A VARIABLE-BOUNDARY NUMERICAL TIDAL MODEL

APPROVED:

Chairman

Department Head

APPROVED: Date Dec. 5, 1969

Dean of the College of Mathematics, Physical Sciences and Engineering.

Vice President for Research and Advanced Study

A VARIABLE-BOUNDARY NUMERICAL TIDAL MODEL

A

THESIS

Presented to the Faculty of the
University of Alaska in Partial Fulfillment
of the Requirements
for the Degree of
MASTER OF SCIENCE

By
John Christian Hartley Mungall, B.Sc., M.S.
College, Alaska

May, 1970

ABSTRACT

A numerical tidal model using equations developed by Hansen (1952) and Yuen (1967) is automated to the point where a potential user need not undertake extensive reprogramming. The user adds to the program only those cards needed to specify tides at input points as a function of time; the application of the relevant calculations at each grid point being controlled by an integer matrix that corresponds to the inlet boundary.

A sample problem is covered in detail and applications of the model to the M_2 tide of the Gulf of California, and to a hypothetical mean tide in Cook Inlet are shown.

ACKNOWLEDGEMENTS

I would like to express appreciation to my committee chairman, Dr. J. B. Matthews, for his encouragement during the course of this work. In addition, I wish to thank the University of Alaska for financial support.

This study was supported by the Office of Naval Research under contract NONR 3010(05) .

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

To trace the origins of tidal modeling one has to follow the history of the tides through some two thousand years. In the Occident the earliest references to tides are those of Strabo, Pliny, and Pytheas, in the first century A.D.. Such references are understandably rare as the Mediterranean is a region of small tides. The connection between tidal variations and the movement of the sun and moon being obvious, it is not surprising that some rule-of-thumb methods for tidal prediction were found and passed from father to son as closely guarded family secrets. It was not until the seventeenth century, however, that mathematics was applied to the study of tides.

Kepler, with his studies on gravitational effects, provided Newton with the basis for his equilibrium tide theory. This theory explained mathematically such effects as spring and neap tides, priming and lagging, and diurnal inequalities. Newton's theory assumed a non-inertial fluid, the particles of which instantly respond to the attractional forces of the sun and moon. Daniel Bernoulli, with his studies on the mathematics of fluids, paved the way for Laplace who formulated and applied the equations of continuity and motion to the world ocean, and demonstrated the need for harmonic tidal analysis.

The harmonic analysis of tidal records was established by Thomson (later Lord Kelvin), and in 1876 he introduced the first tide predicting machine. Further improvements in the practice of harmonic analysis were made by G. Darwin and Doodson. A new approach to tidal analysis and prediction appeared in 1965 when Munk and Cartwright presented a paper on tidal spectroscopy and prediction. This technique, the so-called "response method", allows the inclusion of input functions other than gravitational forces.

With the harmonic method well established, analytical studies were made on the dynamics of water movement in canals and oceans. With these studies are associated such names as Airy, Kelvin, Lamb, Poincaré, Rayleigh, Taylor, Jeffreys, Proudman, and others. The first actual model (as opposed to analytical solutions) appears to be one on the Red Sea by Blondel (1912), based on the calculus of variations. Efforts were then directed by people such as Sterneck (1914), Defant (1920), Grace (1936), and Proudman (1953), to models involving the numerical solution of the equations of motion and continuity from which the time dependency has been removed. During this period all calculations had to be performed by hand. Considerable advances in the calculation of water movements in rivers and canals were made by the Dutch, who tended more towards solutions of a mathematical nature as opposed to numerical solutions. The post-war advent of the digital computer made feasible the time-dependent solution of the hydrodynamic equations. The result of the withdrawing of the time-dependency restriction was to allow solutions of a non-linear nature to be obtained. This is particularly desirable when tides in shallow waters are being studied. Furthermore the computer made possible calculations in two dimensions, so that cross-currents and Coriolis force effects could be included.

The first application of a two-dimensional tidal model was to the North Sea (Hansen, 1952). A further application of Hansen's explicit technique was made by Yuen (1967) to the tides of the Bay of Fundy. Both these models were, however, specifically tailored to the area being studied and were not general, i.e. the model could not conveniently be applied to other areas. This situation showed an obvious need for a variable-geometry model that could be adapted to new outlines without extensive reprogramming.

A sophisticated model of variable-geometry nature was devised by Leendertse (1967). It is based on the implicit method, which is considerably more complicated than the explicit method on account of the need for the solution of sets of simultaneous equations at each time step. It is felt that the approach used in this model is too complex for the method to be easily understood (and hence modified if desired) by users not possessing a strong background in the techniques of numerical models. In the past the users of two-dimensional tidal models seem to have been physical oceanographers or possibly civil engineers. A need now exists for a model that is not only capable of handling variable geometries, but that is also conceptually simple, well documented, and easy to use. On these points it is felt that Leendertse's model falls short of the ideal.

In the chapters that follow, a model is developed that uses Yuen's equations in an automated form. The equations are applied as necessary by a process that monitors an integer matrix based on the positions of the inlet boundaries.

The prospective user is warned that certain stability criteria must be adhered to during the computations. These are covered in Chapters II and III.

CHAPTER II

THE CALCULATION OF TIDES IN INLETS

1. Introduction--.

The prediction of tides of an astronomical origin at points close to deep seas and oceans is now, within specified limits, a routine matter. However the problem becomes more complicated when attention is turned towards shallow semi-enclosed coastal areas (henceforth referred to as inlets).

Statistical methods are now in existence that seem to be adequate for the prediction of tides in inlets, provided that long-term records are available. If it is desired that the effects of storms and changes in local topography (land reclamation, shipping channels and canals, hydro-electric projects, etc.) are to be reliably forecasted then the approach must generally involve the solution of the basic hydrodynamic equations. The simplified equations of continuity and motion are, for one dimension, from Proudman (1953):

$$\frac{\partial (Au)}{\partial x} + b\frac{\partial h}{\partial t} = 0 \qquad (2.1)$$

and

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + g\frac{\partial h}{\partial x} + \sum F_i = 0, \qquad (2.2)$$

where:

x=distance	b=width
A=cross-sectional area	u=velocity
h=total water depth	t=time
g=gravitational acceleration	F_i=i^{th} force .

The equations to be solved are further simplified by assuming homogeneous flow of a long wave nature (shallow water wave), except for the case of the tidal bore. They are complicated by the inclusion of a frictional term that is essentially non-linear. The term $u\frac{\partial u}{\partial x}$ is normally neglected as being small in comparison with the other terms.

When shallow water waves are being considered, the wave motion is generally assumed to be such that the vertical accelerations and velocities are negligible, i.e. the orbital motions of particles in the vertical plane are no longer circular or elliptical as with deep water waves. Once it has been assumed that the velocity vector is restricted to lie only in the horizontal plane, the depth mean velocity can be used. If vertical current profiles for a given region are available then it may be that the mean current can be extrapolated to provide a prediction for the overall current profile.

The effect of friction is included in the equations of motion via the application of the formulae of De Chezy (in Europe) or Manning (in the United States) which were developed for the study of uniform flow in channels. When the inlet is wide compared to its depth (say, in a ratio of 10:1) it is customary to use for the frictional force per unit mass

$$F = \frac{g u |u|}{C^2 h} , \qquad (2.3)$$

where C=De Chezy's coefficient,

which makes the friction opposite in direction to the current. In the m.k.s. system C is approximately equal to 50 meters$^{\frac{1}{2}}$ sec^{-1} , so that

$$F \simeq \frac{0.004 \; u|u|}{h} \quad . \qquad (2.4)$$

The above-mentioned equations, (2.1) and (2.2), may be dealt with in three main ways: harmonic methods, characteristic methods, and finite difference methods. For the purposes of background each method will be covered in some detail in the sections that follow.

2. Harmonic methods--.

By the use of Fourier series, the tide is divided up into various constituents whose periods result from the relative motions of the earth, sun, and moon. The equations are linearised (Lorentz, 1926) by neglecting the convection term $u\frac{\partial u}{\partial x}$ and by replacing the friction term by

$$F = \frac{g}{C^2 h} \frac{8}{3\pi} u \bar{u} \quad , \qquad (2.5)$$

where $\bar{u}$=maximum amplitude of current,

and the solutions for height and current are assumed sinusoidal. The time-dependence of the equations may then be removed, leaving a pair of simultaneous linear partial differential equations. It is however necessary to estimate the maximum amplitude of the current at the start of the calculations. The method becomes considerably more complicated when more than one constituent is considered at a time.

The simplest example of a harmonic type calculation is that of the solution of the tides in an inlet of constant cross section (Sverdrup, Johnson, and Fleming, 1942). If convective and frictional terms are neglected, the equations of motion and continuity become

$$\frac{\partial u}{\partial t} + g\frac{\partial z}{\partial x} = 0 \qquad (2.6)$$

and

$$\frac{\partial z}{\partial t} + d\frac{\partial u}{\partial x} = 0 \, , \qquad (2.7)$$

where Z=height above mean sea level
d=depth of water below mean sea level.

If the solution is assumed to vary sinusoidally with time,

$$z = \bar{z}\,\sin\left(\frac{2\pi t}{T}\right) \qquad (2.8)$$

and

$$u = \bar{u}\,\cos\left(\frac{2\pi t}{T}\right) , \qquad (2.9)$$

where $\bar{Z}$=maximum amplitude of tide
T=period of tide,

and these quantities are substituted into (2.6) and (2.7), then

$$-\bar{u}\,\frac{2\pi}{T} + g\frac{\partial \bar{z}}{\partial x} = 0 \qquad (2.10)$$

and

$$\bar{z}\,\frac{2\pi}{T} + d\,\frac{\partial \bar{u}}{\partial x} = 0 \, . \qquad (2.11)$$

This leads to

$$\bar{z} = B\,\cos\left(\frac{2\pi x}{\mathcal{L}}\right) , \qquad (2.12)$$

where $\mathcal{L} = T\sqrt{gd}$
B=constant (to be determined).

Thus

$$z = B\cos\left(\frac{2\pi x}{\mathcal{L}}\right)\sin\left(\frac{2\pi t}{T}\right) \tag{2.13}$$

and

$$u = -\frac{Bg}{\sqrt{gd}}\sin\left(\frac{2\pi x}{\mathcal{L}}\right)\cos\left(\frac{2\pi t}{T}\right). \tag{2.14}$$

If x=0 at the closed end of the inlet and the tide is specified at x=L (with the maximum amplitude of the tide being H) then

$$z = \frac{H}{\cos\left(\frac{2\pi L}{\mathcal{L}}\right)}\cos\left(\frac{2\pi x}{\mathcal{L}}\right)\sin\left(\frac{2\pi t}{T}\right) \tag{2.15}$$

and

$$u = \frac{-Hg}{\sqrt{gd}\cos\left(\frac{2\pi L}{\mathcal{L}}\right)}\sin\left(\frac{2\pi x}{\mathcal{L}}\right)\cos\left(\frac{2\pi t}{T}\right). \tag{2.16}$$

Equation (2.15) shows clearly that nodes, or points of zero tidal amplitude, can exist whenever $x=\mathcal{L}(2n+1)/4$, n=0,1,2,.... . Furthermore, infinite tidal amplitudes will result should $L=\mathcal{L}(2n+1)/4$, n=0,1,2,...., i.e. whenever a node coincides with the mouth of the inlet. Practically, of course, friction will limit the infinite amplitudes; nevertheless, considerable amplification of a tidal constituent can occur should the length of the inlet be near one of its resonant lengths for that particular period.

For a comprehensive presentation of the method, the reader is directed to the book by Dronkers (1964).

3. Characteristic methods--.

The material in this section was taken chiefly from the book by Stoker (1957).

The equations of continuity and motion, (2.1) and (2.2), (neglecting all forces other than hydrostatic) may be rewritten in terms of the variables u and c (where $c=\sqrt{gh}$). Two ordinary differential equations result:

$$C_1: \frac{dx}{dt} = u+c, \quad \text{with} \quad u+2c=k_1 \quad \text{for a given curve ,} \tag{2.17}$$

and

$$C_2: \frac{dx}{dt} = u-c, \quad \text{with} \quad u-2c=k_2 \quad \text{for a given curve.} \tag{2.18}$$

These equations represent two sets of curves on the x-t plane: the set C_1 being referred to as 'forward characteristics' and the set C_2 as 'backward characteristics'. The equations are written for a point moving relative to the bottom. If the axis is shifted to a point (x_1, t_1) moving with constant velocity $V(x_1, t_1)$, then C_1 and C_2 become:

$$\frac{dx}{dt} = \pm c \, . \tag{2.19}$$

The importance of this is that the process may now be seen to be one of the propagation of disturbances away from the point in question with a velocity, or celerity, c.

The characteristic method is particularly useful when aperiodic conditions exist (storm surges, dam failures, lock closures, etc.), and for situations where the flow becomes critical or supercritical, i.e. $u \geqslant \sqrt{gh}$. This situation is similar to supersonic flow in gases. In water the phenomenon is associated with hydraulic jumps and tidal bores. It should be mentioned that the characteristic method itself cannot deal with the discontinuity region. However, it is useful for indicating the time and place of occurence of the bore, and the conditions on either side of the discontinuity. The reason for this is that at the actual discontinuity the above equations break down owing to the existence of energy losses and vertical accelerations. As far as the practicality of calculations is concerned, the characteristic method is too complicated for most exploratory calculations, but is of greater interest when certain complicated situations are to be analysed. A further use of characteristic theory is to indicate the sufficiency of boundary conditions for a given problem.

The basic approach by which the method of characteristics is used to solve a simple initial value problem, in which the depth is constant, is as follows; If u and c ($c=\sqrt{g(d+z)}$) are known for points A and B, then the slopes of the characteristics through these points are known from

$$\frac{dx}{dt} = u \pm c \quad . \qquad (2.20)$$

If the distance AB is small the curved characteristics may be approximated by straight lines. When the forward characteristic through A and the backward characteristic through B are drawn, they will intersect at Q, as in Figure 2.1.

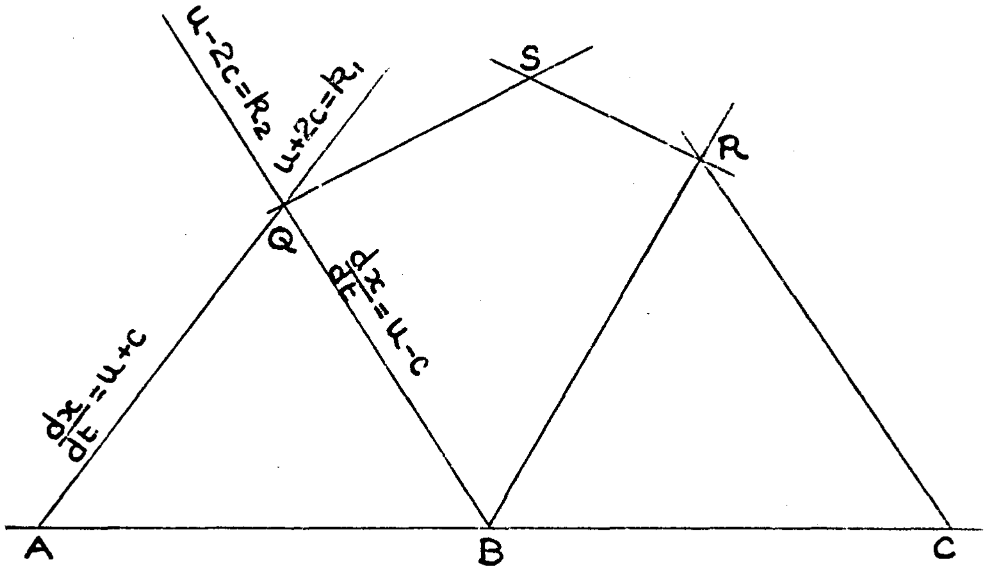


Figure 2.1. Part of characteristic net.

With the initial conditions known, it is also possible to evaluate the constants k_1 and k_2. Therefore two equations may be solved to give the values of u and c at Q. Similarly, points R and S may be found, and so on for the network, provided that the boundaries are at infinity.

It is important to note that conditions at S are influenced by conditions between A and C. The area SAC is known as the zone of determinacy of S. In most cases of interest it is necessary to include the effects of boundaries. Suppose a left-hand boundary exists at x=a (see Figure 2.2). A backward characteristic from B is assumed to intersect the t-axis at (a,τ) and hence if both u and c were known at B, then k_2 is known. Thus at (a,τ) we have

$$u(a,\tau) - 2.c(a,\tau) = k_2 \tag{2.21}$$

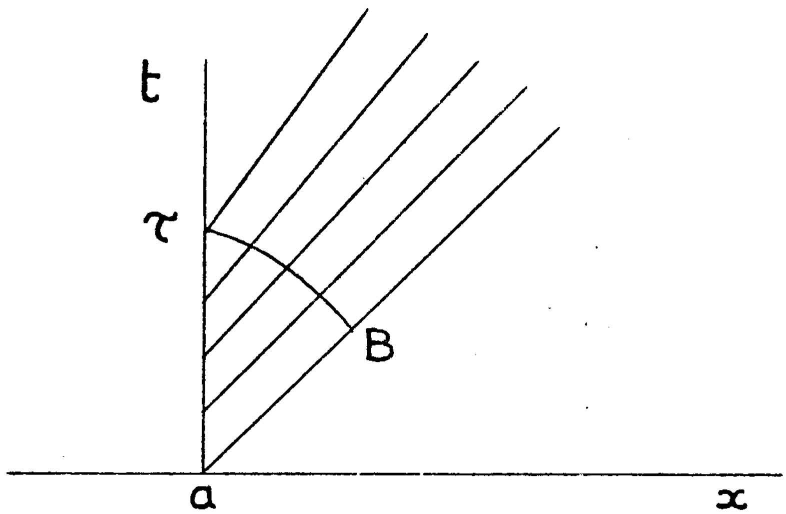


Figure 2.2. Characteristics at a boundary (x=a).

To evaluate the slope of the forward characteristic through (a,τ) it is necessary to evaluate

$$\frac{dx}{dt} = u(a,\tau) + c(a,\tau) \tag{2.22}$$

and

$$k_1 = u(a,\tau) + 2.c(a,\tau) \tag{2.23}$$

Using (2.21), (2.22) and (2.23) may be written in two ways:

$$\frac{dx}{dt} = 3.c(a,\tau) + k_2 ; \quad k_1 = k_2 + 4.c(a,\tau) \tag{2.24}$$

and

$$\frac{dx}{dt} = \frac{3}{2}.u(a,\tau) - \frac{k_2}{2} ; \quad k_1 = 2.u(a,\tau) - k_2 \tag{2.25}$$

Thus if either $u(a,\tau)$ or $c(a,\tau)$ (where c is a function of z) are known, the forward characteristic through (a,τ) may be drawn. We therefore reach the important conclusion that it is only necessary to specify height or current,but not both, at a boundary. It has been tacitly assumed so far that the backward characteristic through B does indeed intersect the t-axis, i.e. that

$$[u(a,\tau) - c(a,\tau)] < 0$$

or

$$u(a,\tau) < \sqrt{gh} \qquad (2.26)$$

If $u(a,\tau)$ is greater than $\sqrt{gh}$ there will be no intersection, and hence to draw the forward characteristic through (a,τ), both $u(a,\tau)$ and $c(a,\tau)$ must be specified. Such disturbances can not propagate to the left, and so conditions at x=a will not propagate downstream. This flow is said to be supercritical, or in the case of a gas, supersonic.

A major difficulty of the characteristic method is also evident from the above discription. If values of u and c are required at equi-spaced intervals in time and space, it is necessary to carry out a series of interpolations.

One further case of interest is one that can arise when a disturbance is propagated into lower-lying water. If the forward characteristics should intersect, as in Figure 2.3, with the first intersection at I, a situation is encountered wherein two different heights exist at the same point, i.e. a bore or a hydraulic jump has formed.

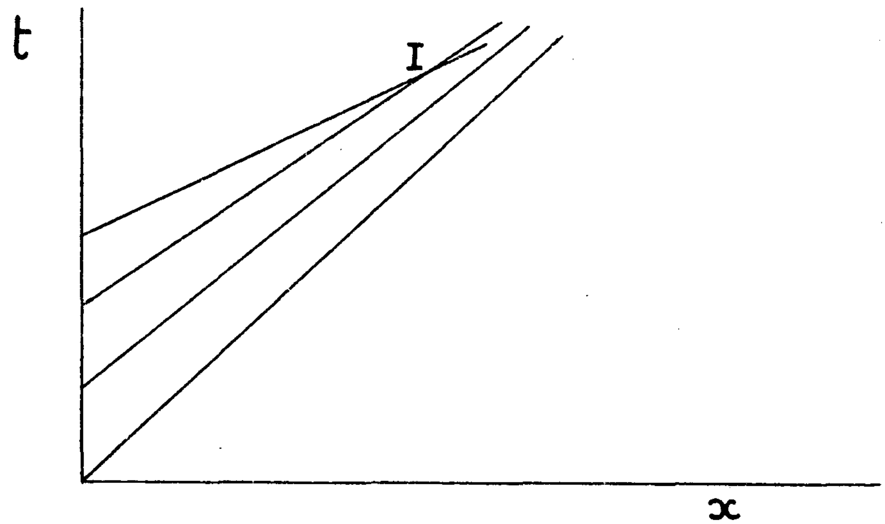


Figure 2.3. A set of intersecting forward characteristics.

For this point I, and all others lying within the forward and backward characteristics from a point just before I, calculations are no longer possible using this theory alone. A theory involving shock fronts must be used.

4. Finite difference methods--.

The various quantities in the equations of motion and continuity are replaced by their forward, centered, off-centered, or backward finite difference equivalents (these in turn being derived from Taylor series expansions). A time-space grid is prepared and the components of the finite difference equations are evaluated at the grid intersections. The solution of the finite difference equations must be stable. Thus the solution must approach the true solution of the original equations (as evaluated at the grid points) as the mesh size approaches zero. Unfortunately this is not always guaranted, so it is necessary to concern oneself with establishing the stability criteria (generally involving the time step τ, the distance increment ℓ, and the velocity of propagation of the disturbance c) for each proposed finite difference scheme.

Following the procedure of Richtmyer and Morton (1967), difference quotients are introduced in the following manner.

$$\frac{\partial z}{\partial x} = (1-\theta)\frac{\left(Z^{\tau}_{m+1} - Z^{\tau}_{m}\right)}{\ell} + \theta\,\frac{\left(Z^{\tau}_{m} - Z^{\tau}_{m-1}\right)}{\ell} \tag{2.27}$$

where $Z^{\tau}_{m} = Z[m\ell, r\tau]$, m and r integer counting indices that correspond to grid lines (see Figure 2.4), and $0 \leq \theta \leq 1$.

The difference quotient is termed forward, centered, or backward if $\theta = 0$, 1/2, or 1 respectively. Using such methods the equations of motion and continuity may be rewritten in finite difference form in several ways. In the discussion of the two schemes that follow, considerable use was made of the report by Leendertse (1967).

The Leap Frog method

The first example of a finite difference scheme that will be discussed is the so-called leap frog method. It is an example of a staggered grid. Using the following simplified equations of continuity and motion,

$$\frac{\partial z}{\partial t} + h\frac{\partial u}{\partial x} = 0 \tag{2.28}$$

and

$$\frac{\partial u}{\partial t} + g\frac{\partial z}{\partial x} = 0 , \tag{2.29}$$

the finite difference equations are written as

$$\frac{Z_m^{r+1} - Z_m^{r-1}}{2\tau} + h\,\frac{U_{m+1}^{r} - U_{m-1}^{r}}{2\ell} = 0 \tag{2.30}$$

and

$$\frac{U_{m+1}^{r+2} - U_{m+1}^{r}}{2\tau} + g\,\frac{Z_{m+2}^{r+1} - Z_m^{r+1}}{2\ell} = 0 . \tag{2.31}$$

On the time-space grid, the grid points concerned are shown in Figure 2.4.

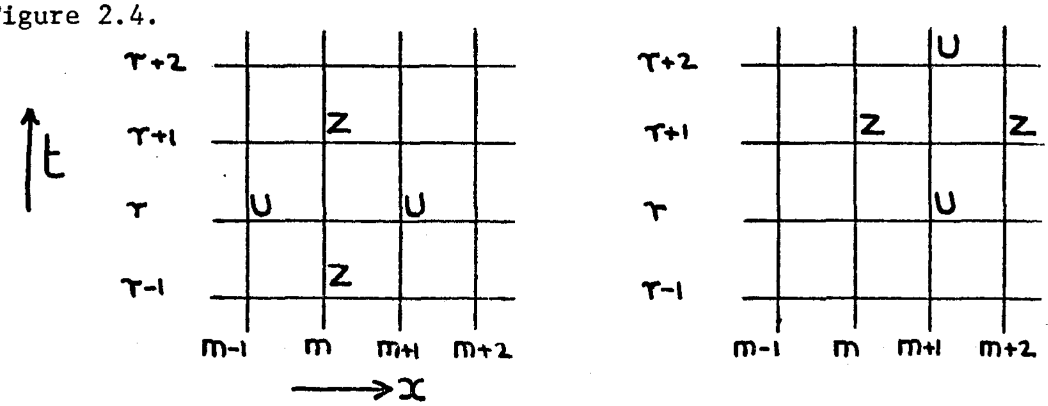


Figure 2.4. Grid points used in the leap frog method.

If m and r are taken as being odd, it will be seen that heights are calculated at even-numbered time steps and odd-numbered space steps, while currents are calculated at odd time steps and even space steps. For an inlet whose open end is on column 1, and closed end on column 10, the order in which the calculations are performed is as follows. The normal routine will be to calculate all the Z's along a particular grid row, to assign Z_1 equal to the value of the tide height corresponding to that particular time step, and to assign $U_{10} = 0$; Z_1 and U_{10} are thus boundary conditions. To initiate the computations (the calculation of Z_3^2 , Z_5^2 ,....., Z_9^2) it is necessary to supply initial conditions for Z along row 0, and for U along row 1. For calculations concerned with inlets it is convenient to start the calculations at a time corresponding to high tide at the mouth of the inlet. In this situation the currents will all be zero if a standing wave solution is assumed ((2.15) and (2.16)) and the initial tide heights may be estimated or obtained from a simple calculation of the harmonic type. So far no preparatory check has been made as to whether the scheme will be stable.

One way of approaching the investigation of stability is to assume a particular error wave at a given time step. The wave may then be represented by a Fourier series composed of terms such as

$$U = U^* e^{i\beta t} e^{i\sigma x} \tag{2.32}$$

and

$$Z = Z^{*} e^{i\beta t} e^{i\sigma x} \tag{2.33}$$

where β =wave frequency
σ =wave number
U^{*}, Z^{*}=Fourier series components.

If a linear system such as the above is being examined, only one term of the Fourier series need be investigated. As the solution is only valid at certain grid points, we assume that

$$U = U^{*} e^{i\beta r\tau} e^{i\sigma m\ell} \tag{2.34}$$

and

$$Z = Z^{*} e^{i\beta r\tau} e^{i\sigma m\ell} . \tag{2.35}$$

When equations (2.34) and (2.35) are substituted into the finite difference equations (2.30) and (2.31), the following equation results;

$$\left[e^{i\beta\tau}\right]^{2} - 2 + 4\frac{\tau^{2}}{\ell^{2}} gh.\sin^{2}(\sigma\ell) + \left[e^{i\beta\tau}\right]^{-2} = 0 . \tag{2.36}$$

Putting

$$b = 1 - 2\frac{\tau^{2}}{\ell^{2}} gh.\sin^{2}(\sigma\ell) , \tag{2.37}$$

we get

$$\left(e^{i\beta\tau}\right) = \pm\left(b \pm \sqrt{b^{2}-1}\right)^{1/2} = \lambda_{1,2,3,4} . \tag{2.38}$$

The requirement for stability is that $|\lambda| \leq 1$. It therefore follows that the stability condition for this scheme is

$-1 \leq b \leq 1$, or

$$\left(\frac{\sqrt{gh}}{\ell/\tau}\right) < 1. \tag{2.39}$$

This stability condition must be adhered to whenever this particular finite difference scheme is used. Note that $\sqrt{gh}$ is the speed of the long, surface gravity wave, and that ℓ/τ is the maximum velocity that can be resolved by the grid. One might call the term (ℓ/τ) the grid resolution velocity (E. Berg, personal communication). Thus the stability criterion, equation (2.39), takes on a new aspect; the maximum expected velocity of propagation must be less than the grid resolution velocity for stability to be ensured.

