, 5: Quasilinear hyperbolic systems and waves*. Publ. Pitman Ltd.

Leendertse J.J. (1970) A water quality simulation model for well-mixed estuaries and coastal seas, The Rand Corporation, Santa Monica.

Leendertse J.J. & Gritton E.C. (1971) A water quality simulation model for well-mixed estuaries and coastal seas, VOL 2, Computation Procedures, The New York City Rand Institute.

Lynch D.R. and Gray W.G. (1980) 'Finite element simulation of flow in deforming regions', *Journal of Conputational Physics*, 36, pp. 135-153.

McDowell D.M. & O'Connor B.A. (1977) *Hydraulic Behaviour of Estuaries*. Macmillan, London

Owens P.H. (1984) Numerical modelling of two-dimensional flows in coastal waters. *MSc thesis*, Univ. of Birmingham. Unpublished.

Proudman J. (1953) *Dynamical Oceanography*. London.

Reid R.O. & Bodine B.R. (1968) Numerical model for storm surges in Galveston Bay. *Proc. of the Am. Soc. of Civ. Engrs. J. of the Waterways and Harbours Div.* February.

Sielecki A. & Wurtele M.G. (1970) The numerical integration of the non-linear shallow water equations with sloping boundaries. *Journal of Computational Physics*, 6, 219-236.

Smith G.D. (1978) '*Numerical Solution of Partial Differential Equations: Finite Difference Methods*', Oxford University Press, 2nd Edition.

Stelling G.S., Wiersma A.K. & Willemse J.B.T.M. (1986) Practical aspects of accurate tidal computations. *J. of Hydraulic Engineering*, Vol. 112, No. 9, September.

Stripling S., Dyer K.R. & Huntley D.A. (1994) A review of Transport models in the Marine Environment. For Westlakes Research Institute, Cumbria. Confidential

Svendsen I.A., & Jonsson I.G. (1980) Hydrodynamics of coastal regions. Published by Den Private Ingeniørfond Technical University of Denmark.

Wallace H.M. (1994) 'COSMOS-2D (Ver. 2.3): Nearshore sediment transport Model, Description of Model Structure and Input' HR Wallingford Report No. IT388 (issue B).

Xanthopoulos Th. and Koutitas Ch. (1976) 'Numerical simulation of a two-dimensional flood wave propagation due to dam failure', *J. Hydraulic Research*, 14(4), pp. 321-331.

Yeh G.-T. and Chou F.-K. (1979) 'Moving boundary numerical surge model', *J. of the Waterway, Port, Coastal and Ocean Division, Proc. A.S.C.E.*, Vol. 105, No. WW3, pp. 247-263.

APPENDIX 1

LIST OF TERMS USED IN MODK.FTN77

(upper case ≡ lower case for all terms)

A - Amplitude of tide, m
CD - Stability constraint on the water depth value
CDRAG - Coefficient of drag
CORX - Zonal Coriolis component, ms^{-2}
CORY - Meridional Coriolis component, ms^{-2}
CV - Stability constraint on the velocity magnitude
DA - Change in amplitude
DDX - Slope of water over distance ∂x
DDY - Slope of water over distance ∂y
DE - Depth of water at U-point east of test-cell
DEDT - Change of tidal elevation with time, $\partial\zeta/\partial t$
DEPTHX - Water depth at U-point
DEPTHY - Water depth at V-point
DN - Total water depth at V-point north of test cell
DP - Change in tidal phase
DS - Total water depth at V-point south of test cell
DT - Time step, ∂t
DTC - Total water depth in test cell
DUDT - Temporal acceleration (zonal component), $\partial U/\partial t$
DVDT - Temporal acceleration (meridional component), $\partial V/\partial t$
DW - Total depth of water at U-point west of test cell
DX - Zonal finite difference grid size
DY - Meridional finite difference grid size
D1 - Total water depth at I or J on a ∂-point
D2 - Total water depth at I+1 or J+1 on a ∂-point
E - Tidal elevation
EN - New value of E
EDDYX - Zonal component of eddy viscosity
EDDYY - Meridional component of eddy viscosity
ELEVE - Average depth along the eastern edge of cell
ELEVN - Average depth along the northern edge of cell
ELEVS - Average depth along the southern edge of cell
ELEVW - Average depth along the western edge of cell
EMEAN - Mean value of elevations surrounding cell
E1 - Tidal elevation at I or J
E2 - Tidal elevation at I+1 or J+1
F - Coriolis parameter
FDL - Counter for output routines
FP6 - Counter for every 6^{th} value of ψ
FP10 - Counter for every 10^{th} value of ψ
FRICX - Zonal frictional component
FRICY - Meridional frictional component
GDEDX - Slope term in the east-west direction
GDEDY - Slope term in the north-south direction
H - Depth of sea bed at h-point
HE - Depth of sea bed at ζ-point

HEAST - Depth of sea bed at U-point east of cell
HED - Horizontal eddy diffusion
HEDD - Horizontal eddy diffusion as a function of depth
HN - Depth of sea bed at V-point north of cell
HS - Depth of sea bed at V-point south of cell
HW - Depth of sea bed at U-point west of cell
H1 - Depth of sea bed at I or J on a ∂-point
H2 - Depth of sea bed at I+1 or J+1 on a ∂-point
I - North to south coordinate
I - Counter in DO loops
IPSI - Integer value of ψ
IP6 - Every 6^{th} value of ψ
IP10 - Every 10^{th} value of ψ
ITC - Number of tidal cycle
ITCA - Number of tidal cycle for beginning of analysis
ITCR - Tidal cycle in which shoal is raised
ITS - Number of time step
J - East to west coordinate
KEY - Switch to indicate instability
KHE - Counter to specify depths or to read from file
KIE - Counter for specifying elevations or to read from file
KIS - Counter for 'cold' or 'hot' start
KN - Switch to indicate dry cell
KOP - Switch for output to file
KWET - Switch to indicate wetted cell
K4 - Switch to locate a maximum of four surrounding wet cells
NDAL - Counter for choice of drying algorithm
NOC - Output channel to mainframe file space
NRUN - Run number
OPSI - Counter for screen dump every 36^{th} value of ψ
P - Tidal phase
PSI - Tidal time angle, $\psi°$
PSIN - New value of ψ
PSIR - Tidal time angle in radians
PSI8 - Double precision value of PSI
PSI36 - Every 36° of ψ
RATEX - Stream rate at U-point
RATEY - Stream rate at V-point
RR - Key for flat or Gaussian bathymetry
R2 - Value of I (coordinate) used to create shoal
U - Zonal depth averaged velocity
UATVP - Zonal velocity component at V-point
UDUDX - Advective term
UDVDX - Advective term
UN - New value of U
UV - U or V value
V - Meridional depth averaged velocity
VATUP - Meridional velocity at U-point
VDUDY - Advective term

