MEMORANDUM
RM-5294-PR
MAY 1967

ASPECTS OF A COMPUTATIONAL MODEL FOR LONG-PERIOD WATER-WAVE PROPAGATION

Jan J. Leendertse

PREPARED FOR:
UNITED STATES AIR FORCE PROJECT RAND

The RAND Corporation
SANTA MONICA, CALIFORNIA
MEMORANDUM
RM-5294-PR
MAY 1967

ASPECTS OF A COMPUTATIONAL MODEL FOR
LONG-PERIOD WATER-WAVE PROPAGATION
Jan J. Leendertse

This research is supported by the United States Air Force under Project RAND—Con
tact No. F41620-67-C-0045—monitored by the Directorate of Operational Requirements
and Development Plans, Deputy Chief of Staff, Research and Development. Hq USAF.
Views or conclusions contained in this Memorandum should not be interpreted as
representing the official opinion or policy of the United States Air Force.

DISTRIBUTION STATEMENT
Distribution of this document is unlimited.
PREFACE

The broad objective of this study is to model water waves generated by large nuclear explosions, near the surface of the ocean or under water, during propagation towards shore. With such models, assessment of damage from these waves in coastal areas can be evaluated.

The characteristics of these waves near shore and the extent of shore inundation are related in a very complex manner to the energy yield of the explosion, the depth of the explosion, the water depth at the blast location, and the topography of the ocean floor.

This Memorandum describes the development of a computational method for studying the propagation of long-period waves in relatively shallow water at some distance from the explosion. The accuracy of the computation scheme has been extensively tested on tidal-flow models of the estuary of the Rhine River and of the North Sea by comparing measured data with computed results. This work was done in cooperation with the Hydraulic Research Division of the Delta Works (Netherlands Government).

The numerical techniques developed in this study can also be applied in principle to other large-scale hydrodynamic problems, such as the prediction of flooding of coastal installations due to natural storms, and to the propagation of very-low-frequency sound waves.
SUMMARY

This Memorandum presents various aspects of a computational model for the calculation of long-period water waves. The model is based upon numerical integration of the hydrodynamic equations governing the long-period wave motions of the sea.

The basis of computation is the vertically integrated equations of motion and continuity in an Eulerian system. The partial-differential equations, which include the effects of earth rotation and bottom roughness, are approximated by two sets of difference equations. The two sets are used in succession for a step-by-step solution in time. An analytical investigation of simplified sets of difference equations indicates that this multioperation method is unconditionally stable.

The discreteness of the representation of the waves in time and in two spatial dimensions influences the computed velocity of wave propagation but not (or insignificantly, through terms of a second order of magnitude) the amplitude of the wave.

The computational method, which is presented in FORTRAN, permits modeling of long waves, such as tides, surges, seiches, and tsunamis, in areas with complicated boundaries in an expedient manner and is particularly suited for hydraulic engineering research. A guide is given for use of the model, and computational effects are discussed in detail.

The use of the computational procedure is illustrated with results of tidal computations of the southern North Sea and of the Haringvliet in the estuary of the Rhine River.
Acknowledgments

The author wishes to thank the following persons for their help in the preparation of this Memorandum: J. J. Dronkers, Hydraulic Research Division of the Delta Works, Netherlands Rijkswaterstaat, for his valuable suggestions and for providing all data concerning the Haringvliet and North Sea computational models; his staff for critical analyses of the results and the program description; E. van Spiegel, Technological University, Delft, for his helpful discussions and careful review of the manuscript; K. W. Brown, Warren Hollis, R. H. Mayall, Miss Gail Burkholz, and Miss Tiina Repnau for their assistance in program developments.
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREFACE</td>
<td></td>
<td>iii</td>
</tr>
<tr>
<td>SUMMARY</td>
<td></td>
<td>v</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td></td>
<td>vii</td>
</tr>
<tr>
<td>SYMBOLS</td>
<td></td>
<td>xi</td>
</tr>
</tbody>
</table>

### Section 1: The Basic, Partial-Differential Equations for Long-Period Gravity Waves

1.0 Introduction ............................................ 1
1.1 Basic Equations ....................................... 1
1.2 Equations for Long-Period Wave Motion .............. 2

### Section 2: Analytical Methods for Investigation of Approximation, Stability, and Convergence of Difference Equations

2.0 Introduction ............................................ 7
2.1 Approximation .......................................... 7
2.2 Stability ............................................... 12
2.3 Wave Deformation ...................................... 17
2.4 Generation of Spurious Modes ......................... 27
2.5 Effects of Nonlinear Terms ............................ 31
2.6 Effects of Bottom Stress .............................. 35
2.7 Stability Aspects of the Convective-Inertia Terms 41
2.8 Conservation of Mass and Momentum ................. 48

### Section 3: Wave Motion in Two Spatial Dimensions

3.0 Introduction ............................................ 52
3.1 The System of the Finite-Difference Equations in Two Dimensions ............................................ 52
3.2 Convective-Inertia Terms and Bottom-Effect Terms 60
3.3 Approximation Problems ................................ 65
3.4 Stability ............................................... 70
3.5 Wave Deformation ...................................... 74
3.6 Conservation of Mass and Momentum ................. 78

### Section 4: Numerical Experiments

4.0 Use of Numerical Experiments ......................... 81
4.1 Tokyo Bay Model ....................................... 81
4.2 The Haringvliet Model: Effect of Step Size ........ 83
4.3 The Haringvliet Model: Effect of Iteration .......... 93
4.4 The Haringvliet Model: Effect of Approximation .... 100
4.5 North Sea Model ...................................... 107

### Section 5: Concluding Remarks

.......................................................... 116
Appendix
A. COMPUTATION FORMULAS ........................................ 117
B. DETERMINATION OF ROOTS WITHIN THE UNIT CIRCLE OF A
   CUBIC EQUATION WITH REAL COEFFICIENTS ................. 122
C. PROGRAM DESCRIPTION ......................................... 127

REFERENCES .......................................................... 163
SYMBOLS

\( A = \) coefficient

\( [A] = \) matrix

\( A_j^n = \) relation parameter of the multioperation method

\( \vec{A}_n = \) Fourier-series component vector (time dependent)

\( a = \) constant

\( B = \) coefficient

\( [B] = \) matrix

\( B_j^{j/2} = \) relation coefficient of the multioperation method

\( C = \) coefficient of De Chézy

\( [C] = \) matrix

\( D = \) coefficient

\( [D] = \) matrix

\( F = \) function representing influence of wind and atmospheric pressure

\( \vec{F} = \) forcing-function vector

\( \vec{F}(\cdot) = \) forcing-function vector

\( f = \) Coriolis parameter

\( G(t, r) = \) amplification matrix

\( g = \) acceleration of gravity

\( h = \) distance between bottom and reference plane

\( \text{Im}(\cdot) = \) imaginary part of ( )

\( j = 0, \pm \frac{1}{2}, \pm 1, \pm \frac{3}{2} \)

\( K = \) coefficient

\( K(\cdot) = \) tide-generating force

\( \vec{K} = \) vector

\( k = 0, \pm \frac{1}{2}, \pm 1, \pm \frac{3}{2}; \) also, coefficient of linearized bottom friction
\( k' \) = coefficient of linearized bottom friction in a computational system

\( L \) = wavelength

\( L_1, L_2 \) = bounds in spatial dimension

\( \mathcal{L} \) = operator

\( \ell \) = grid size

\( M \) = positive, finite, and nonzero number

\( m \) = integer value

\( n = 0, \frac{1}{2}, 1, \frac{3}{2}, 2, \ldots \)

\( P \) = recursion coefficient

\( p \) = order of approximation

\( p_0 \) = atmospheric pressure

\( Q \) = recursion coefficient

\( R \) = expression for quadratic friction term

\( \text{Re}(\ldots) \) = real part of ( )

\( R_j \) = recursion coefficient

\( r \) = integer value

\( r_j \) = coefficient

\( S \) = recursion coefficient

\( s \) = distance

\( T(\mathcal{L}) \) = propagation factor

\( T_1, T_2 \) = boundary in time

\( U \) = vertically averaged velocity component

\( U^\kappa \) = Fourier-series components

\( \overline{U} \) = vector, or symbolic form of the solution of the differential equation

\( \overline{U}^\kappa \) = Fourier-series component vector
\( \bar{\Omega}(z) \) = symbolic form of the solution of the difference equation

\( U_0 \) = velocity of the basic flow

\( u \) = velocity component in x-direction

\( u' \) = velocity distribution component

\( V \) = vertically averaged velocity component

\( v \) = velocity component in y-direction

\( v' \) = velocity distribution component

\( \omega \) = velocity component in z-direction

\( X \) = force per unit mass in horizontal direction

\( x \) = horizontal distance

\( Y \) = force per unit mass in horizontal direction

\( y \) = horizontal distance

\( Z \) = force per unit mass in vertical direction

\( z \) = vertical distance

\( \alpha \) = coefficient

\( \beta \) = wave frequency

\( \beta' \) = computed-wave frequency

\( \gamma \) = direction of wave propagation

\( \sigma \) = distance between reference plane and water level

\( \psi \) = Fourier-series component

\( \vartheta \) = weighting function

\( \lambda \) = eigenvalue

\( \nu \) = number of time steps

\( \rho \) = density of water

\( \sigma \) = wave number = \( 2\pi/L \)

\( \tau \) = time step of each of the two operations of the multioperation method
\( \tau_b \) = bottom stress

\( \gamma \) = coefficient

\( [ ]_l \) = grid function with grid size

\( || \ | \ || \) = norm

0( ) = indication for order of approximation of a series of terms
1. THE BASIC, PARTIAL-DIFFERENTIAL EQUATIONS FOR LONG-PERIOD GRAVITY WAVES

1.0 INTRODUCTION

This section reviews the derivation of the partial-differential equations for the nonsteady motion of long waves. This derivation is based upon the vertical integration of the equation of motion and the continuity equation for an incompressible fluid in Eulerian coordinates. The effects of viscosity are introduced into the equations of motion, and special attention is given to the engineering practice of using empirical values.

In the classical theory, which is described briefly in this chapter, the effects of the vertical acceleration and velocities are neglected.

1.1 BASIC EQUATIONS

The basis of this research is the fluid-flow, partial-differential equations representing the conditions of our investigation, such as velocities and pressures on a fixed-coordinate system; thus, these equations are in the Eulerian form (see Lamb\(^{(1)}\)). The Cartesian axes \(X\) and \(Y\) are taken counterclockwise in a horizontal plane of the undisturbed water surface, with the \(Z\)-axis vertically upward. The components of the velocity are \(u, v, \) and \(w,\) which are parallel to the coordinate axis at the point \((x,y,z)\). For nonviscous flow the dynamical equations may be written

\[
\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} + \frac{1}{\rho} \frac{\partial p}{\partial x} = X \tag{1.1.1}
\]

\[
\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} + \frac{1}{\rho} \frac{\partial p}{\partial y} = Y \tag{1.1.2}
\]

\[
\frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} + \frac{1}{\rho} \frac{\partial p}{\partial z} = Z \tag{1.1.3}
\]
where

\[
\begin{align*}
\rho &= \text{pressure} \\
\rho &= \text{density} \\
X, Y, Z &= \text{the components of extraneous forces per unit of mass}
\end{align*}
\]

The extraneous forces are the forces generated by the rotation of the earth, the tide-generating forces which are caused by celestial bodies, and the gravity force in the z-direction.

The continuity equation for incompressible flow used in the derivation is

\[
\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \quad (1.1.4)
\]

1.2 EQUATIONS FOR LONG-PERIOD WAVE MOTION

In the classical theory of long waves, the vertical acceleration of the fluid particles is neglected because these accelerations are very small with respect to the acceleration of the gravity field. Also, the velocities of the water particles in the z-direction may be neglected in dealing with long waves. Thus, all terms containing w in Eqs. (1.1.1), (1.1.2), and (1.1.3) are omitted.

The Cartesian coordinate system is taken in the horizontal plane of the undisturbed water surface. The distance between this reference plane and the bottom is indicated by h, and the distance between this reference plane and the water surface at the considered time is indicated by ζ.

Following Hansen,\(^2\) vertically averaged velocity components can be introduced according to

\[
\begin{align*}
U &= \frac{1}{(h + \zeta)} \int_{-h}^{\zeta} u \, dz \\[2ex]
V &= \frac{1}{(h + \zeta)} \int_{-h}^{\zeta} v \, dz
\end{align*} \quad (1.2.1)\]

The distributions of the velocity components \( u \) and \( v \) over the vertical at a certain location can be expressed as a function of the averaged velocity by introduction of distribution coefficients, as follows:

\[
\begin{align*}
 u(z) &= U[1 + u'(z)] \\
 v(z) &= V[1 + v'(z)]
\end{align*}
\]  
(1.2.3)  
(1.2.4)

For these distribution coefficients, the following relations are valid:

\[
\begin{align*}
 \int_{-h}^{z} u'(z) \, dz &= 0 \\
 \int_{-h}^{z} v'(z) \, dz &= 0
\end{align*}
\]  
(1.2.5)  
(1.2.6)

As mentioned in Section 1.1, the horizontal components of the extraneous forces are the effects of earth rotation and the tide-generating force, which may be expressed

\[
\begin{align*}
 X &= fv + K(x) \\
 Y &= -fu + K(y)
\end{align*}
\]  
(1.2.7)  
(1.2.8)

where

\[
\begin{align*}
 f &= \text{Coriolis parameter} \\
 K(x), K(y) &= \text{tide-generating force}
\end{align*}
\]

The Coriolis parameter is a function of the latitude. (See Proudman\(^3\) and Dronkers\(^4\) for the derivation of this parameter.)

Since the effects of the vertical acceleration and velocity are neglected, Eq. (1.1.3) may be written
\[ \frac{1}{\rho} \frac{\partial p}{\partial z} = z \] (1.2.9)

In the vertical direction the external forces \( z \) are the gravity force, the component of the forces induced by earth rotation, and the tide-generating force. The latter two are very small compared to gravity and are neglected in this analysis, which concerns shallow water with a depth that is a fraction of the wavelength. It is assumed that the density is uniform. Consequently, in this theory the pressure is assumed to be hydrostatic and a linear function of the depth, as follows:

\[ p(z) = \rho g (\zeta - z) + p_o \] (1.2.10)

where

\[ p_o = \text{atmospheric pressure} \]
\[ \rho = \text{density of water} \]

The derivatives of the pressure in the horizontal directions now become a function of the water level and the atmospheric pressure,

\[ \frac{\partial p}{\partial x} = \rho g \frac{\partial \zeta}{\partial x} + \frac{\partial p_o}{\partial x} \] (1.2.11)

\[ \frac{\partial p}{\partial y} = \rho g \frac{\partial \zeta}{\partial y} + \frac{\partial p_o}{\partial y} \] (1.2.12)

Integration of Eqs. (1.1.1) and (1.1.2) over the region \( z = -h(x,y) \) to \( z = \zeta(x,y) \) and introduction of Eqs. (1.2.1) through (1.2.8), (1.2.11), and (1.2.12) after dividing by \( (h + \zeta) \) give

\[ \frac{\partial U}{\partial t} + U \frac{\partial U}{\partial x} + V \frac{\partial U}{\partial y} - f V + g \frac{\partial \zeta}{\partial x} = - \frac{1}{\rho} \frac{\partial p_o}{\partial x} + A(x) \] (1.2.13)
\[
\frac{\partial \mathbf{v}}{\partial t} + U \frac{\partial \mathbf{v}}{\partial x} + v \frac{\partial \mathbf{v}}{\partial y} + f \mathbf{u} + g \frac{\partial \mathbf{w}}{\partial y} = - \frac{1}{\rho} \frac{\partial p}{\partial x} + A(y) \tag{1.2.14}
\]

The terms \(A(x)\) and \(A(y)\) contain the effects of the tide-generating forces and derivatives introduced by the vertical integration. Numerical computations and estimates based upon theoretical considerations show that these terms can generally be omitted in actual computation (see Welander;\(^{(5)}\) Hansen\(^{(6)}\)). It is necessary, however, that the velocity distributions over the vertical be fairly constant.

Up to this point in the discussion, an inviscid fluid has been considered. It can be shown that if viscosity is taken into account, shear-stress terms can be derived from the vertical eddy viscosity in the equation of motion representing effects of wind at the surface and friction at the bottom.\(^{(1)}\)

Hansen and Uusitalo also introduce the effects of the horizontal (lateral) eddy viscosity into the equation of motion for computational reasons.\(^{(6,7)}\) The terms are very small, and they are neglected in this study.