If the above conditions for b are met, the four roots of λ will lie on the unit circle in the complex plane. This means that error waves will not tend to die out with increasing time. One way of ensuring that they do die out is to include a bottom friction term.

With the equation of motion modified to

$$\frac{\partial u}{\partial t} + g\frac{\partial z}{\partial x} + ku = 0 , \tag{2.40}$$

and using equations (2.28), (2.34), and (2.35), we get

$$i\beta Z^* + ih\sigma U^* = 0, \tag{2.41}$$

and

$$ig\sigma Z^* + (i\beta + k)U^* = 0 \tag{2.42}$$

Thus

$$\beta = \sigma\left\{ i\frac{k}{2\sigma} \pm \sqrt{gh - \left(\frac{k}{2\sigma}\right)^2} \right\}, \tag{2.43}$$

or

$$z = z^* e^{-\frac{k}{2}t} e^{\pm i\sigma\sqrt{gh-\left(\frac{k}{2\sigma}\right)^2}\,t}\, e^{i\sigma x} , \tag{2.44}$$

so that the effect of bottom friction is to decrease the amplitude of the error wave. In general, the effect of friction will be to improve stability as friction represents an energy loss.

If the four roots of λ that lie on the unit circle are closely inspected it will be seen that two of them have positive real parts and two negative. The effect of the former is to provide a term $\cos(\beta r \tau)$, which is as one would expect. The two negative ones cause a term of the type $(-1)^r \cos(\beta r \tau)$. This oscillates to positive and negative values with each consecutive time step providing a spurious solution of period 2τ modulated by a wave of period T, where T is the period of the computed wave.

An Implicit scheme

The second scheme to be considered has its finite difference equations written in the following form;

$$Z_m^{r+1} - Z_m^r + \frac{h\tau}{2\ell}\left(U_{m+1}^{r+1} - U_{m-1}^{r+1}\right) = 0 \tag{2.45}$$

and

$$U_{m+1}^{r+1} - U_{m+1}^r + \frac{g\tau}{2\ell}\left(Z_{m+2}^{r+1} - Z_m^{r+1}\right) = 0 . \tag{2.46}$$

The grid points at which quantities must be evaluated are shown in Figure 2.5.

Taking again an inlet whose length has been divided up into nine equal intervals of length ℓ, with the entrance lying on column 1 and closed end on column 10, the values that have to be calculated along each row are

Z U Z U Z U Z U Z U .

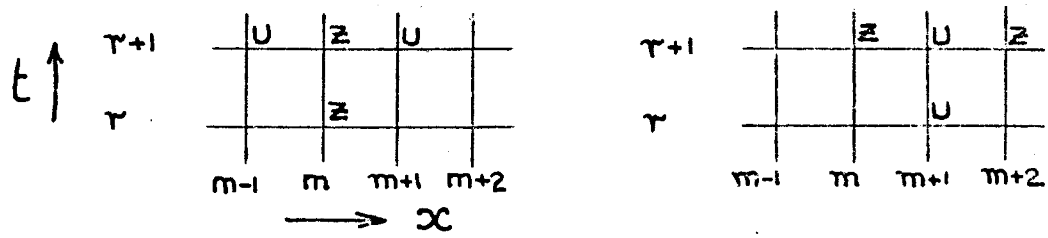


Figure 2.5. Grid points used in the implicit method.

If the values of Z and U are known at time step r, one cannot immediately calculate U_2^{r+1}, even though Z_2^{r+1} is available as a boundary condition, for it depends on Z_3^{r+1}. It is however possible to write 8 equations involving the five U^{r+1}'s and the five Z^{r+1}'s. There are only 8 unknowns as 2 of the 10 values are boundary conditions. It is thus necessary to solve 8 simultaneous equations for 8 unknowns in order to obtain all the values for time (r+1). For this reason the above system of difference equations is known as <u>implicit</u>. The equations to be solved are

$$\begin{Bmatrix} 1 & 0 & 0 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ -a & 1 & a & 0 & 0 & & & & & 0 \\ 0 & -b & 1 & b & 0 & & & & & 0 \\ 0 & 0 & -a & 1 & a & & & & & 0 \\ \cdot & \cdot & \cdot & \cdot & & & & & & \cdot \\ \cdot & \cdot & \cdot & \cdot & & & & & & \cdot \\ \cdot & \cdot & \cdot & \cdot & & & & & & \cdot \\ 0 & 0 & 0 & 0 & & & & -b & 1 & b \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{Bmatrix} \begin{Bmatrix} Z_1^{r+1} \\ U_2^{r+1} \\ Z_3^{r+1} \\ U_4^{r+1} \\ \cdot \\ \cdot \\ \cdot \\ Z_9^{r+1} \\ U_{10}^{r+1} \end{Bmatrix} = \begin{Bmatrix} 0 \\ U_2^{r} \\ Z_3^{r} \\ U_4^{r} \\ \cdot \\ \cdot \\ \cdot \\ Z_9^{r} \\ 0 \end{Bmatrix} + \begin{Bmatrix} Z_1(t) \\ 0 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ U_{10}(t) \end{Bmatrix} \tag{2.47}$$

where $a = g\tau/2\ell$, $b = h\tau/2\ell$
$Z_1(t)$=conditions at the inlet entrance
$U_{10}(t) = 0$.

The above equations may be solved by the use of an algorithm. The equation for $U_2^{\tau+1}$ is written in terms of $Z_3^{\tau+1}$ plus known quantities; $Z_3^{\tau+1}$ is written in terms of $U_4^{\tau+1}$ etc. until $Z_9^{\tau+1}$ is written in terms of $U_{10}^{\tau+1}$, which is known. The values for $Z_9^{\tau+1}$,$U_8^{\tau+1}$,..... $U_2^{\tau+1}$ may then be found in reverse order.

If a stability analysis is performed for this implicit method as was previously done for the leap frog method, it is found that

$$e^{i\beta\tau} = \frac{1 \pm i\frac{\tau}{\ell}\sqrt{gh}\,.\sin(\sigma\ell)}{1 + \frac{\tau^2}{\ell^2}gh\,.\sin^2(\sigma\ell)} , \qquad (2.48)$$

so that

$$|\lambda| = e^{-\mathrm{Im}(\beta\tau)} = \left[1 + \frac{\tau^2}{\ell^2}gh\,.\sin^2(\sigma\ell)\right]^{-1/2} . \qquad (2.49)$$

Hence $|\lambda| < 1$ for all non-trivial values of τ and ℓ , and the important fact is established that this implicit scheme is unconditionally stable.

Stability criteria based on characteristic theory

It is interesting to consider the problem of stability utilising characteristic theory (Abbott, 1966). This will often allow one to estimate stability criteria from a visual inspection of the grid layout. Considering part of a time-space grid layout for the leap frog method (in which conditions at P are calculated from a knowlege of those at A and B). the following approach may be used (see Figure 2.6.).

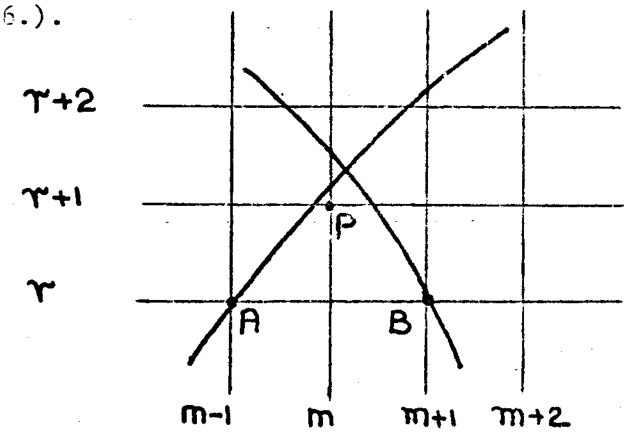


Figure 2.6. Section of time-space grid.

If AX and BY represent the forward and backward characteristics through A and B respectively, then the domain of determinacy of AB is the area bounded by AB and the lines AX and BY, i.e. any point within this region will be such that the forward and backward characteristics through it will both intersect row r between the limits A and B. For the leap frog scheme to be stable it is therefore necessary that point P lies within this zone of determinacy. As the term $u\frac{\partial u}{\partial x}$ has been neglected, the slope of the characteristics is such that

$$\frac{dx}{dt} = \pm c \quad . \tag{2.50}$$

Thus for stability

$$\frac{\tau}{\ell} < \frac{1}{c} , \tag{2.51}$$

i.e.

$$\Delta t < \frac{\Delta x}{\sqrt{gh}}, \qquad (2.52)$$

which is the same condition as that derived earlier (equation (2.39)).

When considering the second (implicit) scheme from the point of view of the method of characteristics, the reason for the unconditional stability may be seen to be due to the fact that it is possible to construct all the characteristics that intersect row (r+1), for the calculation of conditions at time (r+1) depends on the simultaneous application of conditions at time r along with boundary conditions at time (r+1).

CHAPTER III

THE FINITE DIFFERENCE EQUATIONS

1. The basic equations--.

The equations used are the same as those used by Yuen (1967) and are as follows (with axes as in Figure 3.1):

$$\frac{\partial U}{\partial t} + \gamma(U^2+V^2)^{1/2}\frac{U}{H} - fV + g\frac{\partial Z}{\partial x} = 0, \tag{3.1}$$

$$\frac{\partial V}{\partial t} + \gamma(U^2+V^2)^{1/2}\frac{V}{H} + fU + g\frac{\partial Z}{\partial y} = 0, \tag{3.2}$$

and

$$\frac{\partial (HU)}{\partial x} + \frac{\partial (HV)}{\partial y} + \frac{\partial Z}{\partial t} = 0, \tag{3.3}$$

where U=x-component of depth-mean velocity
V=y-component of depth-mean velocity
Z=vertical tide measured (positive upwards) from mean sea level
D=depth of water beneath mean sea level
H=total depth of water (H=D+Z)
γ=friction coefficient
f=Coriolis parameter (f=2Ω sin(latitude))
g=acceleration due to gravity
Ω=angular rotational speed of the earth .

The above equations will be solved by the method of finite differences. A choice exists between the two different approaches, the explicit method and the implicit method. On account of the availability of literature on the subject, it was decided that efforts would be directed to the development of a variable boundary model using the explicit method.

Although covered by Yuen, the derivation of the finite difference form of the equations will be covered in detail during the rest of the chapter. This is done so that a sound base will be available on which to base the program, and also because Yuens work contains some printing errors which are misleading.

2. The grid network--.

The grid system used is one first alluded to by Richardson (1922), and is staggered in time and space. It is thus an extension of the leap frog method. A projection of the grid onto the x-y plane can be seen in Figure 3.1.

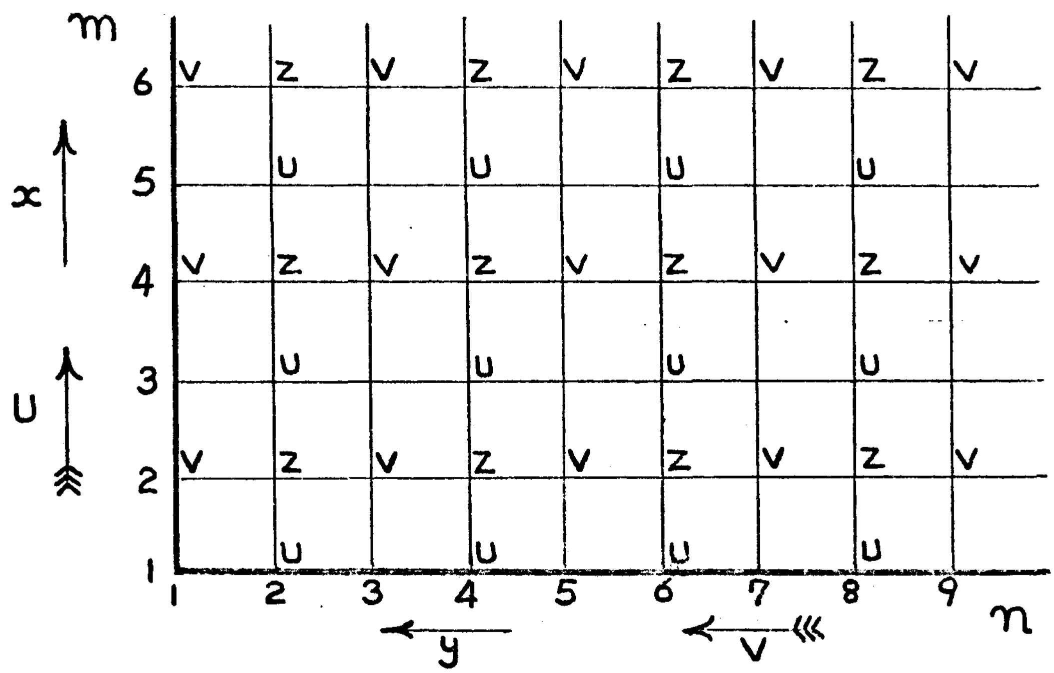


Figure 3.1. Section of staggered grid.

U and V are calculated at odd time steps, Z at even numbered steps.

3. U-point calculation--.

Equation (3.1) is first written in the form

$$\frac{\partial U}{\partial t} = -\left[\frac{U \mp (U^2+V^2)^{1/2}}{H} - fV + g\frac{\partial Z}{\partial x}\right]. \tag{3.4}$$

It is replaced by a two-point centered finite difference relation as follows;

$$\frac{\partial U^{(r)}}{\partial t} = \frac{U^{(r+1)} - U^{(r-1)}}{2\tau},$$

where the superscript r refers to time step r , and τ is the interval between time steps. In a similar fashion,

$$\frac{\partial Z(m,n)}{\partial x} = \frac{Z(m+1,n) - Z(m-1,n)}{2\ell},$$

where the subscript (m,n) refers to 'east-west' grid line m , and 'north-south' grid line n . ℓ is the interval between grid lines on the x-y plane.

It will be seen that in equation (3.4) it is necessary to have available the values of V and H at the U-point. These are estimated by interpolation from surrounding V- and Z-points (see Chapter III, section 6). To calculate U at the point (m,n), equation (3.4) is first represented in finite difference form by

$$\frac{U^{(r+1)}_{(m,n)} - U^{(r-1)}_{(m,n)}}{2\tau} = -\left[\frac{U^{(r-1)}_{(m,n)} \mp \left(U^{2\,(r-1)}_{(m,n)} + V^{2\,(r-1)}_{(m,n)}\right)^{1/2}}{H^{(r)}_{(m,n)}} - fV^{(r-1)}_{(m,n)} + g\frac{\left(Z^{(r)}_{(m+1,n)} - Z^{(r)}_{(m-1,n)}\right)}{2\ell}\right]. \tag{3.5}$$

It will be observed that in the representation of the right hand side of equation (3.4), terms U and V should have been evaluated at time step r. As U and V are calculated only at time steps (r-3), (r-1), (r+1), etc., they are approximated by taking the most recent values available, i.e. from time step (r-1). In terms of $U^{(r+1)}_{(m,n)}$, equation (4.5) can be written;

$$U^{(r+1)}_{(m,n)} = U^{(r-1)}_{(m,n)} + 2\tau \left\{ \frac{-U^{(r-1)}_{(m,n)} \mp \left(U^{2\,(r-1)}_{(m,n)} + V^{2\,(r-1)}_{(m,n)}\right)^{1/2}}{H^{(r)}_{(m,n)}} + f V^{(r-1)}_{(m,n)} - g\frac{\left(Z^{(r)}_{(m+1,n)} - Z^{(r)}_{(m-1,n)}\right)}{2\ell} \right\} . \qquad (3.6)$$

At this stage a stability factor is applied to the two leading $U^{(r-1)}_{(m,n)}$ terms (a weighted average of surrounding points);

$$\overline{U^{(r-1)}_{(m,n)}} = \alpha U^{(r-1)}_{(m,n)} + \frac{(1-\alpha)}{4}\left\{ U^{(r-1)}_{(m+1,n+1)} + U^{(r-1)}_{(m-1,n+1)} + U^{(r-1)}_{(m-1,n-1)} + U^{(r-1)}_{(m+1,n-1)} \right\} , \qquad (3.7)$$

with $0 \leqslant \alpha \leqslant 1$.

Again, the U terms within the $\{\ \}$ are all interpolated values. This stabilisation differs from that used by Yuen, in that he used only values of U calculated at U-points and not interpolated U values as in equation (3.7). The alteration has been made so that more complex boundary shapes may be dealt with without having to adjust the stabilisation process to suit the outline of the inlet, as did Yuen.

The final form of equation (3.4) before programming is thus:

$$U_{(m,n)}^{(\tau+1)} = \overline{U_{(m,n)}^{(\tau-1)}} + 2\tau \left\{ \frac{-U_{(m,n)}^{(\tau-1)} \mp \left(U^{2\,(\tau-1)}_{(m,n)} + V^{2\,(\tau-1)}_{(m,n)} \right)^{1/2}}{H_{(m,n)}^{(\tau)}} + fV_{(m,n)}^{(\tau-1)} - g\,\frac{\left(Z_{(m+1,n)}^{(\tau)} - Z_{(m-1,n)}^{(\tau)}\right)}{2\,\ell} \right\}. \tag{3.8}$$

4. V-point calculation--.

Equation (3.2) is first written in the form

$$\frac{\partial V}{\partial t} = -\left[\frac{V \mp (U^2 + V^2)^{1/2}}{H} + fU + g\,\frac{\partial Z}{\partial y} \right]. \tag{3.9}$$

In exactly the same fashion as with the finite difference evaluation of equation (3.4) , replacing -fV by +fU and $g\frac{\partial Z}{\partial x}$ by $g\frac{\partial Z}{\partial y}$, the final form of equation (3.9) is

$$V_{(m,n)}^{(\tau+1)} = \overline{V_{(m,n)}^{(\tau-1)}} + 2\tau \left\{ \frac{-V_{(m,n)}^{(\tau-1)} \mp \left(U^{2\,(\tau-1)}_{(m,n)} + V^{2\,(\tau-1)}_{(m,n)} \right)^{1/2}}{H_{(m,n)}^{(\tau)}} - fU_{(m,n)}^{(\tau-1)} - g\,\frac{\left(Z_{(m,n-1)}^{(\tau)} - Z_{(m,n+1)}^{(\tau)}\right)}{2\,\ell} \right\}, \tag{3.10}$$

with

$$\overline{V_{(m,n)}^{(\tau-1)}} = \alpha V_{(m,n)}^{(\tau-1)} + \frac{(1-\alpha)}{4}\left\{V_{(m+1,n+1)}^{(\tau-1)} + V_{(m-1,n+1)}^{(\tau-1)} + V_{(m-1,n-1)}^{(\tau-1)} + V_{(m+1,n-1)}^{(\tau-1)}\right\}. \tag{3.11}$$

It will be seen that in equation (3.10) the expression for $\frac{\partial z}{\partial x}$ is evaluated with the x-axis going from right to left. As the grid columns are numbered from left to right (see Figure 3.1.) the form of $\frac{\partial z}{\partial x}$ in equation (3.10) does not agree precisely with that of $\frac{\partial z}{\partial y}$ in equation (3.8).

5. Z-point calculation--.

Equation (3.3) is first written in the form

$$\frac{\partial z}{\partial t} = -\frac{\partial (HU)}{\partial x} - \frac{\partial (HV)}{\partial y}. \tag{3.12}$$

Equation (3.12) is then rewritten in finite difference form;

$$\frac{Z_{(m,n)}^{(\tau+2)} - Z_{(m,n)}^{(\tau)}}{2\tau} = -\frac{\left(H_{(m,n-1)}^{(\tau)} U_{(m,n-1)}^{(\tau+1)} - H_{(m,n+1)}^{(\tau)} U_{(m,n+1)}^{(\tau+1)}\right)}{2\ell} - \frac{\left(H_{(m+1,n)}^{(\tau)} U_{(m+1,n)}^{(\tau+1)} - H_{(m-1,n)}^{(\tau)} U_{(m-1,n)}^{(\tau+1)}\right)}{2\ell}. \tag{3.13}$$

It is seen that H should have been evaluated at time step (r+1).

It is approximated by making use of the value for H calculated at time step (r). The error is considered negligible, of the order of 3 cms. in (say) 20 or more meters .

Equation (3.13), written in terms of $Z^{(r+2)}_{(m,n)}$, becomes

$$Z^{(r+2)}_{(m,n)} = \overline{Z^{(r)}_{(m,n)}} - 2\tau \left\{ \frac{\left(H^{(r)}_{(m,n-1)} U^{(r+1)}_{(m,n-1)} - H^{(r)}_{(m,n+1)} U^{(r+1)}_{(m,n+1)} \right)}{2\ell} + \frac{\left(H^{(r)}_{(m+1,n)} U^{(r+1)}_{(m+1,n)} - H^{(r)}_{(m-1,n)} U^{(r+1)}_{(m-1,n)} \right)}{2\ell} \right\}, \tag{3.14}$$

where

$$\overline{Z^{(r)}_{(m,n)}} = \alpha Z^{(r)}_{(m,n)} + \frac{(1-\alpha)}{4} \left\{ Z^{(r)}_{(m+1,n)} + Z^{(r)}_{(m-1,n)} + Z^{(r)}_{(m,n-1)} + Z^{(r)}_{(m,n+1)} \right\}. \tag{3.15}$$

Notice again that the terms in the $\{\quad\}$ are interpolated. We are now left with the interpolations of V and Z at U-points, and of U and Z at V-points.

6. Interpolation of values at U- and V-points--.

In the previous sections it has been mentioned that interpolated values are necessary at U- and V-points. These are approximated by linear interpolations. A more sophisticated approximation could have been used at the expense of calculation time and of generality of the model.

a) At U-points away from boundaries (see Figure 3.2).

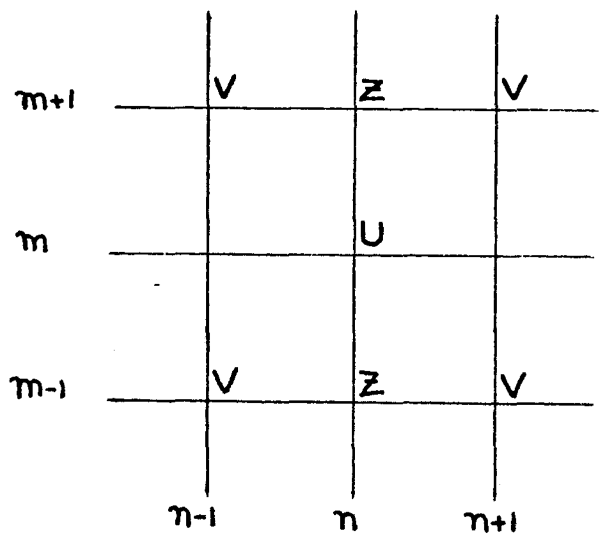


Figure 3.2. Values required for interpolations at a U-point.

$$V(m,n) = \frac{1}{4}\left\{V(m+1,n+1) + V(m-1,n+1) + V(m-1,n-1) + V(m+1,n-1)\right\}, \quad (3.16)$$

and

$$Z(m,n) = \frac{1}{2}\left(Z(m+1,n) + Z(m-1,n)\right) \quad . \quad (3.17)$$

b) At U-points lying on boundaries (see Figure 3.3).

Boundaries through U-points are always horizontal (i.e. pass through grid points of equal m). For the case of solid land lying to the 'north' of the water, V(m,n) is found by obtaining an interpolated value for V(m-1,n) and then performing a second interpolation using V(m-2,n) and V(m-1,n). Thus

$$V(m,n) = \left(V(m-1,n-1) + V(m-1,n+1)\right) - V(m-2,n) \quad . \quad (3.18)$$

It should be noted that V(m-2,n) must have been computed before equation (3.18) can be evaluated.

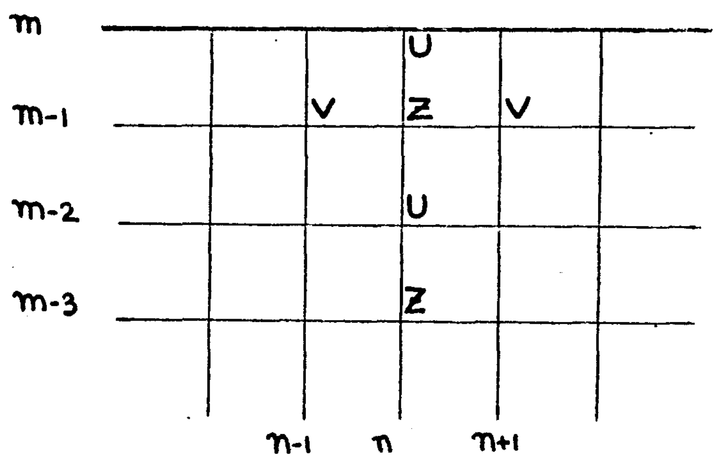


Figure 3.3. Values required for interpolations at U-points on a boundary.

Z(m,n) is found by using the values for Z(m-1,n) and Z(m-3,n):

$$Z(m,n) = 1.5\ Z(m-1,n) - 0.5\ Z(m-3,n) \quad . \qquad (3.19)$$

In a similar fashion, when land occurs to the 'south' of the water:

$$V(m,n) = \left(V(m+1,n-1) + V(m+1,n+1)\right) - V(m+2,n) , \qquad (3.20)$$

and

$$Z(m,n) = 1.5\ Z(m+1,n) - 0.5\ Z(m+3,n) \quad . \qquad (3.21)$$

c) At V-points away from boundaries.

$$U(m,n) \doteq \frac{1}{4}\left(U(m+1,n+1) + U(m-1,n+1) + U(m-1,n-1) + U(m+1,n-1)\right) \qquad (3.22)$$

$$Z(m,n) = \frac{1}{2}\left(Z(m,n+1) + Z(m,n-1)\right) \quad . \qquad (3.23)$$

d) At V-points lying on boundaries.

For the case of solid land lying to the 'west':

$$U(m,n) = \left(U(m+1,n+1) + U(m-1,n+1)\right) - U(m,n+2) \quad . \qquad (3.24)$$

$$Z(m,n) = 1.5\ Z(m,n+1) - 0.5\ Z(m,n+3) \quad . \qquad (3.25)$$

For the case of solid land lying to the 'east':

$$U(m,n) = \left(U(m+1,n-1) + U(m-1,n-1)\right) - U(m,n-2) , \qquad (3.26)$$

and

$$Z(m,n) = 1.5\ Z(m,n-1) - 0.5\ Z(m,n-3) \quad . \qquad (3.27)$$

7. Calculation for a special (narrow) case--.

Provision is made for making calculations in the case when part or all of an inlet is represented by a width of 2 ℓ. In this case there are two possibilities. The narrow axis lies 'north-south' or 'east-west'.

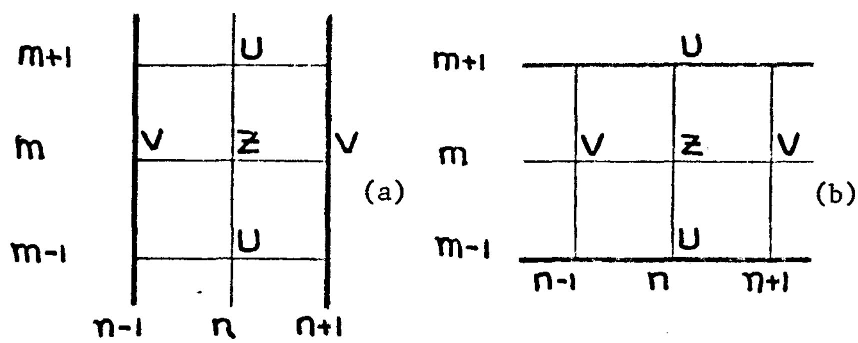


Figure 3.4. Narrow channel case.

a) 'North-south' narrow axis direction.