VDVDY - Advective term
VN - New value of V
WIND - Value for 'winding up'

```
c-----------------------------------------------------------------
c-----------------------------------------------------------------
c-----------------------------------------------------------------
c-----    ******************************************    -----
c-----    *  2-D MODEL USING HYDRODYNAMIC EQUATIONS  *    -----
c-----    *                                          *    -----
c-----    *  UTILIZING 4 DIFFERENT 'DRYING' METHODS  *    -----
c-----    ******************************************    -----
c-----                                                    -----
c-----                                                    -----
c-----                       BY                           -----
c-----                                                    -----
c-----                  MR. S STRIPLING                   -----
c-----                                                    -----
c-----               UNIVERSITY OF PLYMOUTH               -----
c-----                                                    -----
c-----                                                    -----
c-----------------------------------------------------------------
c-----------------------------------------------------------------
c-----------------------------------------------------------------

      program model

      parameter(mpti=40,mptj=40)
      dimension u(mpti,mptj),v(mpti,mptj),un(mpti,mptj),vn(mpti,mptj)
     &,kwet(mpti,mptj),e(mpti,mptj),en(mpti,mptj),he(mpti,mptj)
     &,deepp(mpti,mptj),h(mpti,mptj),kn(mpti,mptj)
      dimension dudt(mpti,mptj),corx(mpti,mptj),gdedx(mpti,mptj)
     &,fricx(mpti,mptj),dvdt(mpti,mptj),cory(mpti,mptj)
     &,gdedy(mpti,mptj),fricy(mpti,mptj)
      dimension a(mpti,mptj),p(mpti,mptj),dec(mpti,mptj)
     &,hs(mpti,mptj),em(mpti,mptj),um(mpti,mptj),vm(mpti,mptj)
      common /lim/ ny,nx
      common /coord/ if,jf,dx,dy
      common /time/ itc,psi,its,dt,hour,psin
      common /grid/ ngs,nsbc,ndal
      common /numb/ pi,rad
      common /stab/ cd,cv,key
      common /cons/ fdl,hed,cdrag,f
      common /start/ kie,kis,nrun,rr

      integer hour
      character wash*1
      real etime, dtime, tim(2)

      psi8   =  0.d0

c----         To force the grid-size

      ngs    =  3

c----         To set switch for output

      kop    =  1

c----         To specify the number of the run

      write(*,*) '      '
      write(*,*) '      '
      write(*,*) ' Input the run-number'
      write(*,*) '      '
      write(*,*) '      '
      read(*,1)nrun
    1 format(i5)
c      nrun   =  115
```

```
c----        To specify counter for 'full' boundary levels at the start

      kie    =  0

c----        To gather information on the run

      write(*,*) '        '
      write(*,*) 'Do you wish to run the Wash model ? (y/n)'
      write(*,*) '        '
      read(*,'(a1)')wash
      write(*,*) '        '
      write(*,*) 'Is this to be (1) a hot run, or (2) a cold run ?'
      write(*,*) '        '
      read(*,*)kis

c----        To determine the dimensions of the grid

      ny     =  40
      nx     =  40

      if(wash.eq.'y')then

         ny     =  60
         nx     =  65

      endif

c----        To set stability limits

      cv     =  9.9
      cd     =  6.0

      rr     =  -1.0

c----        To specify constants

      rad    =  0.0174532925
      pi     =  3.141592654
      f      =  2.0*7.29e-05*sin(50.*rad)
      hed    =  10.0
      cdrag  =  0.0025
      fdl    =  0.1
      hour   =  0
      if     =  18
      jf     =  21

      if(wash.eq.'y')then

         dy     =  741.3
         dx     =  741.3
         dt     =  62.1

      else

      write(*,*) 'This model may be run with a grid size of:'
      write(*,*) '        '
      write(*,*) '(1) 5000m   or   (2) 1000m'
      write(*,*) '        '
      write(*,*) 'Which would you prefer, (1) or (2) ? '
      write(*,*) '        '
      read(*,*)ngs

         if(ngs.eq.1)then

            dy     =  5000.
            dx     =  5000.
```

```
            dt      = 124.2

         else

            dy      =  1000.
            dx      =  1000.
            dt      =  62.1

         endif

      endif

c-----      Opening files
                  open(25,file='knout')
      open(54,file='fintop')
      open(56,file='newstore')
      open(22,file='ddl')
      open(23,file='ddf')
      open(24,file='ddo')
      open(99,file='bathys')
      open(13,file='slp-depths')
      open(50,file='hhestore')
      open(11,file='wasdep')
      open(43,file='washhe')
      open(63,file='dds')
      open(64,file='ddse')
      open(30,file='magn00')
      open(70,file='dir00')
      open(31,file='magn01')
      open(71,file='dir01')
      open(32,file='magn02')
      open(72,file='dir02')
      open(33,file='magn03')
      open(73,file='dir03')
      open(34,file='magn04')
      open(74,file='dir04')
      open(35,file='magn05')
      open(75,file='dir05')
      open(36,file='magn06')
      open(76,file='dir06')
      open(37,file='magn07')
      open(77,file='dir07')
      open(38,file='magn08')
      open(78,file='dir08')
      open(39,file='magn09')
      open(79,file='dir09')
      open(40,file='magn10')
      open(80,file='dir10')
      open(41,file='magn11')
      open(81,file='dir11')

c-----      Specifying bathymetry

      if(wash.eq.'y')then

         nsbc   =  3

         if(kis.eq.2)call batwas(i,j,h,he)
         if(kis.eq.1)call washhe(h,he)

      else

         write(*,*) 'You may choose a sea-bed configuration of:'
         write(*,*) ' '
         write(*,*) ' '
         write(*,*) '(1) A basin with a sloping beach'
```

```
         write(*,*) ' '
         write(*,*) '                or'
         write(*,*) ' '
         write(*,*) '(2) A basin with a Gaussian Island.'
         write(*,*) ' '
         read(*,*)nsbc

         if(kis.eq.1)call sethhe(he,h)

         if(kis.eq.2)then

            if(nsbc.eq.1)call bathys(he,h)
            if(nsbc.eq.2)call depths(h,he,kwet)

         endif

      endif

c-----      Setting initial conditions

      if(wash.eq.'y')then

         call wasic(un,vn,en,e,u,v,a,p)

      else

         call setic(un,vn,en,e,u,v,h,he)

      endif

      if(kis.eq.1)then

         itc    =  5

      else

         itc    =  0

      endif

      its    =  0

      if(ngs.eq.3)then

         itca   =  4

      else

         itca   =  5

      endif

c-----      To select a drying algorithm

      write(*,*) 'Would you like to run the model using:'
      write(*,*) ' '
      write(*,*) '1: Flather & Heaps (1975) drying algorithm'
      write(*,*) '                       or'
      write(*,*) '2: Leendertses (1970) drying algorithm'
      write(*,*) '                       or'
      write(*,*) '3: Owens (1984) drying algorithm'
      write(*,*) '                       or'
      write(*,*) '4: George & Stripling (1994) drying system ?'
      write(*,*) ' '
      read(*,*)ndal
```

```
      print *,'elapsed:',etime,' user:',tim(1),', sys:',tim(2)

      if(ndal.eq.1)call setwet(kn)

c-----
c-----        Solution of the hydrodynamic equations:
c-----

  100 continue

      if(ndal.eq.4)then

         call dpoint(h,deepp,he)
         call calcus(dudt,corx,gdedx,fricx,un,he,e,h,v,
     &                u,dudt1,corx1,gdedx1,fricx1)
         call calcvs(dvdt,cory,gdedy,fricy,vn,he,e,h,u,
     &                v,dvdt1,cory1,gdedy1,fricy1)

      else

         call calcu(dudt,corx,gdedx,fricx,kwet,e,he,un,v,
     &               u,dudt1,corx1,gdedx1,fricx1,h,kn)
         call calcv(vn,dvdt,cory,gdedy,fricy,kwet,e,he,u,
     &               v,dvdt1,cory1,gdedy1,fricy1,h,kn)

      endif

      call expluv(un,vn)
      call smooth(un,vn)

      if(ngs.eq.1)psin    =  real(psi8) + 1.
      if(ngs.eq.2)psin    =  real(psi8) + 0.25
      if(ngs.eq.3)psin    =  real(psi8) + 0.5

      ip10    =  int(psin/10.)
      fp10    =  (psin/10.) - ip10

      if(ngs.eq.3)then

         call wasen(e,a,p)

      else

         call seten(en)

      endif

      if(ndal.eq.4)then

         call calces(he,un,vn,en,e,deepp,h,
     &                u,v,dudt,corx,gdedx,fricx,dvdt,cory,
     &                gdedy,fricy,dec)

      else

         call calce(en,e,he,h,kn,un,vn,dudt,corx,
     &               gdedx,fricx,dvdt,cory,gdedy,fricy,u,v,
     &               kwet)


      endif

c-----        To check for stability

c       call chstab(en,un,vn)
c       if(key.eq.1)go to 90
```

```
c----
c----       Output Options
c----
c----       To call subroutine quiver for output

         psi    =  real(psi8)

      if(itc.eq.6)then


         p10    =  psi - 10.
         p30    =  int((psi - 10.)/30.)

         if(p30.ge.0)then

            op     =  ((psi - 10.)/30.) - p30
            ip     =  int(psi)

            if(abs(op).lt.1.0e-06)then

               call quiver(u,un,v,vn,e,he)
               hour   =  hour + 1

            endif

         endif

      endif


c----       To output information into fintops

      ip6    =  int(psi/6.)
      fp6    =  (psi/6.) - ip6
      if(abs(fp6).lt.1.0e-06)call intop(e,u,un,v,vn,
     &                                  em,um,vm)

c----       To store e,u,v,h,he at a set time

      if((itc.eq.4).and.(psi.ge.359.9))then

         call store(he,e,u,v,hs)
         call stohhe(he,h)

      endif

c     if(itc.eq.itcr)call raise(h,he,deepp)
      if(itc.eq.itca+2)go to 90

c----       Up-dating

      if(ngs.eq.1)psi8   =  psi8 + 1.0d0
      if(ngs.eq.2)psi8   =  psi8 + 0.25d0
      if(ngs.eq.3)psi8   =  psi8 + 0.5d0

      if(psi8.gt.360.d0)then

         itc    =  itc + 1
         psi8   =  psi8 - 360.d0

      endif

      its    =  its + 1

      do 300 i = 1,ny
```

```
      do 300 j = 1,nx

         u(i,j)  =  un(i,j)
         v(i,j)  =  vn(i,j)
  300    e(i,j)  =  en(i,j)

      go to 100

   90 call intop(e,u,un,v,vn,em,um,vm)

      etime = dtime(tim)
      print *,'elapsed:',etime,' user:',tim(1),', sys:',tim(2)

      stop
      end


      subroutine areaf(ec,hc,hne,hnw,hse,hsw,af)
c-----
c-----       To establish the wetness of a cell by areal factors
c-----

      dimension hk(4)

      hk(1)    = hne
      hk(2)    = hnw
      hk(3)    = hse
      hk(4)    = hsw
      z4       = -99.

      do 2 k = 1,4

         if(hk(k).gt.z4)then

            z4     = hk(k)
            k4     = k

         endif

    2 continue

      z1       = 99.

      do 3 k = 1,4

         if(hk(k).lt.z1)then

            z1     = hk(k)
            k1     = k

         endif

    3 continue

      if(ec.ge.hne.and.ec.ge.hnw.and.ec.ge.hse.and.ec.ge.hsw)then

         af     = 1.
         return

      elseif(ec.le.hne.and.ec.le.hnw.and.ec.le.hse.and.ec.le.hsw)then

         af     = 0.
         return

      endif
```

```
      do 4 k = 1,4

         if(k.eq.k1.or.k.eq.k4)then

            go to 4

         else

            k2   = k
            go to 5

         endif

    4 continue

    5 do 6 k = 1,4

         if(k.eq.k1.or.k.eq.k2.or.k.eq.k4)then

            go to 6

         else

            k3   = k
            go to 7

         endif

    6 continue

    7 if(hk(k3).le.hk(k2))then

         kt   = k2
         k2   = k3
         k3   = kt

      else

      continue

      endif

c-----      to evaluate areal factor for case (i)

      if(ec.lt.hk(k2))then

         af  = 0.5*(ec - hk(k1))**2/((hk(k2) - hk(k1))*(hk(k3)
     &         - hk(k1)))
         return

      endif

c-----      to evaluate areal factor for case (ii)

      if(ec.lt.hk(k3))then

         y2   = (hk(k3) - ec)/(hk(k3) - hk(k1))
         x3   = (hk(k3) - ec)/(hk(k3) - hk(k2))
         y3   = x3
         call heron(0.,0.,0.,y2,x3,y3,at1)

         y2   = (ec - hk(k2))/(hk(k4) - hk(k2))
         x3   = (ec - hk(k2))/(hk(k3) - hk(k2))
         y3   = x3
         call heron(0.,0.,0.,y2,x3,y3,at2)
```

```
         af    = 0.5 - at1 + at2
         return

      else

c-----       to evaluate areal factor for case (iii)

         af1   = 0.5*(hk(k4) - ec)**2/((hk(k4) - hk(k2))*(hk(k4)
     &           - hk(k3)))
         af    = 1.0 - af1
         return

      endif

      end


      subroutine bathys(he,h)
c-----
c-----       To input a sloping bathymetry
c-----

      dimension he(ny,nx),h(ny,nx)
      common /lim/ ny,nx

      do 200 i = 1,ny
      do 200 j = 1,nx

         he(i,j) =  5.0
         he(i,j) =  he(i,j) - (i-1) + (drand() - 0.5)

         if(i.gt.25)then

            he(i,j) =  -50. + (drand() - 0.5)

         elseif(j.lt.11.or.j.gt.29)then

            he(i,j) =  5.0

         endif

  200 continue

      do 400 i = 1,ny-1
      do 400 j = 1,nx-1

         h(i,j)  =  5.0
         h(i,j)  =  h(i,j) - (i-0.5) + (drand() - 0.5)

         if(i.ge.25)then

            h(i,j)  =  -50. + (drand() - 0.5)

         elseif(j.lt.10.or.j.gt.29)then

            h(i,j)  =  5.0

         endif

  400 continue

      do 600 i = 1,ny

         write(99,5)(he(i,j),j = 1,nx)
  600    write(99,5)(h(i,j),j = 1,nx)
```

```
    5    format(10f8.2)

      return
      end


      subroutine batwas(i,j,h,he)
c-----
c-----       To read bathymetry for the Wash from wasdep
c-----       and to evaluate he(i,j).
c-----

      dimension h(ny,nx), he(ny,nx)
      common /lim/ ny,nx

      do 200 i = 1,ny

         read(11,75)(h(i,j),j = 1,nx)

         do 300 j = 1,nx

            h(i,j)  =  h(i,j)*0.1 - 3.4

  300    continue

  200 continue

      do 1 j = 1,nx

         he(1,j)  =  h(1,j)
         he(ny,j) =  200.

    1 continue

      do 2 i = 1,ny

         he(i,1)  =  h(i,1)
         he(i,nx) =  h(i,nx)

    2 continue

      do 3 i = 2,ny
         do 4 j = 2,nx

            he(i,j)  =  0.25*(h(i,j) + h(i-1,j) + h(i-1,j-1)
     &                        + h(i,j-1))

    4    continue
    3 continue

      open(43,file='washhe')

      do 5 i = 1,ny

         write(43,100)(h(i,j),j = 1,nx)

    5 continue

      do 6 i = 1,ny

         write(43,100)(he(i,j),j = 1,nx)

    6 continue

   75 format(65f5.0)
  100 format(65f5.1)
```

```
      return
      end


      subroutine calce(en,e,he,h,kn,un,vn,dudt,corx,
     &                 gdedx,fricx,dvdt,cory,gdedy,fricy,u,v,
     &                 kwet)
c-----
c-----      To calculate the elevation at a grid-point using
c-----      the conservation of mass equation.
c-----

      dimension u(ny,nx), v(ny,nx), un(ny,nx), vn(ny,nx)
      dimension e(ny,nx), en(ny,nx), he(ny,nx), h(ny,nx)
      dimension dudt(ny,nx), corx(ny,nx), gdedx(ny,nx),
     &          fricx(ny,nx), dvdt(ny,nx), cory(ny,nx),
     &          gdedy(ny,nx), fricy(ny,nx), kwet(ny,nx),kn(ny,nx)
      common /lim/ ny,nx
      common /time/ itc,psi,its,dt,hour,psin
      common /coord/ if,jf,dx,dy
      common /grid/ ngs,nsbc,ndal
      common /start/ kie,kis,nrun,rr

      do 20 i = 2,ny-1
      do 20 j = 2,nx-1

         en(i,j) =  0.

c-----      Depths at e-points for plotting purposes:

         dtc     =  e(i,j) - he(i,j)
         dee     =  e(i,j+1) - he(i,j+1)
         dew     =  e(i,j-1) - he(i,j-1)
         den     =  e(i-1,j) - he(i-1,j)
         des     =  e(i+1,j) - he(i+1,j)

c-----      Depths at gates for calculation purposes:

         de      =  0.5*(e(i,j) + e(i,j+1) - h(i-1,j) - h(i,j))
         dw      =  0.5*(e(i,j) + e(i,j-1) - h(i,j-1) - h(i-1,j-1))
         dn      =  0.5*(e(i,j) + e(i-1,j) - h(i-1,j-1) - h(i-1,j))
         ds      =  0.5*(e(i,j) + e(i+1,j) - h(i,j-1) - h(i,j))

c-----      To calculate e using Flather & Heaps' method:

         if(ndal.eq.1)then

            if(kn(i,j).eq.0.and.he(i,j).gt.-6.)then

               en(i,j) = he(i,j)
               go to 10

            endif

c-----      To calculate e using Leendertse's method:

         elseif(ndal.eq.2)then

            if(kwet(i,j).eq.0)then

               call wetest(i,j,kwet,e,he,dn,ds,de,dw)
               if(kwet(i,j).eq.0)then

                  en(i,j)  =  he(i,j)
                  go to 10
```

```
            endif

         elseif(de.le.0..or.dw.le.0..or.dn.le.0..or.ds.le.0.)
     &          then

            kwet(i,j)  =  0
            en(i,j)    =  he(i,j)
            go to 10

         endif

c-----      To calculate e using Owen's method:
       elseif(ndal.eq.3)then

            hmax  =  h(i,j)
            if(h(i,j-1).lt.hmax)hmax   = h(i,j-1)
            if(h(i-1,j-1).lt.hmax)hmax = h(i-1,j-1)
            if(h(i-1,j).lt.hmax)hmax   = h(i-1,j)
            dc    =  e(i,j) - hmax

c-----      N.B.  The value dc may seem a bit odd, but it is
c-----      merely a representation of the volume of a cell,
c-----      not it's depth.

            if(kwet(i,j).eq.0)then

               call wetowe(i,j,kwet,e,he,dn,ds,de,dw)
               if(kwet(i,j).eq.0)then

                  en(i,j)  =  he(i,j)
                  go to 10

               endif

            elseif(dc.le.0.)then

               kwet(i,j)  =  0
               en(i,j)    =  he(i,j)
               go to 10

            endif

         endif

         ddx      =  (de*un(i,j) - dw*un(i,j-1))/dx
         ddy      =  (dn*vn(i-1,j) - ds*vn(i,j))/dy
         dedt     =  -ddx-ddy
         en(i,j)  =  dedt*dt + e(i,j)

         if(en(i,j).le.he(i,j))then

            if(ndal.eq.1)then

               en(i,j) = he(i,j)

            elseif(ndal.eq.2.or.ndal.eq.3)then

               kwet(i,j)  =  0
               en(i,j)    =  he(i,j)

            endif

         elseif(ndal.eq.2)then

            call leen3(i,j,kwet,h,en,he)
```

```
        elseif(ndal.eq.3)then

           call owen3(i,j,h,en,he,kwet)

        endif

   10   if(i.eq.if.and.j.eq.jf)then

           if(ndal.eq.1)then

              call focus(23,dee,dew,den,des,dtc,e(i,j+1),e(i,j-1),
     &                   e(i-1,j),e(i+1,j),e(i,j),un(i,j),dudt(i,j),
     &                   corx(i,j),gdedx(i,j),fricx(i,j),vn(i,j),
     &                   dvdt(i,j),cory(i,j),gdedy(i,j),fricy(i,j),
     &                   he(if,jf),he(if,jf+1),he(if,jf-1),
     &                   he(if-1,jf),he(if+1,jf))

           elseif(ndal.eq.2)then

              call focus(22,dee,dew,den,des,dtc,e(i,j+1),e(i,j-1),
     &                   e(i-1,j),e(i+1,j),e(i,j),un(i,j),dudt(i,j),
     &                   corx(i,j),gdedx(i,j),fricx(i,j),vn(i,j),
     &                   dvdt(i,j),cory(i,j),gdedy(i,j),fricy(i,j),
     &                   he(if,jf),he(if,jf+1),he(if,jf-1),
     &                   he(if-1,jf),he(if+1,jf))

           elseif(ndal.eq.3)then

              call focus(24,dee,dew,den,des,dtc,e(i,j+1),e(i,j-1),
     &                   e(i-1,j),e(i+1,j),e(i,j),un(i,j),dudt(i,j),
     &                   corx(i,j),gdedx(i,j),fricx(i,j),vn(i,j),
     &                   dvdt(i,j),cory(i,j),gdedy(i,j),fricy(i,j),
     &                   he(if,jf),he(if,jf+1),he(if,jf-1),
     &                   he(if-1,jf),he(if+1,jf))

           endif

        endif

   20 continue
  200 format(f6.2)
      return
      end


      subroutine calces(he,un,vn,en,e,deepp,h,
     &                  u,v,dudt,corx,gdedx,fricx,dvdt,cory,
     &                  gdedy,fricy,dec)
c-----
c-----      To calculate the elevation of a grid-point using the
c-----      momentum equation, and Stripling & George's drying
c-----      algorithm
c-----

      dimension u(ny,nx), v(ny,nx), un(ny,nx),vn(ny,nx), dec(ny,nx),
     &          en(ny,nx), he(ny,nx), deepp(ny,nx), e(ny,nx),
     &          h(ny,nx), dudt(ny,nx), corx(ny,nx),
     &          gdedx(ny,nx), fricx(ny,nx), dvdt(ny,nx),
     &          cory(ny,nx), gdedy(ny,nx), fricy(ny,nx)
      common /lim/ ny,nx
      common /time/ itc,psi,its,dt,hour,psin
      common /coord/if,jf,dx,dy
      common /grid/ ngs,nsbc,ndal
      common /start/ kie,kis,nrun,rr
```

```
      do 20 i = 2,ny-1
      do 20 j = 2,nx-1

         kslp  =  1
         if(he(i,j).lt.-6.)kslp = 0
         if((he(i,j).gt.5..and.ngs.lt.3).or.(he(i,j).ge.20..and.
     &       ngs.eq.3))then

            un(i,j)  =  0.
            vn(i,j)  =  0.
            en(i,j)  =  0.
            go to 20

         endif

c-----      To calculate surrounding elevations for plotting purposes

         dtc       =  e(i,j) - deepp(i,j)
         dec(i,j)  =  e(i,j) - he(i,j)
         if(dec(i,j).le.0.)dec(i,j) = 0.
         dee       =  e(i,j+1) - he(i,j+1)
         if(dee.le.0.)dee = 0.
         dew       =  e(i,j-1) - he(i,j-1)
         if(dew.le.0.)dew = 0.
         den       =  e(i-1,j) - he(i-1,j)
         if(den.le.0.)den = 0.
         des       =  e(i+1,j) - he(i+1,j)