The wind stress is a forcing function in the system of equations, but its dependency upon wind velocity and direction is not discussed here. The bottom stress \((\tau_b)\) is proportional to the squared velocity and hence affects the behavior of the long waves considerably. The frictional resistance factor, which is used to establish the relation between the squared velocity and the bottom stress, can be found only by observation. This coefficient depends on the roughness of the bottom, the bottom material, and the depth. Following Dronkers,\(^{(4)}\) the relation for flow in one direction is expressed

\[
\tau_b = \rho g C^2 \mathbf{v} |\mathbf{v}| 
\]

where

- \(\rho\) = density of the fluid
- \(\mathbf{v}\) = velocity
- \(C\) = De Chézy coefficient
Introduction of this bottom-stress term into the two-dimensional system results in

$$\frac{\partial U}{\partial t} + U \frac{\partial U}{\partial x} + V \frac{\partial U}{\partial y} - fV + g \frac{\partial \zeta}{\partial x} + g \frac{U(U^2 + V^2)^{\frac{1}{2}}}{c^2(h + \zeta)} = F(x) \quad (1.2.15)$$

$$\frac{\partial V}{\partial t} + U \frac{\partial V}{\partial x} + V \frac{\partial V}{\partial y} + fU + g \frac{\partial \zeta}{\partial y} + g \frac{V(U^2 + V^2)^{\frac{1}{2}}}{c^2(h + \zeta)} = F(y) \quad (1.2.16)$$

where $F(x)$ and $F(y)$ are the forcing functions of wind stress and barometric pressures in the $x$- and $y$-directions, respectively.

In a similar manner, the equation of continuity can be integrated over the vertical. The boundary condition for the free surface is

$$w(\zeta) = \frac{\partial \zeta}{\partial t} + U \frac{\partial \zeta}{\partial x} + V \frac{\partial \zeta}{\partial y} \quad (1.2.17)$$

and at the bottom,

$$w(-h) + u \frac{\partial h}{\partial x} + v \frac{\partial h}{\partial y} = 0 \quad (1.2.18)$$

With these boundaries, vertical integration of the continuity equation (Eq. (1.1.4)) results in

$$\frac{\partial \zeta}{\partial t} + \frac{\partial}{\partial x}(h + \zeta) U \frac{\partial U}{\partial x} + \frac{\partial}{\partial y}(h + \zeta) V \frac{\partial V}{\partial y} = 0 \quad (1.2.19)$$
2. ANALYTICAL METHODS FOR INVESTIGATION OF APPROXIMATION, STABILITY, AND CONVERGENCE OF DIFFERENCE EQUATIONS

2.0 INTRODUCTION

The problems associated with the numerical solution of the partial-differential equations representing wave motion will now be considered. The following system of linear partial-differential equations representing wave motion in one direction is used as an example:

\[
\frac{\partial \zeta}{\partial t} + h \frac{\partial u}{\partial x} = 0 \quad (2.0.1)
\]

\[
\frac{\partial u}{\partial t} + g \frac{\partial \zeta}{\partial x} = -\frac{\partial p_o}{\partial x} \quad (2.0.2)
\]

where

\[\zeta(x,t) = \text{water level}\]
\[u(x,t) = \text{velocity}\]
\[p_o(x,t) = \text{atmospheric pressure}\]
\[h = \text{water depth}\]

Wave motion is completely described in time and distance by these formulas, provided that initial values and boundary conditions are given.

2.1 APPROXIMATION

In developing a method for numerical integration of the above equations, we replace the differential equations with difference equations to operate in spatial and time coordinates on definite points of a grid system. If the grid size approaches zero, the solution of the difference equations must converge to that of the differential equations. Consequently, a method must be formulated to express the difference between the exact solution and that obtained by the approximation. The solution of the differential equations, represented by the symbolic form \( \bar{U} \), is determined continuously over the whole interval considered (in this case \( x,t \)); the solution of the difference equations being developed, \( \bar{U}(t) \), is determined only at a finite number of grid points.
To overcome the difficulty of comparison between the continuous function and the function at a finite number of points, the exact solution of the differential equations is taken as the grid function \([U]_L\) (which is in correspondence with a grid of size \(L\), upon which the solution of the difference equation is obtained). The difference between \([U]_L\) and the solution of the difference equation \(\tilde{U}_L\) serves as the required measure, and the norm \(|[U]_L - \tilde{U}(L)|\) is introduced to estimate this difference. This norm is taken as the maximum modulus of all components over all points in the study of systems of equations.

Similarly, in the study of the approximation of the differential equation, which we represent as \(\tilde{U} = \tilde{F}\), the relation between the differential operator \([\tilde{U}]_L\) (operating upon \(\tilde{U}\) at the grid points) and the difference operator \(L_{\tilde{U}}(L)\) can be expressed by norms. Also, the relation between the forcing function \([F]_L\) of the differential equation as a grid function and the forcing function \(\tilde{F}(L)\) of the difference equation can be expressed by norms.

With these concepts, the order of approximation of the systems can be defined. The order of approximation of the difference equation \(L_{\tilde{U}}(L) = \tilde{F}(L)\) to the differential equation \(\tilde{U} = \tilde{F}\), with respect to the function \(\tilde{U}(x,t)\), is given by \(p\) if

\[
|L_{\tilde{U}}[U]_L - [L\tilde{U}]_L| \leq M_1 \xi^p \tag{2.1.1}
\]

\[
|\tilde{F}(L) - [\tilde{F}]_L| \leq M_2 \xi^p \tag{2.1.2}
\]

where \(M_1\) and \(M_2\) are positive, finite, and nonzero and do not depend on \(\xi\).

The order of approximation of the boundary conditions may be defined in a similar manner. Godunov and Ryabenki\(^8\) show that the order of the norm of the difference \(|[U]_L - \tilde{U}(L)|\) is the same as the order of the approximation of the difference equation, provided that the difference scheme used is stable. They also show that in this case the solution of the difference equation converges to that of the differential equation. The stability aspects are discussed in Section 2.2.
The order of approximation of difference schemes can now be determined. The differential operator of the system of Eqs. (2.0.1) and (2.0.2) is written in symbolic form

\[ \mathcal{L} = \begin{vmatrix} \frac{\partial}{\partial t} & h \frac{\partial}{\partial x} \\ g \frac{\partial}{\partial x} & \frac{\partial}{\partial t} \end{vmatrix} \]  

(2.1.3)

which operates on the column vector

\[ \overline{u} = \begin{bmatrix} \xi \\ u \end{bmatrix} \]  

(2.1.4)

The forcing function is also a column vector

\[ \overline{f} = \begin{bmatrix} 0 \\ - \frac{\partial p}{\partial x} \end{bmatrix} \]  

(2.1.5)

In the approximation of the forcing function, it is assumed that this vector is known exactly in time and space; consequently, we are able to set the values of the function \( \overline{f} \) on the grid point \((x, t)\) exactly in the difference approximation of the system \( \mathcal{L} \overline{u}(t) = 0 \). Thus, in this case, the only concern is the approximation of the difference operator which is found by using Taylor expansions.

Consider the difference equation \( \mathcal{L} \overline{u}(t) = 0 \), using the implicit system

\[ \frac{1}{\tau} \left( \zeta^r_m + \zeta^r_m \right) + \frac{1}{4\xi} \left( u^r_{m+1} + u^r_{m-1} + u^r_{m+1} + u^r_{m-1} \right) = 0 \]  

(2.1.6)

\[ \frac{1}{\tau} \left( u^r_m + u^r_m \right) + \frac{1}{4\xi} \left( \zeta^r_{m+1} + \zeta^r_{m-1} + \zeta^r_{m+1} + \zeta^r_{m-1} \right) = 0 \]  

(2.1.7)

at the time \( t = rt \) and at the location \( x = mx \). The Taylor expansion at the time \( t' = rt + (\xi t) \) for the first two terms of Eq. (2.1.6) is written
\[ \zeta^{r+1}_m = \zeta(x, t') + \frac{\tau}{2} \zeta_t(x, t') + \frac{1}{8} \tau^2 \zeta_{tt}(x, t') + \frac{1}{48} \tau^3 \zeta_{ttt}(x, t') + R_1 \]  
\( (2.1.8) \)

\[ \zeta^r_m = \zeta(x, t') - \frac{\tau}{2} \zeta_t(x, t') + \frac{1}{8} \tau^2 \zeta_{tt}(x, t') - \frac{1}{48} \tau^3 \zeta_{ttt}(x, t') + R_2 \]  
\( (2.1.9) \)

Thus,

\[ \frac{1}{\tau} (\zeta^{r+1}_m - \zeta^r_m) = \zeta_t(x, t') + \frac{1}{24} \tau^2 \zeta_{ttt}(x, t') + O(\tau^3) \]  
\( (2.1.10) \)

where \( O(\cdot) \) is an indication for the order of approximation of a series of terms. Similarly, for the next two terms of Eq. \( (2.1.6) \)

\[ \frac{1}{2\ell} (U^{r+1}_m - U^{r+1}_{m-1}) = U_x(x, t') + \frac{1}{2} \tau U_{xt}(x, t') + \frac{1}{6} \ell^2 U_{xxx}(x, t') + \frac{1}{2} \tau + O(\ell^3) \]  
\( (2.1.11) \)

\[ \frac{1}{2\ell} (U^r_m - U^r_{m-1}) = U_x(x, t') - \frac{1}{2} \tau U_{xt}(x, t') + \frac{1}{6} \ell^2 U_{xxx}(x, t') - \frac{1}{2} \tau + O(\ell^3) \]  
\( (2.1.12) \)

The expansions for these last two terms can now be made at time \( t' \):

\[ \frac{1}{2\ell} (U^{r+1}_m - U^{r+1}_{m-1}) = U_x(x, t') + \frac{1}{2} \tau U_{xt}(x, t') + \frac{1}{8} \tau^2 U_{xtt}(x, t') \]

\[ + \frac{1}{48} \tau^3 U_{xttt}(x, t') + \frac{1}{6} \ell^2 U_{xxx}(x, t') + \frac{1}{2} \tau U_{xxx}(x, t') \]

\[ + \frac{1}{8} \tau^2 U_{xxx}(x, t') + \frac{1}{48} \tau^3 U_{xxxtt}(x, t') + \cdots \]  
\( (2.1.13) \)
\[
\frac{1}{2 \ell} (U_{m+1}^\tau - U_{m-1}^\tau) = U_x(x, t') - \frac{1}{2} \tau U_{\tau \tau \tau} (x, t') + \frac{1}{8} \tau^2 U_{\tau \tau \tau \tau} (x, t') \\
- \frac{1}{48} \tau^3 U_{\tau \tau \tau \tau} (x, t') + \frac{1}{6} \ell^2 [U_{xxx} (x, t') - \frac{1}{2} \tau U_{xxxt} (x, t')] \\
+ \frac{1}{8} \tau^2 U_{x x x x} (x, t') - \frac{1}{48} \tau^3 U_{x x x x x x} (x, t') + \cdots
\] (2.1.14)

Inserting Eqs. (2.1.10), (2.1.13), and (2.1.14) into Eq. (2.1.6) and maintaining only the lower derivatives of the series, we then obtain the difference between the differential and difference equations of the first set

\[
+ \frac{1}{24} \tau^2 U_{\tau \tau \tau} (x, t') + \frac{1}{8} \tau^2 h U_{\tau} (x, t') + \frac{1}{6} \ell^2 h U_{xxx} (x, t') + 0(\tau^3, \ell^3)
\] (2.1.15)

and the differences between the differential and difference equations of the second set

\[
+ \frac{1}{24} \tau^2 U_{\tau \tau \tau} (x, t') + \frac{1}{8} \tau^2 h U_{\tau} (x, t') + \frac{1}{6} \ell^2 h U_{xxx} (x, t') + 0(\tau^3, \ell^3)
\] (2.1.16)

From the definition, it is clear that this system \( (U, \ell) \) has a second-order accuracy. Regardless of the relation between the space step and the time step, this system approximates the system of differential equations if both \( \tau \) and \( \ell \) are approaching zero.

The order of approximation is a good tool when different systems are compared; however, it gives no means for a numerical evaluation of accuracy.
2.2 STABILITY

The system of finite-difference equations, which is employed for approximating the system of differential equations, also requires that the numerical errors which are introduced in the computational method do not amplify in an unlimited manner. The simplest form of investigation of this stability requirement is to follow a Fourier expansion of a line of errors as time progresses. This method, introduced by von Neumann, gives insight into what happens during computation.

A numerical solution of the system of Eqs. (2.0.1) and (2.0.2) is attempted, using the following implicit scheme and omitting the forcing function:

\[
\chi_{m+1}^r - \chi_m^r + \frac{1}{2} \frac{T}{L} h (U_{m+1}^{r+1} - U_{m-1}^{r+1}) = 0
\]  \hspace{1cm} (2.2.1)

\[
U_m^{r+1} - U_m^r + \frac{1}{2} \frac{T}{L} g (\chi_{m+1}^{r+1} - \chi_{m-1}^{r+1}) = 0
\]  \hspace{1cm} (2.2.2)

Suppose that a line of errors \( \Delta \bar{U}(x) \) exists at a particular time \( t = 0 \) in the spatial grid of the vector:

\[
\bar{U} = \begin{bmatrix} U \\ \chi \end{bmatrix}
\]  \hspace{1cm} (2.2.3)

It is now possible to make a finite Fourier decomposition of this error in the following manner:

\[
\Delta \bar{U}(x) = \sum_n \bar{\chi}_n e^{i \sigma_n x}
\]  \hspace{1cm} (2.2.4)

Since a finite number of points (N) exists in the x-direction, the number of terms (n) of this decomposition equals N. The system under consideration is a linear system, and thus the behavior of only one term of the Fourier series can be considered.
The vector \( \overline{A}_n \) is time dependent, and in order to satisfy the expression of the particular error wave at \( t = 0 \), the vector must take the form

\[
\overline{A}_n(t) = U_n^{*} e^{i \xi n t}
\]  

(2.2.5)

where \( U_n^{*} \) and \( \xi n \) are constants. Thus, the expression for the errors at \((t,x)\) of the vector \( \overline{U} \) should satisfy the form

\[
\delta \overline{U}(x,t) = U_n^{*} e^{i \xi t} e^{i \omega x}
\]  

(2.2.6)

where

\[
U_n^{*} = \begin{Bmatrix} U_n^{*} \\ \zeta_n^{*} \end{Bmatrix}
\]  

(2.2.7)

It is assumed that the errors are perturbations imposed on the solution of the linear system. If we subtract the exact equations from the difference equations with the perturbations, we obtain among the errors' components a set of relations that are identical to the relations for the components of \( \overline{U} \), as we are dealing with a linear system.

Introducing Eq. (2.2.6) into the set of equations which represents the relation of the errors and using only values of \( \xi \) and \( \tau \) on the grid results in relations between \( \overline{U} \) and \( \overline{\zeta} \). Using \( \lambda = e^{i \sigma \tau} \), then

\[
(\lambda - 1) \zeta_n^{*} + \frac{1}{2} \frac{\tau}{\xi} h \lambda (e^{i \sigma \xi} - e^{-i \sigma \xi}) U_n^{*} = 0
\]  

(2.2.8)

\[
(\lambda - 1) U_n^{*} + \frac{1}{2} \frac{\tau}{\xi} h \lambda (e^{i \sigma \xi} - e^{-i \sigma \xi}) \zeta_n^{*} = 0
\]  

(2.2.9)

Equations (2.2.8) and (2.2.9) represent two homogeneous linear equations in \( \zeta_n^{*} \) and \( U_n^{*} \). Since \( \zeta_n^{*} \) and \( U_n^{*} \) do not vanish identically, the determinant of this system must vanish, yielding a quadratic equation in \( \lambda \), from which we find
\[\lambda_{1,2} = \frac{1 \pm i \frac{t}{\ell} \sqrt{gh} \sin (\sigma \ell)}{1 + \frac{\pi^2}{\ell^2} gh \sin^2 (\sigma \ell)}\] (2.2.10)

and

\[|\lambda_{1,2}| < 1\] (2.2.11)

Thus, the line of errors which is introduced will decay with time, and the system is unconditionally stable.

In the investigation of stability, a linear homogeneous system is used. Thus, the wave representing the solution of the partial-differential equation will behave as the error waves described above. Rather than studying the propagation of the error wave, the growth or decay of the solutions of the system will determine the stability of computation for this particular case, as it is a homogeneous system.