A situation exists here such that the problem is locally reduced to a one-dimensional situation. No cross currents exist, so that all the V's are zero and no surface slope due to Coriolis force will occur (see Figure 3.4.a). The interpolations are then

$$U(m,n-1) = U(m,n+1) = \frac{U(m+1,n) + U(m-1,n)}{2} \quad , \qquad (3.28)$$

and

$$Z(m,n-1) = Z(m,n+1) = Z(m,n) \quad . \qquad (3.29)$$

b) 'East-west' narrow axis direction.

The same type of situation exists here (see Figure 3.4.b). The interpolations become

$$V(m+1,n) = V(m-1,n) = \frac{V(m,n-1) + V(m,n+1)}{2} \quad , \qquad (3.30)$$

and

$$Z(m+1,n) = Z(m-1,n) = Z(m,n) \quad . \qquad (3.31)$$

At this point all the types of calculations necessary for the estimation of tides in an inlet are in finite difference form, if only to a certain degree of sophistication. Boundary conditions have still to be added.

Velocities normal to the boundaries are put equal to zero whenever the transition water to land occurs. Thus U=0 along 'east-west' solid boundaries (m= constant), and V=0 along 'north-south' solid boundaries (n= constant). There remains the problem of open boundaries. These occur whenever the boundaries of the model coincide with open water. In Chapter II it was shown by the method of characteristics that either height or current needs to be given as a boundary condition provided that the flow velocity is less than critical. As little is usually known about currents, it is normal to specify heights as a function of time for the various Z-points lying on open boundaries. However, in order to evaluate the bottom friction term near the open boundary, one has to know the currents along the input line. To do this, a minor assumption is made that $\frac{\partial U}{\partial x}=0$ on 'east-west' open boundaries and $\frac{\partial V}{\partial y}=0$ on 'north-south' open boundaries.

8. The finite difference equations expressed in FORTRAN IV--.

In this section mention is made only of the variable names used in the program. Details of the instructions themselves may be seen in the actual program (Appendix I).

As it was desirable to program for the greatest possible grid size compatible with a 16K single precision word memory (as then available at the University of Alaska Computer Center), an inspection was made of the matrices necessary for the performance of the calculations. The matrices first considered necessary were those for U, V, Z, H, D, and for use in a later phase of the program, an integer matrix. An inspection of the grid configuration suggested that U and V, and D and H might easily be interleaved. For this purpose, interleaving was performed in the following fashion:

V(m,n) is stored in U(m,n+1)

and D(m,n) is stored in H(m,n+1) .

Table 3.1 shows the original variables along with their corresponding array names.

Original Name	Array Name
U(m,n)	U1(M,N)
V(m,n)	U1(M,N+1)
Z(m,n)	Z1(M,N)
H(m,n)	H(M,N)
D(m,n)	H(M,N+1)
Integer Matrix	IU(M,N)

Table 3.1. Array names.

The integer array, IU, was limited to two bytes instead of the customary four as no number larger than a '3' needed storing (two bytes can contain a positive integer of up to 127).

In such a manner the array storage requirements were reduced in the approximate ratio 12:7. Taking into account the computer core limitations, the maximum grid size that could be handled was 65 x 29.

Taking the three equations for the prediction of U, V, and Z (i.e. equations (3.8), (3.10), and (3.14)), the instructions were simplified by using the following:

Equation (3.8)

$$\text{USTAB} = \overline{U_{(m,n)}^{(\tau-1)}} \qquad \text{(see equation (3.7)).} \qquad (3.33)$$

$$\text{ZXATU} = \frac{\partial Z}{\partial x} = \frac{Z^{(r)}_{(m+1,n)} - Z^{(r)}_{(m-1,n)}}{2\ \ell} \quad . \tag{3.34}$$

Equation (3.10)

$$\text{VSTAB} = V^{(r-1)}_{(m,n)} \quad \text{(see equation (3.11))} . \tag{3.35}$$

$$\text{ZYATU} = \frac{\partial Z}{\partial y} = \frac{Z^{(r)}_{(m,n-1)} - Z^{(r)}_{(m,n+1)}}{2\ \ell} \quad . \tag{3.36}$$

Equation (3.14)

$$\text{Z1(M,N)} = Z^{(r)}_{(m,n)} \quad \text{(see equation (3.15))} . \tag{3.37}$$

$$\text{HUX} = \frac{H^{(r)}_{(m,n-1)} U^{(r+1)}_{(m,n-1)} - H^{(r)}_{(m,n+1)} U^{(r+1)}_{(m,n+1)}}{2\ \ell} \quad . \tag{3.38}$$

$$\text{HVY} = \frac{H^{(r)}_{(m+1,n)} U^{(r+1)}_{(m+1,n)} - H^{(r)}_{(m-1,n)} U^{(r+1)}_{(m-1,n)}}{2\ \ell} \quad . \tag{3.39}$$

The transposition of some of the more important variables may be seen in Table 3.2.

Original Symbol	Variable Name
r	R
f	F
g	GEE
α	Y .

Table 3.2. Transposition of some major variables.

The stability factor α was put equal to 0.99 following the practice of Yuen.

9. Stability of the finite difference equations in two space dimensions--.

It is tempting to use the same approach as was used for considering stability criteria for the one space dimension explicit scheme of Chapter II. A section of the grid network as used for the calculation of V is seen in Figure 3.5.

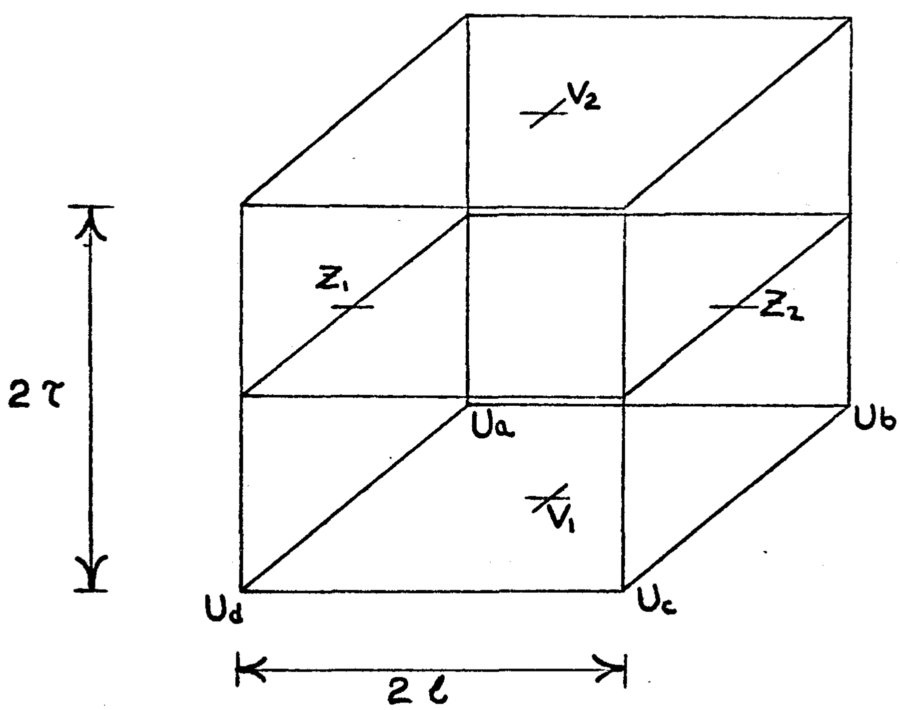


Figure 3.5. Grid points required for V-point calculation.

For stability V_2 must lie within the domain of determinacy of points U_a, U_b, U_c, U_d, Z_1, and Z_2. The U-points therefore are more likely to cause instability (on account of the steepness of the slope $U_i V_2$).

The value of this slope is easily seen to be $\frac{2\tau}{\ell\sqrt{2}}$.

For stability this value must be less than the slope of the characteristic cone through U_i , viz 1/c .

i.e.
$$\frac{\tau\sqrt{2}}{\ell} < \frac{1}{c} ,$$

or
$$\tau < \frac{\ell}{\sqrt{2gh}} \quad . \qquad (3.40)$$

If a similar diagram is drawn for a Z-point calculation (see Figure 3.6) it is seen that the stability requirement comes to

$$\tau < \frac{\ell}{\sqrt{gh}} \quad .$$

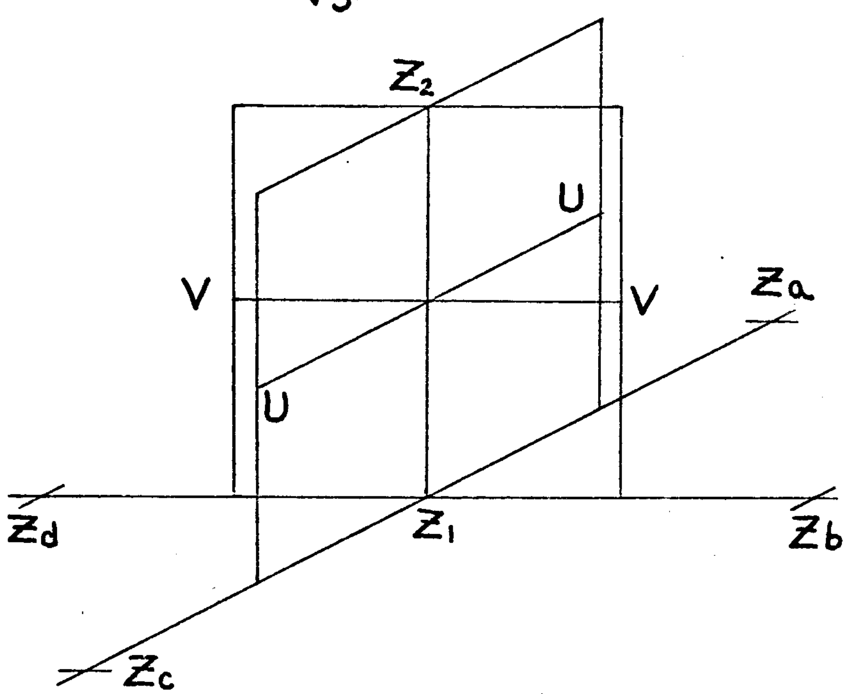


Figure 3.6. Grid points required for Z-point calculation.

The same stability requirements result for the U-points as for the V-points on account of the similar grid configuration. The most stringent requirement, as far as time is concerned, is thus that in equation (3.40).

CHAPTER IV

AUTOMATION OF THE SEQUENCE OF CALCULATIONS

1. The basic sequence of calculations--.

With the basic forms of calculation in FORTRAN form, the next and most crucial step ahead is their sequential control. Instructions must be developed that apply the basic types of calculation to each appropriate grid point as determined by the nature of the boundary.

First of all it is instructive to consider what might be called the conventional approach to the arrangement of the order in which the finite difference calculations are performed. Having chosen a suitable grid boundary, one might then arrange for the assignment of depths, initial tide heights, and zero velocities. The next step is the interpolation of tide heights and currents. Then follows the calculation of currents and heights, the input of new boundary values, and the repetition of the calculations. One way in which this might be done (for the case of a rectangular grid) is as follows:

Interpolation

a) Starting at the 'southwest' corner, one line from the bottom (m = 2), write an instruction for calculating U, Z, and H at V-points lying within the boundaries. Repeat this for all even-numbered rows.

b) Starting with the second line from the bottom ($m = 3$), write a similar type of instruction for calculating V, Z, and H at U-points. Repeat this for all odd numbered rows except for the top and bottom rows.

c) Apply equations (3.20) and (3.21) to U-points on the bottom row, and (3.18) and (3.19) to U-points on the top row.

d) Apply equations (3.24) and (3.25) to V-points on the left boundary, and (3.26) and (3.27) to V-points on the right boundary.

Current and height calculations

e) Apply U, V, and Z calculations at U-, V-, and Z-points respectively, row by row.

Boundary conditions and time increment

f) At this point it is convenient to apply the boundary conditions; along water - land boundaries U and V are put equal to zero as necessary. Along the line(s) where the inlet meets the open sea it is necessary to specify tide heights. These tide heights will replace those calculated in the Z-point calculations of step (e). The false values for Z that were calculated do not in any way effect the rest of the calculation. As mentioned in section 7, Chapter III, $\frac{\partial V}{\partial Y} = 0$ and $\frac{\partial U}{\partial X} = 0$ are applied along open boundaries as necessary.

g) The time step is now checked to see if the end of the tidal cycle has been reached. If not, the time is increased by 2τ, and the program returns to step (a).

h) The process is then repeated for the desired number of tidal cycles, values of U, V, and Z being printed whenever desirable.

It will be seen that the above method is straightforward as long as the grid boundary is strictly rectangular. If, however, the boundaries are irregular, the number of instructions will be greatly increased, and the amount of time to be spent in programming will be correspondingly large.

If a series of inlets are to be studied, perhaps with each involving two or more different grid spacings, it is obvious that any modifications to the program that result in reducing programming will be of considerable value. After programming several inlets in the manner above, as a result of the experience so gained, an approach was found that reduced the programming of any inlet to the few instructions necessary to specify the tide height at input points as a function of time.

2. Automation of the inlet-tide program--.

An inspection of the grid layout and of the various calculation types reveals a simple means by which the program may be automated. The new program is centered round the scanning of an integer-matrix which contains information as to the location of the solid and open boundaries.

Referring to Figure 4.1, an example of a grid network of irregular boundary configuration is shown with two perpendicular lines (crossing at a Z-point) emphasized. Starting with the row (m = 2), it will be seen that the following types of standard calculation may be inferred from the boundary limits:

* V = 0 at (m = 2, n = 1) and at (2,9)
* Conventional interpolation of U and Z at V-points (2,3) through (2,7)
* Special boundary-case interpolation of U and Z at V-points (2,1) and (2,9)
* V calculations at V-points (2,3) through (2,7)
* Z calculations at Z-points (2,2) through (2,8)

Similarly, along the column (n =6), the following calculation-types may be inferred:

* U (9,6) = 0
* Conventional interpolation of V and Z at U-points (3,6) through (7,6)
* Special boundary-case interpolation of V and Z at U-points (1,6) and (9,6)
* U calculations at U-points (3,6) through (7,6)
* U (1,6) = U (3,6) (application of $\frac{\partial U}{\partial X} = 0$ on open boundary)

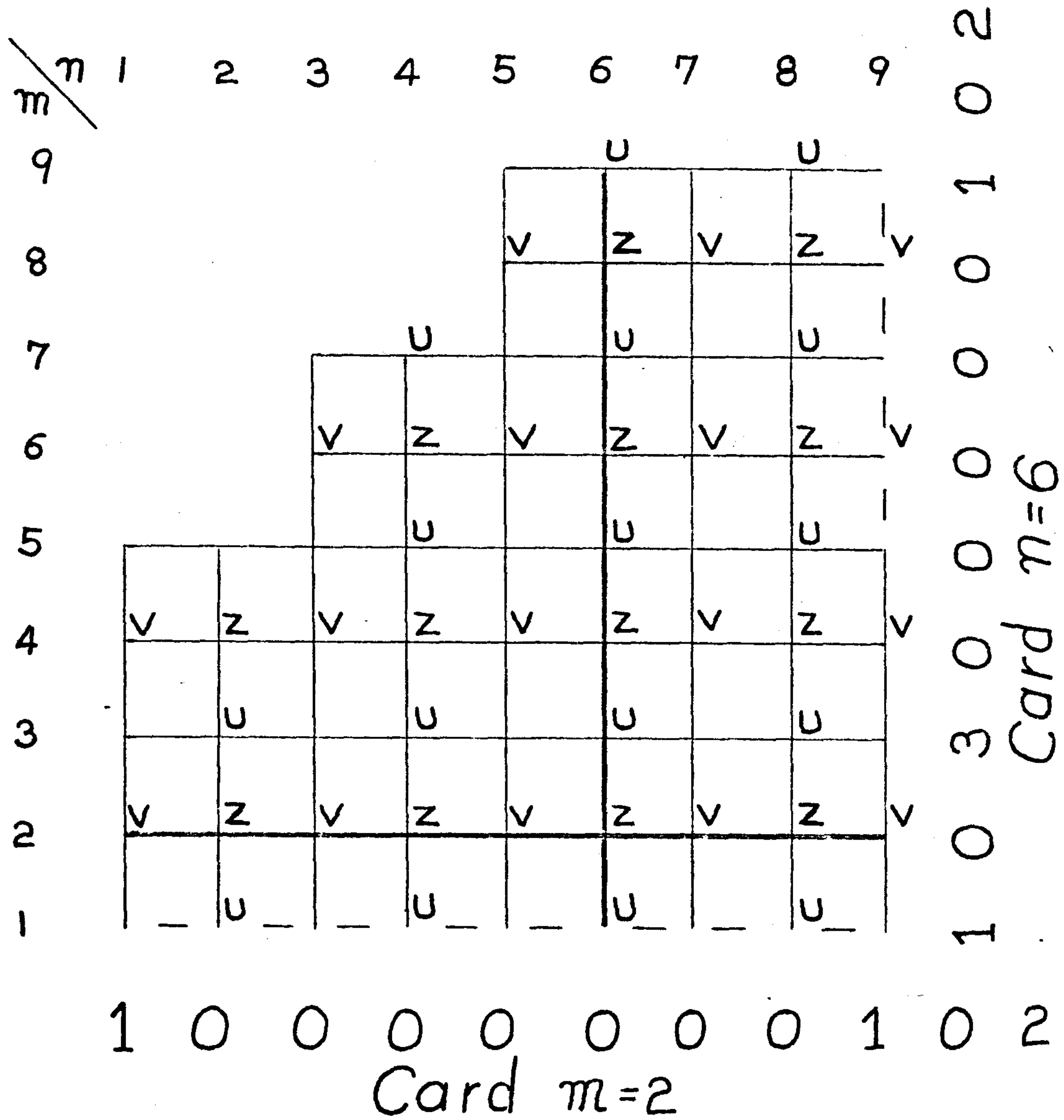


Figure 4.1. Typical column and row through Z-point, with associated integer matrix input cards (see text for explanation).

It will be noted that Z calculations are not needed along this column, as all Z-points can be covered when traversing the rows. This approach will be seen to include all possible boundary cases as long as the interpolations used are those previously referred to. At this point, it is possible to inspect the rows and columns visually, and thus specify the various calculation types. The next step is to perform this function automatically.

Boundary limits are specified in the form of integer numbers (see figure 4.1). Starting (for example, along a row containing V-points) from the left, the integer 1 is punched in odd-numbered columns of the card whenever a solid boundary is encountered. It is assumed that land extends to the left of the first integer. The next 1 indicates that solid land has once again been reached. This process of alternating land and water may be continued until the maximum allowable grid network size has been reached. In this program the limits are 29 in the horizontal direction.

An even number of 1's must be specified in order for the calculations to be bounded. In the case that no solid boundary exists, the 1 must still be used, as it serves as a limit for the grid-point calculations in that particular row. A 3 is placed 2 spaces to the inlet side of the boundary. This indicates to the program that the velocity V at the point 1 (to which the 3 applies) will be changed from zero to that at the matrix point containing the 3, i.e. we have applied $\frac{\partial V}{\partial Y} = 0$.

In order for the '3' not to cause confusion in the program, it is necessary that, in the particular row to which the '3' applies, there be a 1 two spaces away on the punched card on one side only of the '3'.

When the last (even-numbered) boundary has been reached, a 2 is placed two places to the right of the last 1. This indicates to the program that no further values of the integer matrix need be scanned along this row. The integers are punched on cards, one card corresponding to one row.

'East - west' boundaries are specified in precisely the same fashion as for 'north - south' boundaries. In this case, the grid is scanned from 'south' to 'north' along grid columns containing U- and Z- points, the limit being 65 grid points.

3. Input of boundary conditions--.

The boundary values are read into the computer first along columns of constant n, starting from the 'west', then along rows of constant m, starting from the 'south' (see Figure 4.2).

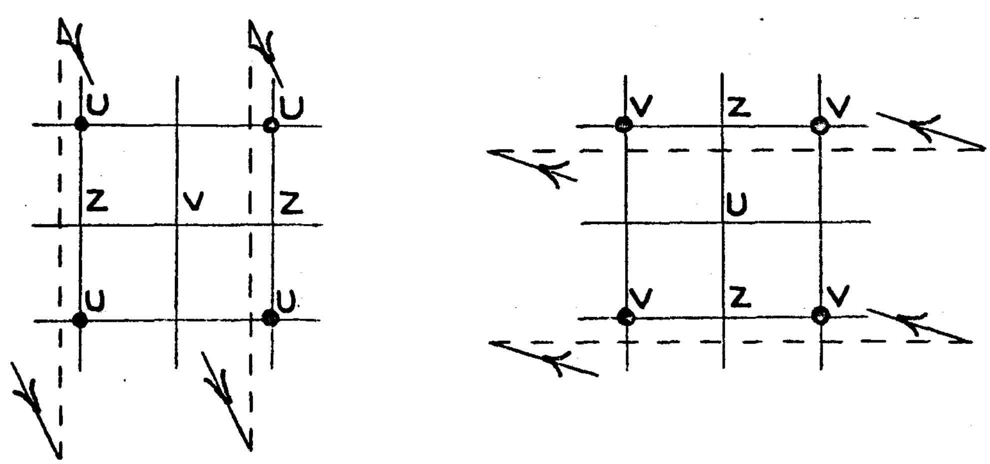


Figure 4.2. Order in which grid boundaries are read.

The half-word integer matrix IU previously referred to is thus built up column by column, row by row. The dimensions of this matrix exceed 65 x 29 by 3 and 2, making a 68 x 31 matrix: The 2 in each direction is to include the integer '2' at the end of each column and row; the extra 1 is to cause the array storage area to begin and end on a full-word boundary in the computer core.

This integer matrix is monitored during all parts of the program. Input of depths and initial tide heights, current and height calculations, interpolations, printout, and later in the analysis of the raw U, V, and Z output data.

This pattern followed is in all cases similar, and will be outlined in some detail.

4. Description of boundary-monitoring process--.

The procedure will be illustrated for the case of one of the rows during U calculations at U-points (see Figure 4.3).

At the start of the calculation of each row, a flag, IFL, is put equal to zero. This signifies that solid land lies to the left, i.e. that the first boundary met will indicate a transition from land to water. The first odd numbered column (n = 1) is then inspected for a 0, 1, 2, or 3:

If a 3 is found, the column number is increased by 2 and the process repeated

If a 2 is found, this indicates that no more columns need be scanned, so the sequence jumps to the next row

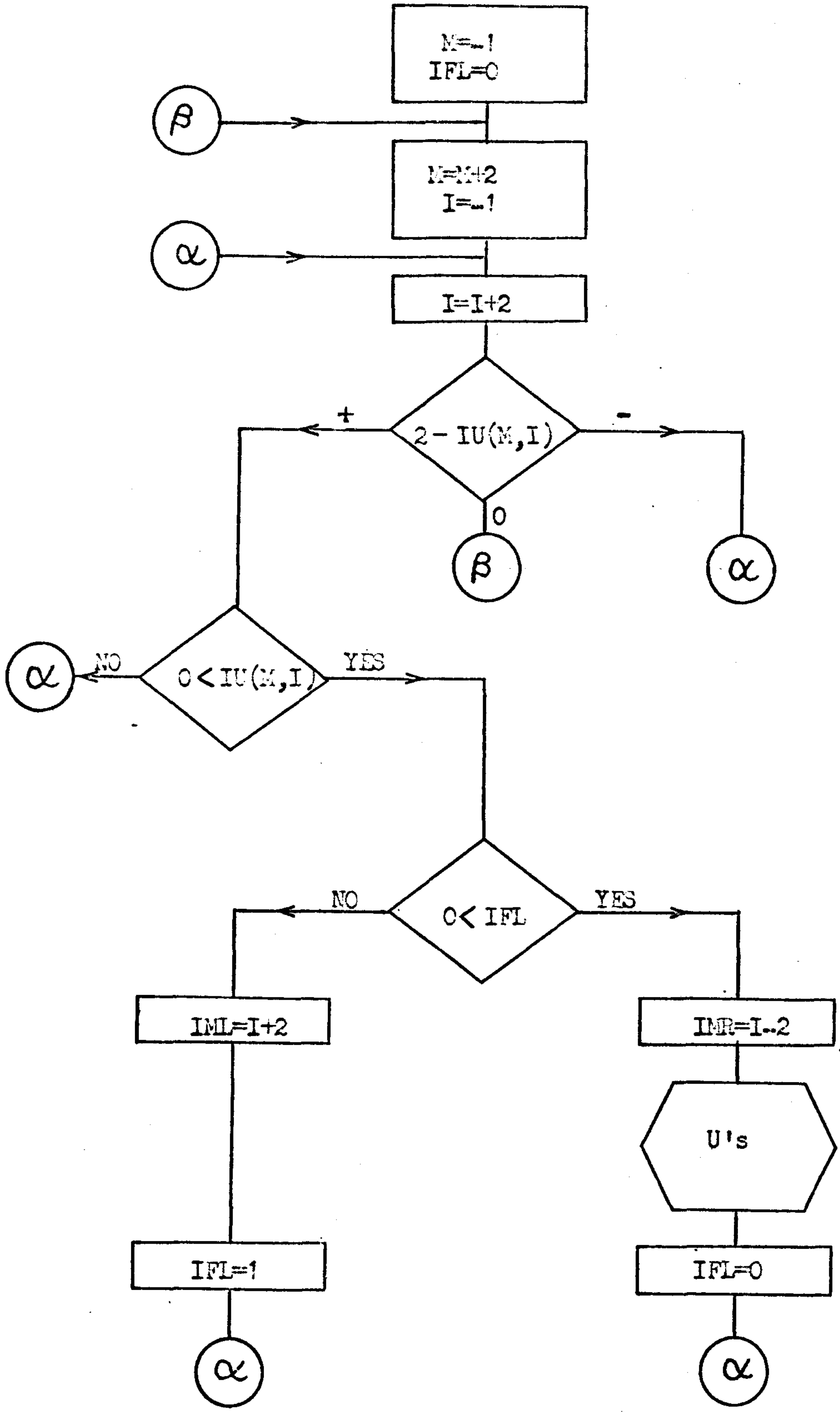


Figure 4.3. Flow chart for boundary-monitoring process.

If the integer is less than 2, the integer is checked for a
1 or a 0

If a 0 is found, the column number is increased by 2 and the process repeated

If a 1 is found, the flag is checked to see whether a left or right boundary has been arrived at

If the flag is 0, the boundary is a left-hand one. In this case the left-hand limit IML is set equal to (column number + 2). The flag value is changed to 1, and the process repeated.

If the flag is 1, the boundary is a right-hand one. In this case, the right hand limit IMR is set equal to (column number - 2). At this point, as may be seen from Figure 4.3, the limits of the U at U-point calculations for this section of the row have been ascertained. The calculations are then performed. The flag is then changed back to 0, and the process repeated. When all of the rows have been checked, the next phase of the program is entered (not shown in the flow chart).

The above process is modified by the use of extra 'IF' statements to deal with the various situations of special-case interpolations, unusually narrow conditions, etc.

CHAPTER V

PROGRAM ARRANGEMENT

1. Division of the program into subroutines--.

To simplify programming, and to divide the program up so that it would fit into the available core space, the full program was split up into several subroutines. Two of them are used once only, the remainder are called whenever necessary. The main program is responsible for calling the various subroutines when required. A flow chart of the main program, and of the subroutines may be seen the the pages that follow.

The flow chart (Figure 5.1) shows just sufficient information to enable the reader to follow the program through the steps of initialisation and then through the instructions that monitor the time steps and the tide cycles. Within the latter, on the second page of the flow chart, are the statements that control the times at which tide heights and currents are printed out. To trace the various branches in the full printout of the program (see Appendix I), the number of each instruction lying at the end of a branch line is written to the left of the corresponding instruction.

2. Overlays--.

The total program length including the FORTRAN program, array storage, and supervisor exceeded the available core space.