         if(des.le.0.)des = 0.

c-----      To evaluate water depths at the edges of the cell:

         if(kslp.eq.0)then

c-----      without sloping facets;

            de  =  0.5*(e(i,j) + e(i,j+1) - h(i-1,j) - h(i,j))
            dw  =  0.5*(e(i,j) + e(i,j-1) - h(i,j-1) - h(i-1,j-1))
            dn  =  0.5*(e(i,j) + e(i-1,j) - h(i-1,j-1) - h(i-1,j))
            ds  =  0.5*(e(i,j) + e(i+1,j) - h(i,j-1) - h(i,j))

         else

c-----      with sloping facets;

            eeg  =  0.5*(e(i,j) + e(i,j+1))
            ewg  =  0.5*(e(i,j) + e(i,j-1))
            eng  =  0.5*(e(i,j) + e(i-1,j))
            esg  =  0.5*(e(i,j) + e(i+1,j))

            call dedgec(eeg,h(i-1,j),h(i,j),de)
            call dedgec(ewg,h(i,j-1),h(i-1,j-1),dw)
            call dedgec(eng,h(i-1,j-1),h(i-1,j),dn)
            call dedgec(esg,h(i,j-1),h(i,j),ds)

         endif

         ddx       =  (de*un(i,j) - dw*un(i,j-1))/dx
         ddy       =  (dn*vn(i-1,j) - ds*vn(i,j))/dy
         af        =  1.

         if(kslp.eq.1)then

            call areaf(e(i,j),he(i,j),h(i-1,j),h(i-1,j-1),
     &                 h(i,j),h(i,j-1),af)

            if(af.lt.0.01)af = 0.01
```

```
      endif

      dedt     =  (-ddx-ddy)*af
      en(i,j)  =  dedt*dt + e(i,j)

      if(en(i,j).le.deepp(i,j).and.kslp.eq.1)en(i,j) = deepp(i,j)

      if(i.eq.if.and.j.eq.jf)then

            call focus(63,dee,dew,den,des,dec,e(i,j+1),e(i,j-1),
     &                 e(i-1,j),e(i+1,j),e(i,j),un(i,j),dudt(i,j),
     &                 corx(i,j),gdedx(i,j),fricx(i,j),vn(i,j),
     &                 dvdt(i,j),cory(i,j),gdedy(i,j),fricy(i,j),
     &                 he(if,jf),he(if,jf+1),he(if,jf-1),
     &                 he(if-1,jf),he(if+1,jf))

      endif

   20 continue

      return
      end


      subroutine calcu(dudt,corx,gdedx,fricx,kwet,e,he,un,v,
     &                 u,dudt1,corx1,gdedx1,fricx1,h,kn)
c----
c----       To calculate the u-component of the depth mean velocity
c----       using the easterly direction equation for the continuity
c----       of momentum.
c----

      dimension u(ny,nx), v(ny,nx), un(ny,nx),
     &          kwet(ny,nx), e(ny,nx), he(ny,nx),
     &          h(ny,nx), dudt(ny,nx), corx(ny,nx),
     &          gdedx(ny,nx), fricx(ny,nx), kn(ny,nx)
      common /lim/ ny,nx
      common /time/ itc,psi,its,dt,hour,psin
      common /coord/ if,jf,dx,dy
      common /grid/ ngs,nsbc,ndal
      common /cons/ fdl,hed,cdrag,f

      do 50 i = 1,ny-1
      do 50 j = 1,nx-1

         un(i,j)     =  0.
         eddyx       =  0.
         ududx       =  0.
         vdudy       =  0.
         dudt(i,j)   =  0.
         corx(i,j)   =  0.
         gdedx(i,j)  =  0.
         fricx(i,j)  =  0.

         if(kwet(i,j).eq.0.and.ndal.ne.1)go to 30
         if(i.eq.1)go to 50

         depthx  =  0.5*(-h(i,j+1) - h(i,j) + e(i,j) + e(i,j+1))
         if(depthx.le.0.)then

            if(ndal.eq.1)kn(i,j) = 0
            go to 30

         endif
```

```
          if(ndal.eq.1)then

             call wdtest(i,j,e(i,j),he(i,j),e(i,j+1),he(i,j+1),kn(i,j)
     &                   ,un(i,j))
             if(kn(i,j).eq.0.and.he(i,j).gt.-6.)go to 30

          endif

c          if(e(i,j).le.he(i,j))go to 30
          if(depthx.lt.fdl)depthx  =  fdl

c----        Coriolis

          vatup      =  (v(i-1,j) + v(i-1,j+1) + v(i,j) + v(i,j+1))*0.25
          corx(i,j)  =  f*vatup

c----        Pressure Gradient

          gdedx(i,j) =  9.81*(e(i,j+1) - e(i,j))/dx

c----        Eddy viscosity

          if(i.gt.(ny-1))go to 10
          if(j.eq.1.or.j.gt.(nx-2))go to 10

          hedd       =  hed*depthx
          call ceddyx(i,j,u,hedd,eddyx)
          call cududx(i,j,u,ududx)
          call cvdudy(i,j,u,vdudy,vatup)

c----        Friction

   10     ratex      =  sqrt(u(i,j)*u(i,j) + vatup*vatup)
          fricx(i,j) =  cdrag*u(i,j)*ratex/depthx

c----        Temporal

          dudt(i,j)  =  - gdedx(i,j) - fricx(i,j) + eddyx - ududx - vdudy
     &                  + corx(i,j)

c----        New value of u

          un(i,j)    =  u(i,j) + dt*dudt(i,j)

   30     dudt1      =  dudt(if,jf)
          corx1      =  corx(if,jf)
          gdedx1     =  gdedx(if,jf)
          fricx1     =  fricx(if,jf)

   50 continue

      return
      end


      subroutine calcus(dudt,corx,gdedx,fricx,un,he,e,h,v,
     &                  u,dudt1,corx1,gdedx1,fricx1)
c----
c----        As for subroutine calcu but in case of sloping facets.
c----

      dimension u(ny,nx), v(ny,nx), un(ny,nx), e(ny,nx),
     &          he(ny,nx), h(ny,nx), dudt(ny,nx),
     &          corx(ny,nx),gdedx(ny,nx), fricx(ny,nx)
      common /lim/ ny,nx
      common /time/ itc,psi,its,dt,hour,psin
```

```
      common /coord/ if,jf,dx,dy
      common /grid/ ngs,nsbc,ndal
      common /cons/ fdl,hed,cdrag,f

      do 50 i = 1,ny-1
      do 50 j = 1,nx-1

         eddyx      =  0.
         ududx      =  0.
         vdudy      =  0.
         dudt(i,j)  =  0.
         corx(i,j)  =  0.
         gdedx(i,j) =  0.
         fricx(i,j) =  0.
         un(i,j)    =  0.

         if(he(i,j).ge.5..or.he(i,j+1).ge.5..and.nsbc.ne.3)go to 50
         if(he(i,j).ge.20..or.he(i,j+1).ge.20.)go to 50
         if(i.eq.1)go to 50

         kslp       =  1
         if(he(i,j).lt.-6..and.he(i,j+1).lt.-6.)kslp = 0

         if(kslp.eq.0)then

            dg   =  0.5*(e(i,j) + e(i,j+1) - h(i-1,j) - h(i,j))

         else

            eg   =  0.5*(e(i,j) + e(i,j+1))
            call dedgec(eg,h(i-1,j),h(i,j),dg)
            if(dg.lt.fdl)dg = fdl

         endif

         if(dg.le.0.)go to 30

c-----      Coriolis

         vatup      =  (v(i-1,j) + v(i-1,j+1) + v(i,j) + v(i,j+1))*0.25
         corx(i,j)  =  f*vatup

c----       Pressure Gradient

         gdedx(i,j) =  9.81*(e(i,j+1) - e(i,j))/dx

c----       Eddy viscosity

         if(i.gt.(ny-1))go to 10
         if(j.eq.1.or.j.gt.(nx-2))go to 10

         hedd       =  hed*dg
         call ceddyx(i,j,u,hedd,eddyx)
         call cududx(i,j,u,ududx)
         call cvdudy(i,j,u,vdudy,vatup)

c----       Friction

   10    ratex      =  sqrt(u(i,j)*u(i,j) + vatup*vatup)
         fricx(i,j) =  cdrag*u(i,j)*ratex/dg

c----       Temporal

         dudt(i,j)  =  - gdedx(i,j) - fricx(i,j) + eddyx - ududx - vdudy
     &                 + corx(i,j)
```

```
c-----       New value of u

         un(i,j)     =  u(i,j) + dt*dudt(i,j)

   30    dudt1       =  dudt(if,jf)
         corx1       =  corx(if,jf)
         gdedx1      =  gdedx(i,j)
         fricx1      =  fricx(i,j)

   50 continue

      return
      end


      subroutine calcv(vn,dvdt,cory,gdedy,fricy,kwet,e,he,u,
     &                 v,dvdt1,cory1,gdedy1,fricy1,h,kn)
c-----
c-----       To calculate the v-component of the depth mean velocity
c-----       using the continuity of momentum equation (depth averaged)
c-----       in the northerly direction.
c-----

      dimension u(ny,nx), v(ny,nx), vn(ny,nx),
     &          kwet(ny,nx), e(ny,nx), he(ny,nx),
     &          h(ny,nx), dvdt(ny,nx), cory(ny,nx),
     &          gdedy(ny,nx), fricy(ny,nx), kn(ny,nx)
      common /lim/ ny,nx
      common /time/ itc,psi,its,dt,hour,psin
      common /coord/ if,jf,dx,dy
      common /grid/ ngs,nsbc,ndal
      common /cons/ fdl,hed,cdrag,f

      do 50 i = 1,ny-1
      do 50 j = 1,nx-1

         vn(i,j)     =  0.
         eddyy       =  0.
         vdvdy       =  0.
         udvdx       =  0.
         dvdt(i,j)   =  0.
         cory(i,j)   =  0.
         gdedy(i,j)  =  0.
         fricy(i,j)  =  0.

         if(kwet(i,j).eq.0.and.ndal.gt.1)go to 30
         if(j.eq.1)go to 50

c         kn(i,j)          =  0.

         depthy  =  0.5*(e(i,j) + e(i+1,j) - h(i+1,j) - h(i,j))
         if(depthy.le.0.)then

            if(ndal.eq.1)kn(i,j) = 0.
            go to 30

         endif

         if(ndal.eq.1)then

            call wdtest(i,j,e(i,j),he(i,j),e(i+1,j),he(i+1,j),kn(i,j)
     &                  ,vn(i,j))
            if(kn(i,j).eq.0.and.he(i,j).gt.-6.)go to 30

         endif
```

```
c          if(e(i,j).le.he(i,j))go to 30
           if(depthy.lt.fdl)depthy  =  fdl

c----         Coriolis

           uatvp       =  (u(i,j-1) + u(i+1,j-1) + u(i+1,j) + u(i,j))*0.25
           cory(i,j)   =  f*uatvp

c----         Pressure Gradient

           gdedy(i,j) =  9.81*(e(i,j) - e(i+1,j))/dy

c----         Eddy Viscosity

           if(i.lt.2.or.i.gt.(ny-2).or.j.gt.(nx-1))go to 10
           hedd        =  hed*depthy
           call ceddyy(i,j,v,hedd,eddyy)
           call cvdvdy(i,j,v,vdvdy)
           call cudvdx(i,j,v,udvdx,uatvp)

c----         Friction

   10      ratey       =  sqrt(uatvp*uatvp + v(i,j)*v(i,j))
           fricy(i,j) =  cdrag*v(i,j)*ratey/depthy

c----         Temporal

           dvdt(i,j)   =  - gdedy(i,j) - fricy(i,j) + eddyy - udvdx
     &                    - vdvdy - cory(i,j)

c----         New value of v

           vn(i,j)     =  v(i,j) + dt*dvdt(i,j)

   30      dvdt1       =  dvdt(if,jf)
           cory1       =  cory(if,jf)
           gdedy1      =  gdedy(if,jf)
           fricy1      =  fricy(if,jf)

   50 continue

      return
      end


      subroutine calcvs(dvdt,cory,gdedy,fricy,vn,he,e,h,u,
     &                  v,dvdt1,cory1,gdedy1,fricy1)
c----
c----         As for calcv but in case of sloping facets
c----

      dimension u(ny,nx), v(ny,nx), vn(ny,nx), e(ny,nx),
     &          he(ny,nx), h(ny,nx), dvdt(ny,nx),
     &          cory(ny,nx), gdedy(ny,nx), fricy(ny,nx)
      common /lim/ ny,nx
      common /time/ itc,psi,its,dt,hour,psin
      common /coord/ if,jf,dx,dy
      common /grid/ ngs,nsbc,ndal
      common /cons/ fdl,hed,cdrag,f

      do 50 i = 1,ny-1
      do 50 j = 1,nx-1

         eddyx       =  0.
         udvdx       =  0.
         vdvdy       =  0.
```

```
          dvdt(i,j)  =  0.
          cory(i,j)  =  0.
          gdedy(i,j) =  0.
          fricy(i,j) =  0.
          vn(i,j)    =  0.

          if(he(i,j).ge.5..or.he(i+1,j).ge.5..and.nsbc.lt.3)go to 50
          if(he(i,j).ge.20..or.he(i+1,j).ge.20.)go to 50

          if(j.eq.1)go to 50

          kslp       =  1
          if(he(i,j).lt.-6..and.he(i+1,j).lt.-6.)kslp = 0

          if(kslp.eq.0)then

              dg  =  0.5*(e(i,j) + e(i+1,j) - h(i,j-1) - h(i,j))

          else

              eg  =  0.5*(e(i,j) + e(i+1,j))
              call dedgec(eg,h(i,j),h(i,j-1),dg)
              if(dg.lt.fdl)dg = fdl

          endif

          if(dg.le.0.)go to 30

c-----        Coriolis

          uatvp      =  (u(i,j-1) + u(i+1,j-1) + u(i+1,j) + u(i,j))*0.25
          cory(i,j)  =  f*uatvp

c-----        Pressure Gradient

          gdedy(i,j) =  9.81*(e(i,j) - e(i+1,j))/dy

c-----        Eddy Viscosity

          if(i.lt.2.or.i.gt.(ny-2).or.j.gt.(nx-1))go to 10
          hedd       =  hed*dg
          call ceddyy(i,j,v,hedd,eddyy)
          call cvdvdy(i,j,v,vdvdy)
          call cudvdx(i,j,v,udvdx,uatvp)

c-----        Friction

   10     ratey      =  sqrt(uatvp*uatvp + v(i,j)*v(i,j))
          fricy(i,j) =  cdrag*v(i,j)*ratey/dg

c-----        Temporal

          dvdt(i,j)  =  - gdedy(i,j) - fricy(i,j) + eddyy - udvdx
     &                  - vdvdy - cory(i,j)

c-----        New value of v

          vn(i,j)    =  v(i,j) + dt*dvdt(i,j)

   30     dvdt1      =  dvdt(if,jf)
          cory1      =  cory(if,jf)
          gdedy1     =  gdedy(if,jf)
          fricy1     =  fricy(if,jf)

   50 continue
```

```
      return
      end


      subroutine ceddyx(i,j,u,hedd,eddyx)
c-----
c-----       To calculate eddy viscosity in the x-direction
c-----

      dimension u(ny,nx)
      common /lim/ ny,nx
      common /coord/ if,jf,dx,dy

      d2udx2  =  (u(i,j-1) + u(i,j+1) - 2.*u(i,j))/(dx*dx)
      d2udy2  =  (u(i-1,j) + u(i+1,j) - 2.*u(i,j))/(dy*dy)

      eddyx   =  hedd*(d2udx2 + d2udy2)

      return
      end


      subroutine ceddyy(i,j,v,hedd,eddyy)
c-----
c-----       To calculate eddy viscosity in the y-direction
c-----

      dimension v(ny,nx)
      common /lim/ ny,nx
      common /coord/ if,jf,dx,dy

      d2vdx2  =  (v(i,j+1) + v(i,j-1) -2.*v(i,j))/(dx*dx)
      d2vdy2  =  (v(i-1,j) + v(i+1,j) -2.*v(i,j))/(dy*dy)

      eddyy   =  hedd*(d2vdx2 + d2vdy2)

      return
      end


      subroutine chstab(en,un,vn)
c-----
c-----       To check the stability of the model
c-----

      dimension en(ny,nx), un(ny,nx), vn(ny,nx)
      common /lim/ ny,nx
      common /time/ itc,psi,its,dt,hour,psin
      common /stab/ cd,cv,key

      key  =  0

      do 1 i = 1,ny
      do 1 j = 1,nx

         if(en(i,j).lt.-cd.or.en(i,j).gt.cd.or.un(i,j).lt.-cv.
     &      or.un(i,j).gt.cv.or.vn(i,j).lt.-cv.or.vn(i,j).gt.cv
     &      )then

            key  =  1

            write(*,20)i,j
            write(*,25)un(i,j),vn(i,j),en(i,j),psi

            return
```

```
         endif

    1 continue

   20 format(/' Incipient Instability at i =',i3,',  j =',i3)
   25 format(/'un=',f9.3,' vn=',f9.3,' en =',f7.3,' psi=',f5.0)

      return
      end


      subroutine cududx(i,j,u,ududx)
c-----
c-----       To calculate ududx
c-----

      dimension u(ny,nx)
      common /lim/ ny,nx
      common /coord/ if,jf,dx,dy

      small  =  1.0e-06

      if(abs(u(i,j-1)).lt.small.and.abs(u(i,j+1)).lt.small)then

         ududx  =  0.
         return

      elseif(abs(u(i,j-1)).lt.small)then

c-----
c-----       To evaluate ududx near to an east-facing coastline
c-----

         if(u(i,j).gt.0.)then

            ududx  =  0.
            return

         else

c-----       central and eastern point

            ududx  =  u(i,j)*(u(i,j+1) - u(i,j))/dx
            return

         endif

      elseif(abs(u(i,j+1)).lt.small)then

c-----
c-----       To evaluate ududx near to a west-facing coastline
c-----

         if(u(i,j).lt.0.)then

            ududx  =  0.
            return

         else

c-----       central and western point

            ududx  =  u(i,j)*(u(i,j) - u(i,j-1))/dx
            return

         endif
```

```
      else

         ududx  =  u(i,j)*(u(i,j+1) - u(i,j-1))/(2.*dx)
         return

      endif

      end


      subroutine cudvdx(i,j,v,udvdx,uatvp)
c-----
c-----       To calculate udvdx
c-----

      dimension v(ny,nx)
      common /lim/ ny,nx
      common /coord/ if,jf,dx,dy

      small  =  1.0e-06

      if(abs(v(i,j-1)).lt.small.and.abs(v(i,j+1)).lt.small)then

c-----
c-----       If the flow is through a channel then udvdx is zero.
c-----

         udvdx  =  0.
         return

      elseif(abs(v(i,j-1)).lt.small)then

c-----
c-----       To evaluate udv/dx for an east-facing coastline.
c-----

         udvdx  =  uatvp*(v(i,j+1) - v(i,j))/dx
         return

      elseif(abs(v(i,j+1)).lt.small)then

c-----
c-----       To evaluate udv/dx for a west-facing coastline.
c-----

         udvdx  =  uatvp*(v(i,j) - v(i,j-1))/dx
         return

      else

         udvdx  =  uatvp*(v(i,j+1) - v(i,j-1))/(2.*dx)

      endif

      end


      subroutine cvdudy(i,j,u,vdudy,vatup)
c-----
c-----       To calculate vdudy
c-----

      dimension u(ny,nx)
      common /lim/ ny,nx
      common /coord/ if,jf,dx,dy
```

```
      small  =  1.0e-06

      if(abs(u(i-1,j)).lt.small.and.abs(u(i+1,j)).lt.small)then

c-----
c-----       If the flow is through a channel then vdudy is zero
c-----

         vdudy  =  0.
         return

      elseif(abs(u(i+1,j)).lt.small)then

c-----
c-----       To evaluate vdu/dy for a north-facing coastline
c-----

         vdudy  =  vatup*(u(i-1,j) - u(i,j))/dy
         return

      elseif(abs(u(i-1,j)).lt.small)then

c-----
c-----       To evaluate vdu/dy for a south-facing coastline
c-----

         vdudy  =  vatup*(u(i,j) - u(i+1,j))/dy
         return

      else

         vdudy  =  vatup*(u(i-1,j) - u(i+1,j))/(2.*dy)
         return

      endif

      end


      subroutine cvdvdy(i,j,v,vdvdy)
c-----
c-----       To calculate vdvdy
c-----

      dimension v(ny,nx)
      common /lim/ ny,nx
      common /coord/ if,jf,dx,dy

      small  =  1.0e-06

      if(abs(v(i-1,j)).lt.small.and.abs(v(i+1,j)).lt.small)then

         vdvdy  =  0.
         return

      elseif(abs(v(i-1,j)).lt.small)then

c-----
c-----        To evaluate vdvdy for a south-facing coastline
c-----

         if(v(i,j).lt.0.)then

            vdvdy  =  0.
            return
```

```
         else

c-----      central and southern points

            vdvdy  =  v(i,j)*(v(i,j) - v(i+1,j))/dy
            return

         endif

      elseif(abs(v(i+1,j)).lt.small)then

c-----
c-----      To evaluate vdvdy for a north-facing coastline
c-----

         if(v(i,j).gt.0.)then

            vdvdy  =  0.
            return

         else

c-----      central and northern points

            vdvdy  =  v(i,j)*(v(i-1,j) - v(i,j))/dy
            return

         endif

      else

         vdvdy  =  v(i,j)*(v(i-1,j) - v(i+1,j))/(2.*dy)
         return

      endif

      end


      subroutine dedgec(eg,h1,h2,dg)
c-----
c-----      To evaluate the depth along the edge of a cell
c-----

      if(eg.gt.h1.and.eg.gt.h2)then

         dg   = eg - 0.5*(h1 + h2)
         return

      elseif(eg.le.h1.and.eg.le.h2)then

         dg   = 0.
         return

      elseif(h1.gt.h2)then

         dg   = (eg - h2)*(eg - h2)*0.5/(h1 - h2)
         return

      else

         dg   = (eg - h1)*(eg - h1)*0.5/(h2 - h1)
         return

      endif
```

```
      end

      subroutine depths(h,he,kwet)
c-----
c-----      To specify depths
c-----

      dimension he(ny,nx), h(ny,nx), kwet(ny,nx)
      common /lim/ ny,nx
      common /start/kie,kis,nrun,rr

      if(rr.gt.0)then

c-----      To create a flat sea-bed, with slight random variation

         do 50 i = 1,ny
         do 50 j = 1,nx

            h(i,j)  =  0.
            he(i,j) =  -50. + (drand() - 0.5)*rr
   50       h(i,j)  =  0.25*(he(i,j) + he(i,j+1) + he(i+1,j)
     &                + he(i+1,j+1))

      else

c-----      To create a Gaussian shoal

         do 60 i = 1,ny
         do 60 j = 1,nx

            r2he    =  (real(i-19.5))**2 + (real(j-19.5))**2
            he(i,j) =  -50.*(1 - exp(-r2he*0.01))

            r2h     =  (real((i+0.5) - 19.5))**2
     &                + (real((j+0.5) - 19.5))**2
            h(i,j)  =  -50.*(1 - exp(-r2h*0.01))

            if(he(i,j).lt.0.)then
               kwet(i,j)  =  1
            else
               kwet(i,j)  =  0
            endif

   60    continue

      endif

      do 70 i = 15,25

   70    write(*,20)(h(i,j), j = 15,25)

      do 80 i = 1,ny

         write(13,30)(h(i,j), j = 1,nx)
   80    write(13,30)(he(i,j), j = 1,nx)

   20 format(11f7.2)
   30 format(10f8.2)

      return
      end


      subroutine dpoint(h,deepp,he)
```

```
c----
c----       To establish the deepest point of a cell.
c----         (Based around the elevation point).
c----

      dimension he(ny,nx), deepp(ny,nx), h(ny,nx)
      common /lim/ ny,nx
      common /coord/ if,jf,dx,dy
      common /time/ itc,psi,its,dt,hour,psin
      common /grid/ ngs,nsbc,ndal

      do 50 i = 2,ny-1
      do 50 j = 2,nx-1

      deepp(i,1)    =  he(i,1)
      deepp(i,nx)   =  h(i,nx)
      deepp(1,j)    =  h(1,j)
      deepp(ny,j)   =  h(ny,j)
      deepp(1,1)    =  he(1,1)
      deepp(1,nx)   =  h(1,nx)
      deepp(ny,nx)  =  h(ny,nx)
      deepp(ny,1)   =  h(ny,1)

      if(nsbc.eq.1)deepp(ny,1)   =  5.0


         if(h(i,j).ge.5..and.nsbc.lt.3)then

            deepp(i,j) =  h(i,j)

         elseif(h(i,j).ge.20..and.nsbc.eq.3)then

            deepp(i,j) =  h(i,j)

         else

            deepp(i,j) =  h(i,j)

            if(h(i-1,j).lt.deepp(i,j))deepp(i,j)   =  h(i-1,j)
            if(h(i-1,j-1).lt.deepp(i,j))deepp(i,j) =  h(i-1,j-1)
            if(h(i,j-1).lt.deepp(i,j))deepp(i,j)   =  h(i,j-1)

         endif

   50 continue

c----       To display depths of cells taken for analysis

      if(its.eq.0)write(*,200)h(if,jf),he(if,jf)
  200 format(' h(if,jf) = ',f6.2,'m',9x,' he(if,jf) = ',f6.2,'m.')

      if(its.eq.0)then

         write(*,*)
         write(*,*)
         write(*,*)'      Deepest Points'
         write(*,*)
         write(*,*)

         do 15 i = 15,25

            write(*,16)(deepp(i,j),j = 15,25)

   15    continue

         write(*,*)
```

```
      write(*,*)
      write(*,*)'      H-values'
      write(*,*)
      write(*,*)

      do 17 i = 15,25

         write(*,16)(h(i,j),j = 15,25)

   17 continue

   16 format(11f7.2)

      endif



      return
      end


      subroutine expluv(un,vn)
c-----
c-----      To extrapolate u and v on the boundaries
c-----

      dimension un(ny,nx), vn(ny,nx)