The solutions of the system of difference equations expressed by the Fourier series are

\[\zeta(x,t) = \sum_n \zeta_n e^{i(\sigma_n x + \delta_n t)}\] (2.2.12)

\[U(x,t) = \sum_n U_n e^{i(\sigma_n x + \delta_n t)}\] (2.2.13)

For the numerical computations, these equations are used only on the grid points. As we are dealing with a linear system, only one term of the Fourier series need be considered. If these series are substituted into the difference equations, Eqs. (2.2.1) and (2.2.2), a common factor, \(e^{i\sigma t}\), can be canceled out, and only expressions with \(e^{i\sigma t}\) will be maintained.
The substitution can be written in the general matrix form

\[ [A] \overline{U}^{r+1} = [B] \overline{U}^r \]  \hspace{2cm} (2.2.14)

where

\[ \overline{U}^{r+1} = \begin{Bmatrix} \xi_{m+1}^{r+1} \\ \eta_{m+1}^{r+1} \end{Bmatrix} \]  \hspace{2cm} (2.2.15)

The matrices \([A]\) and \([B]\) for the difference equations (2.2.1) and (2.2.2) are

\[ [A] = \begin{bmatrix} 1 & \frac{1}{2} \frac{T}{\ell} h (e^{i\sigma \ell} - e^{-i\sigma \ell}) \\ + \frac{1}{2} \frac{T}{\ell} g (e^{i\sigma \ell} - e^{-i\sigma \ell}) & 1 \end{bmatrix} = \begin{bmatrix} 1 & i \frac{\tau}{\ell} h \sin (\sigma \ell) \\ i \frac{\tau}{\ell} g \sin (\sigma \ell) & 1 \end{bmatrix} \]  \hspace{2cm} (2.2.16)

\[ [B] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \]  \hspace{2cm} (2.2.17)

The matrices \([A]\) and \([B]\) of Eq. (2.2.14) depend on the time step \(\tau\) and the wave number \(\sigma\). Equation (2.2.14) can be written as

\[ U^{r+1} = [G(\tau, \sigma)] \overline{U}^r \]  \hspace{2cm} (2.2.18)

where

\[ [G(\tau, \sigma)] = [A]^{-1}[B] \]

The matrix \([G(\tau, \sigma)]\) is called the amplification matrix by Lax (see Richtmyer\(^9\)). In the simple case of the implicit system (Eqs. (2.2.1) and (2.2.2)), the amplification matrix is
\[
G(\tau, \sigma) = \begin{bmatrix}
\frac{1}{[1 + \tau^2 L^{-2} gh \sin^2 (\sigma \ell)]} & \frac{-i\tau L^{-1} h \sin (\sigma \ell)}{[1 + \tau^2 L^{-2} gh \sin^2 (\sigma \ell)]} \\
\frac{-i\tau L^{-1} h \sin (\sigma \ell)}{[1 + \tau^2 L^{-2} gh \sin^2 (\sigma \ell)]} & \frac{1}{[1 + \tau^2 L^{-2} gh \sin^2 (\sigma \ell)]}
\end{bmatrix}
\] (2.2.19)

For all values of \( \sigma \), stability requires that the \( r \)th power of the amplification matrix be bounded, where \( r \) is the finite number of operations performed with a finite time step \( \tau \). Several conditions of stability can be derived from the boundedness of the amplification matrix. (9)

In the analysis of the stability of computational methods, only the necessary condition and the third and fourth sufficient conditions of von Neumann will be used. The von Neumann necessary condition for stability states that the time step must be finite and that for all \( \sigma \) the eigenvalues of the amplification matrix must satisfy

\[ |\lambda| \leq 1 + 0(\tau) \] (2.2.20)

The third sufficient condition states that if there exists a constant \( a \) such that \( |\Delta| \geq a > 0 \) where \( \Delta \) is the determinant of the normalized eigenvectors of \( G \), the von Neumann condition is sufficient as well as necessary for stability. The fourth sufficient condition states that if the elements of the amplification matrix are bounded for the finite time step \( \tau \) and for all values of \( \sigma \) and that if all eigenvalues of this matrix, with the possible exception of one, are inside a unit circle, then the von Neumann condition is sufficient as well as necessary for stability. The one eigenvalue which may be outside the unit circle must satisfy the necessary condition (Eq. (2.2.20)). The eigenvalue of the amplification matrix of Eq. (2.2.19) of the sample system is

\[ \lambda_{1,2} = \frac{1 \pm i \frac{\tau}{L} \sqrt{gh} \sin (\sigma \ell)}{1 + \frac{\tau^2}{L^2} gh \sin^2 (\sigma \ell)} \] (2.2.21)
Thus, according to the fourth sufficient condition this system is unconditionally stable, as the absolute value of all the eigenvalues is always less than one.

2.3 WAVE DEFORMATION

According to Section 2.1, the solution of the simple implicit-difference equations converges to the solution of the differential equation if the space step and time step approach zero. The rate of convergence is generally derived experimentally from consideration of the truncation errors with decreasing step size and from sample cases executed with different time steps. In some instances important information concerning the numerical solution can be obtained by a factor which compares amplitude and phase of the components of the Fourier series representing the computed wave with the real solution, which is called the physical wave. Assuming a linear system, the only concern, then, is the relation of the components in amplitude and phase after a certain time interval.

Consider again the linear system

\[
\frac{\partial \sigma}{\partial t} + h \frac{\partial u}{\partial x} = 0 \tag{2.0.1}
\]

\[
\frac{\partial u}{\partial t} + g \frac{\partial \sigma}{\partial x} = - \frac{\partial \rho}{\partial x} \tag{2.0.2}
\]

The general solution of this system (omitting the forcing function) consists of the Fourier series

\[
\zeta = \zeta^* e^{i(\sigma x + \beta t)} \tag{2.3.1}
\]

\[
u = \nu^* e^{i(\sigma x + \beta t)} \tag{2.3.2}
\]

where \(\sigma\) is the wave number, and \(\zeta^*\) and \(\nu^*\) are constants. After inserting Eqs. (2.3.1) and (2.3.2) into Eqs. (2.0.1) and (2.0.2), the following relations are valid:
\[ \frac{S}{\sigma} = \pm \sqrt{8h} \]  

(2.3.3)

The term on the right-hand side is the velocity of wave propagation. Two waves can be in existence: a progressive and a retrogressive wave.

In numerical procedures the partial derivatives are replaced by differences in space and time \((\Delta x = \Delta l, \text{ and } \Delta t = \tau)\). Calculations are made at values of a network so that \( x = ml \text{ and } t = r\tau \), where \( m \) and \( r \) are integers. Three finite-difference schemes are now discussed.

Consider an implicit scheme with the difference operator

\[ \zeta_m^{r+1} - \zeta_m^r + \frac{1}{4} \frac{\tau}{l} h(u_m^{r+1} - u_m^{r+1} + \zeta_m^{r+1} - \zeta_m^r) = 0 \]  

(2.3.4)

\[ u_m^{r+1} - u_m^{r+1} + \frac{1}{4} \frac{\tau}{l} g(\zeta_m^{r+1} - \zeta_m^r + \zeta_m^{r+2} - \zeta_m^r) = 0 \]  

(2.3.5)

The general solution of the linear system of Eqs. (2.0.1) and (2.0.2) can be represented on the points of the lattice by

\[ \zeta_m^r = \zeta_m^* e^{i(\omega ml + \phi r\tau)} \]  

(2.3.6)

\[ u_m^r = u_m^* e^{i(\omega ml + \phi r\tau)} \]  

(2.3.7)

Introduction of these relations into Eqs. (2.3.4) and (2.3.5) results in two homogeneous equations:

\[ (e^{i\phi r\tau} - 1)\zeta_m^* + \frac{1}{4} \frac{\tau}{l} h(e^{i\phi r\tau} + 1)(e^{i\sigma l} - e^{-i\sigma l})u_m^* = 0 \]  

(2.3.8)

\[ (e^{i\phi r\tau} - 1)u_m^* + \frac{1}{4} \frac{\tau}{l} g(e^{i\phi r\tau} + 1)(e^{i\sigma l} - e^{-i\sigma l})\zeta_m^* = 0 \]  

(2.3.9)
For nontrivial values of $\zeta^*$ and $\vartheta^*$, two solutions are obtained for $e^{i\vartheta\tau}$. These are the eigenvalues discussed in previous sections:

$$e^{i\vartheta_1,\tau} = \frac{1 - A \pm 2i\sqrt{A}}{1 + A}$$

(2.3.10)

where

$$A = \frac{1}{4} \frac{\tau^2}{\ell^2} gh \sin^2 (\sigma \ell)$$

(2.3.11)

The imaginary part of $\Phi_1$ and $\Phi_2$ is zero, representing an absolute value of the eigenvalue equal to one; thus, $\Phi_1$ and $\Phi_2$ (which are real) are then found from

$$\sin (\Phi_1,\tau) = \frac{\pm 2\sqrt{A}}{1 + A} = \pm \frac{\tau}{\ell} \sqrt{gh} \sin (\sigma \ell) / \left[ 1 + \frac{\tau^2}{4\ell^2} gh \sin^2 (\sigma \ell) \right]$$

(2.3.12)

or

$$\Phi_{1,2,\tau} = \sin^{-1} \left[ \frac{\pm \frac{\tau}{\ell} \sqrt{gh} \sin (\sigma \ell)}{1 + \frac{\tau^2}{4\ell^2} gh \sin^2 (\sigma \ell)} \right]$$

(2.3.13)

Using an expansion of the inverse trigonometric function, we obtain for $A < 1$

$$\Phi_{1,2,\tau} = \pm \frac{\tau}{\ell} \sqrt{gh} \sin (\sigma \ell) \left[ \frac{1}{(1 + A)} + \frac{4}{6} \frac{A}{(1 + A)}^2 + \left( \frac{16}{2} \frac{3}{20} \frac{A^2}{(1 + A)^3} \right) \ldots \right]$$

(2.3.14)

and finally, by replacing $\sin (\sigma \ell)$ by its power series, we find
\[ \frac{B_{1,2}}{\sigma} = \pm \sqrt{gh} \left[ 1 - \frac{(\sigma t)^2}{3!} + \frac{(\sigma t)^4}{5!} - \frac{(\sigma t)^6}{7!} + \cdots \right] \]

\[ \times \left[ \frac{1}{1 + A} + \frac{4}{6} \frac{A}{(1 + A)^3} + \left( \frac{16}{2} \left( \frac{3}{20} \right) \frac{A^2}{(1 + A)^5} + \cdots \right) \right] \quad (2.3.15) \]

Both series in Eq. (2.3.15) are smaller than unity and approach unity when \( \tau \to 0 \) and also \( L \to 0 \). Thus, for all values of \( (\tau/L)\sqrt{gh} \), the computed wave propagates slower than the physical wave. Consequently, the wave frequency of the computed wave, now designated by the frequency \( B' \) in the following discussion, is lower than the frequency of the physical wave. However, as mentioned previously, the amplitude of the computed wave does not change for this system because the moduli of the eigenvalues are equal to unity.

In order to compare the behavior of various difference schemes, the concept of the complex propagation factor \( T(\sigma t) \) is introduced. This factor refers to a wave with a length \( L \), which is approximated by finite differences in spatial coordinates. The wave is followed over the wavelength. In some numerical processes the amplitude of the computed wave changes, and the velocity of propagation may differ from that of the physical wave. The propagation factor, expressed in the dimensionless parameter \( (\sigma t) \), is defined as the complex ratio of the computed wave in amplitude and phase to the physical wave after a time interval in which the physical wave propagates over its wavelength. Thus, the modulus of the propagation factor is a measure for the decay of the amplitude of a wave during computation, while the argument of this factor is a measure of the computed phase shift. A propagation factor exists for each wave in the system, in this case a forward and a backward wave. In the simple cases considered in this section, both factors are the same and can be expressed as

\[ T(\sigma t) = \frac{e^{i(\beta't + \omega x)}}{e^{i(\beta t + \omega x)}} \quad \text{for} \quad x = \frac{2\pi}{\sigma} = L \quad \text{and} \quad t = \frac{2\pi}{\beta} \]

\[ = e^{i2\pi(\beta'/\beta - 1)} \quad (2.3.16) \]
The modulus of the factor for the system discussed equals one, and as shown below, its argument represents the phase shift

\[
\arg[T(\sigma t)] = 2\pi \left[ \frac{\sin^{-1}\left\{ \frac{\tau}{\ell} \sqrt{g h} \sin (\sigma t) / \left[ 1 + \frac{1}{4} \frac{\tau^2}{\ell^2} gh \sin^2 (\sigma t) \right] \right\}}{\frac{\tau}{\ell} \sqrt{g h} (\sigma t)} - 1 \right]
\]

or

\[
\arg[T(\sigma t)] = 2\pi \left\{ \left[ 1 - \frac{(\sigma t)^2}{3!} + \frac{(\sigma t)^4}{5!} - \frac{(\sigma t)^6}{7!} + \ldots \right] \times \left[ \frac{1}{1 + A} + \frac{4}{6} \frac{A}{(1 + A)^3} + \left( \frac{16}{20} \frac{A^2}{(1 + A)^5} + \ldots \right) \right] - 1 \right\}
\]

(2.3.17) (2.3.18)

A positive value of this argument represents an acceleration of the computed wave. The amplification matrix G of the computational system given by Eqs. (2.3.4) and (2.3.5) has a complete set of linearly independent eigenvectors. Thus, the system is unconditionally stable according to the third sufficient condition.

As the second example, the unconditionally stable forward implicit scheme discussed in Section 2.2 is investigated:

\[
\zeta_{m+1}^{r+1} - \zeta_{m}^{r} + \frac{1}{2} \frac{\tau}{\ell} h(U_{m+1}^{r+1} - U_{m-1}^{r+1}) = 0
\]

(2.3.19)

\[
U_{m+1}^{r+1} - U_{m+1}^{r} + \frac{1}{2} \frac{\tau}{\ell} g (\zeta_{m+2}^{r+1} - \zeta_{m}^{r+1}) = 0
\]

(2.3.20)
Using Eqs. (2.3.1) and (2.3.2) results in the following homogeneous equations, with the designation $\beta'$ for the frequency of the computed wave:

\[
(e^{i\beta'\tau} - 1)e^{i\pi L} + \frac{1}{2} \frac{\tau}{L} \sin L \left( e^{i\pi L} - e^{-i\pi L} \right) \gamma = 0 \tag{2.3.21}
\]

\[
(e^{i\beta'\tau} - 1)\gamma + \frac{1}{2} \frac{\tau}{L} \cos L \left( e^{i\pi L} - e^{-i\pi L} \right) \gamma = 0 \tag{2.3.22}
\]

From these equations we find

\[
e^{i\beta'\tau} = \frac{1 \pm i \frac{\tau}{L} \sqrt{gh \sin (\pi L)}}{1 + \frac{\tau^2}{L^2} gh \sin^2 (\pi L)} \tag{2.3.23}
\]

Thus,

\[
\text{Im}(\beta'\tau) = \frac{1}{2} \ln \left[ 1 + \frac{\tau^2}{L^2} gh \sin^2 (\pi L) \right] \tag{2.3.24}
\]

and, consequently, the modulus of the eigenvalue is

\[
|\lambda| = e^{-\text{Im}(\beta'\tau)} \tag{2.3.25}
\]

Also, from Eq. (2.3.23)

\[
\text{Re}(\beta'\tau) = \tan^{-1} \left[ \frac{\pm \frac{\tau}{L} \sqrt{gh \sin (\pi L)}}{1 \pm \frac{\tau^2}{L^2} \sqrt{gh \sin (\pi L)}} \right] \tag{2.3.26}
\]

Consequently, the phase shift can be found in the following manner:

\[
\arg[T(\pi L)] = 2\pi \left\{ \tan^{-1} \left[ \frac{\pm \frac{\tau}{L} \sqrt{gh \sin (\pi L)}}{1 \pm \frac{\tau^2}{L^2} \sqrt{gh (\pi L)}} \right] - 1 \right\} \tag{2.3.27}
\]
If a time step \( \tau \) is used in a numerical calculation, the computation of the propagation of a physical wave having a frequency \( \sigma \) over its wavelength requires \( T/\tau = 2\pi/\sqrt{gh} \) time steps; if the frequency is expressed as a spatial frequency \( \sigma L \) by use of Eq. (2.3.3), the number of time steps required is \( \nu = 2\pi/(\tau/\sqrt{gh}) \sigma L \).

Consequently, the modulus of the propagation factor in this case is

\[
|T(\sigma L)| = |\lambda|^\nu = \left[ 1 + \frac{\tau^2}{L^2} gh \sin^2 (\sigma L) \right]^{-\frac{1}{2\nu}} \tag{2.3.28}
\]

which is smaller than unity. Thus, with this scheme the computed wave will decay with time.