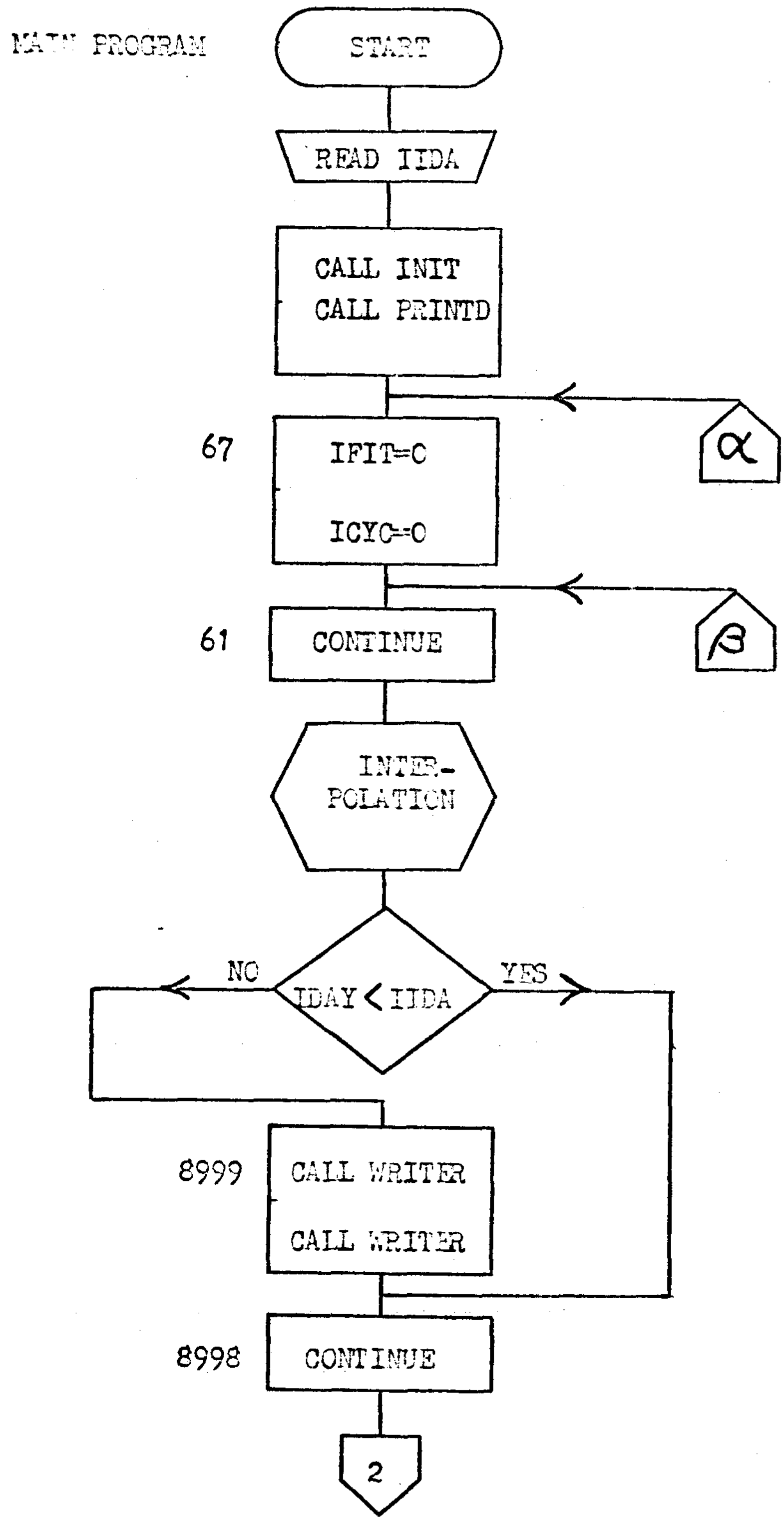


Figure 5.1. Program flow chart (1/6).

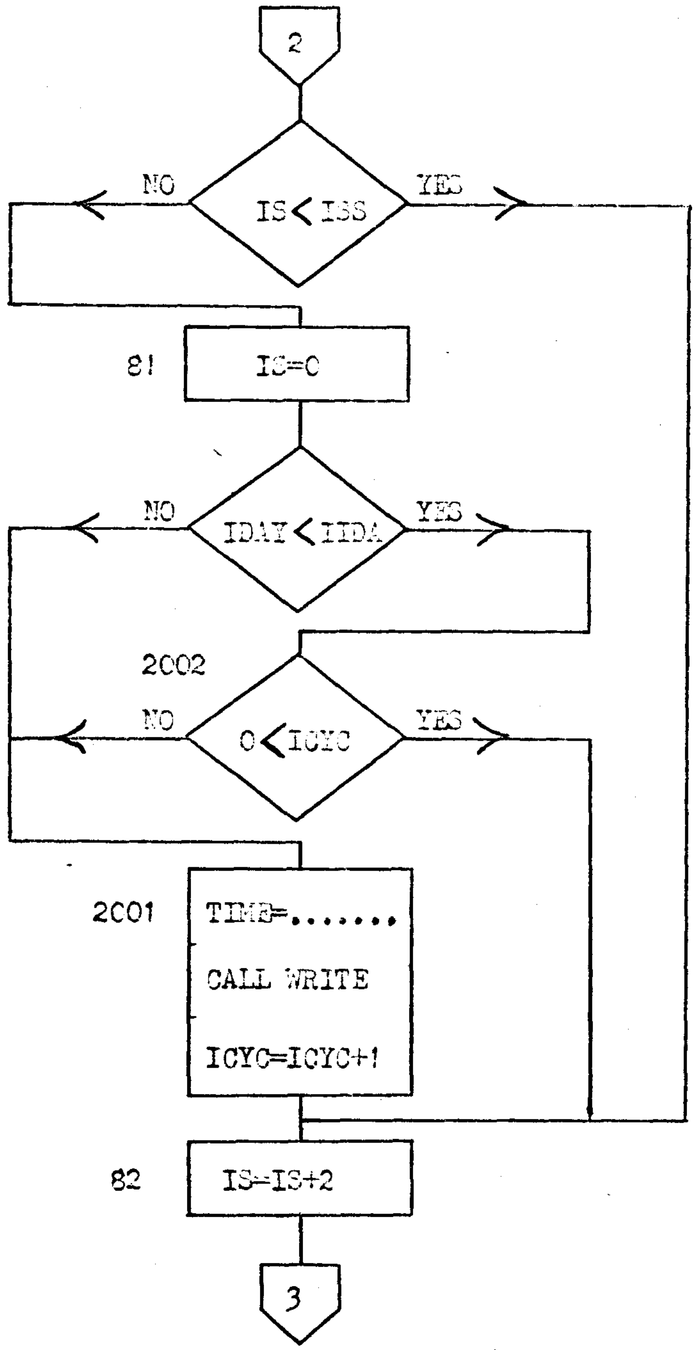


Figure 5.1. Program flow chart (2/6).

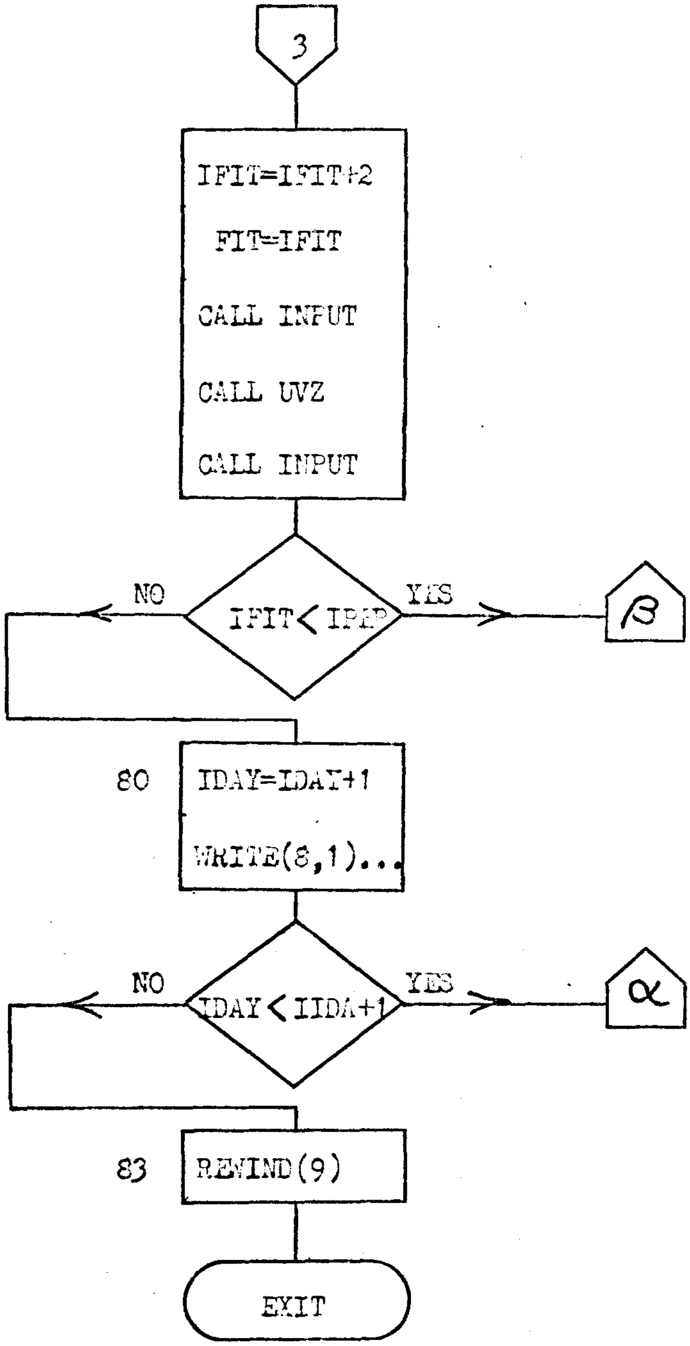


Figure 5.1. Program flow chart (3/6).

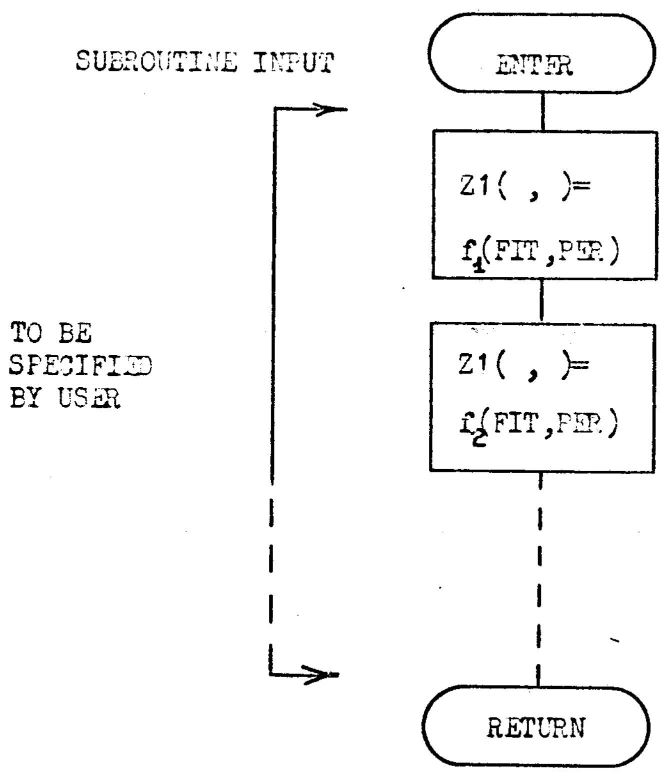


Figure 5.1. Program flow chart (4/6).

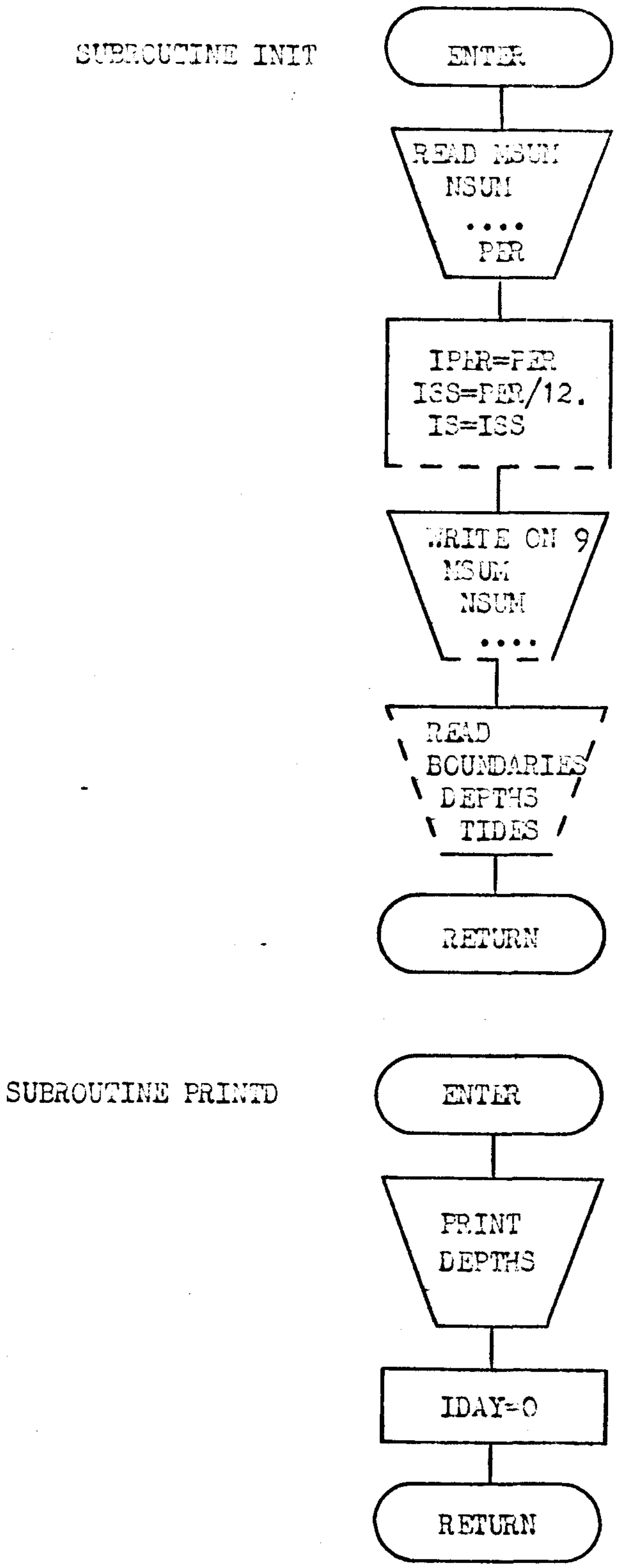
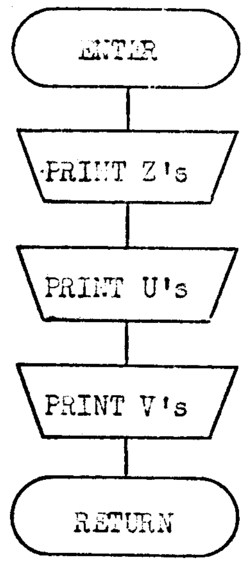


Figure 5.1. Program flow chart (5/6).

SUBROUTINE WRITE



SUBROUTINE UVZ

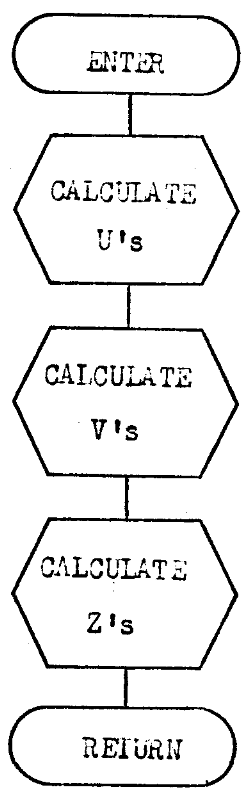


Figure 5.1. Program flow chart (6/6).

In order to run the program, it was necessary to split the program into several 'phases'. The process involves the storage of all the phases, with the exception of the main calling program (the 'root' phase), on disc. The root phase calls the particular phase required off the disc into core, where it is placed starting at a particular location.

For convenience, each of the phases consists of one of the main subroutines:

SUBROUTINE	PHASE NAME
INIT	PHASNME1
PRINTD	PHASNME4
WRITE	PHASNME2
UVZ	PHASNME3

Table 5.1. Phase Names.

The subroutines WRITER and INPUT were not split up thus, as they are continually being called by the root phase.

Once the phase corresponding to a particular subroutine has been placed in the core, it is called one as would a conventional subroutine. The additional instructions necessary are as follows:

* The main program is preceded by a card:

```
1234
    PHASE PHASNME∅,ROOT
```

* The next phase is preceded by:

```
1234
   PHASE PHASNME1,*
```

where the asterisk signifies that the program is to be placed into the first available location following the root phase.

* Each successive phase is preceded by a card of the type

```
1234
   PHASE PHASNME2,PHASNME1
```

The second name, after the ',' signifies that this phase is to be loaded into the core starting at the same location as PHASNME1.

* To call any particular phase, the necessary instruction is, for example;

```
1234567
      CALL OPSYS ('LOAD','PHASNME3')
```

* At any later point, the subroutine associated with PHASNME3 may be called as usual.

It is obvious that a subroutine may only be called when it has been previously loaded into the core.

The layout of the phases is conveniently shown by a diagram (see Figure (5.2). The numbers to the left of the main tree are the corresponding core locations in hexadecimal arithmetic, for one given length of the INPUT subroutine.

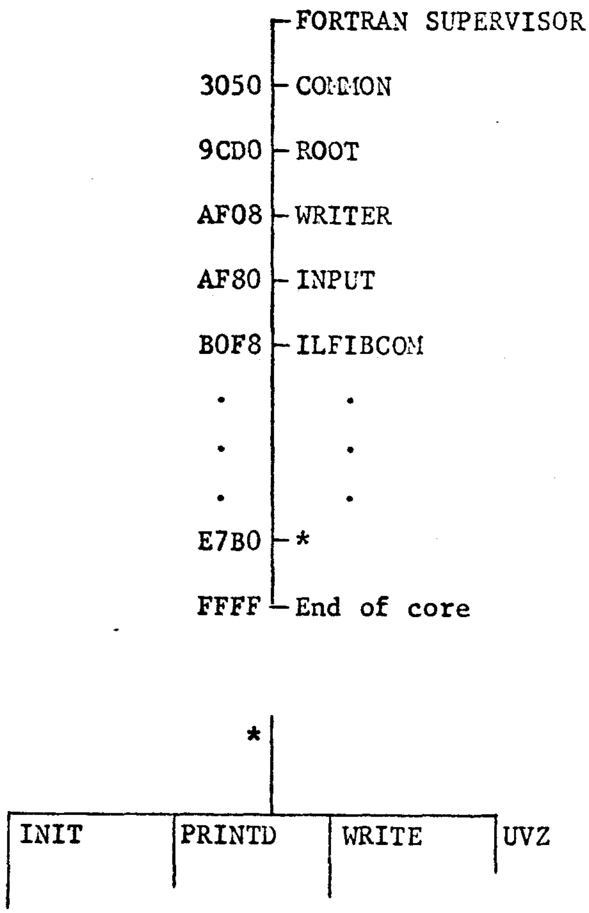


Figure 5.2. Overlay tree.

With this overlay system, with the longest phase (INIT) in core, the program extends to F551. A few additional bytes are reserved for buffer storage when various input/output devices are encountered during the program. No information as to their extent is printed out. If insuffient core space is available, an error message will be printed out, and the job terminated. In this particular computer, sufficient space was evidently available.

For more information on the overlay system, the reader is referred to the relevant IBM manual (IBM, 1968).

CHAPTER VI

GRID SELECTION AND DATA ARRANGEMENT

1. Grid selection--.

When a particular inlet is selected for tidal studies using the numerical model described above, the first thing to do is to ascertain the stability requirements. The accepted criterion for the stability of the staggered-grid model is

$$\ell > \tau \sqrt{2g \, D_{max}} . \tag{6.1}$$

The variable boundary model requires that the quantity (number of intervals)/(tidal period) be a multiple of 12 (this is to satisfy a part of the program that is responsible for printing out heights and current information 12 times during the last tidal cycle). The number of intervals normally used has been 360 or 720 (i.e. respectively 180 and 360 different times at which Z's are calculated at Z-points). The former gives a resolution of (ideally) 2° for the phase of the tide.

Using this type of calculation, a compromise may be found between a grid spacing that appears to represent the inlet satisfactorily, time intervals, and resolution.

A convenient method for fitting the grid to the inlet shape is as follows:

Draw a 65 x 29 grid on a sheet of paper and photograph it so as to obtain a slide.

Project an image of the grid onto a wall, and adjust the projector to give approximately the correct interval between grid lines.

Tape the map to the wall so that a reasonable alignment exists between the major axis of the inlet and the grid.

Final adjustments may then be made so as to achieve the best fit possible, consistent with stability and cost limitations.

The above method, although it has inaccuracies in it arising from optical distortion, heating up of the projector etc., gives a good first approximation. For small maps, some more convenient methods may be found.

<u>The left-most edge of the inlet must be on column n=1, the bottom of the inlet must be on row m=1.</u>

Having decided on a suitable grid configuration, the grid should then be transferred to the map. The use of any form of tracing paper (other than transparent mylar) as an overlay makes the work to follow more awkward. All the grid lines should be drawn in, and U-, V-, and Z-points suitably labelled.

2. Basic data cards--.

The next step is to prepare the data cards. Considering for example the grid in Figure 6.1.

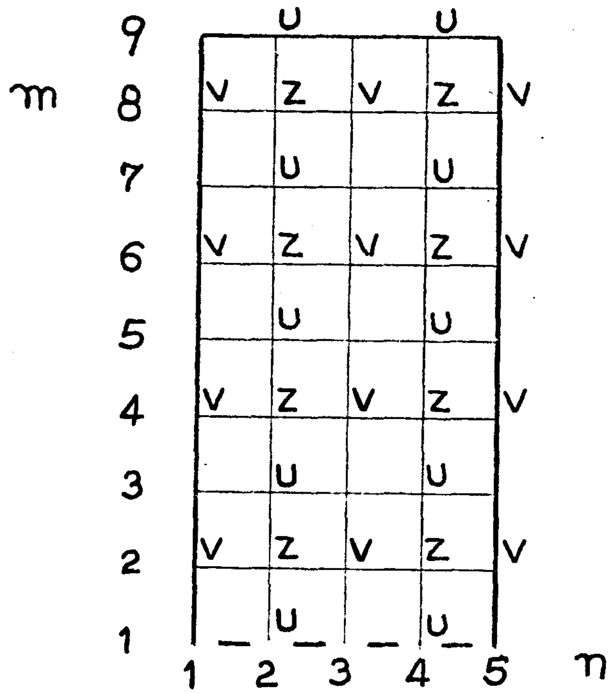


Figure 6.1 Example of simple grid.

The first data cards are those that specify the maximum dimensions of the grid, number of tidal cycles to be calculated, etc. These 8 cards are placed immediately behind the first // EXEC card. The order and format of the cards are as follows:

Card	Variable Name	Format	Example	Units
1	IIDA	I2	05	
2	MSUM	I2	09	
3	NSUM	I2	05	
4	DL	F12.4	50000.0	meters
5	T	F12.4	12.42	hours
6	R	F12.4	.003	
7	ALAT	F12.4	5.0	degrees*
8	PER	F12.4	360.0	

* North positive

Table 6.1. Example of input data cards.

The above cards specify the following:

1. 5 complete tidal cycles are to be calculated, starting at 01, ending at 05
2. Number (m) of top row (from example)
3. Number (n) of right column (from example)
4. Grid spacing in <u>meters</u> = 50 Km.
5. Period of tide in hours (M_2 tide)
6. Friction coefficient, generally 0.003
7. Latitude in degrees (5° N)
8. Intervals per tidal period.

3. Boundary data cards--.

Then follows a series of cards specifying boundaries along columns, i.e. points where U=0 or $\frac{\partial U}{\partial X} = 0$.

In this case we have two cards:

Column	1	2	3	4	5	6	7	8	9	10	11	12		80
Card														
1	1	0	3	0	0	0	0	0	1	0	2	0. . . .		0 0
2	1	0	3	0	0	0	0	0	1	0	2	0		0 0

The second series of cards specifies boundaries along rows, i.e. points where V=0 or $\frac{\partial V}{\partial Y} = 0$;

There are 4 cards:

Column	1	2	3	4	5	6	7	8	9	10	11	12		80
Card														
1	1	0	0	0	1	0	2	0	0	0	0	0 0 0		0 0
2	1	0	0	0	1	0	2	.	.	.	.			0 0
3	1	0	0	0	1	0	2	.	.	.	.			0 0
4	1	0	0	0	1	0	2	.	.	.	.			0 0

With the integer matrix in the core, the depths at V- and U-points may now be read in and automatically allocated.

4. Depth data cards--.

In this case, depths are read in at V points, starting in our example with the depth at (m = 2, n = 1). This depth is punched on a single card, in the format F12.4 (i.e. in decimal), the units being FATHOMS. (Meters were not used as most American and English charts are in fathom units). No depth should be less than the maximum expected tide amplitude -- one might say that no depth should be less than 4 fathoms. The next card contains the depth at (2,3), following with those at (2,5), (4,1), (4,3), (4.5),.... (8,3), (8,5), one depth to each card. The order is thus as in Figure 6.2.

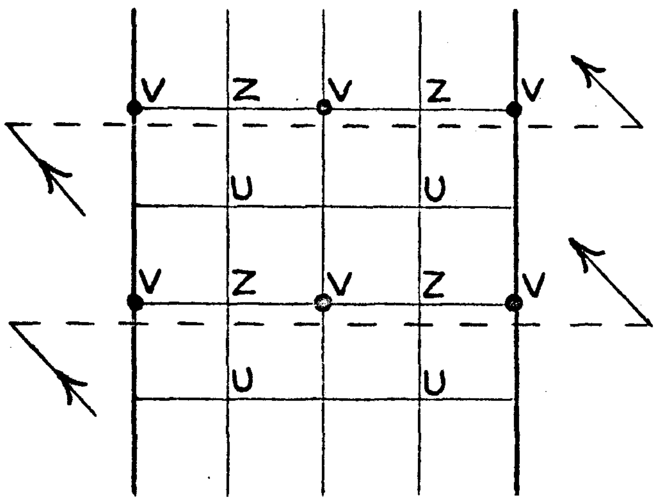


Figure 6.2. Order of specifying depths at V-points.

The next group of cards contain depths at U-points, the procedure being the same as for the V-points. The order of the cards is, for our example: (2,1), (2,3), (2,5), (2,7), (2,9), (4,1),......, (4,7), (4,9). See Figure 6.3.

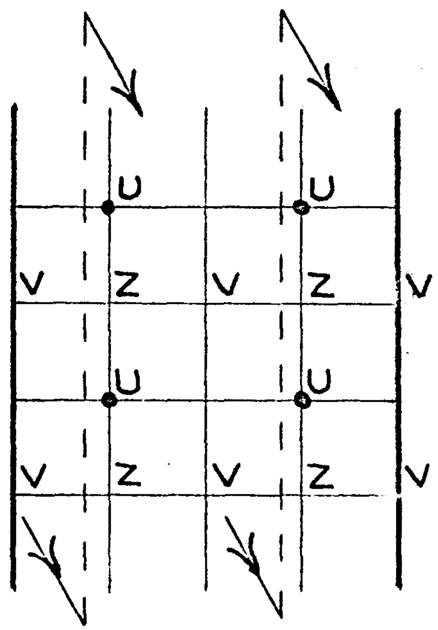


Figure 6.3. Order of specifying depths at U-points.

5. Initial tide-height and boundary-value cards--.

With the depth cards all prepared, we then proceed to the initial tide heights at Z-points. These are prepared from the best available distribution of tide amplitudes and phases over the inlet. the tide is considered to be at its maximum height across the input. Heights along the other V and Z rows are estimated by taking (amplitude) x cos(phase lag), where the phase lag is the delay of arrival time of maximum tide height compared with the input.