      common /lim/ ny,nx

      do 1 j = 2,nx-2

         un(1,j)     =  2.*un(2,j) - un(3,j)
         un(ny,j)    =  2.*un(ny-1,j) - un(ny-2,j)
    1 continue

      do 2 i = 2,ny-2

         vn(i,1)     =  2.*vn(i,2) - vn(i,3)
         vn(i,nx)    =  2.*vn(i,nx-1) - vn(i,nx-2)
    2 continue

c-----       at the corners:

      un(1,1)     =  2.*un(2,2) - un(3,3)
      vn(1,1)     =  2.*vn(2,2) - vn(3,3)
      un(ny,1)    =  2.*un(ny-1,2) - un(ny-2,3)
      vn(ny-1,1)  =  2.*vn(ny-2,2) - vn(ny-3,3)
      un(1,nx)    =  2.*un(2,nx-2) - un(3,nx-3)
      vn(1,nx)    =  2.*vn(2,nx-1) - vn(3,nx-2)
      un(ny,nx-1) =  2.*un(ny-1,nx-2) - un(ny-2,nx-3)
      vn(ny-1,nx) =  2.*vn(ny-2,nx-1) - vn(ny-3,nx-2)

      return
      end


      subroutine focus(nc,t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12,
     &                 t13,t14,t15,t16,t17,t18,t19,t20,
     &                 hx,he1,he2,he3,he4)
c-----
c-----      To focus on values in and surrounding the test cell
c-----
      common /grid/ ngs,nsbc,ndal
      common /start/kie,kis,nrun,rr

      ta  =  t1*100.
```

```
      tb  =  t2*100.
      tc  =  t3*100.
      td  =  t4*100.
      te  =  t5*100.
      tf  =  t6*100.
      tg  =  t7*100.
      th  =  t8*100.
      ti  =  t9*100.
      tj  =  t10*100.
      tk  =  t11*100.
      tl  =  t12*1.0e+05
      tm  =  t13*1.0e+05
      tn  =  t14*1.0e+05
      to  =  t15*1.0e+05
      tp  =  t16*100
      tq  =  t17*1.0e+05
      tr  =  t18*1.0e+05
      ts  =  t19*1.0e+05
      tt  =  t20*1.0e+05

      if(ndal.eq.4)then

         if(t10.le.hx)then

            ta  =  0.
            tb  =  0.
            tc  =  0.
            td  =  0.
            te  =  0.
            tj  =  hx*100.
            tk  =  0.
            tl  =  0.
            tm  =  0.
            tn  =  0.
            to  =  0.
            tp  =  0.
            tq  =  0.
            tr  =  0.
            ts  =  0.
            tt  =  0.

         endif

         if(t6.le.he1)tf = he1*100.
         if(t7.le.he2)tg = he2*100.
         if(t8.le.he3)th = he3*100.
         if(t9.le.he4)ti = he4*100.

      endif

      write(nc,10)ta,tb,tc,td,te,tf,tg,th,ti,tj,tk,tl,tm,tn,to,tp,tq,
     &            tr,ts,tt,nrun
   10 format(11f7.1,2f7.2,2(f8.3,1x),f7.1,1x,2f7.2,2(f8.3,1x),i5)

      return
      end


      subroutine heron(x1,y1,x2,y2,x3,y3,area)
c-----
c-----       To find the area of a triangle using Heron's formula
c-----

      a  =  sqrt((x2 - x1)**2 + (y2 - y1)**2)
      b  =  sqrt((x3 - x2)**2 + (y3 - y2)**2)
      c  =  sqrt((x1 - x3)**2 + (y1 - y3)**2)
```

```
      s  =  0.5*(a + b + c)
      d  =  s*(s - a)*(s - b)*(s - c)

      if(d.le.0.)then

         area  =  0.
         return

      else

         area  =  sqrt(d)
         return

      endif

      end


      subroutine intop(e,u,un,v,vn,em,um,vm)
c-----
c-----       Tp provide an interim output at the time horizon for e
c-----

      dimension u(ny,nx), v(ny,nx), un(ny,nx),
     &          vn(ny,nx), e(ny,nx)
      integer um(ny,nx), vm(ny,nx), em(ny,nx)
      common /lim/ ny,nx
      common /time/ itc,psi,its,dt,hour,psin
      common /start/ kie,kis,nrun,rr

      do 1 i = 1,ny
      do 1 j = 1,nx

         em(i,j)  =  int(100.*e(i,j))
         um(i,j)  =  int(50.*(u(i,j) + un(i,j)))
         vm(i,j)  =  int(50.*(v(i,j) + vn(i,j)))

    1 continue



      jf  =  40
      ji  =  jf - 6

      do 2 k = 1,1

         write(54,20)nrun,itc,psi
         write(54,21)(j,j,j = ji,jf)

      do 2 i = 34,40

         write(54,22)i,(em(i,j),um(i,j),j = ji,jf)
         write(54,23)i,(vm(i,j),j = ji,jf)

    2 continue

   20 format(1h1,'  run no. ',i3,9x,'Tidal Cycle no. ',i2,9x,
     &'psi = ',f6.2//)
   21 format(5x,22i5/)
   22 format(i3,2x,22i5)
   23 format(i3,2x,11(i5,5x))

      return
      end
```

```
      subroutine leen3(i,j,kwet,h,en,he)
c-----
c-----       to execute Leendertses' (1970) 3rd drying test
c-----

      dimension en(ny,nx), he(ny,nx), h(ny,nx), kwet(ny,nx)
      common /lim/ ny,nx

      if(kwet(i,j).eq.0)then

         return

      else

         kwet(i,j)  =  1
         hn         =  0.5*(h(i-1,j-1) + h(i-1,j))
         hmin       =  hn
         hs         =  0.5*(h(i,j-1) + h(i,j))

         if(hs.gt.hmin)    hmin = hs

         heast      =  0.5*(h(i-1,j) + h(i,j))

         if(heast.gt.hmin) hmin = heast

         hw         =  0.5*(h(i-1,j-1) + h(i,j-1))

         if(hw.gt.hmin)    hmin = hw

         ecrit      =  0.02 + hmin

         if(en(i,j).lt.ecrit)then

            kwet(i,j) = 0
            en(i,j)   = he(i,j)

         endif

      endif

      return
      end


      subroutine owen3(i,j,h,en,he,kwet)
c-----
c-----       to execute Owens' (1984) 3rd drying test
c-----

      dimension en(ny,nx), he(ny,nx), h(ny,nx),
     &          kwet(ny,nx)
      common /lim/ ny,nx

      if(kwet(i,j).eq.0)then

         return

      else

         kwet(i,j)  =  1
         hn         =  0.5*(h(i-1,j-1) + h(i-1,j))
         hmax       =  hn
         hs         =  0.5*(h(i,j-1) + h(i,j))

         if(hs.lt.hmax)    hmax = hs
```

```
         heast      =  0.5*(h(i-1,j) + h(i,j))

         if(heast.lt.hmax) hmax = heast

         hw         =  0.5*(h(i-1,j-1) + h(i,j-1))

         if(hw.lt.hmax)    hmax = hw

         ecrit      =  0.1 + hmax

         if(en(i,j).lt.ecrit)then

            en(i,j)   = he(i,j)
            kwet(i,j) = 0.

         endif

      endif

      return
      end


      subroutine quiver(u,un,v,vn,e,he)
c-----
c-----       To determine the magnitude and direction of the current
c-----       vector for production of quivers

      dimension u(ny,nx), v(ny,nx), un(ny,nx),
     &          vn(ny,nx), e(ny,nx), he(ny,nx)
      common /lim/ ny,nx
      common /time/ itc,psi,its,dt,hour,psin
      common /numb/ pi,rad

      return
      end


      subroutine raise(h,he,deepp)
c-----
c-----       To raise the sea-bed by 0.05m per time-step
c-----

      dimension he(ny,nx), deepp(ny,nx), h(ny,nx)
      common /lim/ ny,nx
      common /time/ itc,psi,its,dt,hour,psin

      if(psi.lt.270..or.psi.ge.350.)then

         return

      else

         do 1 i = 1,ny
         do 1 j = 1,nx

            h(i,j)  =  h(i,j) + 0.05
            he(i,j) =  he(i,j) + 0.05

    1    continue

         do 2 i = 1,ny-1
         do 2 j = 1,nx-1

            deepp(i,j)  =  deepp(i,j) + 0.05
```

```
    2    continue

         return

       endif

       end


       subroutine seten(en)
c----
c----       To set en on the boundaries
c----

       dimension en(ny,nx)
       common /lim/ ny,nx
       common /time/ itc,psi,its,dt,hour,psin
       common /grid/ ngs,nsbc,ndal
       common /numb/ pi,rad

       psir  =  psin*rad
       wind  =  1.
       if(itc.eq.0)wind = psin/360.

       if(ngs.eq.2)then

c----
c----       North and South boundaries (1000m)
c----
          p0  =  0.
          do 1 j = 1,nx

             dp  =  (real(j) - 20.5)*0.005
             if(nsbc.eq.2)en(1,j) = 2.8*cos(psir - p0 - dp)*wind
             en(ny,j)  =  3.2*cos(psir - p0 - dp)*wind

    1     continue

c----
c----       East and West boundaries (1000m)
c----
          if(nsbc.eq.1)then

             do 2 i = 20,40

                a          =  3.+real(i-20)*0.01
                en(i,nx)   =  a*cos(psir - p0 - 0.1)*wind

    2        continue

             return

          else

             do 3 i = 2,ny-1

             a          =  3.+(real(i) - 20.5)*0.01
             if(nsbc.eq.2)en(i,1) = a*cos(psir - p0 + 0.1)*wind
             en(i,nx)   =  a*cos(psir - p0 - 0.1)*wind

    3        continue

             return

          endif
```

```
      else

c-----
c-----      North and South boundaries (5000m)
c-----
         p0  =  0.
         do 4 j = 1,nx

            dp  =  (real(j) - 20.5)*0.025
            if(nsbc.eq.2)en(1,j) = 2.0*cos(psir - p0 -dp)*wind
            en(ny,j)  =  4.0*cos(psir - p0 -dp)*wind

    4    continue

c-----
c-----      East and West boundaries (5000m)
c-----
         if(nsbc.eq.1)then

            do 5 i = 10,40

               a          =  3.+real(i-20)*0.05
               en(i,nx)   =  a*cos(psir - p0 - 0.5)*wind

    5       continue

            return

         else

            do 6 i = 2,ny

               a          =  3.+(real(i) - 20.5)*0.05
               en(i,1)    =  a*cos(psir - p0 + 0.5)*wind
               en(i,nx)   =  a*cos(psir - p0 - 0.5)*wind

    6       continue

            return

         endif

      endif

      end


      subroutine sethhe(he,h)
c-----
c-----      To set h and he in case of hot-start
c-----

      dimension he(ny,nx), h(ny,nx)
      common /lim/ ny,nx
      common /grid/ ngs,nsbc,ndal

      if(nsbc.eq.1)then

         do 100 i = 1,ny

  100       read(99,20)(he(i,j), j = 1,nx)

         do 200 i = 1,ny

  200       read(99,20)(h(i,j), j = 1,nx)
```

```
      else

         do 500 i = 1,ny

  500       read(50,20)(he(i,j), j = 1,nx)

         do 600 i = 1,ny

  600       read(50,20)(h(i,j), j = 1,nx)

      endif

   20 format(5e16.5)

      return
      end


      subroutine setic(un,vn,en,e,u,v,h,he)
c-----
c-----       To set initial conditions
c-----

      dimension u(ny,nx), v(ny,nx), un(ny,nx),
     &          vn(ny,nx), e(ny,nx), en(ny,nx),
     &          he(ny,nx), h(ny,nx)
      common /lim/ ny,nx
      common /grid/ ngs,nsbc,ndal
      common /start/kie,kis,nrun,rr

      do 50 i = 1,ny
      do 50 j = 1,nx

         en(i,j)  =  0.
         un(i,j)  =  0.
         vn(i,j)  =  0.

   50 continue

      if(kis.eq.1)then

c-----       Hot-start; e,u and v fields read from euvstore
c-----       and modified for dry cells.

         do 60 i = 1,ny

            read(56,65)(e(i,j), j = 1,nx)

         do 60 j = 1,nx

            if(e(i,j).lt.he(i,j))e(i,j)  =  he(i,j)

   60    continue

         do 70 i = 1,ny

            read(56,65)(u(i,j), j = 1,nx)

         do 70 j = 1,nx-1

            if((e(i,j) - he(i,j)).lt.1.0e-06)u(i,j)  =  0.

            if((e(i,j+1) - he(i,j+1)).lt.1.0e-06)u(i,j)  =  0.

   70    continue
```

```
         do 80 i = 1,ny-1

            read(56,65)(v(i,j), j = 1,nx)

         do 80 j = 1,nx

            if((e(i,j) - he(i,J)).lt.1.0e-06)v(i,j)  =  0.
            if((e(i+1,j) - he(i+1,j)).lt.1.0e-06)v(i,j)  =  0.

   80    continue

      else

c-----      Cold-start; e = u = v = 0 everywhere.

         do 150 i = 1,ny
         do 150 j = 1,nx

            e(i,j)  =  0.
            u(i,j)  =  0.
            v(i,j)  =  0.

  150    continue

         if(kie.eq.0)return

         call setice(e)
         if(h(i,j).gt.5..or.he(i,j).gt.5.)then

            do 160 i = 1,ny
            do 160 j = 1,nx

               e(i,j) = 0.
               u(i,j) = 0.
               v(i,j) = 0.

  160       continue

         endif

      endif

   65 format(5e16.5)

      return
      end


      subroutine setice(e)
c-----
c-----      To set the initial conditions on the boundaries.
c-----

      dimension e(ny,nx)
      common /lim/ ny,nx
      common /time/ itc,psi,its,dt,hour,psin
      common /grid/ ngs,nsbc,ndal
      common /numb/ pi,rad

      psir =  psi*rad

      do 100 i = 1,ny

         if(ngs.eq.1)da = real(i-20.5)*0.05
         if(ngs.eq.2)da = real(i-20.5)*0.01
```

```
         a =  3. + da

      do 100 j = 1,nx

         if(ngs.eq.1)dp = real(j-20.5)*0.025
         if(ngs.eq.2)dp = real(j-20.5)*0.005

         p =  (180.*rad) + dp

         e(i,j) = a*cos(psir - p)

  100 continue

      return
      end


      subroutine setwet(kn)
c-----
c-----       To set everywhere wet
c-----

      dimension kn(ny,nx)
      common /lim/ ny,nx

      do 100 i = 1,ny
      do 100 j = 1,nx

         kn(i,j) = 1

  100 continue

      return
      end



      subroutine smool(x,is,js,iif,jjf)
c-----
c-----       To smooth a line of values
c-----

      dimension x(ny,nx), ws(70)
      common /lim/ ny,nx

      small  =  1.0e-06

      if(js.eq.jjf)then

c-----       Meridional smoothing

         ws(is)     =  x(is,jjf)
         ws(is+1)   =  0.25*(x(is,jjf) + 2.*x(is+1,jjf) + x(is+2,jjf))
         ws(iif-1)  =  0.25*(x(iif-2,jjf) + 2.*x(iif-1,jjf) +
     &                  x(iif,jjf))
         ws(iif)    =  x(iif,jjf)

         do 8 i = is+2,iif-2
         do 6 ii = i-2,i+2

            if(x(ii,jjf).gt.20..or.abs(x(ii,jjf)).lt.small)go to 7

    6    continue

            ws(i)  =  (6.*x(i,jjf) + 4.*(x(i-1,jjf) + x(i+1,jjf)) +
```

```
     &                  x(i-2,jjf) + x(i+2,jjf))/16.

            go to 8

     7      ws(i)  =  x(i,jjf)
     8    continue

          do 9 i = is,iif

             x(i,jjf)  =  ws(i)

     9    continue

          return

        else

c----       Zonal smoothing

          if(is.ne.iif)return

          ws(js)     =  x(is,js)
          ws(js+1)   =  0.25*(x(is,js) + 2.*x(is,js+1) + x(is,js+2))
          ws(jjf-1)  =  0.25*(x(is,jjf-2) + 2.*x(is,jjf-1) + x(is,jjf))
          ws(jjf)    =  x(is,jjf)

          do 3 j = js+2,jjf-2
          do 1 jj = j-2,j+2

             if(x(is,jj).gt.20..or.abs(x(is,jj)).lt.small)go to 2

     1    continue

             ws(j)  =  (6.*x(is,j) + 4.*(x(is,j-1) + x(is,j+1))
     &                  + x(is,j-2) + x(is,j+2))/16.
             go to 3

     2       ws(j)  =  x(is,j)

     3    continue

          do 4 j = js,jjf

             x(is,j)  =  ws(j)

     4    continue

          return

        endif

        end


        subroutine smooth(un,vn)
c----
c----       To smooth values of un and vn on the boundaries
c----

        dimension un(ny,nx), vn(ny,nx)
        common /lim/ ny,nx

        call smool(un,1,1,1,nx-1)
        call smool(un,ny,1,ny,nx-1)
        call smool(un,1,nx-1,ny,nx-1)
        call smool(un,1,1,ny,1)
```

```
      call smool(vn,1,1,1,nx)
      call smool(vn,ny-1,1,ny-1,nx)
      call smool(vn,1,nx,ny-1,nx)
      call smool(vn,1,1,ny-1,1)

      return
      end


      subroutine stohhe(he,h)
c-----
c-----       To store he and h at the end of the nth cycle
c-----

      dimension he(ny,nx), h(ny,nx)
      common /lim/ ny,nx

      rewind 50

      do 100 i = 1,ny

         write(50,200)(he(i,j),j = 1,nx)

  100 continue

      do 150 i = 1,ny

         write(50,200)(h(i,j), j = 1,nx)

  150 continue

  200 format(5e16.5)

      return
      end


      subroutine store(he,e,u,v,hs)
c-----
c-----       To store e, u, v at the end of the nth cycle
c-----

      dimension e(ny,nx), he(ny,nx), u(ny,nx),
     &          v(ny,nx), hs(ny,nx)
      common /lim/ ny,nx

      rewind 56

      do 10 i = 1,ny
      do 10 j = 1,nx

         hs(i,j)  =  0.

   10 continue

      do 60 i = 1,ny

         write(56,65)(e(i,j),j = 1,nx)

   60 continue

      do 61 i = 1,ny

         write(56,65)(u(i,j),j = 1,nx)

   61 continue
```

```
      do 62 i = 1,ny

         write(56,65)(v(i,j),j = 1,nx)

   62 continue

      do 63 i = 1,ny

         write(56,65)(hs(i,j),j = 1,nx)

   63 continue

      write(*,20)

   20 format(//'Data have been stored in file newstore')
   65 format(5e16.5)

      return
      end


      subroutine wasen(e,a,p)
c----
c----      To specify e on the boundaries
c----

      dimension e(ny,nx), a(ny,nx), p(ny,nx)
      common /lim/ ny,nx
      common /time/ itc,psi,its,dt,hour,psin
      common /numb/ pi,rad

      wind  =  1.
      wm2   =  28.98410422
      t     =  12.42*psin/360.

      if(itc.eq.0)wind = psin/360.

      do 1 j = 40,nx

         e(1,j)  =  a(1,j)*cos((wm2*t - p(i,j))*rad)*wind

    1 continue

      do 2 i = 2,24

         e(i,nx)  =  a(i,nx)*cos((wm2*t - p(i,65))*rad)*wind

    2 continue

      return
      end


      subroutine washhe(h,he)
c----
c----       To read h and he from washhe in case of
c----       a hot start.
c----

      dimension h(ny,nx), he(ny,nx)
      common /lim/ ny,nx

      open(43,file='washhe')

      do 100 i = 1,ny
```

```
         read(43,75)(h(i,j),j = 1,nx)

  100 continue

      do 200 i = 1,ny

         read(43,75)(he(i,j),j = 1,nx)

  200 continue

   75 format(65f5.1)

      return
      end


      subroutine wasic(un,vn,en,e,u,v,a,p)
c-----
c-----       To set the initial conditions for the Wash.
c-----

      dimension un(ny,nx), vn(ny,nx), en(ny,nx),
     &          e(ny,nx),u(ny,nx), v(ny,nx),
     &          a(ny,nx), p(ny,nx)
      common /lim/ ny,nx
      common /numb/ pi,rad

      do 100 i = 1,nx
      do 100 j = 1,ny

         un(i,j)  =  0.
         vn(i,j)  =  0.
         u(i,j)   =  0.
         v(i,j)   =  0.
         e(i,j)   =  0.

  100 continue

      call wasice(i,j,e,a,p)

      return
      end


      subroutine wasice(i,j,e,a,p)
c-----
c-----       To read the boundary values of a(i,j) and p(i,j)
c-----       for the Wash model.
c-----

      dimension e(ny,nx), a(ny,nx), p(ny,nx)
      real amp,phase
      common /lim/ ny,nx

      open(10,file='was-bcez')

      do 50 m = 1,49

         read(10,111)i,j,amp
         a(i,j)  =  amp*0.001

   50 continue

         rewind 10
```

```
      do 100 m = 1,49

         read(10,112)i,j,phase
         p(i,j)  =  phase*0.1

  100 continue

  111 format(3x,i2,3x,i2,31x,f5.0)
  112 format(3x,i2,3x,i2,36x,f5.0)

      return
      end


      subroutine wdtest(i,j,e1,h1,e2,h2,kn,uv)
c-----
c-----       Wet or dry test using:
c-----       Flather & Heaps' (1975) drying algorithm
c-----

      dimension kn(ny,nx)
      common /lim/ ny,nx
      kn(i,j)  =  0
      uv       =  0.

c      go to 1000

      if(e2.gt.20..or.e1.gt.20.)then
         kn(i,j) = 0
         return

      else

        d1  =  e1 - h1
        d2  =  e2 - h2

      endif

      if(d1.le.0..and.d2.le.0.)then
         kn(i,j) = 0
         return

      elseif(d1.le.0.)then

         if((e2 - e1).le.0.1)then
            kn(i,j) = 0
            return

         else

            kn(i,j)  =  1
            return

         endif

      elseif(d2.le.0.)then

         if((e1 - e2).le.0.1)then
            kn(i,j) = 0
            return

         else

            kn(i,j)  =  1
            return
```

```
         endif

      else

         kn(i,j)  =  1
         return

      endif

      return

 1000 if((e2.gt.20..or.e1.gt.20.))return

      d1 = e1 - h1
      d2 = e2 - h2
      if((d1.le.0..and.d2.le.0.))return
      if(d1.le.0.)go to 2
      if(d2.le.0.)go to 3
      go to 5
    2 if((e2 - e1).le.0.1)return
      go to 5
    3 if((e1 - e2).le.0.1)return
    5 kn(i,j) = 1
      return
      end


      subroutine wetest(i,j,kwet,e,he,dn,ds,de,dw)
c-----
c-----       to test whether a dry cell has become wet (Leendertse 1970)
c-----

      dimension e(ny,nx), he(ny,nx), kwet(ny,nx)
      common /lim/ ny,nx

      kwet(i,j)  =  0
      k4         =  0
      emean      =  0.

      if(e(i-1,j).gt.he(i-1,j).or.dn.gt.0.)then

         k4      =  k4 + 1
         emean   =  emean + e(i-1,j)

      endif

      if(e(i+1,j).gt.he(i+1,j).or.ds.gt.0.)then

         k4      =  k4 + 1
         emean   =  emean + e(i+1,j)

      endif

      if(e(i,j+1).gt.he(i,j+1).or.de.gt.0.)then

         k4      =  k4 + 1
         emean   =  emean + e(i,j+1)

      endif

      if(e(i,j-1).gt.he(i,j-1).or.dw.gt.0.)then

         k4      =  k4 + 1
         emean   =  emean + e(i,j-1)

      endif
```

```
      if(k4.ne.0)then

         emean   =  emean/real(k4)
         if(emean.ge.e(i,j))kwet(i,j) = 1

      endif

      return
      end


      subroutine wetowe(i,j,kwet,e,he,dn,ds,de,dw)
c-----
c-----       To test for a flooding cell using Owewns' (1984)
c-----       improved method
c-----

      dimension e(ny,nx), he(ny,nx), kwet(ny,nx)
      common /lim/ ny,nx

      kwet(i,j) = 0
      kx        = 0
      emean     = 0.

      if(e(i-1,j).gt.he(i-1,j).or.dn.gt.0..or.(-he(i-1,j) + e(i-1,j)).gt
     &    .(-he(i,j) + e(i,j)))then

         kx     = kx + 1
         emean  = emean + e(i-1,j)

      endif

      if(e(i+1,j).gt.he(i+1,j).or.ds.gt.0..or.(-he(i+1,j) + e(i+1,j)).gt
     &    .(-he(i,j) + e(i,j)))then

         kx     = kx + 1
         emean  = emean + e(i+1,j)

      endif

      if(e(i,j+1).gt.he(i,j+1).or.de.gt.0..or.(-he(i,j+1) + e(i,j+1)).gt
     &    .(-he(i,j) + e(i,j)))then

         kx     = kx + 1
         emean  = emean + e(i,j+1)

      endif
      if(e(i,j-1).gt.he(i,j-1).or.dw.gt.0..or.(-he(i,j-1) + e(i,j-1)).gt
     &    .(-he(i,j) + e(i,j)))then

         kx     = kx + 1
         emean  = emean + e(i,j-1)

      endif

      if(kx.ne.0)then

         emean  = emean/real(kx)
         if(emean.gt.e(i,j))kwet(i,j) = 1

      endif

      return
      end
```