The last example is an explicit scheme with the following equations:

\[
\zeta_m^{r+1} - \zeta_m^r + \frac{1}{2} \frac{\tau}{L} h(U_{m+1}^r - U_{m-1}^r) = 0 \tag{2.3.29}
\]

\[
U_{m+1}^{r+1} - U_{m+1}^r + \frac{1}{2} \frac{\tau}{L} g(\zeta_{m+2}^r - \zeta_m^r) = 0 \tag{2.3.30}
\]

The computed-wave frequencies are expressed in

\[
e^{i\beta^r \tau} = 1 \pm i \frac{\tau}{L} \sqrt{gh} \sin (\sigma L) \tag{2.3.31}
\]

It is clear that operations with this system are always unstable, as the necessary condition does not hold for all values of \( \sigma \). However, if this explicit operation is used in conjunction with the forward implicit method by alternating the two methods of operation, a method is formed with two different difference operators where waves, including the error wave, do not grow or decay. The operations are

\[
\zeta_m^{r+1} - \zeta_m^r + \frac{1}{2} \frac{\tau}{L} h(U_{m+1}^r - U_{m-1}^r) = 0 \tag{2.3.32}
\]
\[
\begin{align*}
U_{m+1}^{r+1} - U_m^r + \frac{1}{2} \frac{\tau}{\ell} \ g(C_{m+2}^r - C_m^r) &= 0 \\
C_m^{r+2} - C_m^{r+1} + \frac{1}{2} \frac{\tau}{\ell} \ h(U_{m+1}^{r+2} - U_m^{r+2}) &= 0 \\
U_{m+1}^{r+2} - U_m^{r+1} + \frac{1}{2} \frac{\tau}{\ell} \ g(C_{m+2}^{r+2} - C_m^{r+2}) &= 0
\end{align*}
\tag{2.3.33, 2.3.34, 2.3.35}
\]

Introduction of Eqs. (2.3.32) and (2.3.33) into Eqs. (2.3.34) and (2.3.35), respectively, results in the central implicit procedure described by Eqs. (2.3.4) and (2.3.5), but with a time step of \(2\tau\).

The truncation error of each of the operations is of the first order; however, the truncation error of the operations combined is of the second order, as described in detail in Section 2.1. The propagation factor and the amplification matrix of the two combined operations is naturally of the central implicit system, but then with a time step \(2\tau\).

From the discussion of the propagation factors of the three examples, it will be clear that the propagation factor is dependent on the discreteness of the variables in time and space.

It is possible to express the propagation factor of a linear system of which the time dimension is considered continuously and the space dimension on grid points. Such a system is the electrical analog of water wave motion in one space dimension. This system is described by the following equations:

\[
\begin{align*}
\left(\frac{\partial^2}{\partial t^2}\right)_m + \frac{h}{2\ell} \ (U_{m+1} - U_{m-1}) &= 0 \\
\left(\frac{\partial U}{\partial t}\right)_{m+1} + \frac{g}{2\ell} \ (C_{m+2} - C_m) &= 0
\end{align*}
\tag{2.3.36, 2.3.37}
\]

Using the general solutions given by Eqs. (2.3.1) and (2.3.2) with the designation \(\xi'\) for the computed-wave frequency results in
\[ \beta' = \pm \frac{(\sin \sigma t)}{l} \sqrt{\frac{g}{h}} \quad (2.3.38) \]

and the propagation factor is

\[ T(\sigma t) = \exp \{ i2\pi [\sin (\sigma t)/(\sigma t) - 1] \} \quad (2.3.39) \]

Thus, the computed wave is retarded.

The phase angle of the propagation factor of the central implicit scheme is shown in Fig. 2.3.1. The vertical scale on the right side indicates the phase lag, and the vertical scale on the left indicates the ratio of the computed-wave velocity to the physical-wave velocity. To show the discreteness of the system in the spatial dimension, the nondimensional quantity \( L/L \) is used to represent the number of parts composing the computed wave. The phase angle is plotted for a few values of the nondimensional parameter \( (\pi/L)\sqrt{gh} \). It will be noted that considerable phase shift will occur if less than 10 points are used for a value of \( (\pi/L)\sqrt{gh} = 1 \). With such a coarse approximation the computed-wave velocity will be much smaller than the physical-wave velocity.

The parameter \( (\pi/L)\sqrt{gh} \) is indicative of the number of operations in time \( (\nu) \) necessary for the computation of the propagation of the wave over its length for a given choice of a number of parts per wavelength \( (L/L) \). The total number of operations for this computation is \( \nu(L/L) \). Thus, after selection of a certain number of operations for this computation, the choice of \( L/L \) (or \( \nu \)) influences the propagation factor, as shown by the lines of equal computational effort in Fig. 2.3.1.

It will be noted that the phase angle is only very weakly influenced near \( (\pi/L)\sqrt{gh} = 2 \). This characteristic is important for the computation of wave propagation in coastal waters. If the nondimensional parameter is near the value of two, then the propagation of the computed wave in relation to the physical wave is hardly influenced by the variation in depth.

The modulus of the propagation factor of the central-implicit-method system equals one and is independent of the discreteness of representation in time or space.
Fig. 2.3.1—Phase angle of the propagation factor of the central implicit method
Figure 2.3.2 presents the phase angle of the propagation factor of the forward implicit method, using the same representation as in Fig. 2.3.1.

It is interesting to observe that when a representation of two parts per wavelength is used, the computed-wave velocity in both systems is zero. This representation is typical for waves which appear to be due to instabilities in computational methods (see Richtmyer(9)). If such waves are generated locally, they will tend to grow, since they cannot move away.

Figure 2.3.3 presents the modulus of the propagation factor of the forward implicit method. Waves computed with this system are damped.

The values of the propagation factors of the forward implicit scheme obtained from Eqs. (2.3.27) and (2.3.28) were confirmed by numerical experiment. As an initial condition a complete sine wave was assumed in a basin closed at both ends and divided into 20 sections (see Fig. 2.3.4), and water levels were computed at 10 points. Then, numerical calculations were made with Eqs. (2.3.19) and (2.3.20), assuming zero velocities at the start. The results obtained by these experiments were in agreement with those shown in Fig. 2.3.2. A typical plot of the decay of the amplitude vector is shown in Fig. 2.3.4.

2.4 GENERATION OF SPURIOUS MODES

Some computational methods have spurious solutions (see Fisher(10) and Lilly(11)). Consider the so-called leapfrog method.(9)

\[ \begin{align*}
\zeta_{m}^{r+1} - \zeta_{m}^{r-1} + \frac{\tau}{k} h(U_{m+1}^{r} - U_{m-1}^{r}) &= 0 \quad (2.4.1) \\
U_{m}^{r+1} - U_{m}^{r-1} + \frac{\tau}{k} \delta(\zeta_{m+1}^{r} - \zeta_{m-1}^{r}) &= 0 \quad (2.4.2)
\end{align*} \]

This system can be employed rather economically by use of staggered sets of the water-level and velocity values. This is achieved by computing the water level on the odd time-sequence numbers and on the even
Fig 2.3.2—Phase angle of the propagation factor of the forward implicit method.

Computed wave velocity/physical-wave velocity

Parts per wavelength (L/2)

Phase lag (deg)

900 operations
400
100
Continuous in time

\( T / \sqrt{gh} = 0.5 \)
Fig. 2.3.3—Modulus of the propagation factor of the forward implicit method.
Fig. 2.3.4—Complex plot of the amplitude during 21 steps of a numerical computation
spatial numbers and by computing the velocity on the even time-sequence numbers and on the odd spatial numbers.

If the general solution (Eqs. (2.3.1) and (2.3.2)) of the linear system (Eqs. (2.0.1) and (2.0.2)) is introduced into this three-level equation, we obtain

\[(e^{i \theta '})^{2} - 2 + 4 \frac{\tau^{2}}{L} gh \sin^{2} (\sigma l) + (e^{i \theta '})^{-2} = 0 \quad (2.4.3)\]

if

\[b = 1 - 2 \frac{\tau^{2}}{L} gh \sin^{2} (\sigma l) \quad (2.4.4)\]

\[(e^{i \theta '})_{1,2,3,4} = \pm \left( b \pm \sqrt{b^{2} - 1} \right)^{\frac{1}{8}} \quad (2.4.5)\]

These four roots are on a unit circle if \(-1 \leq b \leq 1\), as indicated in Fig. 2.4.1. The two roots in the right half plane represent the computed physical modes, and the two in the left half plane represent only computational modes.

Following Eq. (2.4.5), the general solutions of the wave amplitude from Eq. (2.3.1) can be written

\[\zeta(x, t) = [\zeta_{1}^{*} \cos (\sigma x') + \zeta_{2}^{*}(-1)^{r} \cos (\sigma x')]e^{i \omega ml}\]

The solution obtained by this method results in an approximation of the real solution of the differential equation and a spurious solution of the difference equation consisting of a single time-step oscillation modulated by the approximate frequency of the real solution.

2.5 EFFECTS OF NONLINEAR TERMS

Lamb(1) indicates that higher harmonics are generated for wave
Fig. 2.4.1—Complex representation of the roots of the characteristic equation for the leapfrog system.
motion, which is represented by

\[
\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + g \frac{\partial r}{\partial x} = 0 \tag{2.5.1}
\]

\[
\frac{\partial}{\partial x} \left( h + \zeta \right) u + \frac{\partial r}{\partial t} = 0 \tag{2.5.2}
\]

In his example, a harmonic wave is used at the end of a rectangular channel, and a perturbation development shows that when the wave propagates, it will contain higher harmonics. Inversely, it can be shown that when starting from a sinusoidal surface in a channel, waves with wavelength of \( \frac{\lambda}{2}, \frac{\lambda}{4}, \frac{\lambda}{8}, \ldots \) are generated as time progresses. Energy is transferred from longer to shorter waves.

Using a numerical procedure, part of this energy will accumulate in waves with a wavelength of the order of \( 2\lambda \). Shorter waves cannot be represented in the procedure. In addition, as seen in the previous section, the computed-wave velocity of these very short waves is zero or close to zero, thus aggravating this effect.

In order to obtain meaningful solutions, a mechanism is necessary for the dissipation of waves with a wavelength of the order of \( 2\lambda \). Thus it will be required that all the eigenvalues are less than one for these wavelengths. Generally, this is accomplished by the so-called artificial-viscosity terms, which cause smoothing of curvatures. For the explicit methods, these terms are added to the system of difference equations (Hansen\(^6\)) or are "built in" as the well-known Lax-Wendroff schemes. (12,13) Implicit schemes often show eigenvalues less than one for frequencies close to this critical wave frequency. (14) For example, the eigenvalues of the forward implicit scheme in Section 2.2 are less than one, and consequently the modulus of the propagation factor is also less than one (see Fig. 2.3.2).

However, in all these methods some effect is also noticeable for longer waves, as was illustrated in the forward implicit method. For this reason, it seems better not to use methods which show the typical characteristics of small eigenvalues for wavelengths of the order of \( 2\lambda \), but to eliminate the shorter waves by making a Fourier representation
of water levels and velocities from time to time and by omitting all the shorter waves. The latter method has been used by Phillips\(^{(15)}\) in numerical weather forecasting.

A description has been given of the mechanics of generation of short waves by nonlinear equations and the process of accumulation of these short waves—if no physical mechanism of dissipation is present—because of the inability of the grid system to resolve these waves properly. Unfortunately, once the process of accumulation begins, the numerical computations feed the short waves back into the system through the nonlinear terms, and the system becomes nonlinearly unstable. This feedback can be eliminated if the finite-difference expression for the nonconstant parameter is chosen in such a manner that it does not represent waves with a wavelength of \(2\ell\). For example, the finite-difference approximation for the first term of Eq. (2.5.2) can be expressed by

\[
- \frac{1}{2\ell} \left\{ \left[ h_{m-1} + \frac{1}{2} \left( \zeta_{m-2} + \zeta_m \right) \right] u_{m-1} - \left[ h_{m+1} + \frac{1}{2} \left( \zeta_m + \zeta_{m+2} \right) \right] u_{m+1} \right\}
\]

where the water level is computed on the even values of \(m\) and velocities on the odd values of \(m\).

These methods all refer to eliminating part or all of the troublesome short waves for computational reasons.

In the physical-wave systems a free interaction exists between wave motions which we can express with finite differences and those which have wavelengths of the order of the spatial grid size or smaller. The degree of the interaction between the two groups depends naturally on the extent of the nonlinear behavior of the system and upon the choice of grid. For some fluid-mechanics computations like numerical weather forecasting, the interaction between the two may be significant because of limits on grid size (computer memory). Rather than simply eliminate the effects of short waves, it may be possible to simulate the interaction between the two groups. Such procedures have been designed by Arakawa\(^{(16)}\) but are not yet completely published.
2.6 EFFECTS OF BOTTOM STRESS

In many experimental studies concerning long-wave motion, an estimate of bottom roughness is obtained from a comparison of computed data and measured data. It will be shown here that the coefficients used to express the effects of bottom roughness are influenced by the grid size, time step, and approximation.

In its simplest form, the effect of bottom roughness can be introduced by a linearization where the decrease of the water-level gradient is taken as a function of the velocity, as follows:

\[
\frac{\partial \zeta}{\partial t} + h \frac{\partial u}{\partial x} = 0 \tag{2.6.1}
\]

\[
\frac{\partial u}{\partial t} + g \frac{\partial \zeta}{\partial x} + ku = 0 \tag{2.6.2}
\]

The general solutions are of the form

\[
\zeta(x,t) = \zeta^* \exp \{i(\beta t + \sigma x)\} \tag{2.3.1}
\]

\[
u(x,t) = \nu^* \exp \{i(\beta t + \sigma x)\} \tag{2.3.2}
\]

Introduction of Eqs. (2.3.1) and (2.3.2) into Eqs. (2.6.1) and (2.6.2) gives

\[
i\beta \zeta^* + i\sigma \nu^* = 0 \tag{2.6.3}
\]

\[
ig\sigma \zeta^* + (i\beta + k)\nu^* = 0 \tag{2.6.4}
\]

For the relation between \(\beta\) and \(\sigma\), we find

\[
i\beta(i\beta + k) + gh \sigma^2 = 0 \tag{2.6.5}
\]
or

\[
\sigma = \sigma \left\{ \frac{k}{2\sigma} \pm \sqrt{[gh - (k/2\sigma)'^2]} \right\} \tag{2.6.6}
\]

Thus, if a spatial harmonic wave is taken as an initial condition, it will decay with time, or if at a particular point in space a harmonic motion is introduced, it will decay with distance. This decay is for both the progressive wave and the retrogressive wave. After a time \( \tau \), a spatial wave will have an amplitude of \( \exp (-\frac{1}{2}k\tau) \) and will be advanced over a distance \( \tau \sqrt{[gh - (k/2\sigma)'^2]} \).

A typical example of the difference approximation of Eqs. (2.6.1) and (2.6.2) is

\[
\zeta_m^{r+1} - \zeta_m^r + \frac{1}{4} \frac{\tau}{L} h(u_{m+1}^{r+1} - u_m^{r+1} + u_{m+1}^r - u_{m-1}^r) = 0 \tag{2.6.7}
\]

\[
\frac{1}{4} \frac{\tau}{L} g(\zeta_{m+2}^{r+1} - \zeta_m^{r+1} + \zeta_{m+2}^r - \zeta_m^r) + (1 + \frac{1}{2} \tau k')u_m^{r+1} - (1 - \frac{1}{2} \tau k')u_{m+1}^r = 0 \tag{2.6.8}
\]

The frictional coefficient is designated \( k' \) and may be found experimentally from numerical results. The frequency of the computed system is designated \( \beta' \).