Heights are estimated in METERS, and are punched in F12.4 format (decimal), one to a card. The order in which they are taken is from left to right: (2,2), (2,4), (4,2), (4,4),......, (8,2), (8,4). See Figure 6.4.

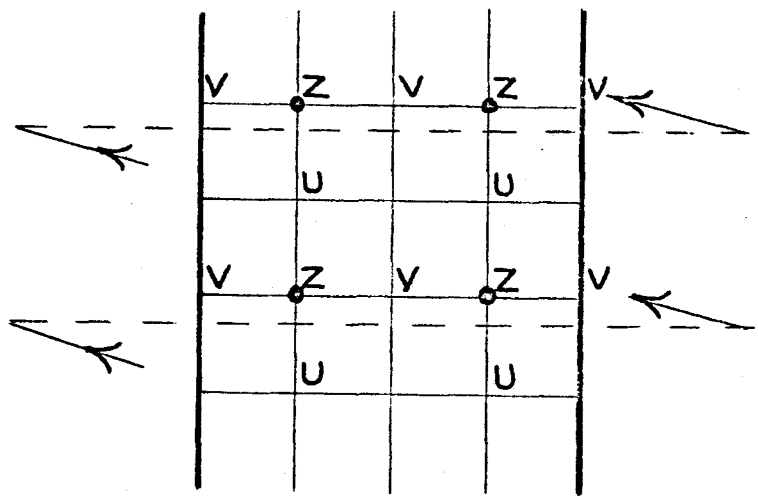


Figure 6.4. Order of specifying initial heights.

We now have the following blocks of data cards:

Type	No. of cards
Grid dimensions, tide information, etc.	8
Boundary positions	6
Depths	22
Initial tide heights	8

Table 6.2. Data arrangement for example.

This fully completes the data cards. The only task remaining is to specify the input conditions. These cards are added to the program in the INPUT subroutine, directly after 'COMMON TIDE'.

As an example, one might use:

Z1(2,2) = 0.743*COS(6.28318*((FIT/PER)-0.0))

Z1(2,4) = -Same-

Here FIT/PER is the point of the tidal cycle that has been reached, expressed as a fraction of 1.0. The last term (in this case -0.0) is the phase delay of the maximum tide compared to that at the input. It will range between -0.0 and -1.0 (a delay of 90° would be -0.25). The number 0.743 indicates a tide amplitude of 74.3 cms (or a range of 148.6 cms).

It is suggested that, as far as sinusoidal tides are concerned, this instruction-type be adheared to, thus only the 0.743 and the -0.0 should be changed.

CHAPTER VII

COMPUTER OUTPUTS AND DATA ANALYSIS

1. Printer output--.

The first page of the computer output (after the // EXEC statement) contains information on the grid interval, tidal period, friction coefficient, latitude, coriolis parameter ($2\Omega \sin\phi$), and the units used in the pages that follow.

The next 1-4 pages contain information as to the distribution of depth (in meters). If the maximum grid width (NSUM) is less than 18, 1 or 2 pages will be printed depending on the value of the grid length (MSUM). If NSUM is greater than 18, one or two additional pages will be printed covering columns 19 to 29. These may be detached and joined to the first one or two pages.

The next pages, in a similar arrangement, will be the (interpolated) values of the initial tide heights. The next two sets of pages will be the initial values for U and V. They will all be zero. As the H, Z1, and U1 matrices were all set to zero at the start of the program, it follows that all untouched elements of the arrays will be printed as zeros. This was done for two reasons (although it may prove confusing at first): to avoid writing complicated format statements, and to serve as a check on the functioning of the program, i.e. if non-zero values show up in unexpected places some error in the boundary-location specification may have occurred.

After this, values of tide height and currents are printed in a similar fashion at the end of each tidal cycle, with the exception of the last.

During the last cycle values are printed out at fractions (1/12) of the tidal period. Thus values will be printed at 1/12, 2/12, 3/12, ..., 11/12 of the period. This provides values of the intermediate tide and current distributions.

2. Tape outputs--.

Two tapes are used during the main program:

* A short tape is placed on unit 8 (a tape I/O device), and has sufficient information read onto it at the end of every tidal cycle so that in the event of an unscheduled termination only a small amount of reprogramming is necessary to restart the program at the beginning of the next cycle. This is useful when, for some reason or other, the program is terminated before the CALL EXIT is reached (such as during a power failure). The tape is discarded in the event of a successful run.

* A long tape is place on I/O unit 9. At the start of the program basic information, such as dimensions, tidal period, boundary positions, etc, are written onto the tape, for details please see Appendix IV. During the last cycle, values of current and height are written onto the tape every time that tide heights are calculated. For convenience, the entire U1 and Z1 matrices are written onto the tape. In order to achieve maximum compression of data, a special

program is used that writes the entire matrix as one continuous record (FORTRAN IV normally limits the maximum record length to 64 single-precision words, then leaves an inter-record gap of 6/10 inch.) The tape is then rewound at the end of the last cycle, and is thus ready for detailed analysis. The program was written by Mr. Don Walker of the University of Alaska Computer Center

3. Data analysis--.

This consists of the analysis of the current and height data on the second tape. Two programs have been joined together to form one standard package:

Program 1: Height and Phase analysis.

This program scans the tide heights at each Z-point. It stores the maximum and minimum tide heights that occur during the last cycle along with the associated phases. These values are then printed out. The output format differs from that used during the main program; asterisks are printed out in land areas, and the spacing between rows has been increased so as to partially offset the distortion of the inlet shape that occurs in the printing. The result is pleasing to the eye.

The program then calculates the mean range from (max tide height-minimum tide height), and mean phase from:

$$\text{mean phase} = \frac{\text{phase of max. height} + \text{phase of min. height}}{2} - 90°, \quad (7.1)$$

provided high tide arrives before low tide during the last cycle. If not, the phase of minimum height first has 360 added to it before equation (7.1) is computed.

Program 2: Current Analysis.

For each current matrix, currents are interpolated at Z-points. These currents are combined to form a vector, and the length and angle (clockwise from the North) are calculated. The current values are checked for maximum and minimum values. The times (in hours) and angles are stored along with the associated maximum or minimum values. At the end of the cycle the values are printed out. From this output it is possible to estimate the dimensions and directions of the current ellipse axes and their sense of rotation. At present, during plotting, it is necessary to assume that the maximum currents are the same at ebb and flood, and that their directions are 180° apart. Similarly with minimum currents at slack water. It should be a simple matter to extend the program to calculate the 2 maximums, and the 2 minimums with their associated angles and times, however, it is arguable whether the present accuracy warrants such detail.

A printout of the two analyses programs will be found in Appendix 3.

CHAPTER VIII

A SAMPLE PROBLEM

To fulfil the need for a sample problem that will serve as a guide for data arrangement and as a test for the program, a simple example will next be presented and solved.

The problem is as follows; An inlet has the following dimensions:

Length	350 km
Width	200 km
Depth	250 fathoms

The inlet will be analysed for a tide of period 12.42 hours, having an amplitude of 0.743 meters at the mouth. In the absence of friction and Coriolis force the application of equation (2.15) shows that the expected amplitude of the tide at the closed end of the inlet should be 1.000 meters. The tide will be considered uniform across the mouth of the inlet for reasons of convenience, although in reality this would be unlikely. To go along with this, a latitude of 5° North will be assumed. If a grid interval of 50 kms. is selected, the application of equation (6.1) results in $\tau \leq 527.9$ seconds. On choosing 360 intervals per tidal period, $\tau = 124.2$ seconds. This might be considered unnecessarily generous, however it will provide good resolution for the phase of the tide. A value for the friction coefficient of 0.003 will be assumed and the program will be allowed to run through five complete cycles.

For this problem the first 8 data cards will be as in Table 6.1. The boundary-value data cards follow as listed in Chapter 6, section 3. As depths throughout the inlet are constant there will follow 22 cards, each with 250.0 punched in the first 5 columns. For the initial tide heights, values are needed for rows 2,4,6, and 8. From equation (2.15) we obtain

$$Z(x) = \cos\left(\frac{360 \cdot x}{2994}\right), \tag{8.1}$$

where x is measured in kms. from the closed end of the inlet.

The approximate initial tide heights are then as in Table (8.1).

Row	Height
8	0.995
6	0.95
4	0.865
2	0.743

Table 8.1. Initial tide heights.

The data cards will therefore be, one number to a card (starting in column 1), 0.743,0.743,0.865,0.865,0.95,0.95,0.995,0.995 . The two cards that have to be added to the INPUT subroutine are as in Chapter 6, section 5.

The program was run on an IBM 360/40 computer and required 7.5 minutes. The two analysis programs required a further 3.5 minutes each. Some of the printed results are shown in Appendix V. The outputs are largely self-explanatory and agree closely with those predicted.

CHAPTER IX

TWO APPLICATIONS OF THE MODEL

1. Application of the model to the M_2 tide of the Gulf of California--.

The Gulf of California has its entrance on the Pacific Ocean and is bounded by Lower California to the west and Mexico proper to the east. The gulf is oriented in a northwest-southeast direction with its northern limit being formed by the Colorado River (Latitude 32° N.). Its mouth lies between Cabo San Lucas and Cabo Corrientes (with a mid-latitude of about 22° N.). The tidal study was confined to that part of the gulf lying to the north of the city of Guaymas (Latitude 28° N.) for reasons of economy of computer time. The bathymetry of the gulf, along with the grid outline finally chosen is shown in Figure 9.1.

The greatest depth that occurs in this restricted region is some 2740 meters. To represent the coast around the locality of Isla Tiburon to an adequate degree, it was found necessary to select a grid interval of 15 km . Owing to the narrowness of the channel lying between Lower California and Isla Angel de la Guarda, it proved impractical to represent the outline of the island with this particular grid scheme. The effect of the island was partially taken into account by assigning an arbitrary depth of 5 fathoms to all grid points lying within the outline of the island.

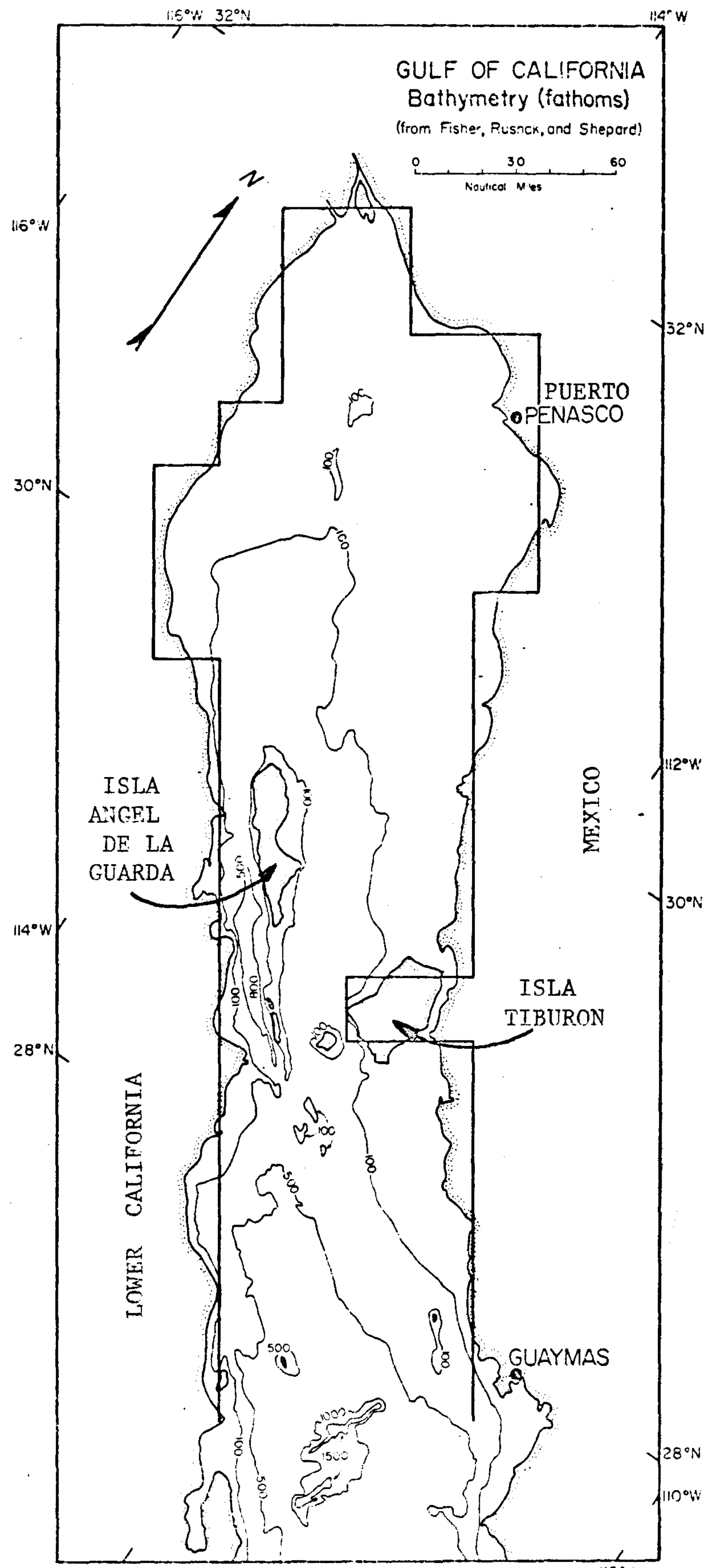

GULF OF CALIFORNIA
Bathymetry (fathoms)
(from Fisher, Rusnak, and Shepard)
0 30 60
Nautical Miles
N
116°W 32°N
114°W
116°W
32°N
PUERTO PENASCO
30°N
112°W
MEXICO
ISLA ANGEL DE LA GUARDA
30°N
114°W
ISLA TIBURON
28°N
LOWER CALIFORNIA
GUAYMAS
28°N
110°W
112°W
110°W
100
500
1000
1500

Figure 9.1

To ensure stability a time step of 62.1 seconds was chosen. This conforms to the stability requirements of the two-dimensional explicit finite difference scheme (equation (6.1)), so that

$$\tau < \frac{15{,}000}{\sqrt{2 \times 9.81 \times 2740}} \qquad (9.1)$$

or $\tau < 65$ seconds.

Input tidal data for this region is scarce. The only places for which adequate tidal data were available consisted of Puerto Penasco in the north, and Guaymas. The amplitudes of the M_2 tide constituents are 157 and 14 cms. respectively, while the difference in phase was taken as 107 degrees (U.N.A.M., 1967). As no reliable information was available for the variation of the M_2 constituents across the input boundary opposite Guaymas, a difference of 1 cm. was assumed for the amplitude (the range being smaller in the west), and zero degrees for the phase. This was based on the values of the mean range and the establishment for San Lucas Cove and Guaymas (Matthews, 1968).

Thus with these assumptions, the cards that had to be added to the INPUT subroutine were as follows:

(column) 1234567

```
      Z1(2,4)=0.130*COS(6.28318*((FIT/PER)-0.000))
      Z1(2,6)=0.133*COS..........................
      Z1(2,8)=0.136*COS..........................
      Z1(2,10)=0.140*COS..........................
      Z1(32,12)=1.570*COS(6.28318*((FIT/PER)-0.297))
```

The total duration of the main and analyses programs was about 65 minutes. The co-tidal and co-range lines, which may be said to be the most useful results, are shown in Figure 9.2 .

The co-tidal lines show that between the sea/sea boundary and Isla Tiburon the tidal wave is essentially of progressive wave type, with the phase of the tide changing by 90 degrees. To the north of Isla Tiburon the wave changes to one of standing wave characteristics. This is supported by the orientation of the co-range lines in this region, which lie across the width of the gulf, and the co-tidal lines, which lie along the axis of the northern part of the gulf (see Defant (1960)). The co-range lines in addition show that almost all of the amplification of the tide occurs between Penasco and Isla Tiburon, the range increasing from 90 to 314 cms.

The nature of the tides in the Gulf of California may be conveniently be indicated by the use of the Formzahl (Courtier, 1938). The Formzahl, F, for any given place is the quantity

$$F = \frac{K_1 + O_1}{M_2 + S_2} . \qquad (9.2)$$

For Penasco F=0.28, falling in the region in which tides are classified "mixed, mainly semi-diurnal" ($0.25 \leq F \leq 1.5$). F for Guaymas is 1.92, and falls under the classification "mixed, mainly diurnal" ($1.5 \leq F \leq 3.0$). The tidal regime of the gulf thus appears to fall into two categories depending on the position north or south of the narrow section; mostly semi-diurnal to the north, mostly diurnal to the south.

Defant (1960) has stated that the overall tidal configuration of the gulf seems to be one of a standing wave with a nodal line

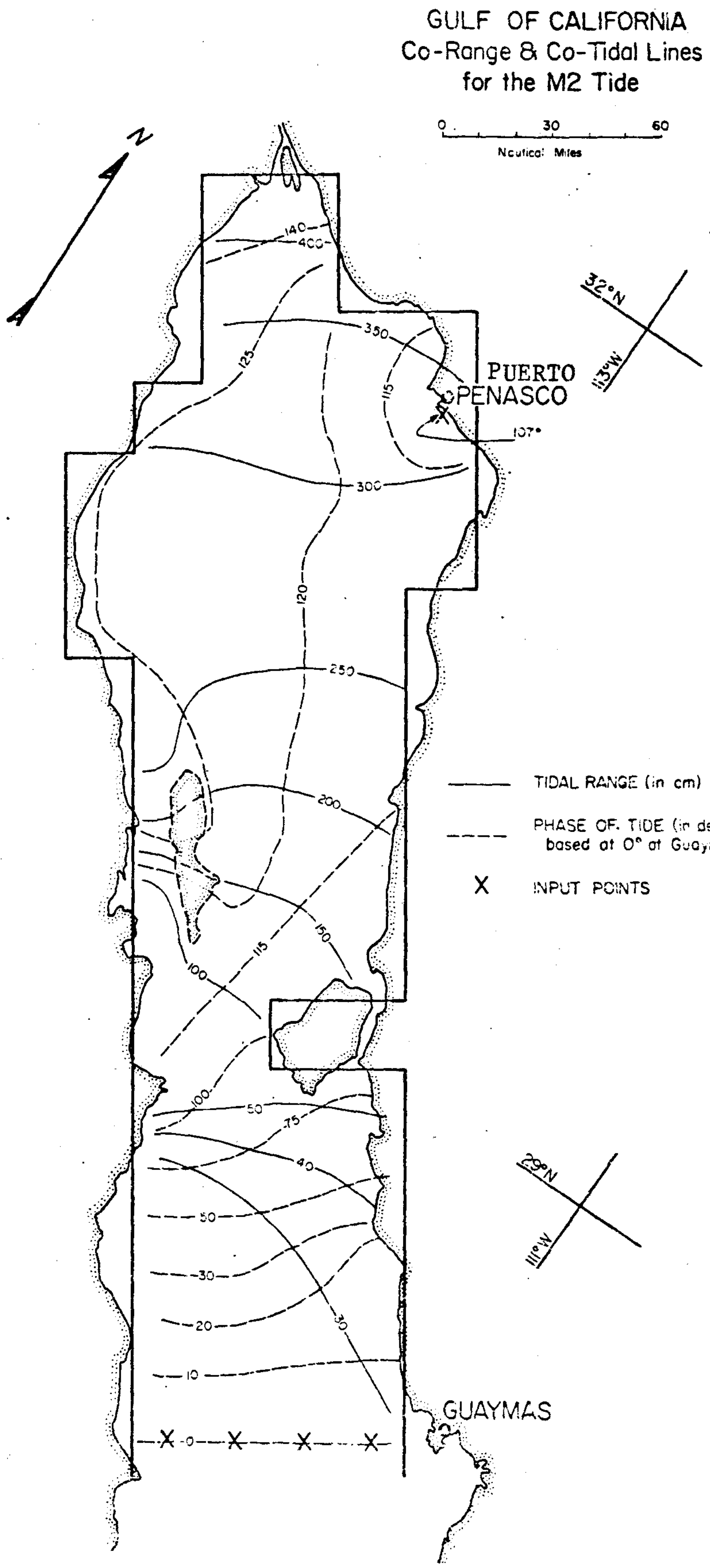

GULF OF CALIFORNIA
Co-Range & Co-Tidal Lines
for the M2 Tide
0
30
60
Nautical Miles
N
32°N
113°W
PUERTO
PENASCO
107°
GUAYMAS
29°N
111°W
TIDAL RANGE (in cm)
PHASE OF TIDE (in degrees, based at 0° at Guaymas)
X
INPUT POINTS
140
400
350
125
115
300
120
250
200
150
115
100
100
50
75
40
50
30
20
10
30
0

Figure 9.2.

occuring near the narrow section. However, as seen above, the variation of predominance of the semi-diurnal and diurnal constituents with latitude warns one not to expect too simple a standing wave pattern. It should therefore not be too surprising that the M_2 co-tidal lines should not agree more closely with the few pieces of tidal data available (Matthews, 1968), which show little difference in the establishment for locations between Guaymas and Isla Tiburon.

The most noticible defect in the output is that of the co-tidal lines in the vicinity of the input point near Puerto Penasco. Here an anomaly in the lines may be seen. However, the fact that the anomalous behavior dies out within a short distance leads one to the conclusion that had this input point been left out, the phase of the tide at this point would have been about 115 degrees. This value differs from the data in the U.N.A.M. tide tables by some 8 degrees. It is interesting to note that there is a difference in the establishment of the M_2 constituent for Guaymas as computed by U.N.A.M. and the U.S.C.&G.S. (unpublished data). The difference can probably be attributed to the small tide amplitudes available for analysis. On account of this it is difficult to distribute the fault between the model and the tide tables without additional tidal records.

An inspection of the combined set of co-range and co-tidal lines leads one to the conclusion that the difference in amplitude across the input boundary should have been nearer to 2 or 3 cms., while the phase difference should have been about 10 degrees, with high tide reaching the east side before the opposite point on the west.

The conclusions suggested by the above application are as follows:

1. In the event that bad data are used at an isolated input point, the fact will be made clear by the distortions in the co-range and co-tidal lines.

2. The effect on the rest of the area will probably become negligible at distances greater than 4 or 5 grid intervals.

2. Application of the model to the tides of Cook Inlet--.

Cook Inlet is located with its entrance on the coast of South-central Alaska. The inlet is some 150 miles long and terminates in two arms, Turnagain Arm and Knik Arm. Cook Inlet is generally shallow, between Homer and Anchorage the greatest depth encountered is of the order of 75 fathoms. At Homer the inlet is 27 miles wide, but narrows locally to 9 miles between the East and West Forelands. North of the Forelands the region becomes increasingly complicated (in the hydrodynamic sense) by the presence of shoals and mud flats, with extensive areas of Turnagain Arm being exposed at low tide.

The tides of the upper part of Cook Inlet are amongst the highest in the world and can be classed with those of the Bay of Fundy, Ungava Bay, and the Straits of Magellan. The tides are predominately semi-diurnal, having a mean range of 25.1 feet at Anchorage (U.S.C.&G.S., 1968) and a value for F (equation (9.2)) of 0.24. In addition the presence of strong currents and seasonal pack ice cause much hinderance to shipping. Long-term measurements of tide heights are complicated by the ice, while velocity measurements are made most difficult by the high currents and rough seas.

It seems customary when using numerical models to investigate the tides in an inlet to use M_2 amplitudes and phases as input conditions. If non-linear equations are used, the resulting currents are largely without significance as one cannot combine the solutions obtained for the various constituents as the model involves non-linear terms. This of course raises questions as to the correctness of restricting the input to one constituent only. The current conditions are of considerable practical interest in the case of Cook Inlet, so it was decided that efforts would be directed towards the ultimate goal of using real tide measurements as input conditions (subject to removal of high frequency components). Since it has been shown that some 120 constituents are needed to reliably predict the tide at Anchorage (Zetler and Cummings, 1967), it was clear that as a first step the model should be tested with a hypothetical tide obtained by assuming a sinusoidal wave of period 12.42 hours, amplitude based upon the mean range as tabulated in the tide tables (U.S.C.&G.S., 1968) for the region, and phase based on the high and low tide arrival times.

After some trial runs on an IBM 360/40 computer, a compromise was reached between computer time and the accuracy with which the outline of the inlet could be represented by straight sections of the grid. The model was restricted to that part of the inlet north of Homer. A grid interval of 3.052 kms. enabled the region of interest to be contained within a grid of dimension 65x29. The final grid outline, along with the bathymetry of the region, may be seen in Figure 9.3. To comply with the accepted stability condition, a time interval of 62.10 seconds was chosen. To arrive at the input conditions across the sea/sea boundary at Homer, an estimate was

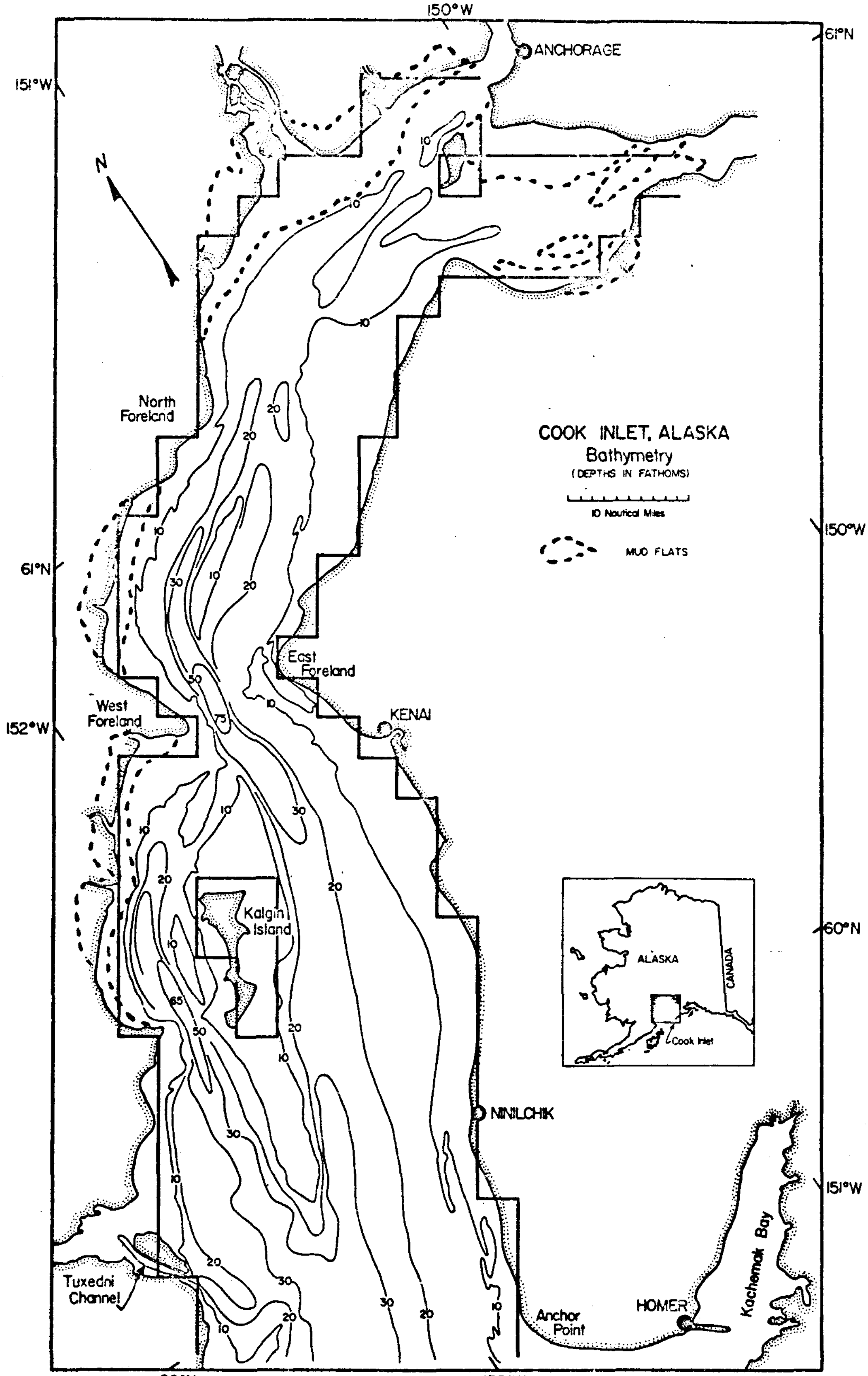

150°W
61°N
ANCHORAGE
151°W
N
North Foreland
COOK INLET, ALASKA
Bathymetry
(DEPTHS IN FATHOMS)
10 Nautical Miles
MUD FLATS
150°W
61°N
East Foreland
West Foreland
152°W
KENAI
Kalgin Island
60°N
ALASKA
CANADA
Cook Inlet
NINILCHIK
151°W
Kachemak Bay
Tuxedni Channel
Anchor Point
HOMER
60°N
152°W

Figure 9.3.

made of the range and phase of the tide on the opposite shore using values for Tuxedni Channel and Iliamna Bay. An interpolation was then performed to obtain the values at each input point. Because of the inability of such a model to handle mud flats (i.e. regions where the depth may occasionally become zero), all such regions were assigned an arbitrary depth of 4 fathoms. Furthermore, to avoid problems with the very shallow conditions that exist in Turnagain Arm and Knik Arm, the northern end of the model was terminated in two sea/sea boundaries. The required cards to be added to the INPUT subroutine are shown in Table 9.1.