APPENDIX 2

```
      PROGRAM GPLOT
C----
C----    To plot terms using GINOGRAF routines.
C----
      DIMENSION TX(360,21),KQ(5),TE(360,11)
      CHARACTER FLOW*1
      CALL GINO
      CALL PLOTTR
      CALL DEVPAP(400.,300.,0)
      CALL SHIFT2(0.,50.)
      CALL PA4F
      NDAL=4
      PRINT *,'                                    '
      PRINT *,'Would you like to analyse the flow'
      PRINT *,'                                    '
      PRINT *,'      around an E-point ? (Y/N) '
      PRINT *,'                                    '
      READ(*,'(A1)')FLOW
      IF(FLOW.EQ.'Y')GO TO 1000
      PRINT *,'Would you like a plot of terms obtained from:'
      PRINT *
      PRINT *,'1: Flather & Heaps (1975) method,'
      PRINT *
      PRINT *,'2: Leendertses (1970) method, '
      PRINT *
      PRINT *,'3: Owens (1984) improved method  '
      PRINT *
      PRINT *,'              or                '
      PRINT *
      PRINT *,'4: George & Striplings (1993) much improved method ?'
      PRINT *
      READ *,NDAL
      IF(NDAL.EQ.1)GO TO 900
      IF(NDAL.EQ.2)GO TO 500
      IF(NDAL.EQ.3)GO TO 159
      OPEN(63,FILE='DDS')
      DO 100 I=1,360
      READ(63,10)(TX(I,J),J=1,10)
   10 FORMAT(10F7.1)
      DO 100 J=1,10
  100 TX(I,J)=TX(I,J)/100.
      READ(63,150)NRUN
  150 FORMAT(148X,I6)
      CALL MOVTO2(100.,180.)
      CALL CHASIZ(3.,3.)
      CALL MOVBY2(10.,0.)
      CALL MOVTO2(70.,185.)
      CALL CHAINT(NRUN,3)
      CALL MOVTO2(90.,185.)
      CALL CHAHOL(30HGEORGE & STRIPLING'S (1993) *.)
      CALL CHAHOL(18HDRYING ALGORITHM*.)
      REWIND 63
      DO 200 I=1,360
      READ(63,11)(TX(I,J),J=11,15)
   11 FORMAT(70X,F7.1,2F7.2,F8.3,F9.3)
      TX(I,11)=TX(I,11)/100.
      IF(TX(I,5).LE.0.)TX(I,11)=0.
      TX(I,12)=TX(I,12)*10.
      IF(TX(I,5).LE.0.)TX(I,12)=0.
      TX(I,13)=TX(I,13)*10.
      IF(TX(I,5).LE.0.)TX(I,13)=0.
      TX(I,14)=TX(I,14)*10.
      IF(TX(I,5).LE.0.)TX(I,14)=0.
      TX(I,15)=TX(I,15)*10.
      IF(TX(I,5).LE.0.)TX(I,15)=0.
  200 CONTINUE
```

```
      REWIND 63
      DO 300 I=1,360
      READ(63,12)(TX(I,J),J=16,20),NRUN
   12 FORMAT(108X,F8.1,F8.2,F7.2,F8.3,F9.3,I6)
      TX(I,16)=TX(I,16)/100.
      IF(TX(I,5).LE.0.)TX(I,16)=0.
      TX(I,17)=TX(I,17)*10.
      IF(TX(I,5).LE.0.)TX(I,17)=0.
      TX(I,18)=TX(I,18)*10.
      IF(TX(I,5).LE.0.)TX(I,18)=0.
      TX(I,19)=TX(I,19)*10.
      IF(TX(I,5).LE.0.)TX(I,19)=0.
      TX(I,20)=TX(I,20)*10.
      IF(TX(I,5).LE.0.)TX(I,20)=0.
  300 CONTINUE
C----     TO CALCULATE A BASIC TIDE-CURVE FOR NO SHOAL PRESENT
      DO 301 I=1,360
      A=REAL(I-1)
  301 TX(I,21)=2.*COS((A-180.)*0.0174532925)
      CALL SCREEN(KQ,KHE,KUV,KAC)
      CALL AXISTT
      CALL AXISHE(KHE)
      CALL AXISUV(KUV,NDAL)
      CALL AXISAC(KAC)
      CALL SHIFT2(90.,100.)
      CALL DRA5L(TX,KQ,NDAL)
      GO TO 90
  500 OPEN(62,FILE='DDLS')
      DO 501 I=1,360
      A=REAL(I-1)
  501 TX(I,21)=2.*COS((A-180.)*0.0174532925)
      DO 600 I=1,360
      READ(62,16)(TX(I,J),J=1,10)
   16 FORMAT(10F7.1)
      DO 600 J=1,10
  600 TX(I,J)=TX(I,J)/100.
      READ(62,650)NRUN
  650 FORMAT(148X,I6)
      CALL MOVTO2(100.,180.)
      CALL CHASIZ(3.,3.)
      CALL MOVBY2(10.,0.)
      CALL MOVTO2(70.,185.)
      CALL CHAINT(NRUN,3)
      CALL MOVTO2(90.,185.)
      CALL CHAHOL(20HLEENDERTSES' 1970 *.)
      CALL CHAHOL(18HDRYING ALGORITHM*.)
      REWIND 62
      DO 700 I=1,360
      READ(62,18)(TX(I,J),J=11,15)
   18 FORMAT(70X,F7.1,2F7.2,F8.3,F9.3)
      TX(I,11)=TX(I,11)/100.
      TX(I,12)=TX(I,12)*10.
      TX(I,13)=TX(I,13)*10.
      TX(I,14)=TX(I,14)*10.
      TX(I,15)=TX(I,15)*10.
  700 CONTINUE
      REWIND 62
      DO 800 I=1,360
      READ(62,19)(TX(I,J),J=16,20),NRUN
   19 FORMAT(108X,F8.1,F8.2,F7.2,F8.3,F9.3,I6)
      TX(I,16)=TX(I,16)/100.
      TX(I,17)=TX(I,17)*10.
      TX(I,18)=TX(I,18)*10.
      TX(I,19)=TX(I,19)*10.
      TX(I,20)=TX(I,20)*10.
  800 CONTINUE
```

```
      CALL SCREEN(KQ,KHE,KUV,KAC)
      CALL AXISTT
      CALL AXISHE(KHE)
      CALL AXISUV(KUV,NDAL)
      CALL AXISAC(KAC)
      CALL SHIFT2(90.,100.)
      CALL DRA5L(TX,KQ,NDAL)
      GO TO 90
C----   To plot terms from Owens (1984) algorithm.
  159 OPEN(64,FILE='DDOS')
      DO 156 I=1,360
      A=REAL(I-1)
  156 TX(I,21)=2.*COS((A-180.)*0.0174532925)
      DO 160 I=1,360
      READ(64,161)(TX(I,J),J=1,10)
  161 FORMAT(10F7.1)
      DO 160 J=1,10
  160 TX(I,J)=TX(I,J)/100.
      READ(64,162)NRUN
  162 FORMAT(148X,I6)
      CALL MOVTO2(100.,180.)
      CALL CHASIZ(3.,3.)
      CALL MOVBY2(10.,0.)
      CALL MOVTO2(70.,185.)
      CALL CHAINT(NRUN,3)
      CALL MOVTO2(90.,185.)
      CALL CHAHOL(23HOWENS' 1984 IMPROVED *.)
      CALL CHAHOL(18HDRYING ALGORITHM*.)
      REWIND 64
      DO 163 I=1,360
      READ(64,164)(TX(I,J),J=11,15)
  164 FORMAT(70X,F7.1,2F7.2,F8.3,F9.3)
      TX(I,11)=TX(I,11)/100.
      TX(I,12)=TX(I,12)*10.
      TX(I,13)=TX(I,13)*10.
      TX(I,14)=TX(I,14)*10.
      TX(I,15)=TX(I,15)*10.
  163 CONTINUE
      REWIND 64
      DO 165 I=1,360
      READ(64,166)(TX(I,J),J=16,20),NRUN
  166 FORMAT(108X,F8.1,F8.2,F7.2,F8.3,F9.3,I6)
      TX(I,16)=TX(I,16)/100.
      TX(I,17)=TX(I,17)*10.
      TX(I,18)=TX(I,18)*10.
      TX(I,19)=TX(I,19)*10.
      TX(I,20)=TX(I,20)*10.
  165 CONTINUE
      CALL SCREEN(KQ,KHE,KUV,KAC)
      CALL AXISTT
      CALL AXISHE(KHE)
      CALL AXISUV(KUV,NDAL)
      CALL AXISAC(KAC)
      CALL SHIFT2(90.,100.)
      CALL DRA5L(TX,KQ,NDAL)
  900 OPEN(23,FILE='DDF')
      DO 904 I=1,360
      READ(23,902)(TX(I,J),J=1,10)
  902 FORMAT(10F7.1)
      DO 904 J=1,10
  904 TX(I,J)=TX(I,J)/100.
      READ(23,950)NRUN
  950 FORMAT(148X,I6)
      CALL MOVTO2(100.,180.)
      CALL CHASIZ(3.,3.)
      CALL MOVBY2(10.,0.)
```

```
      CALL MOVTO2(70.,185.)
      CALL CHAINT(NRUN,3)
      CALL MOVTO2(90.,185.)
      CALL CHAHOL(24HFLATHER & HEAPS' 1975 *.)
      CALL CHAHOL(18HDRYING ALGORITHM*.)
      REWIND 23
      DO 908 I=1,360
      READ(23,910)(TX(I,J),J=11,15)
  910 FORMAT(70X,F7.1,2F7.2,F8.3,F9.3)
      TX(I,11)=TX(I,11)/100.
      TX(I,12)=TX(I,12)*10.
      TX(I,13)=TX(I,13)*10.
      TX(I,14)=TX(I,14)*10.
      TX(I,15)=TX(I,15)*10.
  908 CONTINUE
      REWIND 23
      DO 915 I=1,360
      READ(23,912)(TX(I,J),J=16,20),NRUN
  912 FORMAT(108X,F8.1,F8.2,F7.2,F8.3,F9.3,I6)
      TX(I,16)=TX(I,16)/100.
      TX(I,17)=TX(I,17)*10.
      TX(I,18)=TX(I,18)*10.
      TX(I,19)=TX(I,19)*10.
      TX(I,20)=TX(I,20)*10.
  915 CONTINUE
C-----     TO CALCULATE A BASIC TIDE-CURVE FOR NO SHOAL PRESENT
      DO 920 I=1,360
      A=REAL(I-1)
  920 TX(I,21)=2.*COS((A-180.)*0.0174532925)
      CALL SCREEN(KQ,KHE,KUV,KAC)
      CALL AXISTT
      CALL AXISHE(KHE)
      CALL AXISUV(KUV,NDAL)
      CALL AXISAC(KAC)
      CALL SHIFT2(90.,100.)
      CALL DRA5L(TX,KQ,NDAL)
 1000 OPEN(64,FILE='DDSE')
      DO 1010 I=1,360
 1010 READ(64,1020)(TE(I,J),J=1,10)
 1020 FORMAT(10F6.1,F5.2)
      REWIND 64
      READ(64,1030)NRUN
 1030 FORMAT(65X,I5)
      CALL MOVTO2(100.,180.)
      CALL CHASIZ(3.,3.)
      CALL MOVBY2(10.,0.)
      CALL MOVTO2(70.,185.)
      CALL CHAINT(NRUN,3)
      CALL MOVTO2(90.,185.)
      CALL CHAHOL(24HFLOW AROUND 'E' POINT *.)
C     CALL CHAINT(IF,3)
C     CALL CHAINT(JF,2)
      CALL SCRNE(KQ,KHE,KUV)
      CALL AXISTT
      CALL AXISHE(KHE)
      CALL AXISUV(KUV,NDAL)
      CALL SHIFT2(90.,100.)
      CALL DRA5LE(TE,KQ)
   90 CALL GINEND
      STOP
      END

      SUBROUTINE PA4F
C----  to draw an A4 frame horizontally.
      CALL MOVTO2(0.,0.)
      CALL LINBY2(297.,0.)
```

```
      CALL LINBY2(0.,210.)
      CALL LINBY2(-297.,0.)
      CALL LINBY2(0.,-210.)
      RETURN
      END

      SUBROUTINE SCREEN(KQ,KHE,KUV,KAC)
C---- to choose terms from the SCREEN.
      DIMENSION KQ(5)
      KHE=0
      KUV=0
      KAC=0
      WRITE(1,20)
   20 FORMAT(/'The terms available are as follows:    ,0 = no curve')
      PRINT *
      PRINT *,'1: Depth east of the test-cell'
      PRINT *,'2: Depth west of the test-cell'
      PRINT *,'3: Depth north of the test-cell'
      PRINT *,'4: Depth south of the test-cell'
      PRINT *,'5: Depth of the test-cell'
      PRINT *,'6: Elevation east of the test-cell'
      PRINT *,'7: Elevation west of the test-cell'
      PRINT *,'8: Elevation north of the test-cell'
      PRINT *,'9: Elevation south of the test-cell'
      PRINT *,'10: Elevation in the test-cell'
      PRINT *,'11: Eastward velocity component'
      PRINT *,'12: Eastward temporal acceleration'
      PRINT *,'13: Eastward Coriolis acceleration'
      PRINT *,'14: Eastward pressure force/mass'
      PRINT *,'15: Eastward frictional force/mass'
      PRINT *,'16: Northward velocity component'
      PRINT *,'17: Northward temporal acceleration'
      PRINT *,'18: Northward Coriolis acceleration'
      PRINT *,'19: Northward pressure force/mass'
      PRINT *,'20: Northward frictional force/mass'
      PRINT *,'21: Base curve for elevations'
      WRITE(1,10)
   10 FORMAT(/' Type the code numbers for 5 variables')
      READ(1,*)KQ
      DO 1 K=1,5
      IF(KQ(K).EQ.0)GO TO 1
      IF(KQ(K).LE.10)KHE=1
      IF((KQ(K).EQ.11).OR.(KQ(K).EQ.16))KUV=1
      IF((KQ(K).GE.12).AND.(KQ(K).LE.15))KAC=1
      IF((KQ(K).GE.17).AND.(KQ(K).LE.20))KAC=1
    1 CONTINUE
      RETURN
      END

      SUBROUTINE AXISTT
C---- to plot axis for Tidal Time (degrees)
      CALL SHIFT2(90.,100.)
      CALL AXIPOS(1,0.,0.,180.,1)
      CALL AXISCA(2,36,0.,360.,1)
      CALL AXIDRA(1,1,1)
      CALL AXNSTR('TIDAL TIME ANGLE (degrees)',-90.,1,0)
      CALL SHIFT2(-90.,-100.)
      RETURN
      END

      SUBROUTINE AXISAC(KAC)
C---- to plot AXIS for ACcelerations
      IF(KAC.EQ.0)RETURN
      CALL SHIFT2(30.,100.)
      CALL AXIPOS(1,0.,-75.,150.,2)
      CALL AXISCA(2,6,-0.003,0.003,2)
```

```
      CALL AXIDRA(-2,-1,2)
      CALL AXNSTR('ACCELERATION (m/s****2)',-15.,-2,0)
      CALL SHIFT2(-30.,-100.)
      KAC=0
      RETURN
      END

      SUBROUTINE AXISHE(KHE)
C----   to plot AXIS for water depth
      IF(KHE.EQ.0)RETURN
      CALL SHIFT2(90.,100.)
      CALL AXIPOS(1,0.,-75.,150.,2)
      CALL AXISCA(2,8,-4.0,4.0,2)
      CALL AXIDRA(-2,-1,2)
      CALL AXNSTR('WATER DEPTH, (m)',-15.,-2,0)
      CALL SHIFT2(-90.,-100.)
      KHE=0
      RETURN
      END

      SUBROUTINE AXISUV(KUV,NDAL)
C----   to plot AXIS for plot of velocity
      IF(KUV.EQ.0)RETURN
      CALL SHIFT2(60.,100.)
      CALL AXIPOS(1,0.,-75.,150.,2)
      IF(NDAL.EQ.1)CALL AXISCA(2,20,-1.0,1.0,2)
      IF((NDAL.EQ.2).OR.(NDAL.EQ.3))CALL AXISCA(2,20,-2.0,2.0,2)
      CALL AXIDRA(-2,-1,2)
      CALL AXNSTR('VELOCITY, (m/s)',-15.,-2,0)
      CALL SHIFT2(-60.,-100.)
      KUV=0
      RETURN
      END

      SUBROUTINE DRA5L(TX,KQ,NDAL)
C----   to DRAw up to 5 Lines
      DIMENSION TX(360,21),KQ(5),Q(360),MCOL(30)
      CHARACTER*50 TITEL(30)
      DATA MCOL/6,5,7,2,9,6,5,7,2,9,2,2,4,5,6,7,9,1,0,0,2,2,4,5,6,7,9,0,
     +0,0/
      DO 1 L=1,30
    1 TITEL(L)='                                                  '
      TITEL(1)='Depth east of the test-cell                       '
      TITEL(2)='Depth west of the test-cell                       '
      TITEL(3)='Depth north of the test-cell                      '
      TITEL(4)='Depth south of the test-cell                      '
      TITEL(5)='Depth of the test-cell                            '
      TITEL(6)='Elevation east of the test-cell                   '
      TITEL(7)='Elevation west of the test-cell                   '
      TITEL(8)='Elevation north of the test-cell                  '
      TITEL(9)='Elevation south of the test-cell                  '
      TITEL(10)='Elevation in the test-cell                        '
      TITEL(11)='Eastward velocity component                       '
      TITEL(12)='Eastward temporal acceleration                    '
      TITEL(13)='Eastward Coriolis acceleration                    '
      TITEL(14)='Eastward pressure force/mass                      '
      TITEL(15)='Eastward frictional force/mass                    '
      TITEL(16)='Northward velocity component                      '
      TITEL(17)='Northward temporal acceleration                   '
      TITEL(18)='Northward Coriolis acceleration                   '
      TITEL(19)='Northward pressure force/mass                     '
      TITEL(20)='Northward frictional force/mass                   '
      TITEL(21)='Base curve for elevations                         '
      DO 2 K=1,5
      LQ=KQ(K)
      IF(LQ.EQ.0)GO TO 2
```

```
      DO 3 I=1,360
    3 Q(I)=TX(I,LQ)
      F=19.5
      IF(LQ.GT.10)F=0.125
      IF((LQ.EQ.11).OR.(LQ.EQ.16))F=75.
      IF(LQ.EQ.21)F=19.5
      CALL DELLIN(Q,F,MCOL(LQ))
      Y=-60.-REAL(K-1)*6.
      CALL MOVTO2(10.,Y)
      CALL LINBY2(15.,0.)
      CALL MOVTO2(30.,Y)
      CALL CHASIZ(2.,2.)
      CALL CHASTR(TITEL(LQ))
    2 CONTINUE
      RETURN
      END

      SUBROUTINE DELLIN(Q,F,NCOL)
      DIMENSION Q(360)
      CALL LINCOL(NCOL)
      DO 2 I=1,360
      Y=Q(I)*F
      X=REAL(I-1)*0.5
      IF(I.GT.1)GO TO 1
      CALL MOVTO2(X,Y)
      GO TO 2
    1 CALL LINTO2(X,Y)
    2 CONTINUE
      RETURN
      END

      SUBROUTINE SCRNE(KQ,KHE,KUV)
C---- to choose terms from the SCREEN.
      DIMENSION KQ(5)
      KHE=0
      KUV=0
      KAC=0
      WRITE(1,20)
   20 FORMAT(/'The terms available are as follows:    ,0 = no curve')
      PRINT *
      PRINT *,'1: U(i,j)'
      PRINT *,'2: U(i,j-1)'
      PRINT *,'3: V(i,j)'
      PRINT *,'4: V(i-1,j)'
      PRINT *,'5: Depth at the U-point east of the E-point'
      PRINT *,'6: Depth at the V-point south of the E-point'
      PRINT *,'7: Depth at the U-point west of the E-point'
      PRINT *,'8: Depth at the V-point north of the E-point'
      PRINT *,'9: Depth at the E-point'
      PRINT *,'10: Computational depth of the cell'
      WRITE(1,10)
   10 FORMAT(/' Type the code numbers for up to 5 variables')
      READ(1,*)KQ
      DO 1 K=1,5
      IF(KQ(K).EQ.0)GO TO 1
      IF(KQ(K).LT.5)KUV=1
      IF(KQ(K).GT.4)KHE=1
    1 CONTINUE
      RETURN
      END

      SUBROUTINE DRA5LE(TE,KQ)
C---- to DRAw up to 5 Lines
      DIMENSION TE(360,11),KQ(5),Q(360),MCOL(30)
      CHARACTER*50 TITEL(30)
      DATA MCOL/6,5,7,2,9,6,5,7,2,9,2,2,4,5,6,7,9,1,0,0,2,2,4,5,6,7,9,0,
```

```
 +0,0/
  DO 1 L=1,15
1 TITEL(L)='                                                              '
  TITEL(1)='U(i,j)                                                        '
  TITEL(2)='U(i,j-1)                                                      '
  TITEL(3)='V(i,j)                                                        '
  TITEL(4)='V(i-1,j)                                                      '
  TITEL(5)='Depth at the U-point east of the E-point                      '
  TITEL(6)='Depth at the V-point south of the E-point                     '
  TITEL(7)='Depth at the U-point west of the E-point                      '
  TITEL(8)='Depth at the V-point north of the E-point                     '
  TITEL(9)='Depth at the E-point                                          '
  TITEL(10)='Computational depth of the cell                               '
  DO 2 K=1,5
  LQ=KQ(K)
  IF(LQ.EQ.0)GO TO 2
  DO 3 I=1,360
3 Q(I)=TE(I,LQ)
  F=0.195
  IF((LQ.GE.1).AND.(LQ.LT.5))F=0.75
  CALL DELLIN(Q,F,MCOL(LQ))
  Y=-60.-REAL(K-1)*6.
  CALL MOVTO2(10.,Y)
  CALL LINBY2(15.,0.)
  CALL MOVTO2(30.,Y)
  CALL CHASIZ(2.,2.)
  CALL CHASTR(TITEL(LQ))
2 CONTINUE
  RETURN
  END
```