If the solutions in the form of Eqs. (2.3.6) and (2.3.7) are introduced into the difference equations, Eqs. (2.6.7) and (2.6.8), we obtain

\[
(e^{i \beta' \tau} - 1)\zeta^\infty + \frac{1}{4} \frac{\tau}{L} h(e^{i \beta' \tau} + 1)(e^{i \sigma L} - e^{-i \sigma L})u^* = 0 \tag{2.6.9}
\]

\[
\frac{1}{4} \frac{\tau}{L} (e^{i \beta' \tau} + 1)(e^{i \sigma L} - e^{-i \sigma L})\zeta^\infty + [(1 + \frac{1}{2} \tau k')e^{i \beta' \tau} - (1 - \frac{1}{2} \tau k')]u^* = 0 \tag{2.6.10}
\]
Setting

\[ A = \frac{1}{4} \frac{\tau^2}{l^2} gh \sin^2 (\sigma l) \]  
(2.6.11)

then

\[ e^{i\theta_{1,2}^T} = \frac{1 - A \pm \sqrt{4A - \frac{1}{4} \tau^2 k'^2}}{1 + A + \frac{1}{2} \tau k'} \]  
(2.6.12)

\[ \text{Im}(\theta_{1,2}^T) = -\frac{1}{2} \ln \left( \frac{1 + A - \frac{1}{2} \tau k'}{1 + A + \frac{1}{2} \tau k'} \right) \]  
(2.6.13)

\[ \text{Re}(\theta_{1,2}^T) = \tan^{-1} \left[ \frac{\sqrt{4A - \frac{1}{4} (\tau k')^2}}{(1 - A)} \right] \]  
(2.6.14)

If a time step \( \tau \) is used, the number of operations (\( v \)) to be performed for the time that the physical wave propagates over its wavelength is

\[ v = \frac{2\pi}{\tau \sigma gh - (k/2\sigma)^2} \]  
(2.6.15)

The modulus of the propagation factor is

\[ |T(\sigma l)| = \left[ \frac{\sqrt{(1 + A - \frac{1}{2} \tau k')/(1 + A + \frac{1}{2} \tau k')}}{\exp \left( -\frac{1}{2} \tau k \right)} \right] v \]  
(2.6.16)

If \( k' = k \), and \( \tau > 0 \), then \( T(\sigma l) \neq 1 \).

The phase angle of the propagation factor is
\[ \arg[T(\sigma \phi)] = 2\pi \left\{ \tan^{-1} \left[ \frac{\sqrt{A - \frac{1}{4} (\tau k')^2} / (1 - A)}{(\tau / l) \sqrt{gh - (k/2c)^2} (\sigma f)} \right] - 1 \right\} \] (2.6.17)

If numerical experiments are used to determine the \( k' \) value in comparing field data and computed data, considerable difficulties are encountered. Satisfying the modulus (\( |T| = 1 \)) and the argument (\( \arg(T) = 0 \)) by adjusting the \( k' \) value is usually impossible. If only the modulus (\( |T| = 1 \)) is satisfied, the argument generally will not be satisfied.

A multi-operation method with a time step \( \frac{1}{2} \tau \) is

\[ \zeta^{r+1}_m - \zeta^r_m + \frac{1}{4} \frac{\tau}{l} h (u^{r+1}_{m+1} - u^r_{m-1}) = 0 \] (2.6.18)

\[ (1 + \frac{1}{4} \tau k') u^{r+1}_{m+1} - (1 - \frac{1}{4} \tau k') u^r_{m+1} + \frac{1}{4} \frac{\tau}{l} g (\zeta^{r+1}_{m+2} - \zeta^r_m) = 0 \] (2.6.19)

\[ \zeta^{r+1}_m - \zeta^r_m + \frac{1}{4} \frac{\tau}{l} h (u^{r+1}_{m+1} - u^r_{m-1}) = 0 \] (2.6.20)

\[ (1 + \frac{1}{4} \tau k') u^{r+1}_{m+1} - (1 - \frac{1}{4} \tau k') u^r_{m+1} + \frac{1}{4} \frac{\tau}{l} g (\zeta^{r+1}_{m+2} - \zeta^r_m) = 0 \] (2.6.21)

Substitution of the general solution in Eqs. (2.6.18) and (2.6.19) results in

\[
\begin{vmatrix}
(\lambda - 1) & \frac{1}{2} i \frac{\tau}{l} h \sin (\sigma \phi) \\
\frac{1}{2} i \frac{\tau}{l} g \sin (\sigma \phi) & (1 + \frac{1}{4} \tau k') \lambda - (1 - \frac{1}{4} \tau k')
\end{vmatrix} = 0 \] (2.6.22)
Thus,

\[
\lambda_{1,2} = \frac{1 \pm i\sqrt{\lambda(1 + \frac{1}{4} \tau k') - \frac{1}{16} \tau^2 k'^2}}{1 + \frac{1}{4} \tau k}
\]  \hspace{1cm} (2.6.23)

Substitution of the general solution in Eqs. (2.6.20) and (2.6.21) results in

\[
\begin{vmatrix}
(\lambda - 1) & \frac{1}{2} i \frac{\tau}{\lambda} \sin (\sigma\lambda) \\
& 0 \\
+ \frac{1}{2} i \frac{\tau}{\lambda} g \sin (\sigma\lambda) & (1 + \frac{1}{4} \tau k')\lambda - (1 - \frac{1}{4} \tau k')
\end{vmatrix}
\]  \hspace{1cm} (2.6.24)

and

\[
\lambda_{1,2} = \frac{1 \pm i\sqrt{\lambda(1 - \frac{1}{4} \tau k') - \frac{1}{16} \tau^2 k'^2}}{1 + \lambda + \frac{1}{4} \tau k'}
\]  \hspace{1cm} (2.6.25)

The modulus of the propagation factor can be found by multiplication of the two sets of eigenvalues from Eqs. (2.6.23) and (2.6.25):

\[
|\Gamma(\sigma\lambda)| = \left\{ \begin{array}{c}
\sqrt{\left[1 + A\left(1 - \frac{1}{4} \tau k'\right) - \frac{1}{16} \tau^2 k'^2\right] \left[1 + A\left(1 + \frac{1}{4} \tau k'\right) - \frac{1}{16} \tau^2 k'^2\right]}
\exp\left(-\frac{1}{2} \kappa \tau\right)
\end{array} \right\}^\nu
\]  \hspace{1cm} (2.6.26)

where \(\nu\) is expressed by Eq. (2.6.15).

The phase angle is
\[
\text{arg}[T(\sigma \ell)] = 2\pi \left[ \frac{\tan^{-1}\sqrt{A \left(1 + \frac{1}{4} \tau k'\right)} - \frac{1}{16} \tau^2 k' \tau^2}{(\tau/l\lambda)^2gh - (k/2\sigma)^2(\sigma \ell)} \right] - 1
\]

(2.6.27)

The so-called leapfrog system, which has been used extensively in meteorological calculation, is presented as an example of an explicit system. In this case, however, a linear damping will be added, as in the two systems discussed previously with their implicit feature. The equations of the explicit system are

\[
\zeta^r_{m+\frac{1}{2}} - \zeta^r_{m-\frac{1}{2}} + \frac{1}{2} \frac{\tau}{L} h(U^r_{m+1} - U^r_{m-1}) = 0
\]

(2.6.28)

\[
(1 + \frac{1}{2} \tau k')u^{r+1}_{m+1} - (1 - \frac{1}{2} \tau k')u^r_{m+1} + \frac{1}{2} \frac{\tau}{L} g(C^r_{m+1} - C^r_{m-1}) = 0
\]

(2.6.29)

Using the previous procedures and substituting the general solution into the difference equations, then

\[
\begin{vmatrix}
(\lambda - 1) & + i \frac{\tau}{L} h\sqrt{A} \sin (\sigma \ell) \\
i \frac{\tau}{L} g\sqrt{A} \sin (\sigma \ell) & \left(1 + \frac{1}{2} \tau k'\right)\lambda - \left(1 - \frac{1}{2} \tau k'\right)
\end{vmatrix} = 0
\]

(2.6.30)

Thus

\[
\lambda_{1,2} = \frac{1 - 2A \pm i\sqrt{4A - 4A^2 - \frac{1}{4} \tau^2 k' \tau^2}}{1 + \frac{1}{2} \tau k'}
\]

(2.6.31)
In the stable computational region, the modulus of the propagation factor is

\[
|T(\sigma \xi)| = \left[\frac{\sqrt{(1 - \frac{1}{2} \tau k')/(1 + \frac{1}{2} \tau k')}}{\exp \left(-\frac{1}{2} k \tau\right)}\right]^\nu
\]

(2.6.32)

and the phase angle is

\[
\arg[T(\sigma \xi)] = 2\pi \left[\frac{\tan^{-1}\sqrt{\left(4A - 4A^2 - \frac{1}{4} \tau^2 k'^2\right)/(1 - 2A)}}{(\tau / \xi) \sqrt{gh - (k/2\sigma))^2(\sigma \xi)}} - 1\right]
\]

(2.6.33)

The phase angles and the modulus of the propagation factors of the three methods discussed are presented in Figs. 2.6.1 through 2.6.6. The damping is expressed nondimensionally for a value of \(k/(\sqrt{\sigma \tau}) = 0.4\).

In all three cases the computed wave has a lower velocity of propagation than the physical wave. The graph of the modulus of the leapfrog method indicates that for \(k' = k\), the propagation factor is smaller than unity. Both of the other methods indicate a modulus exceeding unity.

A comparison of Fig. 2.6.1 and Fig. 2.3.1, the latter representing the phase angle of the factor without damping, indicates that the phase angle is not noticeably affected by the linear damping.

The two methods with implicit operations have moduli larger than unity (Figs. 2.6.2 and 2.6.4). Figure 2.6.6 indicates that for the leapfrog method the modulus is smaller than unity.

Note that the choice of approximation for the bottom-stress term has to be made with care. It is probable that the method becomes unstable just because of this term. Such a case is shown in the description of the numerical experiments.

2.7 STABILITY ASPECTS OF THE CONVECTIVE-INERTIA TERMS

Richtmyer has indicated that it is sometimes advantageous to use off-centered differences in implicit equations, whereby a constant
Fig. 2.6.1—Phase angle of the propagation factor of the central implicit method

Fig. 2.6.2—Modulus of the propagation factor of the central implicit method
Fig. 2.6.3—Phase angle of the propagation factor of the multioperation method

Fig. 2.6.4—Modulus of the propagation factor of the multioperation method
Fig. 2.6.5 — Phase angle of the propagation factor of the leapfrog method

Fig. 2.6.6 — Modulus of the propagation factor of the leapfrog method
(θ) is used as a measure of the eccentricity. This constant can be used in the spatial dimension and in the time dimension.

In the spatial dimension, Ito et al.\(^{(19)}\) used off-centered differences for the convective-inertia terms in the long-wave computation and Lelevier used off-centered differences in the one-dimensional fluid flow problems described by Lax.\(^{(9)}\) Both investigators gave the constant (θ) fixed values of zero or unity, depending on the direction of the current. The use of this method can be illustrated by considering the partial-differential equations of the one-dimensional system, which are written

\[
\frac{\partial c}{\partial t} + h \frac{\partial u}{\partial x} = 0
\]  
(2.7.1)

\[
\frac{\partial u}{\partial t} + g \frac{\partial c}{\partial x} + u \frac{\partial u}{\partial x} = 0
\]  
(2.7.2)

Assuming that the flow fluctuations are small compared to the speed of the basic flow \(U_o\), the last equation may be written

\[
\frac{\partial u}{\partial t} + g \frac{\partial c}{\partial x} + U_o \frac{\partial u}{\partial x} = 0
\]  
(2.7.3)

First, the possible error growth of the convective term will be considered, and for this example only the interaction between the spatial derivative and the time derivative will be investigated. Using a weighting function (θ) as a measure of eccentricity of the spatial derivative and omitting the term \(g(\partial c/\partial x)\), Eq. (2.7.3) can be written as follows:

\[
U_m^{r+1} - U_m^r + \frac{1}{2} U_o \left[(1 - \theta)(U_{m+2}^{r+1} - U_m^{r+1}) + \theta(U_m^{r+1} - U_{m-2}^{r+1})\right] = 0
\]  
(2.7.4)

The amplification factor of this limited system is
\[ \lambda = \left\{ 1 - \frac{T}{2L} U_o \left[ (1 - 2\theta) \cos(2\sigma \xi) - (1 - 2\theta) + i \sin(2\sigma \xi) \right] \right\}^{-1} \]

(2.7.5)

If \( U_o < 0 \), a stable system is obtained for \( \theta \) values in the range \( 0 \leq \theta \leq 0.5 \). For the limiting case \( \theta = 0.5 \), the eigenvalue \( \lambda \) satisfies the necessary condition and in this case also the sufficient condition

\[ \lambda \leq 1 + 0(\Delta t) \]

(2.7.6)

If \( U_o > 0 \), then a stable system is obtained for \( \theta \) values in the range \( 0.5 \leq \theta \leq 1 \). Again, for the limiting case \( \theta = 0.5 \), both the necessary and sufficient conditions are satisfied.

Thus the use of a centered spatial derivative of the convective term \( \theta = 0.5 \) poses no particular problems as to stability as far as the middle of the field is concerned.

In an actual computation it is often not possible to determine the spatial derivative centered along a boundary, as in this case velocity values have to be given at two locations. If only one value is given on an upper limit \( M \) and the derivative is taken from inside the field \( (\theta = 1) \), we obtain

\[ \lambda' = \left\{ 1 - \frac{T}{2L} U_o \left[ 1 - \cos(2\sigma \xi) + i \sin(2\sigma \xi) \right] \right\}^{-1} \]

(2.7.7)

If \( U_o > 0 \), then \( \lambda \leq 1 \) is thus stable, but if \( U_o < 0 \), then the necessary condition is not satisfied, and the system will not be stable locally.

Numerical experimentation with the two-dimensional computations discussed in Section 4 confirmed the above analyses. The convective-inertia term was taken centrally in the spatial dimensions. The system becomes less stable along the boundaries if off-centered differences are used.

Next, stability aspects of the finite-difference approximations in
the more complete, nonlinear, wave equations (Eqs. (2.5.1) and (2.5.2)) are investigated. The following multioperation method is considered:

\[
\zeta_{m+1}^{r+1} - \zeta_{m+1}^{r} + \frac{T}{2\ell} \left\{ \left[ h + \frac{1}{2} \left( \zeta_{m+3}^{r+1} + \zeta_{m+1}^{r+1} \right) \right] u_{m+2}^{r+1} \right. \\
- \left. \left[ h + \frac{1}{2} \left( \zeta_{m+1}^{r+1} + \zeta_{m-1}^{r+1} \right) \right] u_{m}^{r+1} \right\} = 0 \quad (2.7.8)
\]

\[
U_{m}^{r+1} - U_{m}^{r} + \frac{T}{2\ell} \left\{ \zeta_{m+1}^{r+1} - \zeta_{m-1}^{r+1} \right. \\
+ \frac{T}{4\ell} U_{m}^{r+1} \left( U_{m+2}^{r+1} - U_{m-2}^{r+1} \right) \left. \right\} = 0 \quad (2.7.9)
\]

\[
\zeta_{m+1}^{r+2} - \zeta_{m+1}^{r+1} + \frac{T}{2\ell} \left\{ \left[ h + \frac{1}{2} \left( \zeta_{m+2}^{r+1} + \zeta_{m+1}^{r+1} \right) \right] u_{m+2}^{r+1} \right. \\
- \left. \left[ h + \frac{1}{2} \left( \zeta_{m+1}^{r+1} + \zeta_{m-1}^{r+1} \right) \right] u_{m}^{r+1} \right\} = 0 \quad (2.7.10)
\]

\[
U_{m}^{r+2} - U_{m}^{r+1} + \frac{T}{2\ell} \left\{ \zeta_{m+1}^{r+1} - \zeta_{m-1}^{r+1} \right. \\
+ \frac{T}{4\ell} U_{m}^{r+1} \left( U_{m+2}^{r+1} - U_{m-2}^{r+1} \right) \left. \right\} = 0 \quad (2.7.11)
\]

Actual computation with this system appears difficult because of the nonlinearities which are involved.

If the assumption is made that the fluctuations in time of the local velocity and the local water depth \((h + \zeta) = H\) are small, the eigenvalues of the amplification matrix for the first two equations can be computed according to Section 2.3:

\[
\lambda_{1,2} = \frac{1 + i(D + A)}{(1 + iD)^2 + A^2} = \frac{1}{1 + i(D + A)} \quad (2.7.12)
\]
where

\[ A = \frac{T}{2} \sqrt{gh} \sin (\alpha l) \]  \hspace{1cm} (2.7.13) \\

\[ D = \frac{1}{2} \frac{T}{l} U_0 \sin (2\alpha l) \]  \hspace{1cm} (2.7.14) \\

and $U_0$ is the speed of the basic flow.

For Eqs. (2.7.10) and (2.7.11) the eigenvalues are

\[ \lambda_{1,2} = 1 - i(D \mp A) \]  \hspace{1cm} (2.7.15) \\

The amplification matrix of the computational scheme given by Eqs. (2.7.8) through (2.7.11) has a complete set of linearly independent eigenvectors, so the system is unconditionally stable according to the third sufficient condition.