Five full tidal cycles were computed, after which time conditions appeared steady. Each cycle required some 20 minutes of computer time. The analysis programs required 15 minutes, and a chart of the resulting co-tidal and co-range lines may be seen in Figure 9.4. It is at once apparent that the tidal regime of Cook Inlet divides the inlet into two distinct regions. For convenience they may be called North Cook Inlet and South Cook Inlet. They are separated from one another by the natural feature of the narrow section that lies between the West and East Forelands.

The tides in South Cook Inlet show the characteristic appearance of a progressive Kelvin wave. The co-range lines lie along the length of the inlet with higher amplitudes occuring to the east. The co-tidal lines lie essentially perpendicular to the co-range lines and slope upwards to the right, thus indicating that the wave is not entirely progressive but tends towards a mixed type of wave

```
      P=0.051+0.0054
      A=2.06-0.06
      DO 69 N=6,20,2
      P=P-0.0054
      A=A+0.06
69    Z1(2,N)=A*COS(6.28318*((FIT/PER)-P))                    INPUT
      Z1(6,4)=2.13*COS(6.28318*((FIT/PER)-0.059))             TUXEDNI
      Z1(12,18)=2.53*COS(6.28318*((FIT/PER)-0.06))            NINILCHK
      Z1(36,8)=2.74*COS(6.28318*((FIT/PER)-0.221))            EASTFORE
      Z1(48,4)=2.79*COS(6.28318*((FIT/PER)-0.314))            NORTHFOR
      Z1(64,18)=3.82*COS(6.28318*((FIT/PER)-0.376))           ANCHORAG
      Z1(60,28)=4.25*COS(6.28318*((FIT/PER)-0.402))           GULLROCK
```

Table 9.1. Cards added to INPUT subroutine for Cook Inlet program.

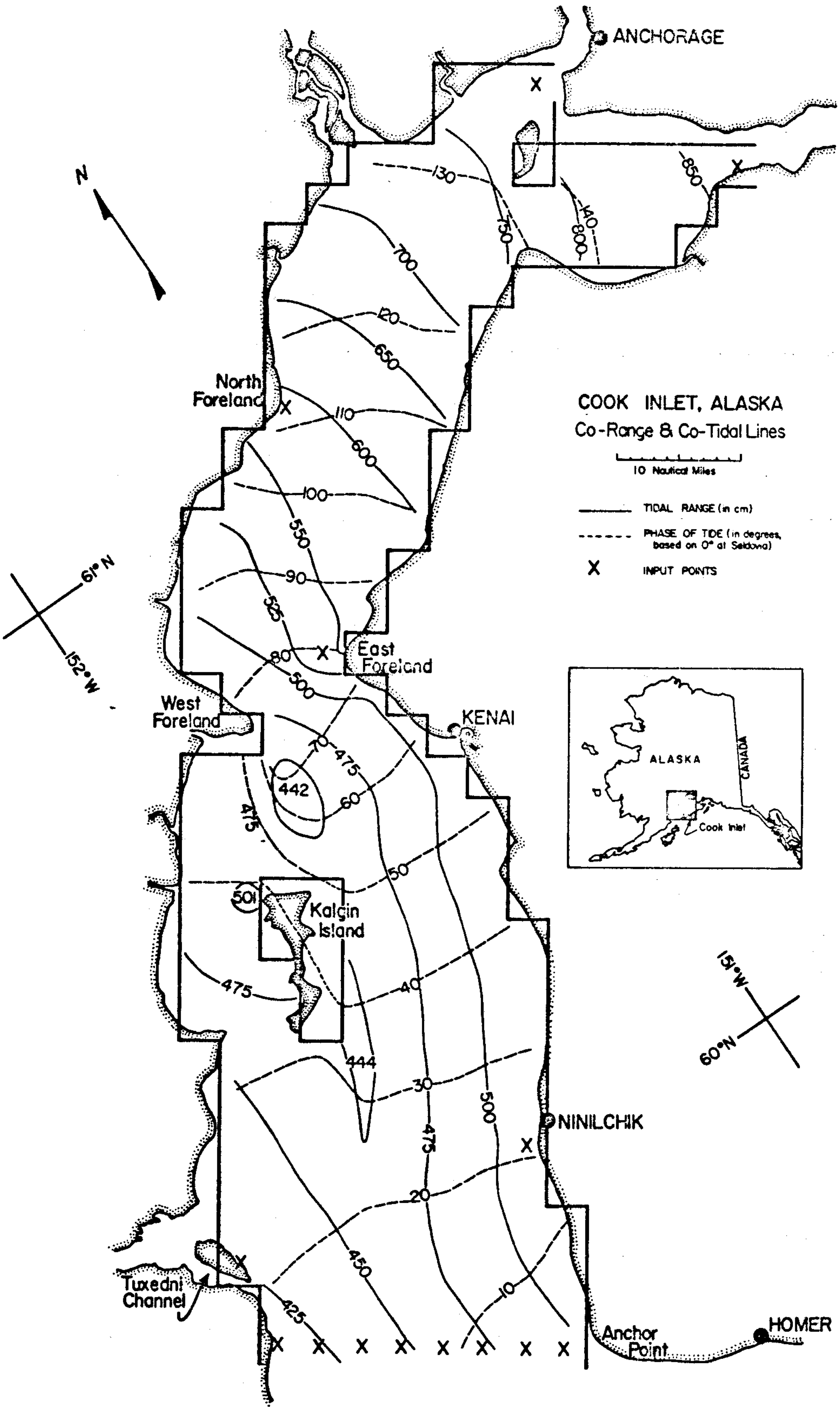

ANCHORAGE
N
North Foreland
COOK INLET, ALASKA
Co-Range & Co-Tidal Lines
10 Nautical Miles
TIDAL RANGE (in cm)
PHASE OF TIDE (in degrees, based on 0° at Seldovia)
X INPUT POINTS
61° N
152° W
East Foreland
West Foreland
KENAI
ALASKA
CANADA
Cook Inlet
Kalgin Island
151° W
60° N
NINILCHIK
Tuxedni Channel
Anchor Point
HOMER

Figure 9.4.

(Defant, 1960). The fact that the two sets of lines are approximately at right angles is an indication that friction probably does not play too important a part in South Cook Inlet.

A feature clearly observed in Figure 9.4 is the speeding up of the tidal wave on the west side of the inlet after Tuxedni Channel has been reached. The explanation for this is to be found in the bathymetry of the region. Depths of some 60 fathoms occur west of Kalgin Island while 20 fathoms is more typical for the part of the inlet lying between Kalgin Island and the Kenai Peninsula. Another point that is worth drawing attention to is that there will be scarcely any change of tidal range with increasing distance up the inlet (as far as the Forelands), not an increase with distance as one would have expected had standing wave behavior been assumed.

The tides of North Cook Inlet have the appearance of the more conventional standing wave. Considerable distortion from the frictionless case is present, as is evidenced by the co-tidal lines not being perpendicular to the co-range lines. If the amplitudes and phases are plotted in the appropriate fashion on Redfield's estuary tidal analysis diagram (Redfield, 1950), a value of about 3 results for μ, the damping coefficient. The reason for the strong frictional effects is certainly to be found in the shallow depths prevalent throughout North Cook Inlet.

Another result of interest is that as one proceeds up North Cook Inlet the difference in the amplitude of the tide across the inlet decreases. This is because the difference in phase between the maximum tide height and the maximum current is approaching 90 degrees. If slack water occurs when the tide is at its highest, there will be no Coriolis force and hence no slope of the water surface across the inlet at this instant.

Because of the fact that no attempt could be made to take into account the varying shore line as the mud flats become exposed, to provide the north end of the model with closed boundaries, or to include the effects of the tidal bore that is said to occur at certain times beyond the model limits in Turnagain Arm (U.S.C.& G.S., 1964), it is almost certain that the reality of the results decreases as one proceeds northwards.

A look at the output of the current analysis shows that the maximum depth-mean currents occur just to the south of the West Foreland - East Foreland narrows. They attain a maximum value of over 200 cms./second (i.e. more than 4 knots) and are counter clockwise. It is to be hoped that in the future a knowlege of the real current profile will be used to estimate the current at any depth, given solely the depth-mean current.

On observing the nature of the co-tidal lines in the region of the input at the southern end of the model, one is led to the conclusion that too great a phase difference was assumed to exist across the open boundary. It is likely that the difference should have been nearer 8 degrees and not 18 degrees, as was used. Furthermore it was probably a mistake to have assumed that the phase of the tide at the input point near Anchorage should have had the same phase as Anchorage. Being half way between Anchorage and Fire Island, the phase should probably have been 7 degrees or so smaller. Finally, on the subject of modifying future input data, it seems that the inclusion of input data for a point near the town of Kenai would have removed the 'awkward' shape of the co-range lines in this region. The predicted range for Kenai is 5.06 meters, some 35 cms. smaller than the tabulated mean range. This input data was specifically left out of the model so that a check would be available as to the veracity of the solution. One concludes from this that all available input data should be used near regions of complex shape.

CHAPTER X

CONCLUSIONS AND FUTURE WORK

1. Conclusions--.

A variable-geometry model has been described in this thesis that is oriented towards the general user. It is designed to stand alone but also to be made part of larger models such as general oceanographic prediction schemes. The method follows the earlier approaches of Hansen and Yuen, and uses Yuen's equations in an automated form. The method of solution is thus already well documented and examples of previous applications of the method may easily be referred to. A background to the solution of tides in inlets is given, and the means by which the finite difference equations are derived is covered step by step. The prospective user is shown clearly the means by which a particular inlet may be studied and how the input data is prepared. A simple example is covered in some detail with all the cards explained and sample computer outputs shown.

For the user's convenience a magnetic tape is prepared during the last tidal cycle computed, on which all heights and currents are stored; this is so that special types of analysis may be performed at later dates as desired. At the end of the last tidal cycle the tape is automatically analysed for tidal range and phase, and maximum and minimum currents: these being probably the most useful results of the computation. It is felt that this approach should make the model described particularly attractive to otherwise wary users.

Two applications to real inlets have been included in the thesis. They were to the Gulf of California and to Cook Inlet. It is the writer's opinion that these have provided a satisfactory test for the model.

2. Future work--.

The inability of the model to deal with mud flats points to the need for work in this area. Although it is tempting to suggest that modifications be made so as to adjust the inlet outline in units of (2 x grid interval) when necessary during the course of the solution, the nature of such a change might prove too gross to deal realistically with the situation. Before such an improvement can be made it seems that efforts should be directed to mathematical studies rather than towards the more tempting "experimental mathematics" approach. A deeper study of the part played by friction would be applicable to shallow regions such as Cook Inlet and the Bering Sea. Jeffreys (1920) has pointed out the importance of the Bering Sea when considering world-wide frictional dissipation for the M_2 tide.

BIBLIOGRAPHY

Abbott, M. B., 1966.
The Method of Characteristics,
American Elsevier, New York, 243 p.

Blondel, A., 1912.
Sur la Theorie des Marees dans un Canal. Appl. a la Mer Rouge,
Ann. Fac. Toulouse, 3.

Courtier, A., 1938.
Marees,
Serv. Hydr. Marine, Paris, 37 p.

Defant, A., 1920.
Die Gezeiten und Gezeiten Stromungen im Irischen Kanal,
Untersuchungen a.s.o., S. B. Weiner Akad. Wiss. (Math. Nature. Kl.),
129, 253.

Defant, A., 1960.
Physical Oceanography, Vol. 2,
Pergamon Press, New York, 598 p.

Dronkers, J. J., 1964.
Tidal Computations in Rivers and Coastal Waters,
North Holland Publishing Company, Amsterdam, 518 p.

Fisher, R. L., Rusnak, G. A., and F. P. Shepard, 1964.
Submarine Topography of the Gulf of California (Chart I),
American Association of Petroleum Geologists.

Grace, S. F., 1936.
Friction in the Tidal Currents of the Bristol Channel,
Geophys. Supp. M.N.R. Astron. Soc., 3, 388-395.

Hansen, W., 1952.
Gezeiten und Gezeitenstrome der Halbtagigen Hauptmondtide M_2 in der Nordsee,
Deutsche Hydr. Zeitschr., Erganzungsheft 1.

I.B.M., 1968.
IBM System/360 Disk Operating System: FORTRAN IV Programmer's Guide,
Form C 28 6397-0, Oct., 96 p.

Jeffreys, H., 1920.
Tidal Friction in Shallow Seas,
Phil. Trans. A. 221, 239.

Leendertse, J. J., 1967.
Aspects of the Computational Model for Long-Period Water Wave propagation,
Rand Memorandum R.M.5294-P.R., Delft, 89 p.

Lorentz, H. A., 1926.
Verslag Staatscommissie Zuiderzee 1918-1926 (Report of the Government Zuiderzee Commission),
Alg. Landsdrukkerij, The Hague.

Matthews, J. B., 1968.
Tides in the Gulf of California,
Thompson, D. A., Editor, Probable Environmental Impact of Heated Brine Effluents from a Nuclear Desalination Plant on the northern Gulf of California, University of Arizona report submitted to the Office of Saline Water, U.S. Department of the Interior, 41-50.

Proudman, J., 1953.
Dynamical Oceanography,
Methuen, London; J. Wiley, New York, 409 p.

Redfield, A., 1950.
The Analyses of Tidal Phenomena in Narrow Embayments,
No. 529, Papers in Phys. Ocean. and Meteor., MIT and Woods Hole Ocean. Inst., 11, no. 4, 36 p.

Richardson, A., 1922.
Weather Prediction by Numerical Process,
Cambridge University Press, 236 p.

Richtmyer, R. D., and K. W. Morton, 1967.
Difference Methods for Initial-Value Problems, 2nd. ed.,
Interscience, New York, 403 p.

Sterneck, R. v., 1914.
Uber die Gezeiten des Aegaischen Meeres,
Akad. Anz. Akad. Wiss. Wien, 10 Dec..

Stoker, J. J., 1957.
Water Waves,
Interscience, New York, 567 p.

Sverdrup, H. V., M. W. Johnson, and R. H. Fleming, 1942.
The Oceans,
Prentice-Hall, New York, 1087 p.

U.S.C.&G.S., 1964.
U.S. Coast Pilot No.9, Pacific and Arctic Coasts, Alaska, Cape Spencer to Beaufort Sea, 7th. ed.,
U.S. Government Printing Office, Washington, D.C., 348 p.

U.S.C.&G.S., 1968.
Tide Tables, West Coast, North and South America,
U.S. Government Printing Office, Washington, D.C., 224 p.

U.N.A.M., 1967.
Tablas de Prediccion de Mareas, Puertos del Oceano Pacifico,
Ap. 1, Parte B, Anales del Instituto de Geofisica, Universidad Nacional Autonoma de Mexico, 13, Mexico, 135 p.

Yuen, K. B., 1967.
The Effects of Tidal Barriers upon the M_2 Tide in the Bay of Fundy,
Manuscript Report Series, No. 5, Marine Sciences Branch,
Department of Energy, Mines and Resources, Ottawa, 146 p.

Zetler, B. D., and R. A. Cummings, 1967.
A Harmonic Method for Predicting Shallow-Water Tides,
J. Mar. Res., 25, 1, 103-114.