APPENDIX 3

TIME	DEPTH	VEL.
Station 0		24.6.93
8.65	0.92	0.047
8.67	0.95	0.065
8.7	1.06	0.093
8.72	1.11	0.112
8.75	1.14	0.102
8.78	1.19	0.064
8.8	1.22	0.072
8.82	1.25	0.07
8.83	1.27	0.071
8.87	1.3	0.057
8.88	1.32	0.058
8.93	1.36	0.054
8.95	1.39	0.054
8.97		0.052
9	1.48	0.058
9.25	1.6	0.06
9.5	1.7	0.081
9.75	1.8	0.075
10	1.8	0.064
10.25	1.6	0.06
10.5	1.54	0.053
10.75	1.25	0.022
10.97	1.24	0.047
11.02	1.16	0.047
11.05	1.12	0.063
11.1	1.08	0.088
11.12	1.06	0.081
11.13	1.05	0.082
11.15	1.03	0.105
11.17	1.02	0.085
11.18	1	0.088
11.19	0.97	0.151
11.21	0.96	0.135
11.24	0.94	0.197
11.26	0.93	0.125
11.28	0.92	0.125
11.29	0.91	0.117
11.31	0.9	0.128
11.33	0.89	0.124

11.34	0.85	0.102

STAT 1 21.6.93

TIME	DEPTH	VEL.
6.67	1.2	0.192
6.75	1.25	0.113
7	1.5	0.071
7.25	1.6	0.082
7.5	1.8	0.07
7.75	1.8	0.063
8	1.7	0.057
8.25	1.6	0.05
8.5	1.4	0.05
8.75	1.2	0.082
9	1.15	0.147
9.25	1.1	0.18

STAT 3(b) 23.6.93

TIME	DEPTH	VEL.	SET
20.05	0.86	0.311	
20.07	0.9	0.417	
20.1	1	0.247	
20.13	1.05	0.196	
20.15	1.1	0.146	
20.17	1.13	0.149	
20.18	1.2	0.182	
20.22	1.25	0.157	
20.23	1.27	0.173	
20.25	1.34	0.166	
20.5	1.6	0.154	350
20.75	1.9	0.155	
21	2.1	0.122	340
21.25	2.3	0.069	
21.5	2.4	0.071	10
21.75	2.4	0.06	
22	2.3	0.058	30
22.25	2.2	0.041	

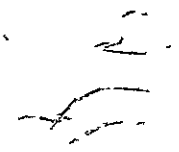
22.5	2.17	0.047	40
22.75	1.9	0.047	
23	1.6	0.075	50
23.25	1.4	0.122	
23.5	1.25	0.167	90
23.58	1.15	0.149	
23.6	1.1	0.134	
23.62	1.06	0.13	
23.63	1.05	0.13	
23.65	1	0.142	
23.68	0.95	0.158	
23.7	0.93	0.116	70
23.72	0.9	0.098	

STAT 4 19.6.93

TIME	DEPTH	VEL.	SET
17.25	1.1	0.135	
17.5	1.5	0.115	
17.75	1.8	0.146	310
18	2	0.144	340
18.25	2.1	0.19	
18.5	2.15	0.155	330
18.75	2.2	0.229	
19	2.1	0.076	10
19.25	2	0.044	
19.5	1.9	0.078	40
19.75	1.7	0.102	
20	1.5	0.108	60
20.25	1.3	0.044	

STAT 5a 21.6.93

TIME	DEPTH	VEL.
18.43	0.88	0.162
18.45	0.93	0.351
18.47	0.98	0.256
18.48	1.02	0.275
18.53	1.08	0.273



18.57	1.12	0.177
18.58	1.15	0.138
18.67	1.29	0.145
18.92	1.6	0.168
19.25	2.05	0.149
19.5	2.3	0.159
19.75	2.35	0.147
20	2.45	0.177
20.25	2.48	0.175
20.5	2.37	0.127
20.75	2.33	0.095
21	2.15	0.09
21.25	1.95	0.091
21.5	1.75	0.123
21.75	1.5	0.126
22	1.28	0.123
22.12	1.1	0.118
22.13	1.08	0.104
22.15	1.05	0.107
22.17	1.03	0.115
22.18	1.01	0.11
22.2	0.99	0.114
22.22	0.97	0.101
22.23	0.95	0.114
22.25	0.93	0.154
22.26	0.91	0.119
22.28	0.9	0.154
22.3	0.88	0.079
22.32	0.86	0.173
22.33	0.84	0.088

STAT 5b 19.6.93

TIME	DEPTH	VEL.
4.75	1	0.14
5	1.4	0.153
5.25	1.6	0.136
5.5	1.9	0.172
5.75	2	0.16
6	2.1	0.19
6.25	2.2	0.183

6.5	2.2	0.113	
6.75	2.1	0.117	
7	2	0.104	
7.25	1.8	0.101	
7.5	1.7	0.1	
7.75	1.5	0.094	
8	1.3	0.091	

STAT 6 22.6.93

TIME	DEPTH	VEL.	
18.8	0.82	0.298	
18.82	0.87	0.318	
18.83	0.96	0.315	
18.85	0.97	0.438	
18.87	1.03	0.432	
18.88	1.05	0.399	
18.9	1.09	0.444	
18.92	1.16	0.424	
18.93	1.22		
18.95	1.29		
18.98	1.31		
19	1.37	0.399	
19.08	1.41	0.373	
19.17	1.57	0.35	
19.25	1.8	0.345	
19.33	1.97	0.323	310
19.42	2.05	0.298	
19.5	2.2	0.275	
19.75	2.65	0.198	300
20	2.8	0.163	
20.25	3.2	0.126	10
20.5	3.3	0.116	
20.75	3.35	0.168	40
21	3.38	0.179	
21.25	3.3	0.189	60
21.5	3.25	0.163	
21.75	3.05	0.136	60
22	2.9		
22.33	2.7	0.123	80
22.5	2.55	0.103	

22.75	2.28	0.122	50
23	2.02	0.149	90
23.25	1.64	0.208	
23.5	1.35	0.229	80
23.67	1.13	0.203	
23.75	0.99	0.163	
23.8	0.93		
23.83	0.91		
23.85	0.89		
23.87	0.87		
23.88	0.84		