2.8 CONSERVATION OF MASS AND MOMENTUM

During numerical computation, the mass and momentum in the bounded system should be conserved when accounting for the increase or decrease of mass and momentum through the bounds. If mass or momentum is added during computation, wave amplitudes will increase in time and the computations will tend to become unstable. Conversely, it may be expected that methods which are computationally stable dissipate mass or momentum.

With linear systems, dissipation of mass or momentum can be checked rather easily. If nonlinear terms are included, finite-difference approximations can be obtained which satisfy the requirements of conservation of mass and momentum, but the approximations are generally complicated. For example, the nonlinear system represented by Eqs. (2.5.1) and (2.5.2) can be written in the vector form

\[ \frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = 0 \]  \hspace{1cm} (2.8.1)
where

\[
\bar{U} = \begin{Bmatrix} \bar{\zeta} \\ \bar{U} \end{Bmatrix}
\]  \hspace{1cm} (2.8.2)

and

\[
\bar{F}(\bar{U}) = \begin{Bmatrix} [h(x) + \zeta(x)]u(x) \\ g_\zeta(x) + \frac{1}{2} [u(x)]^2 \end{Bmatrix}
\]  \hspace{1cm} (2.8.3)

According to the conservation laws, \((12,13,20)\) integration of Eq. \((2.8.1)\) in time and space between their respective boundaries \(T_1, T_2, L_1,\) and \(L_2\) yields zero:

\[
\int_{L_1}^{L_2} \bar{U} \, dx \bigg|_{T_1}^{T_2} + \int_{T_1}^{T_2} \bar{F}(\bar{U}) \, dt \bigg|_{L_1}^{L_2} = 0 \hspace{1cm} (2.8.4)
\]

The difference procedure approximating Eq. \((2.8.1)\) should also satisfy the numerical integration of Eq. \((2.8.4)\). For the approximation, a central implicit system is used. As discussed in Section 2.5, the nonlinear term in the equation of continuity can be expressed in such a manner that feedback of very short waves is eliminated. If the equation of continuity is written

\[
\zeta_{m+1}^r - \zeta_m^r + \frac{\tau}{2L} \left[ [h_{m+\frac{1}{2}}^r + \frac{1}{2} (\zeta_{m+1}^r + \zeta_{m}^r)]u_{m+\frac{1}{2}}^{r+1} - [h_{m-\frac{1}{2}}^r + \frac{1}{2} (\zeta_m^r + \zeta_{m-1}^r)]u_{m-\frac{1}{2}}^{r+1} \right]
\]

\[
+ \left[ h_{m+\frac{1}{2}}^r + \frac{1}{2} (\zeta_{m+1}^r + \zeta_{m}^r) \right]u_{m+\frac{1}{2}}^r - \left[ h_{m-\frac{1}{2}}^r + \frac{1}{2} (\zeta_m^r + \zeta_{m-1}^r) \right]u_{m-\frac{1}{2}}^r = 0
\]

\hspace{1cm} (2.8.5)

then numerical integration over one time step \(\tau\) for the range \(m = 1\) to \(m = M\) results in
\[ \ell \sum_{m=1}^{m=M} \zeta_{m}^{r+1} = \ell \sum_{m=1}^{m=M} \zeta_{m}^{r} + \frac{1}{2} \tau \left( [h_{\frac{1}{2}}^r + \frac{1}{2} (\zeta_{1}^{r+1} + \zeta_{0}^{r+1})] U_{\frac{1}{2}}^{r+1} \right. \\
+ \left. [h_{\frac{1}{2}}^r + \frac{1}{2} (\zeta_{1}^{r} + \zeta_{0}^{r})] U_{\frac{1}{2}}^{r} - [h_{M+\frac{1}{2}}^r + \frac{1}{2} (\zeta_{M+1}^{r+1} + \zeta_{M}^{r+1})] U_{M+\frac{1}{2}}^{r+1} \right) \\
- \left. [h_{M+\frac{1}{2}}^r + \frac{1}{2} (\zeta_{M}^{r} + \zeta_{M+1}^{r})] U_{M+\frac{1}{2}}^{r} \right) \\
(2.8.6) \]

This shows that over one time step, the mass is conserved within the spatial boundaries if one accounts for the mass which enters through the boundaries.

If the equation of motion is written

\[ U_{m}^{r+1} - U_{m}^{r} + \frac{\tau}{2\ell} \left( \zeta_{m+\frac{1}{2}}^{r+1} - \zeta_{m-\frac{1}{2}}^{r+1} + \zeta_{m+\frac{1}{2}}^{r} - \zeta_{m-\frac{1}{2}}^{r} \right) \]

\[ + \frac{\tau}{8\ell} \left[ (U_{m+\frac{1}{2}}^{r+1} + U_{m-\frac{1}{2}}^{r+1})^2 - (U_{m+\frac{1}{2}}^{r} + U_{m-\frac{1}{2}}^{r})^2 \right] = 0 \]

(2.8.7)

then numerical integration over one time step \( \tau \) for the range \( m = 1 \) and \( m = M \) results in

\[ \ell \sum_{m=1}^{m=M} U_{m}^{r+1} = \ell \sum_{m=1}^{m=M} U_{m}^{r} + \frac{1}{2} \tau \left( \zeta_{M+\frac{1}{2}}^{r+1} - \zeta_{M-\frac{1}{2}}^{r+1} - \zeta_{M+1}^{r} + \zeta_{M}^{r} \right) \]

\[ + \frac{1}{8} \tau \left[ (U_{M+\frac{1}{2}}^{r+1} + U_{M-\frac{1}{2}}^{r+1})^2 - (U_{M+\frac{1}{2}}^{r} + U_{M-\frac{1}{2}}^{r})^2 \right] \]

(2.8.8)

Thus, over one time step, the momentum is conserved within the spatial boundaries if one accounts for the impulse caused by the pressures at the boundaries and for the momentum which enters with the flow through the boundaries into the field.

The central implicit system represented by Eqs. (2.8.5) and (2.8.7) is very cumbersome and impractical for actual computation. Not only
do quadratic terms appear in the system of equations, but also computations have to be made for \( U \) and \( \zeta \) on every integer and on every half number of the location because of the convective inertia term in Eq. (2.8.7). If this term is omitted, then the computation can be made by taking \( U \) and \( h \) on the integers and the water levels \((\zeta)\) on the half-number locations. Consequently, an expression for the convective-inertia term by which the spatial derivative is taken over a distance \( 2L \) is desirable. A finite-difference approximation of the convective term will now be chosen in the nondivergence form as presented in the differential equation Eq. (2.5.1), and the equation of motion is now written

\[
U^r_{m} - U^r_m + \frac{\tau}{2L} \left( \zeta^r_{m+\frac{1}{2}} - \zeta^r_{m-\frac{1}{2}} + \zeta^r_{m+\frac{1}{2}} - \zeta^r_{m-\frac{1}{2}} \right) + \frac{\tau}{2L} U^r_m (U_{m+1}^r - U_{m-1}^r) = 0
\]

(2.8.9)

Numerical integration over one time step \( \tau \) for the range \( m = 1 \) to \( m = M \) results in

\[
\tau \sum_{m=1}^{m=M} U^r_{m+1} = \tau \sum_{m=1}^{m=M} U^r_m + \frac{1}{2} \tau \left( \zeta^r_{M+\frac{1}{2}} + \zeta^r_{M+\frac{1}{2}} - \zeta^r_{M-\frac{1}{2}} - \zeta^r_{M-\frac{1}{2}} \right) + \frac{1}{2} \sum_{m=1}^{m=M} U^r_m (U_{m+1}^r - U_{m-1}^r)
\]

(2.8.10)

The last term contains contributions from the middle of the field, and consequently, use of the computational formula given by Eq. (2.8.9) does not conserve momentum.
3. WAVE MOTION IN TWO SPATIAL DIMENSIONS

3.0 INTRODUCTION

This section describes a computational model for wave propagation in two dimensions. The discussion includes (1) the systems of difference equations and the methods used for solving these systems of equations, (2) different aspects of the approximation of the partial-differential equations and the boundaries by finite differences, (3) the stability of the computational method, and (4) the conservation properties of the scheme.

3.1 THE SYSTEM OF THE FINITE-DIFFERENCE EQUATIONS IN TWO DIMENSIONS

The following notation is used in the approximation of the differential equations by the system of difference equations:

\[ u_{j,k}^{(n)} = U(j\Delta x, k\Delta y, n\Delta t); \quad \Delta x = \Delta y = \Delta s \quad (3.1.1) \]

where

\[(x,y) = (j\Delta x, k\Delta y) \text{ and is a spatial grid point}\]

\[ j = 0, \pm \frac{1}{2}, \pm 1, \pm \frac{3}{2}, \cdots \]

\[ k = 0, \pm \frac{1}{2}, \pm 1, \pm \frac{3}{2}, \cdots \]

\[ n = 0, \frac{1}{2}, 1, \frac{3}{2}, 2 \]

Furthermore, the following notation of averages and differences, shown here for \( \zeta \), is introduced:

\[ \zeta_{x,j,k} = \frac{1}{2} (\zeta_{j+\frac{1}{2},k} + \zeta_{j-\frac{1}{2},k}) \]

\[ \zeta_{y,j,k} = \frac{1}{2} (\zeta_{j,k+\frac{1}{2}} + \zeta_{j,k-\frac{1}{2}}) \]

\[ \zeta_{x} = \zeta_{j+\frac{1}{2},k} - \zeta_{j-\frac{1}{2},k} \]

\[ \zeta_{y} = \zeta_{j,k+\frac{1}{2}} - \zeta_{j,k-\frac{1}{2}} \]

\[ \zeta_{j,k} = \frac{1}{4} (\zeta_{j-\frac{1}{2},k-\frac{1}{2}} + \zeta_{j-\frac{1}{2},k+\frac{1}{2}} + \zeta_{j+\frac{1}{2},k-\frac{1}{2}} + \zeta_{j+\frac{1}{2},k+\frac{1}{2}}) \]
A space-staggered scheme is used where velocities, water levels, and depth are described at different grid points (see Fig. 3.1.1). This scheme, first used by Platzman, has the advantage that in the formula for the variable operated upon in time, there is a centrally located spatial derivative for the linear term. The water level $\zeta$ is described at integer values of $j$ and $k$, the velocity $u$ is described

![Diagram](image)

Fig.3.1.1—Space-staggered scheme

at integer-and-one-half values of $j$ and integer values of $k$, and the velocity $v$ is described at integer values of $j$ and integer-and-one-half values of $k$, while the water depth $h$ is described at integer-and-one-half values of $j$ and $k$.

A double time-step operation is used in such a manner that the terms containing space derivatives and the Coriolis force are generally taken as alternating forward and backward. Thus, the time interval is considered over two successive operations, and these terms are either central in time or averaged in time over that interval. This feature is discussed in more detail in Section 3.2, and it will
then be seen that the systems set up in this manner are not easily solvable because of nonlinear terms. Some changes will be necessary to handle these terms on a lower time level, sacrificing a higher approximation for computational speed.

The individual operations each have two time levels. The first operation is taken from time \( n \) to time \( n + \frac{1}{2} \), and the second operation is taken from time \( n + \frac{1}{2} \) to time \( n + 1 \). Values of the fields of \( \zeta^{(n+\frac{1}{2})} \), \( u^{(n+\frac{1}{2})} \), and \( v^{(n+\frac{1}{2})} \) are obtained from the fields of \( \zeta^{(n)} \), \( u^{(n)} \), and \( v^{(n)} \) by an operation which is implicit in \( \zeta \) and \( u \) and explicit in \( v \). Then the fields of \( \zeta^{(n+1)} \), \( u^{(n+1)} \), and \( v^{(n+1)} \) are computed from the fields of \( \zeta^{(n+\frac{1}{2})} \), \( u^{(n+\frac{1}{2})} \), and \( v^{(n+\frac{1}{2})} \) by an operation which is implicit in \( \zeta \) and \( v \) and explicit in \( u \). The two sets of difference equations of this multiproation method are now written with the equation of continuity as the second equation of each set (Eq. (3.1.3)), using an integer value for \( j \) and \( k \) and maintaining the velocity gradients in the convective-inertia terms in differential form within angle brackets \( (\cdot) \).

The effects of bottom roughness are indicated by a function \( R \), and the influence of wind and atmospheric pressure is indicated by a function \( F \). Thus, the individual operations are written

\[
\begin{align*}
 u^{(n+\frac{1}{2})} &= u^{(n)} + \frac{1}{2} \Delta t \frac{\partial u^v(n)}{\partial y} - \frac{1}{2} \Delta t \frac{\partial v^u(n)}{\partial x} - \frac{1}{2} \Delta t \frac{\partial v^v(n)}{\partial y} \\
 &\quad - \frac{1}{2} \frac{\Delta t}{\Delta s} \left[ \frac{\partial u}{\partial s} \right]^{(n+\frac{1}{2})}_x - (R)(n) - (F)(x)^{(n+\frac{1}{2})} & \text{at } j + \frac{1}{2}, \ k
\end{align*}
\]

(3.1.2)

\[
\begin{align*}
 \zeta^{(n+\frac{1}{2})} &= \zeta^{(n)} - \frac{1}{2} \frac{\Delta t}{\Delta s} \left[ (h^v + \zeta^v) u^v(x) \right]^{(n+\frac{1}{2})}_x - \frac{1}{2} \frac{\Delta t}{\Delta s} \left[ (h^x + \zeta^x) v^v(n) \right]_y \\
 &\quad \text{at } j, k
\end{align*}
\]

(3.1.3)
\[
\begin{align*}
v(n+\frac{1}{2}) &= v(n) - \frac{1}{2} \Delta t \, f_u(n+\frac{1}{2}) - \frac{1}{2} \Delta t \, \frac{\partial u}{\partial x}(n+\frac{1}{2}) - \frac{1}{2} \Delta t \, v(n+\frac{1}{2}) \frac{\partial v}{\partial y}(n) \\
&\quad - \frac{1}{2} \frac{\Delta t}{\Delta s} \, g_{xy}^{(n)} - R(y)^*(n+\frac{1}{2}) - F(y)(n) \quad \text{at } j, k + \frac{1}{2} \\
(3.1.4) \\
\end{align*}
\]

\[
\begin{align*}
u(n+1) &= u(n+\frac{1}{2}) + \frac{1}{2} \Delta t \, f_v(n+1) - \frac{1}{2} \Delta t \, u(n+1) \frac{\partial u}{\partial x}(n+\frac{1}{2}) \\
&\quad - \frac{1}{2} \Delta t \, v(n+1) \frac{\partial u}{\partial y}(n+\frac{1}{2}) - \frac{1}{2} \frac{\Delta t}{\Delta s} \, g_{xy}^{(n+\frac{1}{2})} - R(x)^*(n+1) - F(x)(n+\frac{1}{2}) \\
&\quad \text{at } j + \frac{1}{2}, k \\
(3.1.5) \\
\end{align*}
\]

\[
\begin{align*}
\zeta(n+1) &= \zeta(n+\frac{1}{2}) - \frac{1}{2} \frac{\Delta t}{\Delta s} \left[ \frac{\partial h}{\partial y} + \frac{\partial e}{\partial x} \right] u_x(n+\frac{1}{2}) - \frac{1}{2} \frac{\Delta t}{\Delta s} \left[ \frac{\partial h}{\partial x} + \frac{\partial e}{\partial y} \right] v_y(n+1) \\
&\quad \text{at } j, k \\
(3.1.6) \\
\end{align*}
\]

\[
\begin{align*}
v(n+1) &= v(n+\frac{1}{2}) - \frac{1}{2} \Delta t \, f_u(n+\frac{1}{2}) - \frac{1}{2} \Delta t \, u(n+\frac{1}{2}) \frac{\partial u}{\partial x}(n+\frac{1}{2}) \\
&\quad - \frac{1}{2} \Delta t \, v(n+1) \frac{\partial v}{\partial y}(n+\frac{1}{2}) - \frac{1}{2} \frac{\Delta t}{\Delta s} \, g_{xy}^{(n+1)} - R(y)(n+\frac{1}{2}) - F(y)(n+1) \\
&\quad \text{at } j, k + \frac{1}{2} \\
(3.1.7) \\
\end{align*}
\]

A special, additional technique will be employed for the computation of the nonlinear terms marked with an asterisk. To develop a computational system, the forcing function F and the bottom effects are omitted, together with all convective-inertia terms in their partial-differential form. The convective-inertia terms and the resistance terms are developed in Section 3.2.
In each of Eqs. (3.1.2) and (3.1.3) there are three values at the time level \((n + \frac{1}{2})\), which are all situated at the line \(k\) and have to be computed. Writing the continuity equation first and omitting the subscript \(k\), we can write Eqs. (3.1.2) and (3.1.3) in the following form:

\[
- \frac{1}{2} \frac{\Delta t}{\Delta s} \left[ (h'^y + \zeta'^y)^* u_{j-\frac{1}{2}}^{(n+\frac{1}{2})} + \zeta_{j}^{(n+\frac{1}{2})} + \frac{1}{2} \frac{\Delta t}{\Delta s} \left[ (h'^y + \zeta'^y)^* u_{j+\frac{1}{2}}^{(n+\frac{1}{2})} \right] \right] = A_j^{(n)} \quad (3.1.8)
\]

\[
- \frac{1}{2} \frac{\Delta t}{\Delta s} g \zeta_{j}^{(n+\frac{1}{2})} + u_{j-\frac{1}{2}}^{(n+\frac{1}{2})} + \frac{1}{2} \frac{\Delta t}{\Delta s} g \zeta_{j+1}^{(n+\frac{1}{2})} = B_j^{(n)+\frac{1}{2}} \quad (3.1.9)
\]

where

\[
A_j^{(n)} = \zeta_{j}^{(n)} - \frac{1}{2} \frac{\Delta t}{\Delta s} \left[ (h'^y + \zeta'^y)^* \right] \quad \text{at } j \quad (3.1.10)
\]

\[
B_j^{(n)+\frac{1}{2}} = u_{j-\frac{1}{2}}^{(n)} + \frac{1}{2} \Delta t f v^{(n)} \quad \text{at } j + \frac{1}{2} \quad (3.1.11)
\]

Thus, one equation with three unknowns exists for each velocity field point \(u_{j+\frac{1}{2}}^{(n)}\) and each water-level field point \(\zeta_j^{(n)}\) on a line \(k\). If a row of \(N\) water-level points is on the line and velocities are given at the boundaries outside of the water levels concerned, \(N\) water levels and \(N - 1\) velocities at time \((n + \frac{1}{2})\) must be solved from \(2N - 1\) equations.