APPENDIX I

LISTING OF PROGRAM FOR VARIABLE-BOUNDARY TIDAL MODEL

```
C
C
C                  THIS BELONGS TO CHRIS MUNGALL
C                   INSTITUTE OF MARINE SCIENCE
C
C
          INTEGER*2 IU
          DIMENSION U1(65,30),Z1(65,30),H(65,30),IU(68,31)
          DIMENSION PR(20)
          COMMON MSUM,NSUM,DL,T,GEE,R,F,Y,PER,IPER,ISS,IS,DT,IDAY,IFIT,ICYC,
         1FIT,TIME,U1,Z1,H,IU,PR
          COMMON TIDE
          READ(1,1)IIDA
1         FORMAT(I2)
C             IIDA=NUMBER OF TIDAL CYCLES
          CALL OPSYS ('LOAD','PHASNME1')
          CALL INIT
          CALL OPSYS ('LOAD','PHASNME4')
          CALL PRINTD
          CALL OPSYS ('LOAD','PHASNME3')
67        IFIT=0
          ICYC=0
          FIT=IFIT
61        CONTINUE
C
C
C             UATV CALCULATION
          IFL=0
          L2=MSUM-1
          DO 2107 M=2,L2,2
          I=-1
2100      I=I+2
          IF(IU(M,I)-2)2104,2107,2100
C             IF IU=3, TREAT IT AS 0
C             IF IU LESS THAN 2, CHECK FOR 0 OR 1
C             IF IU=2, GO TO NEXT ROW
```

```
 2104 IF(IU(M,I))2100,2100,2101
 C        IF IU=1, CHECK IF IT IS LEFT OR RIGHT BOUNDARY
 2101 IF(IFL)2102,2102,2103
 C        IFL=0 INDICATES LEFT BOUNDARY, IFL=1 RIGHT
 2102 IML=I+2
 C        IF LEFT, SET LEFT (LOWER) LIMIT.  CHANGE IFL VALUE
 C        LEFT BOUNDARY -- VY=0
      IF(IU(M,I+2)-3)2125,2124,2124
 C124 V1(M,I)=V1(M,I+2)
 2124 U1(M,I+1)=U1(M,I+3)
 2125 CONTINUE
      IFL=1
      GO TO 2100
 2103 IMR=I-2
 C        IF RIGHT, SET RIGHT (UPPER) LIMIT.  CHANGE IFL VALUE
 C        RIGHT BOUNDARY -- VY=0
      IF(IU(M,I-2)-3)2127,2126,2126
 C126 V1(M,I)=V1(M,I-2)
 2126 U1(M,I+1)=U1(M,I-1)
 2127 CONTINUE
      IFL=0
 C        NOW WE HAVE LIMITS FOR NORMAL CALCULATION
      IF(IMR-IML)2108,2109,2109
 C        CHECK FOR SPECIAL CASE (IE UNUSUALLY NARROW)
 2109 DO 2106 N=IML,IMR,2
      Z1(M,N)=(Z1(M,N+1)+Z1(M,N-1))/2.
 C    H(M,N)=D(M,N)+Z1(M,N)
      H(M,N)=H(M,N+1)+Z1(M,N)
 2106 U1(M,N)=(U1(M+1,N+1)+U1(M-1,N+1)+U1(M-1,N-1)+U1(M+1,N-1))/4.
 C      THEN CALCULATE VALUES AT SIDES
      Z1(M,IML-2)=(Z1(M,IML-1)-(Z1(M,IML+1)/3.))*1.5
 C    H(M,IML-2)=D(M,IML-2)+Z1(M,IML-2)
      H(M,IML-2)=H(M,IML-1)+Z1(M,IML-2)
      Z1(M,IMR+2)=(Z1(M,IMR+1)-(Z1(M,IMR-1)/3.))*1.5
 C    H(M,IMR+2)=D(M,IMR+2)+Z1(M,IMR+2)
      H(M,IMR+2)=H(M,IMR+3)+Z1(M,IMR+2)
```

```
0045          U1(M,IMR+2)=U1(M+1,IMR+1)+U1(M-1,IMR+1)-U1(M,IMR)
0046          U1(M,IML-2)=U1(M+1,IML-1)+U1(M-1,IML-1)-U1(M,IML)
0047          GO TO 2100
        C          REPEAT PROCESS
        C          NARROW CASE
0048    2108  U1(M,IML-2)=(U1(M+1,IML-1)+U1(M-1,IML-1))/2.
0049          Z1(M,IML-2)=Z1(M,IML-1)
        C     H(M,IML-2)=D(M,IML-2)+Z1(M,IML-2)
0050          H(M,IML-2)=H(M,IML-1)+Z1(M,IML-2)
0051          U1(M,IML)=U1(M,IML-2)
0052          Z1(M,IML)=Z1(M,IML-1)
        C     H(M,IMR+2)=D(M,IMR+2)+Z1(M,IMR+2)
0053          H(M,IMR+2)=H(M,IMR+3)+Z1(M,IMR+2)
0054          GO TO 2100
0055    2107  CONTINUE
        C
        C          VATU CALCULATION
0056          IFL=0
0057          L1=NSUM-1
0058          DO 2117 N=2,L1,2
0059          I=-1
0060    2110  I=I+2
0061          IF(IU(I,N)-2)2114,2117,2110
        C          IF IU=3, TREAT IT AS 0
        C          IF IU LESS THAN 2, CHECK FOR 0 OR 1
        C          IF IU=2, GO TO NEXT COLUMN
0062    2114  IF(IU(I,N))2110,2110,2111
        C          IF IU=1, CHECK IF IT IS BOTTOM OR TOP BOUNDARY
0063    2111  IF(IFL)2112,2112,2113
        C          IFL=0 INDICATES BOTTOM BOUNDARY, IFL=1 TOP
0064    2112  IMB=I+2
        C          IF BOTTOM, SET BOTTOM (LOWER) LIMIT.  CHANGE IFL VALUE
        C          BOTTOM BOUNDARY -- UX=0
0065          IF(IU(I+2,N)-3)2123,2122,2122
0066    2122  U1(I,N)=U1(I+2,N)
0067    2123  CONTINUE
```

```
      IFL=1
      GO TO 2110
 2113 IMT=I-2
C          IF TOP, SET TOP (UPPER) LIMIT.  CHANGE IFL VALUE
C          TOP BOUNDARY -- UX=0
      IF(IU(I-2,N)-3)2121,2120,2120
 2120 U1(I,N)=U1(I-2,N)
 2121 CONTINUE
      IFL=0
C          NOW WE HAVE LIMITS FOR NORMAL CALCULATION
      IF(IMT-IMB)2118,2119,2119
C          CHECK FOR SPECIAL CASE (IE UNUSUALLY NARROW)
 2119 DO 2116 M=IMB,IMT,2
      Z1(M,N)=(Z1(M+1,N)+Z1(M-1,N))/2.
C     H(M,N)=D(M,N)+Z1(M,N)
      H(M,N)=H(M,N+1)+Z1(M,N)
C116  V1(M,N)=(V1(M+1,N+1)+V1(M-1,N+1)+V1(M-1,N-1)+V1(M+1,N-1))/4.
 2116 U1(M,N+1)=(U1(M+1,N+2)+U1(M-1,N+2)+U1(M-1,N)+U1(M+1,N))/4.
C          THEN CALCULATE VALUES AT SIDES
      Z1(IMB-2,N)=(Z1(IMB-1,N)-(Z1(IMB+1,N)/3.))*1.5
C     H(IMB-2,N)=D(IMB-2,N)+Z1(IMB-2,N)
      H(IMB-2,N)=H(IMB-2,N+1)+Z1(IMB-2,N)
      Z1(IMT+2,N)=(Z1(IMT+1,N)-(Z1(IMT-1,N)/3.))*1.5
C     H(IMT+2,N)=D(IMT+2,N)+Z1(IMT+2,N)
      H(IMT+2,N)=H(IMT+2,N+1)+Z1(IMT+2,N)
C     V1(IMT+2,N)=V1(IMT+1,N+1)+V1(IMT+1,N-1)-V1(IMT,N)
      U1(IMT+2,N+1)=U1(IMT+1,N+2)+U1(IMT+1,N)-U1(IMT,N+1)
C     V1(IMB-2,N)=V1(IMB-1,N+1)+V1(IMB-1,N-1)-V1(IMB,N)
      U1(IMB-2,N+1)=U1(IMB-1,N+2)+U1(IMB-1,N)-U1(IMB,N+1)
      GO TO 2110
C          REPEAT PROCESS
C          NARROW CASE
C118  V1(IMB-2,N)=(V1(IMB-1,N+1)+V1(IMB-1,N-1))/2.
 2118 U1(IMB-2,N+1)=(U1(IMB-1,N+2)+U1(IMB-1,N))/2.
      Z1(IMB-2,N)=Z1(IMB-1,N)
C     H(IMB-2,N)=D(IMB-2,N)+Z1(IMB-2,N)
```

```
0089           H(IMB-2,N)=H(IMB-2,N+1)+Z1(IMB-2,N)
         C     V1(IMB,N)=V1(IMB-2,N)
0090           U1(IMB,N+1)=U1(IMB-2,N+1)
0091           Z1(IMB,N)=Z1(IMB-1,N)
         C     H(IMT+2,N)=D(IMT+2,N)+Z1(IMT+2,N)
0092           H(IMT+2,N)=H(IMT+2,N+1)+Z1(IMT+2,N)
0093           GO TO 2110
0094     2117  CONTINUE
0095           IF(IIDA-IDAY)8999,8999,8998
0096     8999  CALL WRITER(9,U1,7800)
0097           CALL WRITER(9,Z1,7800)
0098     8998  CONTINUE
         C
         C
         C PRINT-OUT ONLY 12 TIMES/TIDAL CYCLE
0099           IF(ISS-IS)81,81,82
0100     81    IS=0
0101           IF(IIDA-IDAY)2001,2001,2002
0102     2002  IF(ICYC)2001,2001,82
0103     2001  TIME=FIT*T/(3600.*PER)
         C
         C
0104           CALL OPSYS ('LOAD','PHASNME2')
0105           CALL WRITE
0106           CALL OPSYS ('LOAD','PHASNME3')
0107           ICYC=ICYC+1
0108     82    IS=IS+2
0109           IFIT=IFIT+2
0110           FIT=IFIT
0111           CALL INPUT
         C
         C
0112           CALL UVZ
0113           CALL INPUT
         C         THIS REPLACES DESTROYED INPUT-POINT DATA
0114           IF(IPER-IFIT)80,80,61
```

```
0115     80    IDAY=IDAY+1
         C         THE NEXT 6 INSTRUCTIONS ARE INCLUDED TO LIMIT THE LOSS OF
         C          DATA TO ONE CYCLE ONLY , IN THE EVENT THAT THE PROGRAM IS
         C          CANCELLED DUE TO EXTERNAL CAUSES.  TO RESTART AT THE END OF
         C          THE LAST CYCLE COMPLETED, THE INIT PROGRAM WILL HAVE TO BE
         C          ALTERED SLIGHTLY.
0116           WRITE(8,1)IDAY
0117           CALL WRITER(8,IU,4216)
0118           CALL WRITER(8,Z1,7800)
0119           CALL WRITER(8,U1,7800)
0120           CALL WRITER(8,H,7800)
0121           REWIND 8
0122           IF(IIDA+1-IDAY)83,83,67
0123     83    REWIND 9
0124           CALL EXIT
0125           END
```

```
      SUBROUTINE INPUT
      INTEGER*2 IU
      DIMENSION U1(65,30),Z1(65,30),H(65,30),IU(68,31)
      DIMENSION PR(20)
      COMMON MSUM,NSUM,DL,T,GEE,R,F,Y,PER,IPER,ISS,IS,DT,IDAY,IFIT,ICYC,
     1FIT,TIME,U1,Z1,H,IU,PR
      COMMON TIDE
      Z1(2,4)=0.130*COS(6.28318*((FIT/PER)-0.000))
      Z1(2,6)=0.133*COS(6.28318*((FIT/PER)-0.000))
      Z1(2,8)=0.136*COS(6.28318*((FIT/PER)-0.000))
      Z1(2,10)=0.140*COS(6.28318*((FIT/PER)-0.000))
      Z1(32,12)=1.570*COS(6.28318*((FIT/PER)-0.297))
      RETURN
      END
```

```
      SUBROUTINE INIT
      INTEGER*2 IU
      DIMENSION U1(65,30),Z1(65,30),H(65,30),IU(68,31)
      DIMENSION PR(20)
      COMMON MSUM,NSUM,DL,T,GEE,R,F,Y,PER,IPER,ISS,IS,DT,IDAY,IFIT,ICYC,
     1FIT,TIME,U1,Z1,H,IU,PR
      COMMON TIDE
C
C
C          SET VELOCITIES TO ZERO
      DO 2009 M=1,65
      DO 2009 N=1,30
      Z1(M,N)=0.
      H(M,N)=0.
C009  V1(M,N)=0.
2009  U1(M,N)=0.
C
      READ(1,105)MSUM
105   FORMAT(I2)
      READ(1,105)NSUM
      READ(1,1)DL
      DLL=DL/1000.
      WRITE(3,7000)DLL
7000  FORMAT('1',50X,'GRID INTERVAL=',F6.2,'KILOMETERS')
      READ(1,1)T
C  T=PERIOD IN HOURS
      WRITE(3,7001)T
7001  FORMAT('0',50X,'TIDAL PERIOD=',F6.2,'HOURS')
      T=T*3600.0
      GEE=9.81
C GEE IN M/SEC**
C R=FRICTION COEFFICIENT
      READ(1,1)R
      WRITE(3,7002)R
7002  FORMAT('0',50X,'FRICTION COEFFICIENT=',F6.4)
      READ(1,1)ALAT
```

```
      WRITE(3,7003)ALAT
7003  FORMAT('0',50X,'LATITUDE=',F4.1,'DEGREES')
      PHI=ALAT*3.1416/180.0
      F=(4.0*3.1416*SIN(PHI))/(24.0*3600.0)
C F=CORIOLIS PARAMETER, IN RAD/SEC
      WRITE(3,7004)F
7004  FORMAT('0',50X,'CORIOLIS PARAMETER=',F10.8,'RADIANS/SECOND')
      WRITE(3,7005)
7005  FORMAT('0',50X,'FOLLOWING PRINTOUTS ARE IN METER-SECOND UNITS')
C STABILIZATION FACTOR
      Y=0.99
      READ(1,1)PER
C PER=NUMBER OF TIME INTERVALS/TIDAL PERIOD
      IPER=PER
      IZINT=IPER
      ISS=PER/12.
C         IE PRINT OUT 12 TIMES PER TIDAL CYCLE DURING LAST CYCLE
      IS=ISS
      DT=T/PER
C DT=TIME INCREMENT IN SEDONDS
      WRITE(9,77)MSUM
      WRITE(9,77)NSUM
77    FORMAT(I2)
      WRITE(9,78)DL
78    FORMAT(F12.4)
      WRITE(9,999)IZINT
999   FORMAT(I4)
      WRITE(9,78)T
C
C         READ POINTS AT WHICH U=0
      L1=NSUM-1
      DO 2020 N=2,L1,2
2020  READ(1,104)(IU(M,N),M=1,68)
104   FORMAT(68I1)
C
C         READ POINTS AT WHICH V=0
```

```
          L2=MSUM-1
          DO 2010 M=2,L2,2
    2010  READ(1,100)(IU(M,N),N=1,31)
    100   FORMAT(31I1)
          WRITE(9)((IU(M,N),M=1,67),N=1,31)
    C
    C
    C ALL HEIGHTS IN METERS
    C            READ DATV
          IFL=0
          L2=MSUM-1
          DO 2407 M=2,L2,2
          I=-1
    2400  I=I+2
          IF(IU(M,I)-2)2404,2407,2400
    2404  IF(IU(M,I))2400,2400,2401
    2401  IF(IFL)2402,2402,2403
    2402  IML=I+2
          IFL=1
          GO TO 2400
    2403  IMR=I-2
          IFL=0
          IF(IMR-IML)2408,2409,2409
    C409  READ(1,1)D(M,IML-2)
    2409  READ(1,1)H(M,IML-1)
    1     FORMAT(F12.4)
          H(M,IML-1)=H(M,IML-1)*1.8288
          DO 2406 N=IML,IMR,2
          READ(1,1)H(M,N+1)
    2406  H(M,N+1)=H(M,N+1)*1.8288
    C     READ(1,1)D(M,IMR+2)
          READ(1,1)H(M,IMR+3)
          H(M,IMR+3)=H(M,IMR+3)*1.8288
          GO TO 2400
    C408  READ(1,1)D(M,IML-2)
    2408  READ(1,1)H(M,IML-1)
```

```
            H(M,IML-1)=H(M,IML-1)*1.8288
C           READ(1,1)D(M,IMR+2)
            READ(1,1)H(M,IMR+3)
            H(M,IMR+3)=H(M,IMR+3)*1.8288
            GO TO 2400
 2407       CONTINUE
C
C
C                  READ DATU
            IFL=0
            L1=NSUM-1
            DO 2517 N=2,L1,2
            I=-1
 2510       I=I+2
            IF(IU(I,N)-2)2514,2517,2510
 2514       IF(IU(I,N))2510,2510,2511
 2511       IF(IFL)2512,2512,2513
 2512       IMB=I+2
            IFL=1
            GO TO 2510
 2513       IMT=I-2
            IFL=0
            IF(IMT-IMB)2518,2519,2519
C519        READ(1,1)D(IMB-2,N)
 2519       READ(1,1)H(IMB-2,N+1)
            H(IMB-2,N+1)=H(IMB-2,N+1)*1.8288
            DO 2516 M=IMB,IMT,2
            READ(1,1)H(M,N+1)
 2516       H(M,N+1)=H(M,N+1)*1.8288
C           READ(1,1)D(IMT+2,N)
            READ(1,1)H(IMT+2,N+1)
            H(IMT+2,N+1)=H(IMT+2,N+1)*1.8288
            GO TO 2510
C518        READ(1,1)D(IMB-2,N)
 2518       READ(1,1)H(IMB-2,N+1)
            H(IMB-2,N+1)=H(IMB-2,N+1)*1.8288
```

```
C        READ(1,1)D(IMT+2,N)
         READ(1,1)H(IMT+2,N+1)
         H(IMT+2,N+1)=H(IMT+2,N+1)*1.8288
         GO TO 2510
2517     CONTINUE
C
C
C
C
C               READ Z INITIAL
         IFL=0
         L2=MSUM-1
         DO 2607 M=2,L2,2
         I=-1
2600     I=I+2
         IF(IU(M,I)-2)2604,2607,2600
2604     IF(IU(M,I))2600,2600,2601
2601     IF(IFL)2602,2602,2603
2602     IML=I+1
         IFL=1
         GO TO 2600
2603     IMR=I-1
         IFL=0
         DO 2606 N=IML,IMR,2
2606     READ(1,1)Z1(M,N)
         GO TO 2600
2607     CONTINUE
C
         RETURN
         END
```

```
      SUBROUTINE WRITE
      INTEGER*2 IU
      DIMENSION U1(65,30),Z1(65,30),H(65,30),IU(68,31)
      DIMENSION PR(20)
      COMMON MSUM,NSUM,DL,T,GEE,R,F,Y,PER,IPER,ISS,IS,DT,IDAY,IFIT,ICYC,
     1FIT,TIME,U1,Z1,H,IU,PR
      COMMON TIDE
C         WRITE Z'S
      WRITE(3,110)
110   FORMAT('1',63X,'Z-VALUES')
      WRITE(3,4)TIME
4     FORMAT(' ','CONDITIONS AFTER',2X,F5.2,'HOURS')
      WRITE(3,5)IDAY
5     FORMAT(' ','NUMBER OF TIDAL CYCLES COMPLETED',2X,I2)
      WRITE(3,102)
102   FORMAT('0','          N= 1    N= 2    N= 3    N= 4    N= 5    N= 6    N= 7
     1  N= 8    N= 9    N=10    N=11    N=12    N=13    N=14    N=15    N=16    N
     2=17    N=18')
      DO 5002 J=1,MSUM
      M=MSUM+1-J
5002  WRITE(3,101)M,(Z1(M,N),N=1,18)
101   FORMAT(' ','M=',I2,1X,18(1X,F6.2))
      IF(NSUM-18)5004,5004,5003
5003  WRITE(3,110)
      WRITE(3,4)TIME
      WRITE(3,5)IDAY
      WRITE(3,103)
103   FORMAT('0','  N=19    N=20    N=21    N=22    N=23    N=24    N=25    N=2
     16    N=27    N=28    N=29    N=30')
      DO 5005 J=1,MSUM
      M=MSUM+1-J
5005  WRITE(3,106)(Z1(M,N),N=19,29)
106   FORMAT(' ',11(1X,F6.2))
5004  CONTINUE
C
C
```

```
C          WRITE U'S
           WRITE(3,113)
 113       FORMAT('1',63X,'U-VALUES')
           WRITE(3,4)TIME
           WRITE(3,5)IDAY
           WRITE(3,102)
           M=MSUM
 7008      DO 7012 I=1,18
 7012      PR(I)=0.
           DO 7010 N=2,18,2
 7010      PR(N)=U1(M,N)
           WRITE(3,101)M,(PR(N),N=1,18)
           M=M-1
           IF(M)7016,7016,7009
C          7009 INDICATES THAT M IS EVEN
 7009      DO 7013 I=1,18
 7013      PR(I)=0.
           DO 7014 N=1,17,2
 7014      PR(N)=U1(M,N)
           WRITE(3,101)M,(PR(N),N=1,18)
           M=M-1
           GO TO 7008
 7016      CONTINUE
           IF(NSUM-18)7104,7104,7103
 7103      WRITE(3,113)
           WRITE(3,4)TIME
           WRITE(3,5)IDAY
           WRITE(3,103)
           M=MSUM
 7108      DO 7112 I=1,12
 7112      PR(I)=0.
           DO 7110 N=2,10,2
 7110      PR(N)=U1(M,N+18)
           WRITE(3,106)(PR(J),J=1,11)
           M=M-1
           IF(M)7116,7116,7109
```

```
0064      7109  DO 7113 I=1,12
0065      7113  PR(I)=0.
0066            DO 7114 N=1,11,2
0067      7114  PR(N)=U1(M,N+18)
0068            WRITE(3,106)(PR(J),J=1,11)
0069            M=M-1
0070            GO TO 7108
0071      7116  CONTINUE
0072      7104  CONTINUE
          C
          C
          C           WRITE V'S
0073            WRITE(3,112)
0074      112   FORMAT('1',63X,'V-VALUES')
0075            WRITE(3,4)TIME
0076            WRITE(3,5)IDAY
0077            WRITE(3,102)
0078            M=MSUM
0079      6008  DO 6012 I=1,18
0080      6012  PR(I)=0.
0081            DO 6010 N=2,18,2
          C010  PR(N)=V1(M,N)
0082      6010  PR(N)=U1(M,N+1)
0083            WRITE(3,101)M,(PR(N),N=1,18)
0084            M=M-1
0085            IF(M)6016,6016,6009
          C           6009 INDICATES THAT M IS EVEN
0086      6009  DO 6013 I=1,18
0087      6013  PR(I)=0.
0088            DO 6014 N=1,17,2
          C014  PR(N)=V1(M,N)
0089      6014  PR(N)=U1(M,N+1)
0090            WRITE(3,101)M,(PR(N),N=1,18)
0091            M=M-1
0092            GO TO 6008
0093      6016  CONTINUE
```

```
      IF(NSUM-18)6104,6104,6103
6103  WRITE(3,112)
      WRITE(3,4)TIME
      WRITE(3,5)IDAY
      WRITE(3,103)
      M=MSUM
6108  DO 6112 I=1,12
6112  PR(I)=0.
      DO 6110 N=2,10,2
C110  PR(N)=V1(M,N+18)
6110  PR(N)=U1(M,N+19)
      WRITE(3,106)(PR(J),J=1,11)
      M=M-1
      IF(M)6116,6116,6109
6109  DO 6113 I=1,12
6113  PR(I)=0.
      DO 6114 N=1,11,2
C114  PR(N)=V1(M,N+18)
6114  PR(N)=U1(M,N+19)
      WRITE(3,106)(PR(J),J=1,11)
      M=M-1
      GO TO 6108
6116  CONTINUE
6104  CONTINUE
2     FORMAT(' ',I3)
3     FORMAT('1')
      RETURN
      END
```

```
      SUBROUTINE UVZ
      INTEGER*2 IU
      DIMENSION U1(65,30),Z1(65,30),H(65,30),IU(68,31)
      DIMENSION PR(20)
      COMMON MSUM,NSUM,DL,T,GEE,R,F,Y,PER,IPER,ISS,IS,DT,IDAY,IFIT,ICYC,
     1FIT,TIME,U1,Z1,H,IU,PR
      COMMON TIDE
C          U-POINT CALCULATION
      IFL=0
      L1=NSUM-1
      DO 3117 N=2,L1,2
      I=-1
3110  I=I+2
      IF(IU(I,N)-2)3114,3117,3110
C          IF IU=3, TREAT IT AS 0
C          IF IU LESS THAN 2, CHECK FOR 0 OR 1
C          IF IU=2 GO TO NEXT COLUMN
3114  IF(IU(I,N))3110,3110,3111
C          IF IU=1, CHECK IF IT IS BOTTOM OR TOP BOUNDARY
3111  IF(IFL)3112,3112,3113
C          IFL=0 INDICATES BOTTOM BOUNDARY, IFL=1 TOP
3112  IMB=I+2
C          IF BOTTOM, SET BOTTOM (LOWER) LIMIT.  CHANGE IFL VALUE
      IFL=1
      GO TO 3110
3113  IMT=I-2
C          IF TOP, SET TOP (UPPER) LIMIT.  CHANGE IFL VALUE
      IFL=0
C          NOW WE HAVE LIMITS FOR CALCULATION
      IF(IMT-IMB)3118,3119,3119
C          CHECK FOR SPECIAL CASE (IF UNUSUALLY NARROW)
3119  DO 3116 M=IMB,IMT,2
C CALCULATION OF U AT (M,N)
      ZXATU=(Z1(M+1,N)-Z1(M-1,N))/(2.*DL)
C V1, U1 SHOULD BE FOR TIME (T) -- HERE THEY ARE TAKEN FOR TIME (T-1)
C STABILIZATION OF LEADING U-TERM
```

```
0023            USTAB=(Y*U1(M,N))+((1.-Y)*(U1(M+1,N+1)+U1(M-1,N+1)
              1    +U1(M-1,N-1)+U1(M+1,N-1))/4.)
0024      3116  U1(M,N)=USTAB+(2.*DT*((-USTAB*R*SQRT((U1(M,N)*U1(M,N))
          C     1       +(V1(M,N)*V1(M,N)))/H(M,N))+(F*V1(M,N))-(GEE*ZXATU)))
                1       +(U1(M,N+1)*U1(M,N+1)))/H(M,N))+(F*U1(M,N+1))-(GEE*ZXATU)))
          C END OF U AT (M,N) CALCULATION
0025          . GO TO 3110
          C          NARROW CASE
0026      3118  GO TO 3110
          C          IN NARROW CASE, NO U-POINT CALCULATION IS POSSIBLE
0027      3117  CONTINUE
          C
          C
          C          V-POINT CALCULATION
0028            IFL=0
0029            L2=MSUM-1
0030            DO 3107 M=2,L2,2
0031            I=-1
0032      3100  I=I+2
0033            IF(IU(M,I)-2)3104,3107,3100
          C          IF IU=3, TREAT IT AS 0
          C          IF IU LESS THAN 2, CHECK FOR 0 OR 1
          C          IF IU=2, GO TO NEXT ROW
0034      3104  IF(IU(M,I))3100,3100,3101
          C          IF IU=1, CHECK IF IT IS LEFT OR RIGHT BOUNDARY
0035      3101  IF(IFL)3102,3102,3103
          C          IFL=0 INDICATES LEFT BOUNDARY, IFL=1 RIGHT
0036      3102  IML=I+2
          C          IF LEFT, SET LEFT (LOWER) LIMIT.  CHANGE IFL VALUE
0037            IFL=1
0038            GO TO 3100
0039      3103  IMR=I-2
          C          IF RIGHT, SET RIGHT (UPPER) LIMIT.  CHANGE IFL VALUE
0040            IFL=0
          C     .    NOW WE HAVE LIMITS FOR CALCULATION
0041            IF(IMR-IML)3108,3109,3109
```

```
                 C          CHECK FOR SPECIAL CASE (IE UNUSUALLY NARROW)
0042             3109  DO 3106 N=IML,IMR,2
                 C CALCULATION OF V AT (M,N)
0043                   ZYATV=(Z1(M,N-1)-Z1(M,N+1))/(2.*DL)
                 C V1, U1 SHOULD BE FOR TIME (T) -- HERE THEY ARE TAKEN FOR TIME (T-1)
                 C STABILIZATION OF LEADING V-TERM
                 C     VSTAB=(Y*V1(M,N))+((1.-Y)*(V1(M+1,N+1)+V1(M-1,N+1)
0044                   VSTAB=(Y*U1(M,N+1))+((1.-Y)*(U1(M+1,N+2)+U1(M-1,N+2)
                 C    1   +V1(M-1,N-1)+V1(M+1,N-1))/4.)
                      1   +U1(M-1,N)+U1(M+1,N))/4.)
                 C106  V1(M,N)=VSTAB+(2.*DT*((-VSTAB*R*SQRT((U1(M,N)*U1(M,N))
0045             3106  U1(M,N+1)=VSTAB+(2.*DT*((-VSTAB*R*SQRT((U1(M,N)*U1(M,N))
                 C    1      +(V1(M,N)*V1(M,N)))/H(M,N))-(F*U1(M,N))-(GEE*ZYATV)))
                      1      +(U1(M,N+1)*U1(M,N+1)))/H(M,N))-(F*U1(M,N))-(GEE*ZYATV)))
                 C END OF V AT (M,N) CALCULATION
0046                   GO TO 3100
0047             3108  GO TO 3100
                 C          NARROW CASE
                 C       IN NARROW CASE, NO V-POINT CALCULATION IS POSSIBLE
0048             3107  CONTINUE
                 C
                 C
                 C          Z-POINT CALCULATION
                 C          NOTEO VALUES ARE CALCULATED AT INPUT POINTS--THESE ARE FALSE
0049                   IFL=0
0050                   L2=MSUM-1
0051                   DO 4107 M=2,L2,2
0052                   I=-1
0053             4100  I=I+2
0054                   IF(IU(M,I)-2)4104,4107,4100
                 C          IF IU=3, TREAT IT AS 0
                 C          IF IU LESS THAN 2, CHECK FOR 0 OR 1
                 C          IF IU=2, GO TO NEXT ROW
0055             4104  IF(IU(M,I))4100,4100,4101
                 C          IF IU=1, CHECK IF IT IS LEFT OR RIGHT BOUNDARY
0056             4101  IF(IFL)4102,4102,4103
```

```
C           IFL=0 INDICATES LEFT BOUNDARY, IFL=1 RIGHT
4102  IML=I+1
C          IF LEFT, SET LEFT (LOWER) LIMIT.  CHANGE IFL VALUE
      IFL=1
      GO TO 4100
4103  IMR=I-1
C           IF RIGHT, SET RIGHT (UPPER) LIMIT.  CHANGE IFL VALUE
      IFL=0
C           NOW WE HAVE LIMITS FOR CALCULATION
      DO 4106 N=IML,IMR,2
C CALCULATION OF Z AT (M,N)
      HUX=((H(M+1,N)*U1(M+1,N))-(H(M-1,N)*U1(M-1,N)))/(DL*2.)
C     HVY=((H(M,N-1)*V1(M,N-1))-(H(M,N+1)*V1(M,N+1)))/(DL*2.)
      HVY=((H(M,N-1)*U1(M,N))-(H(M,N+1)*U1(M,N+2)))/(DL*2.)
C           STABILIZATION OF Z1
      Z1(M,N)=(Y*Z1(M,N))+((1.-Y)*(Z1(M+1,N)+Z1(M-1,N)
     1   +Z1(M,N-1)+Z1(M,N+1))/4.)
4106  Z1(M,N)=Z1(M,N)-(2.*DT*(HUX+HVY))
C HUX AND HVY SHOULD INVOLVE Z2 VALUES, BUT HERE THEY ARE
C APPROXIMATED BY Z1.
C END OF Z-CALCULATION
      GO TO 4100
4107  CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PRINTD
      INTEGER*2 IU
      DIMENSION U1(65,30),Z1(65,30),H(65,30),IU(68,31)
      DIMENSION PR(20)
      COMMON MSUM,NSUM,DL,T,GEE,R,F,Y,PER,IPER,ISS,IS,DT,IDAY,IFIT,ICYC,
     1FIT,TIME,U1,Z1,H,IU,PR
      COMMON TIDE
C         WRITE D'S
      WRITE(3,111)
  111 FORMAT('1','                                                 DEPTH-VALUES')
      WRITE(3,102)
  102 FORMAT('0','         N= 1     N= 2     N= 3     N= 4     N= 5     N= 6     N= 7
     1  N= 8     N= 9     N=10     N=11     N=12     N=13     N=14     N=15     N=16     N
     2=17    N=18')
      M=MSUM
 5008 DO 5012 I=1,18
 5012 PR(I)=0.
      DO 5010 N=2,18,2
C010  PR(N)=D(M,N)
 5010 PR(N)=H(M,N+1)
      WRITE(3,108)M,(PR(N),N=1,18)
  108 FORMAT(' ','M=',I2,1X,18(1X,F6.1))
      M=M-1
      IF(M)5016,5016,5009
C         5009 INDICATES THAT M IS EVEN
 5009 DO 5013 I=1,18
 5013 PR(I)=0.
      DO 5014 N=1,17,2
C014  PR(N)=D(M,N)
 5014 PR(N)=H(M,N+1)
      WRITE(3,108)M,(PR(N),N=1,18)
      M=M-1
      GO TO 5008
 5016 CONTINUE
      IF(NSUM-18)5104,5104,5103
 5103 WRITE(3,111)
```

```
      WRITE(3,103)
  103 FORMAT('0',' N=19   N=20
     16  N=27   N=28   N=29   N=
      M=MSUM
 5108 DO 5112 I=1,12
 5112 PR(I)=0.
      DO 5110 N=2,10,2
C110  PR(N)=D(M,N+18)
 5110 PR(N)=H(M,N+19)
      WRITE(3,109)(PR(J),J=1,11)
  109 FORMAT(' ',11(1X,F6.1))
      M=M-1
      IF(M)5116,5116,5109
 5109 DO 5113 I=1,12
 5113 PR(I)=0.
      DO 5114 N=1,11,2
C114  PR(N)=D(M,N+18)
 5114 PR(N)=H(M,N+19)
      WRITE(3,109)(PR(J),J=1,11)
      M=M-1
      GO TO 5108
 5116 CONTINUE
 5104 CONTINUE
      IDAY=0
      RETURN
      END
```

N=21 N=22 N=23 N=24 N=25 N=2
=30')

APPENDIX II

LISTING OF DATA COMPRESSION SUBROUTINE

Note: This subroutine is required by the tidal model program and the two analysis programs.

```
*  THIS ASSEMBLER SUBROUTINE CAN BE USED TO READ OR WRITE LARGE TAPE
*  BLOCKS BY A FORTRAN PROGRAM.  BEFORE CALLING THE SUBROUTINE FOR
*  WRITING, THE USER MUST 'WRITE' AT LEAST ONCE ON TO THE TAPE TO
*INSURE THAT THE TAPE IS PROPERLY OPENED.  NATURALLY THE TAPE WHEN
*  READ BACK, ALSO MUST 'READ' THE TAPE FOR THE SAME REASON.  WHEN
*  FINISHED WRITING A TAPE WITH THIS SUBROUTINE, THE USER MUST 'END-
*  FILE' OR 'REWIND' THE TAPE TO CLOSE IT PROPERLY.
*
*  THE FORMAT FOR THE FORTRAN CALL TO WRITE A RECORD (ASSUMED TO BE
*  A LARGE ARRAY OF DIMENSION (100,10)) ON DATA SET REFERENCE =5
* WOULD BE
*       CALL WRITER (5,ARRAY,4000)
*
*  TO READ THE ARRAY, ONE COULD CODE
*       CALL READER(6,BARRAY,4000)
*
*
*   NOTES.
*  THE FIRST ARGUMENT SPECIFIES THE DATA SET REFERENCE =.  IT MAY BE
*  A CONSTANT OR A FIXED POINT VARIABLE CONTAINING THE DATA SET
*  REFERENCE =.
*
*  ANY NUMBER OF VARIABLES OR ARRAYS MAY BE WRITTEN.  SPECIFY MERELY
*  IN THE SECOND AND THIRD ARGUMENTS THE NAME OF THE FIRST VARIABLE TO
*  BE WRITTEN AND THE ENTIRE LENGTH OF THE VARIABLES TO BE WRITTEN.
*  IT MAY BE NECESSARY TO REFER TO THE STORAGE MAP TO DETERMINE
*  WHICH VARIABLE IS ACTUALLY FIRST IN CORE AND WHAT THE ACTUAL
*  LENGTH IS.
*  THE THIRD ARGUMENT REPRESENTS THE NUMBER OF BYTES TO BE WRITTEN.
*  FORTRAN WORDS OF SINGLE PRECISION CONTAIN 4 BYTES EACH, WHILE
```

```
*  DOUBLE PRECISION VARIABLES
TAPEIO   START 0
WRITER   EQU *
         ENTRY WRITER
         USING *,15
         SAVE  (14,12)
         LA    5,1
GO       LM    2,4,0(1)
         L     2,0(2)
*
         L     4,0(4)
         STH   4,CCWPTR+6
         ST    3,CCWPTR
         STC   5,CCWPTR
         SH    2,=H'3'
         STC   2,CCB+7
         EXCP  CCB
         WAIT  CCB
         RETURN (14,12)
READER   EQU   *
         ENTRY READER
         SAVE  (14,12)
         LA    5,2
         LA    9,READER-WRITER
      SR       15,9
         B     GO
         DS    0F
CCB      CCB   SYS000,CCWPTR
CCWPTR   CCW   0,0,X'20',0
         END
```

CONTAIN 8 BYTES.

```
ESTABLISH ADDRESSABILITY
SAVE ALL FORTRAN REGISTERS
LOAD WRITE OP CODE
LOAD PARM POINTERS
R2= DS REF NO.
R3=  A(IO AREA)
R4=  LENGTH
STORE LENGTH
STORE ADDRESS
STORE OP CODE
GET SYS NO. FROM DS REF NO.
STORE IN CCB
DO I/O OPERATION
WAIT FOR COMPLETION
RETURN

SAVE REGISTERS
LOAD READER OP CODE
GET DIFFERENCE
'TWEEK' BASE REGISTER
```