APPENDIX 4

```
       0.4  53 12.00   0-12.                                           L
       0.4  53 06.00   0-12.                                           L
       0.4  53 00.00   0-12.                                           L
       0.4  52 54.00   0-12.   Welland
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -70
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -70 -70 -70
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -70 -70 -70 -70
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -70 -70 -70-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
       0.4  52 48.00   0-12.                                           L
       0.4  53 12.00  00 00.  North of Boston                          L
       0.4  53 06.00  00 00.  Boston
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -72
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -64 -71-069
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -57 -65 -60
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -56 -56 -57 -59 -54
-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -57 -57 -55 -53 -49 -38
-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -57 -57 -57 -46 -35 -32
-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -51 -47 -45 -43 -35 -33
       0.4  53 00.00  00 00.  (TO BE NAMED)
-200-200-200-200-200-200-200-200-200-200-200-200-200 -67 -55 -49 -45 -42 -28 -24
-200-200-200-200-200-200-200-200-200-200-200-200 -67 -66 -50 -40 -33 -32 -23 -18
-200-200-200-200-200-200-200-200-200-200-200 -66 -68 -56 -49 -38 -27  10  53  50
-200-200-200-200-200-200-200-200-200-200 -65 -65 -55 -50 -48 -50  42  72  50  55
-200-200-200-200-200-200-200-200-200-066-053-051-049-043  56  71  26  31  01
-200-200-200-200-200-200-200-200 -67-061-050-044-050 022  55  05 -08 -22 -18
-200-200-200-200-200-200-200 -68-055-050-046-039 023 061  01 -09 -16 -32 -28
-200-200-200-200-200-200-200-200-068-046-047-021 008 019 009 -22 -14 -35 -26 -18
-200-200-200-200-200-200-200 -67-059-046-024-018-007-003-010 -23 -35 -28 -20 -05
-200-200-200-200-200-200-200-067-038-010-015-015-014 000-027 -35 -28  29  61  92
-200-200-200 -70-067-066-068-037-023-044-047-044-019-017-015  08  22  09  02 -22
-200-200 -72-070-068-066-035-044-040-041-040-041-006-018 016 -10 -21 -35 -30 -28
-200 -69-071-072-066-027-030-041-055-041-034-005-030 000-030 -28 -30 -26 -21 -06
-200 -70-070-064-047-039-048-049-040-046-008-034-031-027-035 -40 -36 -15 -19  06
 -71-071-066-058-064-055-045-047-044-033-044-042-035-036-037 -38 -38 -39 -33 -34
       0.4  52 54.    00 00.  WELLAND CUT
-073-068-052-060 -64 -62 -60-059-057-044-053-051-045-042-045 -45 -38 -43 -39 -28
-070 -70 -67-200-200-200-200 -58 -57-055-066-065-056-050-044 -50 -43 -44 -38 -26
 -70-200-200-200-200-200-200-200-200 -68-067-069-068-067-066 -58 -49 -49 -40 -33
-200-200-200-200-200-200-200-200-200-200-200 -68 -67-072-067 -65 -60 -55 -49 -35
-200-200-200-200-200-200-200-200-200-200-200-200-200 -70 -67 -68 -69 -59 -55 -38
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -70 -69 -66 -47 -41
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -69 -68 -64 -47
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -69 -67 -65
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -65 -53
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -65
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -66
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
```

```
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
       0.4  52 48.00  00 00.  Holbeach                                        L
       0.4  53 12.    00 12.  SKEGNESS
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -27  39  40  46  49
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -26  34  39  43  49
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -25  06  40  50  46
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -24  20  40  50  52
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -10  37  36  42  49
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -10  32  33  42  45
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -10  33  35  42  47
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -13  14  36  39  48
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -13  23  31  42  51
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -23  02  21  45  50
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -28  06  10  24  57  62
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -29  02  26  25  47  54
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -31  08  42  36  25  07
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -30  35  32  20  22  59
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -16 -03  31  03  04  00  60
       0.4  53 06.00  00 12.  Friskney Flats
-200-200-200-200-200-200-200-200-200 -37-037-037 -35-034 008  14  02  00  07  72
-200-200-200-200-200-200-200-200 -40 -37-037-037-037-032-012 017  21  17  38  28  61
-200-200-200-200-200-200 -46-043-041-036-035-033-027 030 002  30  40  13  60  43
-200-200-200-200-200 -53-050-047-046-043-038-035-016 021 040  45  21  45  38  58
-200-200-200-200 -55-052-048-047-045-050-042-039-001 035 060  60  41  19  18  53
-200-200-200 -57-053-050-047-045-044-040-041-037 000 065 056  30 -04 -36  02  66
-200-200 -60-056-051-050-045-042-040-037-036-029 030 094 067  09 -31 -50  08  96
 -76 -77-059-055-051-046-040-038-038-034-026-020 047 106 048 -12 -23 -23  31 120
-075-069-057-053-050-045-039-035-030-025-020 001 088 090 019 -23 -25  06 113  92
-059-062-055-050-046-042-038-029-024-001 030 099 104 045-005 -10  24  77  65 113
-054-055-050-048-041-039-039-023 029 081 101 100 077 029 032  15  27  29  27 178
-046-039-040-040-030-020-004 036 084 070 077 044 013-019-023  00 -05  07 115 274
-038-038-032-027-008 030 063 055 073 098 041-006-020-023-025 -19 -01 114 260 327
-029-029-023 011 015 071 104 098 039 005-010-013-017 000 019  70 134 262 333 330
-040-041-026-006 050 091 117 050-010-020-019-019-006 015 060 120 210 260 320 330
       0.4  53 00.     0 12.   Roaring Middle
-013 013 058 088 100 099 020-025-026-014-014-004 026 080 120 200 270 260 270 240
 057 075 099 063 037-020-019-024-015-008 023 058 100 150 190 230 280 210 250 160
 086 058 018-022-010-015-008 011 032 050 080 100 140 180 210 220 230 210 200 180
 054 020-005-010 004 010 018 028 049 060 120 150 200 220 190 150 200 200 180 150
-025-008 037 027 017 021 030 047 069 100 160 210 220 120 090 150 160 150 140 150
-027-032-016 019 031 030 046 060 070 120 190 190 150 100 100 140 150 150 110 150
-029-022-011-009-001 015 042 070 150 180 170 140 100 080 100 130 130 120 100 140
-012-010-019-011 009 037 051 060 160 160 120 140 080 060 140 140 110  90 110 100
-002 000-019-006 008 032 071 110 130 110 100 070 070 090 130 120  80  70  80 110
 093 052 023 036 049 070 110 130 120 010 070 045 060 110 110  70  60  60  60  90
-020 006 036 034 030 060 100 100 080 080 070 032 080 110 070  60  50  49 149  70
-028-010 004 005 037 070 100 080 070 090 043 061 095 081 051  44  41  26  25  55
-006-012-002 017 039 080 060 050 090 050 028 075 072 062 040  25  22  06  26  37
 019-016 007 022 058 045 037 060 085 027 057 055 058 030 019  10  41  02  21  32
-012 015 020 033 046 028 050 090 048 050 041 051 037 007 006  08  30  00  12  13
       0.4  52 54.     0 12.   (north of Breast Sand)
-020-010 008 020 030 009 065 073 022 030 033 041 004-007-010  28  05 -08  22  04
-019-016-006 041 013 015 099 059 023 011 030 002-021-020-022  28 -16 -20  14  00
-028-011 045-002 003 033 063 062-005 000 039-010-039-032-045  23 -15  19 -25  14
-025-019 027-005 000 057 035 050-024-029 038-027-037-041-041  06 -03  16 -13  10
-027-022 023-013-009 052-002 038 004-018 027-028-040-041-027 -10  11 -10 -05 -14
-031-022 010-010-003 025 022 009 009-017 020-040-039-040-028 -08 -08  07 -14 -15
-033-022-008 005-022-018 022 000-003-010 011-037-036-035-023 -09 -26 -12 -14 -19
-044-010-010-013-022-033-008 010-016 000 012-026-034-034-025 -14 -20 -17 -38 -45
-053-015-022-023-018-031-032-018-020-030-001-011-012-028-027 -24 -21 -28 -30 -46
-077-028-031-035-030-031-032-033-025-031-027-014-007-007-028 -14 -16 -33 -43 -51
-063-044-039-042-037-037-040-031-016-034-034-030-033-018-037 -14 -29 -38 -52 -56
 -64-063-049-047-047-046-042-041-037-046-042-037-040-040-023 -20 -37 -50 -58 -66
-200-056-061-057-058-050-056-049-048-061-060-047-051-025-032 -15 -45 -55 -64 -65
```

```
-200-200-200-200 -63 -60-059-061 -60 -65-200 -66 -63-064-047 -41 -55 -58 -66-200
-200-200-200-200-200-200-200-200-200-200-200-200-200 -64-064 -47 -48 -62-200-200
       0.4  52 48.      0 12.   King's Lynn (Ouse)
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -61 -55 -63-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200 -68 -68-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
       0.4  53 12.      0 24.   east of Skegness
  52  56  60  63  67  68  70  72  76 103 116 150 171 171 171 150 113 124 131 101
  49  49  52  61  67  68  72  90 110 110 116 150 165 200 200 175 150 140 125  82
  40  50  57  63  67  68  74  75 105 111 116 114 160 195 210 200 163 140 124 110
  49  55  61  63  67  71  79  90  97 106 112 135 171 171 162 175 195 150 116 130
  58  56  62  62  67  72  79  90  92 101 101 120 140 175 150 200 158 155 121 121
  47  48  55  63  67  72  79  81 100 106 103 118 120 179 148 220 150 155 167 161
  48  51  56  63  68  74  80  82 106 112 130 151 139 140 145 230 188 230 190 210
  49  53  56  63  75  78  84  84 100 112 140 158 160 180 200 220 192 210 187 170
  52  54  57  65  84  84  88  90 104 136 136 140 150 148 205 210 210 210 165 140
  55  57  63  77  87  91  94 100 112 134 140 150 155 200 232 241 270 227 158 118
 060 060 072 078 090 093 101 103 115 133 142 163 200 250 246 242 242 214 151 150
 067 067 078 083 088 098 103 103 129 142 148 171 230 234 249 230 198 169 132 122
 087 069 066 090 100 108 102 122 142 146 176 200 270 270 235 188 150 124 114 104
 066 057 054 077 088 087 096 123 137 150 195 240 270 225 205 176 151  78  84  72
 060 060 059 071 079 105 113 121 136 150 241 248 247 220 200 180 166  88 148  18
       0.4  53 06.      0 24.   Lynn Knock
 074 076 064 060 090 105 117 128 167 200 260 249 250 215 186 178 143  74  42  22
 070 079 064 057 094 102 124 143 184 247 250 225 215 195 180 175 116  66  50  26
 072 077 081 110 046 107 152 173 210 235 250 208 200 180 173 155 119  68  33  30
 109 066 078 087 048 102 200 200 230 270 230 189 170 180 140 146 118  83  27  33
 099 092 093 120 112 160 216 250 245 240 200 151 154 158 106 143 113  43  20  26
 100 104 098 150 174 206 238 270 290 200 150 154 165 157 130 120  75  20  02  09
 123 123 120 182 222 220 269 285 247 148 120 159 163 137 100  74  15  08  02  05
 097 155 205 260 311 312 297 250 172 118 162 161 140 057 052  30  30  20  02  02
 129 184 300 300 300 265 242 181 147 139 153 127 075 019 039  78  53  32  35  25
 165 265 317 322 270 248 174 157 146 148 131 125 020 014 027  40  45  49  20  43
 335 357 306 253 220 182 150 118 140 133 114 080 006 008 019  20  23  28  20  22
 371 375 277 250 216 150 125 110 105 099 093 023 000 006 008  15  21  26  20  19
 340 276 215 225 200 130 114 103 094 082 052-007-009 000 014  14  14  18  21  21
 287 250 223 203 148 120 097 093 078 058 054-008-016-010 001  08  04  07  02  16
 250 210 230 200 120 090 080 060 044 035 045-004-013-008-003  00  02 -13  01  11
       0.4  53 00.      0 24.   Sunk Sand
 230 200 250 200 090 110 060 035 011 024 024 020 020 025 024  18  22  22  20  16
 180 220 250 110 100 084 051-004 002 004 000 003 020 170 010  04  10  15  09  11
 150 230 150 100 100 025-023-015 002 001-002 014 007-003-010 -27 -30 -35 -20 -10
 170 180 110 100 029-020-011-017-004 006 008 006-012-020 -30 -30 -37 -40 -37 -40
 180 160 110 085-018-009-012-017 001 004 003-011-025-200-200-200-200-200-200-200
 150 100 090 050-011-010-011 020 005 005-008-046 -57-200-200-200-200-200-200-200
 120 054 063 043 000-012-005 033 001-030-045 -45-200-200-200-200-200-200-200-200
 055 024 044 018 040-006 000 009-006-046-200-200-200-200-200-200-200-200-200-200
 033 020 012 008-018 008-006 006-016-200-200-200-200-200-200-200-200-200-200-200
 014 006-006-003-011-006-005 007-025-200-200-200-200-200-200-200-200-200-200-200
 025 013-001-019 008-007 001-010 -30-200-200-200-200-200-200-200-200-200-200-200
 037 012 003-002-002-004-014-023-200-200-200-200-200-200-200-200-200-200-200-200
 045 034 004-006-008-005-010-030-200-200-200-200-200-200-200-200-200-200-200-200
 041 032 005-013-011-020-028 -38-200-200-200-200-200-200-200-200-200-200-200-200
 032 029-001-011-015-021-037 -40-200-200-200-200-200-200-200-200-200-200-200-200
```

```
       0.4  52 54.     0 24.   Bulldog Channel
 025 015-010-009-016-025-037-200-200-200-200-200-200-200-200-200-200-200-200-200
 026 001-015-013-022-027 -32-200-200-200-200-200-200-200-200-200-200-200-200-200
 005-017-017-013-029-030-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-026-032-025-028-030 -45-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-035-045-041-023-037-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-045-049-041-036-043-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-047-050-046-049-050-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-049-052-055-056-059-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-053-053-058-062-065-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-055-065-069 -70-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-062-072 -72-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
 -64-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
       0.4  52 48.     0 24.   (land)                                  L
       0.4  53 12.     0 36.   Docking Shoal
  90 150 200 200 210 220 200 100  60  55  60  65  70  64  70  94 122 155 150 160
 110 186 200 200 210 210 150  55  55  55  58  60  75  70  75  90  95 113 124 145
 104 186 200 210 210 200 140  55  54  40  50  65  81  70  50  72  58  49  61 100
 130 200 200 210 210 200 100  55  57  60  72  50  45  56  60  44  52  60  72  94
 153 205 210 200 200 110  42  36  40  45  50  45  50  37  40  41  57  72  85 127
 194 210 170 164 150 100  42  45  42  40  39  42  50  54  54  54  56  60  80 110
 195 210 120 100  91  54  51  50  50  54  55  60  55  52  51  56  54  62  72 100
 200 167 060  91 148 110  51  55  63  65  60  60  55  54  55  57  49  47  45  78
 164 110  60  90 100 110 118 100  63  65  66  60  65  70  60  54  40  58  68  84
 130  50  60  70  90 110 110 105  80  65  72  72  75  75  50  49  50  60  70  75
  60  51  60  70  84  90  95 107 105 103 110 110  84  75  75  75  52  63  69  69
  17  33  50  54  70  70  70  60  70  75 100 110  80  75  72  74  84  84  70  59
  23  35  40  50  60  63  65  65  65  69  60  65  70  75  74  74  78  72  60  51
  14  30  42  42  42  63  65  65  70  75  70  55  70  75  76  78  80  65  54  52
  15  20  27  30  45  50  63  65  78  80  81  60  51  50  55  60  84  80  62  54
       0.4  53 06.     0 36.   Burnham Flats
  16  23  25  27  48  50  56  60  60  70  70  65  65  55  44  51  71  80  73  75
  20  23  23  31  29  50  47  60  60  63  63  65  67  72  50  54  63  70  84  70
  39  35  33  36  36  40  42  50  57  57  48  50  70  75  60  51  60  60  75  92
  30  33  33  33  31  33  36  39  40  40  50  42  42  50  52  54  57  63  70  94
  20  23  30  25  20  28  25  23  36  36  45  35  30  35  50  50  54  68  65  78
  15  20  25  20  20  23  20  20  27  30  36  36  32  37  25  34  50  55  60  62
  09  13  14  22  20  20  20  20  27  25  23  30  36  39  45  35  41  33  34  36
  07  11  14  20  17  20  20  20  27  25  20  23  30  40  50  60  63  64  63  52
  10  12  20  20  14  20  20  36  30  30  31  33  37  42  50  52  62  78  70  65
  33  33  26  17  30  24  21  28  30  30  32  40  50  58  64  69  68  66  74  78
  23  22  20  20  24  27  24  23  30  30  50  55  60  72  64  70  42  40  41  35
  14  16  12  14  20  27  24  23  24  24  55  66  38  35  30  27  24  20  23  23
  24  23  20  20  18  18  23  23  20  23  31  34  20  20  20  36  34  36  40  42
  33  38  28  23  41  34  50  45  45  50  27  28  17  20  05  20  24  32  45  57
  20  30  45  39  45  45  63  63  60  57  60  49  35  19  11  16  21  27  36  36
       0.4  53 00.     0 36.   Scolt Head
  16  23  29  36  50  44  50  30  35  20  25  30  20  14  20  26  25  24  36  33
  16  16  20  22  16  01 -10  01 -10 -19  05  05  10  14  14  14  17  23  34  31
  05  06  08  12 -15 -10 -25-200-200-200-200 -19 -19 -16 -13 -03 -04 -10  10  07
 -20 -34 -19 -16 -15 -10 -25-200-200-200-200-200-200-200-200 -24 -23 -22 -14 -09
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200-200
       0.4  52 54.     0 36.   (land)                                  L
```

0.4 52 48. 0 36. (land) L

APPENDIX 5

File HWC-BCEZ

i	j		lat.	long.	H of M_2 (mm)	g of M_2 (1/10°)
03	06	8	54 08	0-03.	1780	1230
03	07	8	54 08.	0 00.	1760	1235
03	08	8			1740	1240
03	09	8			1720	1245
03	10	8			1700	1250
03	11	8			1680	1255
03	12	8	54 08.	+0 15.	1660	1260
03	13	8			1640	1265
03	14	8			1620	1270
03	15	8			1600	1275
03	16	8			1580	1280
03	17	8	54 08.	+0 30.	1560	1285
03	18	8			1540	1290
03	19	8			1520	1295
03	20	8			1500	1300
03	21	8			1480	1305
03	22	8	54 08.	+0 45.	1460	1310
03	23	8			1440	1315
03	24	8			1420	1320
03	25	8			1400	1325
03	26	8			1380	1330
03	27	1	54 08.	+1 00.	1360	1335
04	27	2	54 06.	+1 00.	1368	1350
05	27	2	54 04.	+1 00.	1376	1365
06	27	2	54 02.	+1 00.	1384	1380
07	27	2	54 00.	+1 00.	1392	1395
08	27	2			1400	1410
09	27	2			1408	1425
10	27	2			1416	1440
11	27	2			1424	1455
12	27	2	53 50.	+1 00.	1432	1470
13	27	2			1440	1485
14	27	2			1448	1500
15	27	2			1456	1515
16	27	2			1464	1530
17	27	2	53 40.	+1 00.	1472	1545
18	27	2			1480	1560
19	27	2			1488	1575
20	27	2			1496	1590
21	27	2			1504	1605
22	27	2	53 30.	+1 00.	1512	1620

23	27 2			1520	1635
24	27 2			1528	1650
25	27 2			1536	1665
26	27 2			1544	1680
27	27 2	53 20.	+1 00.	1552	1695
28	27 2			1560	1710
29	27 2			1568	1725
30	27 2			1576	1740
31	27 2			1584	1755
32	27 2	53 10.	+1 00.	1592	1770
33	27 2			1600	1785
34	27 2			1608	1800
35	27 2			1616	1815
36	27 2			1624	1830
37	27 2	53 00.	+1 00.	1632	1845
38	27 2	52 58.	+1 00.	1640	1860

File HWC-BCE2

i	j	lat.	long.	H of S_2	g of S_2	H of N_2	g of N_2
03	06 8	54 08	0-03.	668	1730	356	1030
03	07 8	54 08.	0 00.	660	1735	352	1035
03	08 8			652	1740	348	1040
03	09 8			645	1745	344	1045
03	10 8			638	1750	340	1050
03	11 8			630	1755	336	1055
03	12 8	54 08.	+0 15.	622	1760	332	1060
03	13 8			615	1765	328	1065
03	14 8			608	1770	324	1070
03	15 8			600	1775	320	1075
03	16 8			592	1780	316	1080
03	17 8	54 08.	+0 30.	585	1785	312	1085
03	18 8			578	1790	308	1090
03	19 8			570	1795	304	1095
03	20 8			562	1800	300	1100
03	21 8			555	1805	296	1105
03	22 8	54 08.	+0 45.	548	1810	292	1110
03	23 8			540	1815	288	1115
03	24 8			532	1820	284	1120
03	25 8			525	1825	280	1125
03	26 8			518	1830	276	1130
03	27 1	54 08.	+1 00.	510	1835	272	1135
04	27 2	54 06.	+1 00.	513	1850	274	1150
05	27 2	54 04.	+1 00.	516	1865	275	1165
06	27 2	54 02.	+1 00.	519	1880	277	1180

07	27	2	54	00.	+1	00.	522	1895	278 1195
08	27	2					525	1910	280 1210
09	27	2					528	1925	282 1225
10	27	2					531	1940	283 1240
11	27	2					534	1955	285 1255
12	27	2	53	50.	+1	00.	537	1970	286 1270
13	27	2					540	1985	288 1285
14	27	2					543	2000	290 1300
15	27	2					546	2015	291 1315
16	27	2					549	2030	293 1330
17	27	2	53	40.	+1	00.	552	2045	294 1345
18	27	2					555	2060	296 1360
19	27	2					558	2075	298 1375
20	27	2					561	2090	299 1390
21	27	2					564	2105	301 1405
22	27	2	53	30.	+1	00.	567	2120	302 1420
23	27	2					570	2135	304 1435
24	27	2					573	2150	306 1450
25	27	2					576	2165	307 1465
26	27	2					579	2180	308 1480
27	27	2	53	20.	+1	00.	582	2195	310 1495
28	27	2					585	2210	312 1510
29	27	2					588	2225	314 1525
30	27	2					591	2240	315 1540
31	27	2					594	2255	317 1555
32	27	2	53	10.	+1	00.	597	2270	318 1570
33	27	2					600	2285	320 1585
34	27	2					603	2300	322 1600
35	27	2					606	2315	323 1615
36	27	2					609	2330	325 1630
37	27	2	53	00.	+1	00.	612	2345	326 1645
38	27	2	52	58.	+1	00.	615	2360	328 1660

APPENDIX 6

MODEL PREDICTIONS FOR 1993 6 19

All times are in G.M.T.