We now introduce

\[
r_{j+\frac{1}{2}} = \frac{1}{2} \frac{\Delta t}{\Delta s} (h'^y + \zeta'^y)^* ; r_{j+\frac{1}{2}} = \frac{1}{2} \frac{\Delta t}{\Delta s} (h'^y + \zeta'^y)^* ; \ldots
\]

\[
r_{j} = \frac{1}{2} \frac{\Delta t}{\Delta s} g ; r_{j+1} = \frac{1}{2} \Delta t g ; \ldots \quad (3.1.12)
\]

\[
r_j = \frac{1}{2} \frac{\Delta t}{\Delta s} g ; r_{j+1} = \frac{1}{2} \Delta t g ; \ldots \quad (3.1.13)
\]
Equations (3.1.8) and (3.1.9) can be written in matrix form for a line \( k \), assuming that \( u_{J+\frac{1}{2}}^{(n+\frac{1}{2})} \) is a given velocity at the lower boundary and \( u_{I+\frac{1}{2}}^{(n+\frac{1}{2})} \) is a given velocity at the upper boundary:

\[
\begin{bmatrix}
1 & r_{J+\frac{1}{2}} & 0 & 0 & \cdots & 0 \\
-r_J & 1 & r_{J+1} & 0 & \cdots & 0 \\
0 & -r_{J+\frac{1}{2}} & 1 & r_{J+\frac{3}{2}} & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & -r_{I-\frac{1}{2}} & 1 \\
0 & 0 & 0 & \cdots & -r_{I+\frac{1}{2}} & 1
\end{bmatrix}
\begin{bmatrix}
\zeta_J^{(n+\frac{1}{2})} \\
\zeta_J^{(n+\frac{1}{2})} \\
\zeta_{J+1}^{(n+\frac{1}{2})} \\
\cdots \\
\zeta_I^{(n+\frac{1}{2})} \\
\zeta_I^{(n+\frac{1}{2})}
\end{bmatrix}
= 
\begin{bmatrix}
A_J^{(n)} \\
A_J^{(n)} \\
B_{J+\frac{1}{2}}^{(n)} \\
\vdots \\
A_I^{(n)} \\
A_I^{(n)}
\end{bmatrix}
+ 
\begin{bmatrix}
r_{J-\frac{1}{2}}u_{J-\frac{1}{2}}^{(n+\frac{1}{2})} \\
r_{J-\frac{1}{2}}u_{J-\frac{1}{2}}^{(n+\frac{1}{2})} \\
0 \\
0 \\
r_{I+\frac{1}{2}}u_{I+\frac{1}{2}}^{(n+\frac{1}{2})} \\
r_{I+\frac{1}{2}}u_{I+\frac{1}{2}}^{(n+\frac{1}{2})}
\end{bmatrix}
\]

The values of the vector \( (\zeta_J^{(n+\frac{1}{2})}, u_{J+\frac{1}{2}}^{(n+\frac{1}{2})}, \zeta_{J+1}^{(n+\frac{1}{2})}, \ldots, \zeta_I^{(n+\frac{1}{2})}) \) at the \( n + \frac{1}{2} \) time level can be solved with a limited number of operations by a process of elimination of unknowns. Starting with the first equation, the water level \( \zeta_J^{(n+\frac{1}{2})} \) is expressed as a function of the unknown velocity \( u_{J+\frac{1}{2}}^{(n+\frac{1}{2})} \):

\[
\zeta_J^{(n+\frac{1}{2})} = -P_Ju_{J+\frac{1}{2}}^{(n+\frac{1}{2})} + Q_J \quad (3.1.14)
\]

where

\[
P_J = r_{J+\frac{1}{2}} \quad (3.1.15)
\]

\[
Q_J = A_J^{(n)} + r_{J-\frac{1}{2}}u_{J-\frac{1}{2}}^{(n+\frac{1}{2})} \quad (3.1.16)
\]
Substitution of Eq. (3.1.14) into the second equation gives

\[
-r_j \left(-p_j u_j^{(n+\frac{1}{2})} + q_j\right) + u_j^{(n+\frac{1}{2})} + r_{j+1} c_{j+1}^{(n+\frac{1}{2})} = B_{j+\frac{1}{2}}^{(n)} \tag{3.1.17}
\]

or expressing \( u_j^{(n+\frac{1}{2})} \) as a function of \( c_{j+1}^{(n+\frac{1}{2})} \) gives

\[
u_j^{(n+\frac{1}{2})} = -R_j c_{j+1}^{(n+\frac{1}{2})} + S_j \tag{3.1.18}\]

where

\[
R_j = \frac{r_{j+1}}{1 + r_j p_j} \tag{3.1.19}
\]

\[
S_j = \frac{B_{j+\frac{1}{2}}^{(n)} + r_j q_j}{1 + r_j p_j} \tag{3.1.20}\]

Again, the water level can be expressed as a function of the next velocity:

\[
c_{j+1} = -p_{j+1} u_{j+\frac{1}{2}}^{n+\frac{1}{2}} + q_{j+1} \tag{3.1.21}\]

where

\[
p_{j+1} = \frac{r_{j+\frac{3}{2}}}{1 + r_{j+\frac{1}{2}} R_j} \tag{3.1.22}\]

\[
q_{j+1} = \frac{A_{j+1}^{(n)} + r_{j+\frac{3}{2}} S_j}{1 + r_{j+\frac{1}{2}} R_j} \tag{3.1.23}\]
Generally, the following recursion formulas can be written

\[
\zeta_j^{(n+\frac{1}{2})} = -P_j u_j^{(n+\frac{1}{2})} + Q_j \quad (3.1.24)
\]

\[
u_j^{(n+\frac{1}{2})} = -R_j S_j^{(n+\frac{1}{2})} + S_j - 1 \quad (3.1.25)
\]

where

\[
P_j = r_{j+\frac{1}{2}}/(1 + r_{j-\frac{1}{2}}R_{j-1}) \quad (3.1.26)
\]

\[
Q_j = (A_j^{(n)} + r_{j-\frac{1}{2}}S_{j-1})/(1 + r_{j-\frac{1}{2}}R_{j-1}) \quad (3.1.27)
\]

\[
R_j = r_{j+1}/(1 + r_j P_j) \quad (3.1.28)
\]

\[
S_j = (B_j^{(n)} + r_j Q_j)/(1 + r_j P_j) \quad (3.1.29)
\]

The recursion factors \( P, Q, R, \) and \( S \) can be computed in succession until the other bound is reached. If \( u_j^{(n+\frac{1}{2})} \) is a given velocity, the last two factors computed are \( P \) and \( Q \). Since Eq. (3.1.24) expresses \( \zeta_j \) as a known function of the velocity \( u_j^{(n+\frac{1}{2})} \), \( \zeta_j \) can be computed, and all water levels and velocities can be found in descending order by use of Eqs. (3.1.24) and (3.1.25).

The velocity in the other direction at the time level \( n + \frac{1}{2} \) can be found explicitly from Eq. (3.1.4), since the velocity \( u_j^{(n+\frac{1}{2})} \) in the Coriolis term is already known.

For the second operation, going from the time level \( n + \frac{1}{2} \) to \( n + 1 \), Eqs. (3.1.6) and (3.1.7) are solved implicitly in the same manner as described for Eqs. (3.1.2) and (3.1.3). Finally, the velocity in the \( x \)-direction can be computed explicitly from Eq. (3.1.5). It can be seen that no more than two successive fields have to be stored in the computer memory, as only the latest information available is necessary to compute the next step.
The nonlinear terms in the continuity equation can be computed on their proper time level by an iteration procedure. The first estimate of \( \zeta^{(n+\frac{1}{2})} \) is made by the implicit procedure of taking the nonlinear term at the time level \( n \). Next, the value \( \zeta^{(n+\frac{1}{2})} \) thus computed is used in Eq. (3.1.3) for the actual computation. This iteration can be repeated several times.

### 3.2 Convective-Inertia Terms and Bottom-Effect Terms

The space locations of the variables \( u, v, \) and \( \zeta \) were specified in the beginning of Section 3.1. As the operations are dependent on these locations, derivatives of the convective-inertia terms are to be expressed using available locations. Consequently, the spatial derivatives of the convective-inertia terms are expressed as

\[
\begin{align*}
\frac{\partial u}{\partial x}_{j+\frac{1}{2},k} &= \frac{1}{2\Delta s} \left( u_{j+\frac{3}{2},k} - u_{j-\frac{1}{2},k} \right) \\
\frac{\partial u}{\partial y}_{j+\frac{1}{2},k} &= \frac{1}{2\Delta s} \left( u_{j+\frac{1}{2},k+1} - u_{j+\frac{1}{2},k-1} \right) \\
\frac{\partial v}{\partial x}_{j,k+\frac{1}{2}} &= \frac{1}{2\Delta s} \left( v_{j+1,k+\frac{3}{2}} - v_{j-1,k+\frac{3}{2}} \right) \\
\frac{\partial v}{\partial y}_{j,k+\frac{1}{2}} &= \frac{1}{2\Delta s} \left( v_{j,k+\frac{3}{2}} - v_{j,k-\frac{1}{2}} \right)
\end{align*}
\]

As a result of computational considerations, the convective-inertia terms are not taken completely central in time, which would be preferable for accuracy. If the spatial derivative is taken at the higher time level, the matrices contain more than three diagonals, and considerable computational effort is required for solution. Table 3.2.1 indicates the preferred time levels and the time levels used. The spatial derivatives are always chosen in the lower time level, while the velocity is in the higher time level if possible.

A better representation of these terms seems possible if in conjunction with the use of an alternate network by which the water levels
Table 3.2.1

EXPRESSIONS FOR THE CONVECTIVE-INERTIA TERMS

<table>
<thead>
<tr>
<th>Eq.</th>
<th>Preferred</th>
<th>Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1.2</td>
<td>( u \left(n+\frac{1}{2}\right) \frac{\partial u}{\partial x} \left(n+\frac{1}{2}\right) + \frac{\tau}{\nu} \left(n\right) \frac{\partial u}{\partial y} (n) )</td>
<td>( u \left(n+\frac{1}{2}\right) \frac{\partial u}{\partial x} (n) + \frac{\tau}{\nu} (n) \frac{\partial u}{\partial y} (n) )</td>
</tr>
<tr>
<td>3.1.4</td>
<td>( u \left(n+\frac{1}{2}\right) \frac{\partial v}{\partial x} \left(n+\frac{1}{2}\right) + \frac{\tau}{\nu} (n) \frac{\partial v}{\partial y} (n) )</td>
<td>( u \left(n+\frac{1}{2}\right) \frac{\partial v}{\partial x} (n) + \frac{\tau}{\nu} (n+\frac{1}{2}) \frac{\partial v}{\partial y} (n) )</td>
</tr>
<tr>
<td>3.1.5</td>
<td>( u \left(n+\frac{1}{2}\right) \frac{\partial u}{\partial x} \left(n+\frac{1}{2}\right) + \frac{\tau}{\nu} (n+1) \frac{\partial u}{\partial y} (n+1) )</td>
<td>( u \left(n+\frac{1}{2}\right) \frac{\partial u}{\partial x} (n+\frac{1}{2}) + \frac{\tau}{\nu} (n+1) \frac{\partial u}{\partial y} (n+\frac{1}{2}) )</td>
</tr>
<tr>
<td>3.1.7</td>
<td>( u \left(n+\frac{1}{2}\right) \frac{\partial v}{\partial x} \left(n+\frac{1}{2}\right) + \frac{\tau}{\nu} (n+1) \frac{\partial v}{\partial y} (n+1) )</td>
<td>( u \left(n+\frac{1}{2}\right) \frac{\partial v}{\partial x} (n+\frac{1}{2}) + \frac{\tau}{\nu} (n+1) \frac{\partial v}{\partial y} (n+\frac{1}{2}) )</td>
</tr>
</tbody>
</table>

are computed at the locations of the depth of the basic network. The velocity gradients can be taken over a distance \( \Delta s \) rather than over the distance \( 2\Delta s \) as shown in Eqs. (3.2.1) and (3.2.4). The use of such a lattice would also permit the expression of the convective terms in a form where nonlinear feedback in the very short wavelengths is eliminated and at the same time the conservation laws are satisfied.\(^{11,16}\)

Of course, the computation time and computer memory requirements would double, and also troublesome oscillations might occur between the two networks.

As the magnitude of the convective-inertia terms is small compared to the inertia terms in the computations of long waves in coastal waters, a lower order of approximation is justified in favor of a much shorter computation time.

In the four equations of motion (Eqs. (3.1.2) through (3.1.6)), the effects of bottom roughness are approximated by quadratic terms, using Chezy coefficients with values assigned at the water-level locations. The four terms are taken as

\[
R(x)(n) = \frac{1}{2} \Delta t \:\text{gu}(n) \sqrt{\left(\frac{u(n)}{h^2} \frac{\partial u}{\partial x} \right)^2 + \left(\frac{v(n)}{h^2} \frac{\partial v}{\partial x} \right)^2} \quad \text{at} \ j + \frac{1}{2}, \ k \quad (3.2.5)
\]
\[ R_y^{*(n+\frac{1}{2})} = \frac{1}{2} \Delta t \, g_y \, (n+\frac{1}{2}) \sqrt{\frac{(u^{(n+\frac{1}{2})})^2 + (v^{(n)})^2}{(h + v^{(n+\frac{1}{2})}) (c_y^2)}} \quad \text{at } j, k + \frac{1}{2} \]  

(3.2.6)

\[ R_x^{*(n+1)} = \frac{1}{2} \Delta t \, g_x \, (n+1) \sqrt{\frac{(u^{(n+1)})^2 + (v^{(n+1)})^2}{(h^2 + w^{(n+1)}) (c_x^2)}} \quad \text{at } j + \frac{1}{2}, k \]  

(3.2.7)

\[ R_y^{(n+\frac{1}{2})} = \frac{1}{2} \Delta t \, g_y \, (n+\frac{1}{2}) \sqrt{\frac{(u^{(n+\frac{1}{2})})^2 + (v^{(n+\frac{1}{2})})^2}{(h^2 + w^{(n+\frac{1}{2})}) (c_y^2)}} \quad \text{at } j, k + \frac{1}{2} \]  

(3.2.8)

The convective-inertia terms and the expression developed here for the turbulence and the viscosity effects at the sea bottom can be included in the coefficients of the recursion formulas (Eqs. (3.1.24) and (3.1.25)).