APPENDIX III

LISTING OF HEIGHT AND CURRENT ANALYSIS PROGRAMS

```
      INTEGER*2 IU
      DIMENSION IU(68,31),Z1(65,30),ZMAX(65,30),ZMIN(65,30)
      READ(9,77)MSUM
      READ(9,77)NSUM
77    FORMAT(I2)
      READ(9,78)DL
C          DL=GRID INTERVAL IN METERS
      DL=DL/1000.
      READ(9,999)IZINT
999   FORMAT(I4)
      READ(9,78)TT
      TT=TT/3600.0
78    FORMAT(F12.4)
      ZINT=IZINT
      PH=360./ZINT
      DO 9 M=1,65
      DO 9 N=1,30
      ZMAX(M,N)=10000000.0
9     ZMIN(M,N)=10000000.0
      READ(9)((IU(M,N),M=1,67),N=1,31)
      IFL=0
      L2=MSUM-1
      DO 8107 M=2,L2,2
      I=-1
8100  I=I+2
      IF(IU(M,I)-2)8104,8107,8100
8104  IF(IU(M,I))8100,8100,8101
8101  IF(IFL)8102,8102,8103
8102  IML=I+1
      IFL=1
      GO TO 8100
8103  IMR=I-1
      IFL=0
      DO 8884 N=IML,IMR,2
      ZMAX(M,N)=0.0
8884  ZMIN(M,N)=0.0
```

```
              GO TO 8100
      8107    CONTINUE
              T=0.
              IT=0
      15      CALL READER(9,Z1,7800)
      C            READ U1
              CALL READER(9,Z1,7800)
      C            READ Z1
              IFL=0
              L2=MSUM-1
              DO 4107 M=2,L2,2
              I=-1
      4100    I=I+2
              IF(IU(M,I)-2)4104,4107,4100
      4104    IF(IU(M,I))4100,4100,4101
      4101    IF(IFL)4102,4102,4103
      4102    IML=I+1
              IFL=1
              GO TO 4100
      4103    IMR=I-1
              IFL=0
              DO 13 N=IML,IMR,2
              IF(Z1(M,N)-ZMAX(M,N))10,10,11
      11      ZMAX(M,N)=Z1(M,N)
              ZMAX(M-1,N)=T*PH
      C            PUT ASSOCIATED PHASE BELOW Z
              ZMAX(M,N+1)=0.0
              ZMAX(M-1,N+1)=0.0
      10      IF(Z1(M,N)-ZMIN(M,N))12,13,13
      12      ZMIN(M,N)=Z1(M,N)
              ZMIN(M-1,N)=T*PH
              ZMIN(M,N+1)=0.0
              ZMIN(M-1,N+1)=0.0
      13      CONTINUE
              GO TO 4100
      4107    CONTINUE
```

```
            IT=IT+2
            T=T+2.
            IF(IZINT-IT)16,16,15
      C
      C
      16    WRITE(3,200)TT
      200   FORMAT('1',50X,'TIDAL PERIOD=',F6.2,'HOURS')
            WRITE(3,201)DL
      201   FORMAT(' ',50X,'GRID INTERVAL=',F6.2,'KILOMETERS')
            WRITE(3,202)
      202   FORMAT(' ',20X,'OUTPUT DESCRIPTION..')
            WRITE(3,203)
      203   FORMAT(' ',35X,'UNITS..METERS,DEGREES')
            WRITE(3,204)
      204   FORMAT(' ',53X,'N=2,4,6,ETC.')
            WRITE(3,205)
            WRITE(3,205)
            WRITE(3,205)
      205   FORMAT(' ',57X,'*')
            WRITE(3,206)
      206   FORMAT(' ',40X,'M=2,4,ETC*****HEIGHT*****')
            WRITE(3,205)
            WRITE(3,205)
            WRITE(3,205)
            WRITE(3,205)
            WRITE(3,207)
      207   FORMAT(' ',55X,'ANGLE')
      C          WRITE ZMAX AND PHASE
            WRITE(3,110)
      110   FORMAT('1',47X,'MAXIMUM HEIGHTS AND ASSOCIATED PHASE')
            WRITE(3,102)
      102   FORMAT('0','                    N= 2                N= 4                N= 6
           1  N= 8              N=10                N=12                N=14                N=16
           2        N=18')
            L2=MSUM-2
            DO 5002 J=1,L2,2
```

```
                M=MSUM+1-J
                WRITE(3,106)(ZMAX(M,N),N=1,18)
                WRITE(3,3)
      106       FORMAT(' ',5X,18(1X,F6.1))
                M=M-1
                WRITE(3,101)M,(ZMAX(M,N),N=1,18)
      101       FORMAT(' ','M=',I2,1X,18(1X,F6.2))
      3         FORMAT(' ')
      5002      CONTINUE
                M=1
                WRITE(3,106)(ZMAX(M,N),N=1,18)
                IF(NSUM-18)6004,6004,6003
      6003      WRITE(3,110)
                WRITE(3,103)
      103       FORMAT('0','           N=20           N=22           N=24           N=2
               16         N=28          N=30')
                DO 6005 J=1,L2,2
                M=MSUM+1-J
                WRITE(3,166)(ZMAX(M,N),N=19,29)
                WRITE(3,3)
                M=M-1
                WRITE(3,161)(ZMAX(M,N),N=19,29)
      6005      CONTINUE
                M=1
                WRITE(3,166)(ZMAX(M,N),N=19,29)
      6004      CONTINUE
      C
      C
      C               WRITE ZMIN AND PHASE
                WRITE(3,170)
                WRITE(3,102)
                DO 7002 J=1,L2,2
                M=MSUM+1-J
                WRITE(3,106)(ZMIN(M,N),N=1,18)
                WRITE(3,3)
                M=M-1
```

```
      WRITE(3,101)M,(ZMIN(M,N),N=1,18)
 7002 CONTINUE
      M=1
      WRITE(3,106)(ZMIN(M,N),N=1,18)
      IF(NSUM-18)5004,5004,5003
 5003 WRITE(3,170)
 170  FORMAT('1',47X,'MINIMUM HEIGHTS AND ASSOCIATED PHASE')
      WRITE(3,103)
      DO 5005 J=1,L2,2
      M=MSUM+1-J
      WRITE(3,166)(ZMIN(M,N),N=19,29)
      WRITE(3,3)
      M=M-1
      WRITE(3,161)(ZMIN(M,N),N=19,29)
 5005 CONTINUE
      M=1
      WRITE(3,166)(ZMIN(M,N),N=19,29)
 166  FORMAT(' ',11(1X,F6.1))
 161  FORMAT(' ',11(1X,F6.2))
 5004 CONTINUE
C
C
      DO 9000 M=1,65
      DO 9000 N=1,30
 9000 Z1(M,N)=10000000.0
      IFL=0
      L2=MSUM-1
      DO 5107 M=2,L2,2
      I=-1
 5100 I=I+2
      IF(IU(M,I)-2)5104,5107,5100
 5104 IF(IU(M,I))5100,5100,5101
 5101 IF(IFL)5102,5102,5103
 5102 IML=I+1
      IFL=1
      GO TO 5100
```

```
5103  IMR=I-1
      IFL=0
      DO 23 N=IML,IMR,2
C         CONSTRUCT TIDAL RANGE
      Z1(M,N)=ZMAX(M,N)-ZMIN(M,N)
      Z1(M,N+1)=0.0
      Z1(M-1,N+1)=0.0
      IF(ZMIN(M-1,N)-ZMAX(M-1,N))300,301,301
C         IF TZMIN LESS THAN TZMAX, LOW TIDE COMES BEFORE HIGH TIDE
300   ZMIN(M-1,N)=ZMIN(M-1,N)+360.
301   Z1(M-1,N)=((ZMAX(M-1,N)+ZMIN(M-1,N))/2.)-90.
C         CONSTRUCT MEAN PHASE
23    CONTINUE
      GO TO 5100
5107  CONTINUE
C         WRITE TIDE RANGE AND MEAN PHASE
      WRITE(3,210)
210   FORMAT('1',52X,'TIDE RANGE AND MEAN PHASE')
      WRITE(3,102)
      L2=MSUM-2
      DO 5202 J=1,L2,2
      M=MSUM+1-J
      WRITE(3,106)(Z1(M,N),N=1,18)
      WRITE(3,3)
      M=M-1
      WRITE(3,101)M,(Z1(M,N),N=1,18)
5202  CONTINUE
      M=1
      WRITE(3,106)(Z1(M,N),N=1,18)
      IF(NSUM-18)5204,5204,5203
5203  WRITE(3,210)
      WRITE(3,103)
      DO 5205 J=1,L2,2
      M=MSUM+1-J
      WRITE(3,166)(Z1(M,N),N=19,29)
      WRITE(3,3)
```

```
      M=M-1
      WRITE(3,161)(Z1(M,N),N=19,29)
 5205 CONTINUE
      M=1
      WRITE(3,166)(Z1(M,N),N=19,29)
 5204 CONTINUE
      REWIND 9
      CALL EXIT
      END
```

```
      INTEGER*2 IU
      DIMENSION IU(68,31),U1(65,30),RMAX(65,30),RMIN(65,30)
      READ(9,77)MSUM
      READ(9,77)NSUM
77    FORMAT(I2)
      READ(9,78)DL
C         DL=GRID INTERVAL IN METERS
      DL=DL/1000.
      READ(9,999)IZINT
C         IZINT=NUMBER OF INTERVALS -- IE NUMBER OF CURRENT AND
C         HEIGHT CALCULATIONS
999   FORMAT(I4)
      READ(9,78)TT
C         TT=PERIOD IN SECONDS
      TT=TT/3600.
78    FORMAT(F12.4)
      ZINT=IZINT
      READ(9)((IU(M,N),M=1,67),N=1,31)
      DO 10 M=1,65
      DO 10 N=1,30
      RMAX(M,N)=10000000.0
10    RMIN(M,N)=10000000.0
      IFL=0
      L2=MSUM-1
      DO 8107 M=2,L2,2
      I=-1
8100  I=I+2
      IF(IU(M,I)-2)8104,8107,8100
8104  IF(IU(M,I))8100,8100,8101
8101  IF(IFL)8102,8102,8103
8102  IML=I+1
      IFL=1
      GO TO 8100
8103  IMR=I-1
      IFL=0
      DO 8884 N=IML,IMR,2
```

```
              RMAX(M,N)=0.0
      8884    RMIN(M,N)=0.0
              GO TO 8100
      8107    CONTINUE
              T=0.
              IT=0
      15      CALL READER(9,U1,7800)
      C            READ U1
              IFL=0
              L2=MSUM-1
              DO 4107 M=2,L2,2
              I=-1
      4100    I=I+2
              IF(IU(M,I)-2)4104,4107,4100
      4104    IF(IU(M,I))4100,4100,4101
      4101    IF(IFL)4102,4102,4103
      4102    IML=I+1
              IFL=1
              GO TO 4100
      4103    IMR=I-1
              IFL=0
              DO 14 N=IML,IMR,2
              UATZ=(U1(M+1,N)+U1(M-1,N))/2.
      C       VATZ=-(V1(M,N+1)+V1(M,N-1))/2.
              VATZ=-(U1(M,N+2)+U1(M,N))/2.
      C            THIS CHANGES DIRECTION OF +V
              RC=SQRT((UATZ*UATZ)+(VATZ*VATZ))
      C            COMPUTE VECTORIAL CURRENT AT Z(M,N)
              IF(RC-RMAX(M,N))12,11,11
      11      RMAX(M,N)=RC
              RMAX(M-1,N)=TRIG(UATZ,VATZ)
              RMAX(M,N+1)=(TT/ZINT)*T
              RMAX(M-1,N+1)=0.0
      12      IF(IT)1,1,2
      1       RMIN(M,N)=RC
              RMIN(M-1,N)=TRIG(UATZ,VATZ)
```

```
      RMIN(M,N+1)=(TT/ZINT)*T
      RMIN(M-1,N+1)=0.0
      GO TO 14
2     IF(RC-RMIN(M,N))13,13,14
13    RMIN(M,N)=RC
      RMIN(M-1,N)=TRIG(UATZ,VATZ)
      RMIN(M,N+1)=(TT/ZINT)*T
      RMIN(M-1,N+1)=0.0
14    CONTINUE
      GO TO 4100
4107  CONTINUE
      IT=IT+2
      T=T+2.
      CALL READER(9,U1,7800)
C         READ Z1
      IF(IZINT-IT)16,16,15
16    CONTINUE
      WRITE(3,200)TT
200   FORMAT('1',50X,'TIDAL PERIOD=',F6.2,'HOURS')
      WRITE(3,201)DL
201   FORMAT(' ',50X,'GRID INTERVAL=',F6.2,'KILOMETERS')
      WRITE(3,202)
202   FORMAT(' ',20X,'OUTPUT DESCRIPTION..')
      WRITE(3,203)
203   FORMAT(' ',35X,'UNITS..METERS/SEC.,DEGREES,HOURS')
      WRITE(3,204)
204   FORMAT(' ',53X,'N=2,4,6,ETC.')
      WRITE(3,205)
      WRITE(3,205)
      WRITE(3,205)
205   FORMAT(' ',57X,'*')
      WRITE(3,206)
206   FORMAT(' ',40X,'M=2,4,ETC*****CURRENT****TIME****')
      WRITE(3,205)
      WRITE(3,205)
      WRITE(3,205)
```

```
      WRITE(3,205)
      WRITE(3,207)
207   FORMAT(' ',55X,'ANGLE')
C
C          WRITE RMAX,ANGLE,AND TIME
      WRITE(3,110)
110   FORMAT('1',48X,'MAXIMUM CURRENTS, ANGLES, AND TIMES')
      WRITE(3,102)
102   FORMAT('0','                    N= 2              N= 4              N= 6
     1  N= 8              N=10              N=12              N=14              N=16
     2     N=18')
      L2=MSUM-2
      DO 5002 J=1,L2,2
      M=MSUM+1-J
      WRITE(3,106)(RMAX(M,N),N=1,18)
      WRITE(3,3)
106   FORMAT(' ',5X,18(1X,F6.1))
      M=M-1
      WRITE(3,101)M,(RMAX(M,N),N=1,18)
3     FORMAT(' ')
101   FORMAT(' ','M=',I2,1X,18(1X,F6.2))
5002  CONTINUE
      M=1
      WRITE(3,106)(RMAX(M,N),N=1,18)
      IF(NSUM-18)6004,6004,6003
6003  WRITE(3,110)
      WRITE(3,103)
103   FORMAT('0','              N=20              N=22              N=24              N=2
     16              N=28              N=30')
      DO 6005 J=1,L2,2
      M=MSUM+1-J
      WRITE(3,166)(RMAX(M,N),N=19,29)
      WRITE(3,3)
166   FORMAT(' ',11(1X,F6.1))
      M=M-1
      WRITE(3,161)(RMAX(M,N),N=19,29)
```

```
161   FORMAT(' ',11(1X,F6.2))
6005  CONTINUE
      M=1
      WRITE(3,166)(RMAX(M,N),N=19,29)
6004  CONTINUE
C
C
C          WRITE RMIN, ANGLE, AND TIME
      WRITE(3,170)
170   FORMAT('1',48X,'MINIMUM CURRENTS, ANGLES, AND TIMES')
      WRITE(3,102)
      L2=MSUM-2
      DO 7002 J=1,L2,2
      M=MSUM+1-J
      WRITE(3,106)(RMIN(M,N),N=1,18)
      WRITE(3,3)
      M=M-1
      WRITE(3,101)M,(RMIN(M,N),N=1,18)
7002  CONTINUE
      M=1
      WRITE(3,106)(RMIN(M,N),N=1,18)
      IF(NSUM-18)5004,5004,5003
5003  WRITE(3,170)
      WRITE(3,103)
      DO 5005 J=1,L2,2
      M=MSUM+1-J
      WRITE(3,166)(RMIN(M,N),N=19,29)
      WRITE(3,3)
      M=M-1
      WRITE(3,161)(RMIN(M,N),N=19,29)
5005  CONTINUE
      M=1
      WRITE(3,166)(RMIN(M,N),N=19,29)
5004  CONTINUE
      REWIND 9
      CALL EXIT
      END
```

```
0001                  FUNCTION TRIG(UATZ,VATZ)
            C              DECIDE UPON QUADRANT
0002                  IF(UATZ)9000,9001,9001
0003        9001      IF(VATZ)9007,9002,9002
0004        9002      DEG=((ATAN2(VATZ,UATZ))*180.)/3.14159
            C              ANGLE IS BETWEEN 0 AND 90
0005                  GO TO 9009
0006        9000      UATZ=-UATZ
0007                  IF(VATZ)9004,9003,9003
0008        9003      DEG=90.+((ATAN2(UATZ,VATZ))*180.)/3.14159
            C              ANGLE IS BETWEEN 90 AND 180
0009                  GO TO 9009
0010        9004      VATZ=-VATZ
            C              ANGLE IS BETWEEN 180 AND 270
0011                  DEG=180.+((ATAN2(VATZ,UATZ))*180.)/3.14159
0012                  GO TO 9009
0013        9007      VATZ=-VATZ
            C              ANGLE IS BETWEEN 270 AND 360
0014                  DEG=270.+((ATAN2(UATZ,VATZ))*180.)/3.14159
0015        9009      CONTINUE
0016                  TRIG=DEG
0017                  RETURN
0018                  END
```

APPENDIX IV

FORMAT OF OUTPUT TAPE

MSUM	I2	max grid length
NSUM	I2	max grid width
DL	F12.4	grid spacing, meters
IZINT	I4	number of intervals
IT	F12.4	Tidal period in seconds
IU	unformatted	boundary information

IU may be obtained by the statement

```
READ(9)((IU(M,N),M=1,67),N=1,31)
```

NOTE: IU is a half-word integer matrix

U1	unformatted	record 1
Z1	"	2
U1	"	3
Z1	"	4
U1	"	IZINT-1
Z1	"	IZINT

end of file label

It will be advisable to use the same program for reading U1 and Z1 as was used for writing them. This program may be seen in Appendix II.

The program is designed to start at a certain address in the core (in this case at the beginning of the first word of the U1 array) and to continue writing until a certain number of bytes (1/4 single-precision words) have passed. In this case, the number of bytes equals 65 x 30 x 4, or 7800. When reading such data, the reverse process takes place.

If it is considered desirable to write other analysis programs it will be found helpful if either of the two analysis programs are used as examples.

APPENDIX V

SELECTIONS FROM THE SAMPLE PROBLEM COMPUTER OUTPUT

GRID INTERVAL= 50.00KILOMETERS

TIDAL PERIOD= 12.42HOURS

FRICTION COEFFICIENT=0.0030

LATITUDE= 5.0DEGREES

CORIOLIS PARAMETER=0.00001268RADIANS/SECOND

FOLLOWING PRINTOUTS ARE IN METER-SECOND UNITS

DEPTH-VALUES

	N= 1	N= 2	N= 3	N= 4	N= 5	N= 6	N= 7	N= 8	N= 9	N=10	N=11
M= 9	0.0	457.2	0.0	457.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
M= 8	457.2	0.0	457.2	0.0	457.2	0.0	0.0	0.0	0.0	0.0	0.0
M= 7	0.0	457.2	0.0	457.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
M= 6	457.2	0.0	457.2	0.0	457.2	0.0	0.0	0.0	0.0	0.0	0.0
M= 5	0.0	457.2	0.0	457.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
M= 4	457.2	0.0	457.2	0.0	457.2	0.0	0.0	0.0	0.0	0.0	0.0
M= 3	0.0	457.2	0.0	457.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
M= 2	457.2	0.0	457.2	0.0	457.2	0.0	0.0	0.0	0.0	0.0	0.0
M= 1	0.0	457.2	0.0	457.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0

CONDITIONS AFTER 0.0 HOURS
NUMBER OF TIDAL CYCLES COMPLETED 5

	N= 1	N= 2	N= 3	N= 4	N= 5
M= 9	0.0	1.01	0.0	1.01	0.0
M= 8	0.99	0.99	0.99	0.99	0.99
M= 7	0.0	0.97	0.0	0.97	0.0
M= 6	0.95	0.95	0.95	0.95	0.95
M= 5	0.0	0.90	0.0	0.90	0.0
M= 4	0.86	0.86	0.86	0.86	0.86
M= 3	0.0	0.80	0.0	0.80	0.0
M= 2	0.74	0.74	0.74	0.74	0.74
M= 1	0.0	0.68	0.0	0.68	0.0

Z-VALUES

N= 6	N= 7	N= 8	N= 9	N=10	N=11
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0

Z-VALUES

CONDITIONS AFTER 9.31HOURS
NUMBER OF TIDAL CYCLES COMPLETED 5

	N= 1	N= 2	N= 3	N= 4	N= 5	N= 6	N= 7	N= 8	N= 9	N=10	N=11
M= 9	0.0	0.04	0.0	0.04	0.0	0.0	0.0	0.0	0.0	0.0	0.0
M= 8	0.04	0.04	0.04	0.04	0.04	0.0	0.0	0.0	0.0	0.0	0.0
M= 7	0.0	0.03	0.0	0.04	0.0	0.0	0.0	0.0	0.0	0.0	0.0
M= 6	0.03	0.03	0.04	0.04	0.04	0.0	0.0	0.0	0.0	0.0	0.0
M= 5	0.0	0.03	0.0	0.04	0.0	0.0	0.0	0.0	0.0	0.0	0.0
M= 4	0.02	0.03	0.03	0.03	0.04	0.0	0.0	0.0	0.0	0.0	0.0
M= 3	0.0	0.01	0.0	0.02	0.0	0.0	0.0	0.0	0.0	0.0	0.0
M= 2	-0.00	-0.00	-0.00	-0.00	-0.00	0.0	0.0	0.0	0.0	0.0	0.0
M= 1	0.0	-0.01	0.0	-0.02	0.0	0.0	0.0	0.0	0.0	0.0	0.0

U-VALUES

CONDITIONS AFTER 9.31HOURS
NUMBER OF TIDAL CYCLES COMPLETED 5

	N= 1	N= 2	N= 3	N= 4	N= 5	N= 6	N= 7	N= 8	N= 9	N=10	N=11
M= 9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
M= 8	0.02	0.0	0.02	0.0	0.02	0.0	0.0	0.0	0.0	0.0	0.0
M= 7	0.0	0.03	0.0	0.03	0.0	0.0	0.0	0.0	0.0	0.0	0.0
M= 6	0.04	0.0	0.05	0.0	0.05	0.0	0.0	0.0	0.0	0.0	0.0
M= 5	0.0	0.06	0.0	0.06	0.0	0.0	0.0	0.0	0.0	0.0	0.0
M= 4	0.07	0.0	0.07	0.0	0.07	0.0	0.0	0.0	0.0	0.0	0.0
M= 3	0.0	0.09	0.0	0.09	0.0	0.0	0.0	0.0	0.0	0.0	0.0
M= 2	0.09	0.0	0.09	0.0	0.09	0.0	0.0	0.0	0.0	0.0	0.0
M= 1	0.0	0.09	0.0	0.09	0.0	0.0	0.0	0.0	0.0	0.0	0.0

V-VALUES

CONDITIONS AFTER 9.31HOURS
NUMBER OF TIDAL CYCLES COMPLETED 5

	N= 1	N= 2	N= 3	N= 4	N= 5	N= 6	N= 7	N= 8	N= 9	N=10	N=11
M= 9	0.0	0.00	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
M= 8	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
M= 7	0.0	0.00	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
M= 6	0.0	0.0	0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
M= 5	0.0	-0.00	0.0	-0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
M= 4	0.0	0.0	-0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
M= 3	0.0	-0.00	0.0	-0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0
M= 2	0.0	0.0	-0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
M= 1	0.0	-0.00	0.0	-0.00	0.0	0.0	0.0	0.0	0.0	0.0	0.0

```
                                    TIDAL PERIOD= 12.42HOURS
                                    GRID INTERVAL= 50.00KILOMETERS
OUTPUT DESCRIPTION..
          UNITS..METERS,DEGREES
                                      N=2,4,6,ETC.
                                           *
                                           *
                                           *
              M=2,4,ETC*****HEIGHT*****
                                           *
                                           *
                                           *
                                           *
                                         ANGLE
```

		N= 2		N= 4
	******	******	******	******
M= 8	******	0.99	0.0	0.99
	******	0.0	0.0	0.0
M= 6	******	0.95	0.0	0.95
	******	0.0	0.0	358.0
M= 4	******	0.86	0.0	0.86
	******	358.0	0.0	358.0
M= 2	******	0.74	0.0	0.74
	******	0.0	0.0	0.0

MAXIMUM HEIGHTS AND ASSOCIATED PHASE

```
           N= 6             N= 8             N=10
****** ****** ****** ****** ****** ****** ******

  0.0  ****** ****** ****** ****** ****** ******
   0.0 ****** ****** ****** ****** ****** ******

  0.0  ****** ****** ****** ****** ****** ******
   0.0 ****** ****** ****** ****** ****** ******

  0.0  ****** ****** ****** ****** ****** ******
   0.0 ****** ****** ****** ****** ****** ******

  0.0  ****** ****** ****** ****** ****** ******
   0.0 ****** ****** ****** ****** ****** ******
```

		N= 2		N= 4
	******	******	******	******
M= 8	******	-1.01	0.0	-1.01
	******	178.0	0.0	178.0
M= 6	******	-0.96	0.0	-0.96
	******	178.0	0.0	178.0
M= 4	******	-0.87	0.0	-0.87
	******	178.0	0.0	178.0
M= 2	******	-0.74	0.0	-0.74
	******	180.0	0.0	180.0

MINIMUM HEIGHTS AND ASSOCIATED PHASE

	N= 6		N= 8		N=10	
******	******	******	******	******	******	******
0.0	******	******	******	******	******	******
0.0	******	******	******	******	******	******
0.0	******	******	******	******	******	******
0.0	******	******	******	******	******	******
0.0	******	******	******	******	******	******
0.0	******	******	******	******	******	******
0.0	******	******	******	******	******	******
0.0	******	******	******	******	******	******

TIDE RANGE AND MEAN PHASE

		N= 2		N= 4		N= 6		N= 8		N=10	
	******	******	******	******	******	******	******	******	******	******	******
M= 8	******	1.99	0.0	1.99	0.0	******	******	******	******	******	******
	******	-1.0	0.0	-1.0	0.0	******	******	******	******	******	******
M= 6	******	1.91	0.0	1.91	0.0	******	******	******	******	******	******
	******	-1.0	0.0	358.0	0.0	******	******	******	******	******	******
M= 4	******	1.73	0.0	1.73	0.0	******	******	******	******	******	******
	******	358.0	0.0	358.0	0.0	******	******	******	******	******	******
M= 2	******	1.49	0.0	1.49	0.0	******	******	******	******	******	******
	******	0.0	0.0	0.0	0.0	******	******	******	******	******	******

```
                                   TIDAL PERIOD= 12.42HOURS
                                   GRID INTERVAL= 50.00KILOMETERS
OUTPUT DESCRIPTION..
          UNITS..METERS/SEC.,DEGREES,HOURS
                                     N=2,4,6,ETC.
                                         *
                                         *
                                         *
              M=2,4,ETC*****CURRENT****TIME****
                                         *
                                         *
                                         *
                                         *
                                       ANGLE
```

MAXIMUM CURRENTS, ANGLES, AND TIMES

		N= 2		N= 4		N= 6		N= 8		N=10	
	******	******	******	******	******	******	******	******	******	******	******
M= 8	******	0.02	9.18	0.02	9.25	******	******	******	******	******	******
	******	359.7	C.C	359.6	C.0	******	******	******	******	******	******
M= 6	******	0.C5	9.18	C.C5	9.25	******	******	******	******	******	******
	******	359.9	0.C	359.9	C.0	******	******	******	******	******	******
M= 4	******	C.C7	3.17	C.C7	9.18	******	******	******	******	******	******
	******	180.2	0.C	0.0	0.0	******	******	******	******	******	******
M= 2	******	0.C9	3.17	0.C9	9.18	******	******	******	******	******	******
	******	18C.4	C.C	0.1	C.0	******	******	******	******	******	******

		N= 2		N= 4
	******	******	******	******
M= 8	******	0.00	6.14	0.00
	******	107.6	0.0	118.7
M= 6	******	0.00	12.35	0.00
	******	21.8	0.0	52.3
M= 4	******	0.00	0.0	0.00
	******	59.9	0.0	234.8
M= 2	******	0.00	0.0	0.00
	******	85.5	0.0	87.7

MINIMUM CURRENTS, ANGLES, AND TIMES

	N= 6		N= 8		N=10	
******	******	******	******	******	******	******
6.21	******	******	******	******	******	******
0.0	******	******	******	******	******	******
0.0	******	******	******	******	******	******
0.0	******	******	******	******	******	******
6.14	******	******	******	******	******	******
0.0	******	******	******	******	******	******
12.35	******	******	******	******	******	******
0.0	******	******	******	******	******	******