52 56 N 0 06 E

DATE AND TIME	HEIGHT (m)	SET (degT)	RATE (m/s)
1993/06/19 00:00	-1.13	0.	.00
1993/06/19 00:30	-1.13	0.	.00
1993/06/19 01:00	-1.13	0.	.00
1993/06/19 01:30	-1.13	0.	.00
1993/06/19 02:00	-1.13	0.	.00
1993/06/19 02:30	-.51	305.	.45
1993/06/19 03:00	.33	229.	.46
1993/06/19 03:30	1.02	239.	.36
1993/06/19 04:00	1.68	237.	.26
1993/06/19 04:30	2.15	233.	.18
1993/06/19 05:00	2.42	227.	.14
1993/06/19 05:30	2.60	225.	.11
1993/06/19 06:00	2.70	212.	.03
1993/06/19 06:30	2.58	57.	.10
1993/06/19 07:00	2.26	56.	.15
1993/06/19 07:30	1.91	58.	.18
1993/06/19 08:00	1.45	57.	.25
1993/06/19 08:30	.92	56.	.32
1993/06/19 09:00	.36	57.	.38
1993/06/19 09:30	-.19	53.	.39
1993/06/19 10:00	-.71	45.	.43
1993/06/19 10:30	-1.13	0.	.00
1993/06/19 11:00	-1.13	0.	.00
1993/06/19 11:30	-1.13	0.	.00
1993/06/19 12:00	-1.13	0.	.00
1993/06/19 12:30	-1.13	0.	.00
1993/06/19 13:00	-1.13	0.	.00
1993/06/19 13:30	-1.13	0.	.00
1993/06/19 14:00	-1.13	0.	.00
1993/06/19 14:30	-1.13	0.	.00
1993/06/19 15:00	-.31	298.	.44
1993/06/19 15:30	.55	228.	.48
1993/06/19 16:00	1.28	240.	.34
1993/06/19 16:30	1.94	238.	.25
1993/06/19 17:00	2.37	230.	.18
1993/06/19 17:30	2.63	227.	.14
1993/06/19 18:00	2.81	228.	.09

1993/06/19 18:30	2.86	75.	.02
1993/06/19 19:00	2.64	59.	.13
1993/06/19 19:30	2.29	56.	.15
1993/06/19 20:00	1.88	59.	.21
1993/06/19 20:30	1.36	57.	.28
1993/06/19 21:00	.80	55.	.35
1993/06/19 21:30	.21	56.	.40
1993/06/19 22:00	-.34	51.	.42
1993/06/19 22:30	-.86	44.	.46
1993/06/19 23:00	-1.13	0.	.00
1993/06/19 23:30	-1.13	0.	.00

MODEL PREDICTIONS FOR 1993 6 20

All times are in G.M.T.

52 56 N 0 06 E

DATE AND TIME	HEIGHT (m)	SET (degT)	RATE (m/s)
1993/06/20 00:00	-1.13	0.	.00
1993/06/20 00:30	-1.13	0.	.00
1993/06/20 01:00	-1.13	0.	.00
1993/06/20 01:30	-1.13	0.	.00
1993/06/20 02:00	-1.13	0.	.00
1993/06/20 02:30	-1.13	0.	.00
1993/06/20 03:00	-1.13	0.	.00
1993/06/20 03:30	-.06	283.	.41
1993/06/20 04:00	.79	231.	.46
1993/06/20 04:30	1.56	240.	.32
1993/06/20 05:00	2.20	236.	.23
1993/06/20 05:30	2.58	230.	.18
1993/06/20 06:00	2.83	227.	.14
1993/06/20 06:30	3.01	231.	.06
1993/06/20 07:00	2.96	58.	.08
1993/06/20 07:30	2.65	58.	.14
1993/06/20 08:00	2.29	58.	.17
1993/06/20 08:30	1.80	58.	.24
1993/06/20 09:00	1.24	57.	.32
1993/06/20 09:30	.66	56.	.38
1993/06/20 10:00	.06	55.	.41
1993/06/20 10:30	-.50	49.	.45
1993/06/20 11:00	-1.02	43.	.48
1993/06/20 11:30	-1.13	0.	.00
1993/06/20 12:00	-1.13	0.	.00

1993/06/20 12:30	-1.13	0.	.00
1993/06/20 13:00	-1.13	0.	.00
1993/06/20 13:30	-1.13	0.	.00
1993/06/20 14:00	-1.13	0.	.00
1993/06/20 14:30	-1.13	0.	.00
1993/06/20 15:00	-1.13	0.	.00
1993/06/20 15:30	-.89	310.	.61
1993/06/20 16:00	.22	256.	.42
1993/06/20 16:30	1.05	235.	.43
1993/06/20 17:00	1.83	240.	.30
1993/06/20 17:30	2.43	232.	.21
1993/06/20 18:00	2.78	227.	.18
1993/06/20 18:30	3.03	228.	.13
1993/06/20 19:00	3.17	217.	.02
1993/06/20 19:30	3.00	58.	.11
1993/06/20 20:00	2.63	57.	.15
1993/06/20 20:30	2.23	60.	.19
1993/06/20 21:00	1.69	58.	.27
1993/06/20 21:30	1.09	56.	.34
1993/06/20 22:00	.50	56.	.40
1993/06/20 22:30	-.11	54.	.43
1993/06/20 23:00	-.67	48.	.47
1993/06/20 23:30	-1.17	43.	.50

MODEL PREDICTIONS FOR 1993 6 21

All times are in G.M.T.

52 56 N 0 06 E

DATE AND TIME	HEIGHT (m)	SET (degT)	RATE (m/s)
1993/06/21 00:00	-1.13	0.	.00
1993/06/21 00:30	-1.13	0.	.00
1993/06/21 01:00	-1.13	0.	.00
1993/06/21 01:30	-1.13	0.	.00
1993/06/21 02:00	-1.13	0.	.00
1993/06/21 02:30	-1.13	0.	.00
1993/06/21 03:00	-1.13	0.	.00
1993/06/21 03:30	-1.13	0.	.00
1993/06/21 04:00	-.59	307.	.59
1993/06/21 04:30	.51	236.	.50
1993/06/21 05:00	1.33	238.	.39
1993/06/21 05:30	2.12	240.	.28
1993/06/21 06:00	2.63	232.	.20

1993/06/21 06:30	2.96	227.	.17
1993/06/21 07:00	3.21	230.	.10
1993/06/21 07:30	3.26	64.	.04
1993/06/21 08:00	2.99	59.	.14
1993/06/21 08:30	2.61	56.	.16
1993/06/21 09:00	2.13	60.	.22
1993/06/21 09:30	1.55	58.	.30
1993/06/21 10:00	.93	55.	.37
1993/06/21 10:30	.32	56.	.43
1993/06/21 11:00	-.28	52.	.45
1993/06/21 11:30	-.84	46.	.49
1993/06/21 12:00	-1.13	0.	.00
1993/06/21 12:30	-1.13	0.	.00
1993/06/21 13:00	-1.13	0.	.00
1993/06/21 13:30	-1.13	0.	.00
1993/06/21 14:00	-1.13	0.	.00
1993/06/21 14:30	-1.13	0.	.00
1993/06/21 15:00	-1.13	0.	.00
1993/06/21 15:30	-1.13	0.	.00
1993/06/21 16:00	-1.13	0.	.00
1993/06/21 16:30	-.26	299.	.53
1993/06/21 17:00	.79	231.	.52
1993/06/21 17:30	1.61	240.	.36
1993/06/21 18:00	2.38	236.	.25
1993/06/21 18:30	2.83	230.	.20
1993/06/21 19:00	3.13	228.	.16
1993/06/21 19:30	3.36	235.	.06
1993/06/21 20:00	3.28	59.	.09
1993/06/21 20:30	2.93	59.	.14
1993/06/21 21:00	2.54	60.	.18
1993/06/21 21:30	2.00	58.	.25
1993/06/21 22:00	1.38	57.	.33
1993/06/21 22:30	.77	55.	.39
1993/06/21 23:00	.14	56.	.44
1993/06/21 23:30	-.45	51.	.47

MODEL PREDICTIONS FOR 1993 6 22

All times are in G.M.T.

52 56 N 0 06 E

DATE AND TIME	HEIGHT (m)	SET (degT)	RATE (m/s)
1993/06/22 00:00	-.99	45.	.50

1993/06/22 00:30	-1.13	0.	.00
1993/06/22 01:00	-1.13	0.	.00
1993/06/22 01:30	-1.13	0.	.00
1993/06/22 02:00	-1.13	0.	.00
1993/06/22 02:30	-1.13	0.	.00
1993/06/22 03:00	-1.13	0.	.00
1993/06/22 03:30	-1.13	0.	.00
1993/06/22 04:00	-1.13	0.	.00
1993/06/22 04:30	-1.13	0.	.00
1993/06/22 05:00	.09	279.	.46
1993/06/22 05:30	1.06	232.	.48
1993/06/22 06:00	1.89	242.	.33
1993/06/22 06:30	2.58	232.	.23
1993/06/22 07:00	2.99	227.	.20
1993/06/22 07:30	3.28	229.	.14
1993/06/22 08:00	3.45	215.	.01
1993/06/22 08:30	3.25	57.	.12
1993/06/22 09:00	2.86	57.	.15
1993/06/22 09:30	2.42	61.	.20
1993/06/22 10:00	1.84	58.	.28
1993/06/22 10:30	1.20	56.	.35
1993/06/22 11:00	.59	56.	.41
1993/06/22 11:30	-.04	54.	.45
1993/06/22 12:00	-.63	49.	.49
1993/06/22 12:30	-1.13	0.	.00
1993/06/22 13:00	-1.13	0.	.00
1993/06/22 13:30	-1.13	0.	.00
1993/06/22 14:00	-1.13	0.	.00
1993/06/22 14:30	-1.13	0.	.00
1993/06/22 15:00	-1.13	0.	.00
1993/06/22 15:30	-1.13	0.	.00
1993/06/22 16:00	-1.13	0.	.00
1993/06/22 16:30	-1.13	0.	.00
1993/06/22 17:00	-.81	308.	.65
1993/06/22 17:30	.43	248.	.47
1993/06/22 18:00	1.32	235.	.43
1993/06/22 18:30	2.16	242.	.30
1993/06/22 19:00	2.74	233.	.22
1993/06/22 19:30	3.11	227.	.19
1993/06/22 20:00	3.39	231.	.11
1993/06/22 20:30	3.46	63.	.05
1993/06/22 21:00	3.16	59.	.14
1993/06/22 21:30	2.77	57.	.16
1993/06/22 22:00	2.27	60.	.23
1993/06/22 22:30	1.66	58.	.31
1993/06/22 23:00	1.02	55.	.37

1993/06/22 23:30 .40 56. .43

MODEL PREDICTIONS FOR 1993 6 23

All times are in G.M.T.
52 56 N 0 06 E

DATE AND TIME	HEIGHT (m)	SET (degT)	RATE (m/s)
1993/06/23 00:00	-.22	53.	.46
1993/06/23 00:30	-.79	47.	.50
1993/06/23 01:00	-1.13	0.	.00
1993/06/23 01:30	-1.13	0.	.00
1993/06/23 02:00	-1.13	0.	.00
1993/06/23 02:30	-1.13	0.	.00
1993/06/23 03:00	-1.13	0.	.00
1993/06/23 03:30	-1.13	0.	.00
1993/06/23 04:00	-1.13	0.	.00
1993/06/23 04:30	-1.13	0.	.00
1993/06/23 05:00	-1.13	0.	.00
1993/06/23 05:30	-.43	303.	.59
1993/06/23 06:00	.72	233.	.53
1993/06/23 06:30	1.57	238.	.38
1993/06/23 07:00	2.39	237.	.27
1993/06/23 07:30	2.88	231.	.21
1993/06/23 08:00	3.21	228.	.18
1993/06/23 08:30	3.46	235.	.07
1993/06/23 09:00	3.40	60.	.09
1993/06/23 09:30	3.05	60.	.14
1993/06/23 10:00	2.65	60.	.17
1993/06/23 10:30	2.10	59.	.25
1993/06/23 11:00	1.47	57.	.33
1993/06/23 11:30	.84	55.	.39
1993/06/23 12:00	.22	56.	.44
1993/06/23 12:30	-.39	51.	.47
1993/06/23 13:00	-.95	45.	.51
1993/06/23 13:30	-1.13	0.	.00
1993/06/23 14:00	-1.13	0.	.00
1993/06/23 14:30	-1.13	0.	.00
1993/06/23 15:00	-1.13	0.	.00
1993/06/23 15:30	-1.13	0.	.00
1993/06/23 16:00	-1.13	0.	.00
1993/06/23 16:30	-1.13	0.	.00
1993/06/23 17:00	-1.13	0.	.00

1993/06/23 17:30	-1.13	0.	.00
1993/06/23 18:00	-.05	290.	.50
1993/06/23 18:30	.97	231.	.50
1993/06/23 19:00	1.81	241.	.34
1993/06/23 19:30	2.54	233.	.24
1993/06/23 20:00	2.97	228.	.21
1993/06/23 20:30	3.27	229.	.15
1993/06/23 21:00	3.47	234.	.03
1993/06/23 21:30	3.29	57.	.11
1993/06/23 22:00	2.91	58.	.14
1993/06/23 22:30	2.49	61.	.19
1993/06/23 23:00	1.92	58.	.27
1993/06/23 23:30	1.28	56.	.34

MODEL PREDICTIONS FOR 1993 6 24

All times are in G.M.T.

52 56 N 0 06 E

DATE AND TIME	HEIGHT (m)	SET (degT)	RATE (m/s)
1993/06/24 00:00	.66	55.	.41
1993/06/24 00:30	.04	55.	.44
1993/06/24 01:00	-.55	49.	.48
1993/06/24 01:30	-1.08	0.	.00
1993/06/24 02:00	-1.13	0.	.00
1993/06/24 02:30	-1.13	0.	.00
1993/06/24 03:00	-1.13	0.	.00
1993/06/24 03:30	-1.13	0.	.00
1993/06/24 04:00	-1.13	0.	.00
1993/06/24 04:30	-1.13	0.	.00
1993/06/24 05:00	-1.13	0.	.00
1993/06/24 05:30	-1.13	0.	.00
1993/06/24 06:00	-.98	309.	.67
1993/06/24 06:30	.28	261.	.45
1993/06/24 07:00	1.19	234.	.45
1993/06/24 07:30	2.02	242.	.31
1993/06/24 08:00	2.63	232.	.22
1993/06/24 08:30	3.01	226.	.19
1993/06/24 09:00	3.28	229.	.12
1993/06/24 09:30	3.41	70.	.02
1993/06/24 10:00	3.16	58.	.13
1993/06/24 10:30	2.77	56.	.15
1993/06/24 11:00	2.31	61.	.21

1993/06/24 11:30	1.73	58.	.29
1993/06/24 12:00	1.10	56.	.36
1993/06/24 12:30	.49	56.	.42
1993/06/24 13:00	-.13	54.	.45
1993/06/24 13:30	-.71	48.	.49
1993/06/24 14:00	-1.13	0.	.00
1993/06/24 14:30	-1.13	0.	.00
1993/06/24 15:00	-1.13	0.	.00
1993/06/24 15:30	-1.13	0.	.00
1993/06/24 16:00	-1.13	0.	.00
1993/06/24 16:30	-1.13	0.	.00
1993/06/24 17:00	-1.13	0.	.00
1993/06/24 17:30	-1.13	0.	.00
1993/06/24 18:00	-1.13	0.	.00
1993/06/24 18:30	-.59	307.	.61
1993/06/24 19:00	.55	236.	.50
1993/06/24 19:30	1.38	237.	.39
1993/06/24 20:00	2.18	240.	.28
1993/06/24 20:30	2.69	232.	.20
1993/06/24 21:00	3.01	227.	.17
1993/06/24 21:30	3.25	231.	.10
1993/06/24 22:00	3.29	63.	.06
1993/06/24 22:30	2.99	59.	.14
1993/06/24 23:00	2.62	57.	.16
1993/06/24 23:30	2.13	59.	.23