For the implicit computation in the x-direction, the following coefficients are used:

\[ P_j = r_{j+\frac{1}{2}} / (1 + r_{j+\frac{1}{2}} R_{j-1}) \]  

(3.2.9)

\[ Q_j = (A_j^{(n)} + r_{j+\frac{1}{2}} S_{j-1}) / (1 + r_{j+\frac{1}{2}} R_{j-1}) \]  

(3.2.10)

\[ R_j = r_{j+1} / [1 + r_j P_j + \frac{1}{4} \Delta t \Delta s (u_j^{(n)} - u_j^{(n)})] \]  

(3.2.11)

\[ S_j = (A_j^{(n)} + r_j Q_j) / [1 + r_j P_j + \frac{1}{4} \Delta t \Delta s (u_j^{(n)} - u_j^{(n)})] \]  

(3.2.12)
where

\[
A_j^{(n)} = c_j^{(n)} - \frac{1}{2} \frac{\Delta t}{\Delta s} [ \left( \frac{h^x}{c} + \frac{v}{c} \right) v_y^n ] \quad \text{at } j, k \tag{3.2.13}
\]

\[
B_{j+\frac{1}{2}}^{(n)} = \frac{1}{2} \left[ \Delta t \left( f - \frac{\Delta s}{\Delta t} \right) ight] (u_j^{(n)}_{j+1, k+1} - u_j^{(n)}_{j+1, k-1}) y_j^{(n)}
- \frac{1}{2} \Delta t g u_j^{(n)} \frac{\sqrt{(u_j^{(n)} c_j^{(n)} y_j^{(n)})^2 + (v_j^{(n)} c_j^{(n)})^2}}{(h_j^{(n)} + c_j^{(n)}) (c_j^{(n)})^2} \quad \text{at } j+\frac{1}{2}, k \tag{3.2.14}
\]

As in the convective-inertia terms, information on the higher time level is used as soon as it becomes available in the computation. Several choices for expressing these terms do exist, and any term used will affect the stability of the computation.

The explicit operation on the velocity \( v \) becomes

\[
v_{j+\frac{1}{2}}^{(n+\frac{1}{2})} = \left\{ \frac{v_j^{(n+\frac{1}{2})} - \frac{1}{2} \left[ \Delta t f + \frac{1}{2} \frac{\Delta t}{\Delta s} \left( v_j^{(n)}_{j+1, k+1} - v_j^{(n)}_{j-1, k+1} \right) \right] g_j^{(n+\frac{1}{2})} - \frac{1}{2} \frac{\Delta t}{\Delta s} g_j^{(n)} } {1 + \frac{1}{2} \Delta t g \sqrt{(u_j^{(n+\frac{1}{2})})^2 + (v_j^{(n+\frac{1}{2})})^2} + \frac{1}{4} \frac{\Delta t}{\Delta s} (v_j^{(n)}_{j+1, k+1} - v_j^{(n)}_{j, k+1})} \right. \right. \]

\[
(3.2.15)
\]

The coefficients for the recursion formulas in the \( y \)-direction are

\[
P_k = r_{k+\frac{1}{2}} / (1 + r_{k-\frac{1}{2}} r_{k-1}) \tag{3.2.16}
\]

\[
Q_k = (A_k^{(n+\frac{1}{2})} + r_{k-\frac{1}{2}} S_{k-1}) / (1 + r_{k-\frac{1}{2}} R_{k-1}) \tag{3.2.17}
\]

\[
R_k = r_{k+1} / [1 + r_{k-\frac{1}{2}} P_k + \frac{1}{4} \frac{\Delta t}{\Delta s} (v_{k+\frac{1}{2}}^{(n+\frac{1}{2})} - v_{k-\frac{1}{2}}^{(n+\frac{1}{2})})] \tag{3.2.18}
\]

\[
S_k = (B_{k+\frac{1}{2}}^{(n+\frac{1}{2})} + r_{k} Q_k) / [1 + r_{k-\frac{1}{2}} P_k + \frac{1}{4} \frac{\Delta t}{\Delta s} (v_{k+\frac{1}{2}}^{(n+\frac{1}{2})} - v_{k-\frac{1}{2}}^{(n+\frac{1}{2})})] \tag{3.2.19}
\]
where

\[ A_k^{(n+\frac{1}{2})} = c^{(n+\frac{1}{2})} - \frac{1}{2} \frac{\Delta t}{\Delta s} \left[ (h^y + \frac{v}{c})_u^{(n+\frac{1}{2})} \right] \text{ at } j, k \]  

\[ B_k^{(n+\frac{1}{2})} = v^{(n+\frac{1}{2})} - \frac{1}{2} \left[ \Delta t f + \frac{1}{2} \frac{\Delta t}{\Delta s} (v_j^{(n+\frac{1}{2})} - v_{j+1,k}^{(n+\frac{1}{2})} - v_{j-1,k}^{(n+\frac{1}{2})}) \right] \text{ at } j, k + \frac{1}{2} \]

\[ \frac{1}{2} \Delta t g v^{(n+\frac{1}{2})} \sqrt{(h^y + \frac{v}{c})^2 + (v^{(n+\frac{1}{2})})^2} \]  

\[ \frac{1}{2} \Delta t g v^{(n+\frac{1}{2})} \sqrt{(h^y + \frac{v}{c})^2 + (v^{(n+\frac{1}{2})})^2} \]  

\[ \text{at } j, k + \frac{1}{2} \]  

The recursion formulas in the y-direction are

\[ \zeta_k^{(n+1)} = -R_k \zeta_{k-\frac{1}{2}}^{(n+1)} + Q_k \]  

\[ \zeta_{k-\frac{1}{2}}^{(n+1)} = -R_{k-1} \zeta_k^{(n+1)} + S_{k-1} \]  

The explicit operation for the u velocity becomes

\[ u_j^{(n+1)} = \frac{u_j^{(n+\frac{1}{2})} + \frac{1}{2} \left[ \Delta t f - \frac{1}{2} \frac{\Delta t}{\Delta s} \left( u_{j+1,k}^{(n+\frac{1}{2})} - u_{j-1,k}^{(n+\frac{1}{2})} \right) \right] \zeta_j^{(n+1)} - \frac{1}{2} \frac{\Delta t}{\Delta s} \zeta_j^{(n+\frac{1}{2})}} \left[ 1 + \frac{1}{2} \Delta t g \sqrt{(u_j^{(n+\frac{1}{2})})^2 + (v_j^{(n+1)})^2} + \frac{1}{4} \frac{\Delta t}{\Delta s} \left( u_{j+\frac{1}{2},k}^{(n+\frac{1}{2})} - u_{j-\frac{1}{2},k}^{(n+\frac{1}{2})} \right) \right] \]

The formulas developed above cannot be used directly in this form for electronic computation. Coding in FORTRAN presents problems, as coordinate description of the variables can be made only on integers, and integer-and-one-half values do appear in the formulas developed. Multiplication of all coordinate descriptions of the points by a factor
of two would eliminate this problem; however, such a lattice system is uneconomical from the point of view of memory use of the computer because not all locations of an array of \( u, v, \zeta, \) and \( h \) would be used.

It is possible to give each of the variables \( u, v, \zeta, \) and \( h \) a separate coordinate system, which would result in a good use of computer memory and would also allow coordinate description on integers. The computational formulas for these special coordinate systems are presented in Appendix A.

### 3.3 APPROXIMATION PROBLEMS

The formulas developed in Section 3.1 pertain to points inside the field of computation. Along the boundaries, however, the matter becomes complicated when trying to remain within the computational pattern developed. In this discussion of boundaries, only a coastal boundary, which is called a closed bound, and an open boundary in the sea or ocean are considered. A flooding coast, which is considered a moving bound, and the discharge at a certain boundary are not considered.

In the approximation of a closed bound, it is assumed that the velocity perpendicular to the bound is zero, that a finite depth is always available, and that the negative wave amplitude does not exceed this depth. As in the approximation of water depth, the approximation of a coast follows the grid. The coastline is assumed to pass through the locations at which the water depth \( (h) \) is described. Hence, the zero velocities perpendicular to the coast are exactly in the grid function of the velocities \( u \) and \( v \).

At the open bounds the water levels are given as time functions in the boundary going through the water-level points \( (\zeta) \).

Figure 3.3.1 presents the four cases which can be encountered in an implicit computation of water levels and velocities on a row \( k \). Besides these water levels and velocities, which are considered in the implicit step of the computation, all of the other velocity and depth points which appear in the formulas of the computation are also shown.

Cases A and C present two open bounds, and Cases B and D present
Fig. 3.3.1 — Possible boundary conditions

- Water level at open boundary (function of time)
- Water level
- Depth

U velocity
V velocity
closed bounds. Near all of these bounds, computation of the velocities by the derived formulas are impossible, as one or more of the terms for the expression of convective inertia are outside the field of computation. For example, in the computation of the difference equation of motion for the velocity \( U_{j+\frac{1}{2},k} \), the velocity \( U_{j+\frac{1}{2},k+1} \) which appears in the finite-difference approximation of the convective term \( v(\partial u/\partial y) \) is outside the field.

This problem is solved by using a linear approximation of the differential equation in the field concerned. Thus, if the spatial derivatives in the convective-inertia terms cannot be computed, the linear approximation results in omitting these terms. The use of noncentral spatial derivatives on these bounds (extrapolation from the interior field) is not feasible for reasons of stability, as has been discussed in Section 2.7. The convective-inertia terms are of a second order of magnitude in the computations, and their effects on the computational results are shown in the discussion of numerical experiments.

Another approximation problem is associated with the space allocation of the values for the expression of the bottom roughness (C coefficient). Other than being a function of bottom roughness, this coefficient is a function of the total depth and is consequently dependent on the water level. Unfortunately, information about C values is very limited. For flow in one direction, the C value in a cross section with a certain width \( l \) may be considered as the integrated C value, averaged over the width \( l \). The value thus obtained will differ from the C value computed from the average depth. In the two-dimensional computations, the C value used for the computation of the velocity \( U_{j+\frac{1}{2},k} \) (see Fig. 3.3.2) should be expressed as

\[
C_{j+\frac{1}{2},k} = \frac{1}{l^2} \int_{j\ell}^{(j+1)\ell} \int_{(k-\frac{1}{2})\ell}^{(k+\frac{1}{2})\ell} C \, dx \, dy \tag{3.3.1}
\]

where C is, among others, a function of the depth (h) (to the reference plane) and the water level (\( \zeta \)). The latter two terms are functions of location.
Many choices for difference approximations do exist. For example, using a certain water level and depth function for \( C \), it is possible to compute the value when required in our computation. In this method no storage of \( C \) values in the computer memory is required.

In the second approach, the coefficient can be computed only at certain time intervals because the wave amplitude varies slowly with time and the coefficient is only slightly influenced by the water level, particularly in deeper water. The \( C \) values must be stored in the computer memory. Ideally, one \( C \) value should be associated with each velocity point. Thus, if the computational field has \( N \) points, twice this number should be required for storage. Or, with somewhat lesser accuracy, if the coefficients are determined either on the location of the water-level points (\( \zeta \)) or on the location of the depth (\( h \)), only \( N \) values should be required (see Figs. 3.3.3 and 3.3.4). The actual value associated with the velocity components can then be found from
Fig. 3.3.3 — C values at the location of $\zeta$

Fig. 3.3.4 — C values at the location of $h$
the average of the two adjacent C values. In the computational formulas described in this section, the coefficients were determined on the location of the water levels, as shown in Fig. 3.3.3.

3.4 STABILITY

First, the stability of a simplified system of the difference equations (Eqs. (3.1.2) through (3.1.7)) is investigated. In these equations the influence of the convective-inertia terms, of the forcing function, and of the effect of bottom roughness is neglected. In the investigation a linear term is used in the equation of continuity (in accordance with Section 2) for the wave motion in one spatial dimension.

The numerical procedure is the multioperation procedure discussed in Section 3.1. In this procedure the velocity (u) in the x-direction is taken forward in the first operation and backward in the second operation, while velocity (v) in the y-direction is taken backward in the first operation and forward in the second.

In the analyses of the stability, the decay of waves over the combined steps is investigated from considerations of boundedness of the amplification matrix from time level n to time level n + 1. This may be done by use of general solutions of the type

$$\zeta(x,y,t) = \exp [i(\sigma_1 x + \sigma_2 y + \beta t)]$$

(3.4.1)

According to the analyses in Section 2 (Eq. (2.2.14)), the first operation, represented by Eqs. (3.1.2) through (3.1.4), can be written

$$[A] \tilde{U}^{n+\frac{1}{2}} = [B] \tilde{U}^n$$

(3.4.2)

where

$$\tilde{U}^n = \begin{bmatrix} u^n \\ v^n \\ \zeta^n \end{bmatrix}$$

(3.4.3)
The matrices are

\[
[A] = \begin{bmatrix}
1 & 0 & \text{i}g_\alpha \\
\text{c} & 1 & 0 \\
\text{i}h_\alpha & 0 & 1
\end{bmatrix}
\]  
(3.4.4)

\[
[B] = \begin{bmatrix}
1 & \text{c} & 0 \\
0 & 1 & \text{-i}g_\gamma \\
0 & \text{-i}h_\gamma & 1
\end{bmatrix}
\]  
(3.4.5)

where

\[
c = \frac{1}{2} \Delta t \, f \, \cos(\sigma_1 \, t) \, \cos(\sigma_2 \, t)
\]  
(3.4.6)

\[
\alpha = (\Delta t / \Delta s) \, \sin(\sigma_1 \, t)
\]  
(3.4.7)

\[
\gamma = (\Delta t / \Delta s) \, \sin(\sigma_2 \, t)
\]  
(3.4.8)

\[
\lambda = \frac{1}{2} \Delta s
\]  
(3.4.9)

The second operation, represented by Eqs. (3.1.5) through (3.1.7), can be written

\[
[C] \vec{u}^{n+1} = [D] \vec{u}^{n+\frac{1}{2}}
\]  
(3.4.10)

\[
[C] = \begin{bmatrix}
1 & -\text{c} & 0 \\
0 & 1 & \text{i}g_\gamma \\
0 & \text{i}h_\gamma & 1
\end{bmatrix}
\]  
(3.4.11)

\[
[D] = \begin{bmatrix}
1 & 0 & -\text{i}g_\alpha \\
-\text{c} & 1 & 0 \\
-\text{i}h_\alpha & 0 & 1
\end{bmatrix}
\]  
(3.4.12)

From Eqs. (3.4.2) and (3.4.10) we obtain

\[
[D]^{-1}[C] \vec{u}^{n+1} = [A]^{-1}[B] \vec{u}^n
\]  
(3.4.13)
The eigenvalues ($\lambda$) can be obtained from the determinant

$$|\begin{bmatrix} A^{-1}B & -D^{-1}C \end{bmatrix} \lambda | = 0 \quad (3.4.14)$$

where

$$\lambda = \exp (i\theta' \Delta t) \quad (3.4.15)$$

The characteristic equation obtained from Eq. (3.4.14) is

$$(\lambda - 1) \left[ \frac{\lambda^2 + 2 (1 + \rho \rho^2 + \gamma \gamma^2 + \rho \rho^2 \times \gamma \gamma^2 + 2c^2)}{(1 + \rho \rho^2 + \gamma \gamma^2 + \rho \rho^2 \times \gamma \gamma^2)} \lambda + 1 \right]$$

$$= 0 \quad (3.4.16)$$

All moduli of the eigenvalues ($\lambda$) are one when

$$\frac{-1 + \rho \rho^2 + \gamma \gamma^2 + \rho \rho^2 \times \gamma \gamma^2 + 2c^2}{1 + \rho \rho^2 + \gamma \gamma^2 + \rho \rho^2 \times \gamma \gamma^2} \leq 1 \quad (3.4.17)$$

The inequality on the left is always satisfied, and the inequality on the right is satisfied when

$$c^2 < 1 \quad \text{or} \quad \tau f < 2 \quad (3.4.18)$$

The conditions of Eq. (3.4.18) do not directly guarantee computational stability. The von Neumann necessary condition is satisfied, but it is only possible to investigate if any of the sufficient conditions holds for special values of $c, \alpha,$ and $\gamma$.

A stable method is obtained by use of the linearized effect of bottom roughness. Now omitting the Coriolis effect, the matrices $[A]$, $[B]$, $[C]$, and $[D]$ of Eqs. (3.4.2) and (3.4.10) are

$$[A] = \begin{bmatrix} 1 & 0 & i\alpha \\ 0 & 1+i\alpha & 0 \\ i\alpha & 0 & 1 \end{bmatrix} \quad (3.4.19)$$
\[
[B] = \begin{bmatrix}
1-R & 0 & 0 \\
0 & 1 & -ig\psi \\
0 & -i\psi & 1 \\
\end{bmatrix} 
\] (3.4.20)

\[
[C] = \begin{bmatrix}
1+R & 0 & 0 \\
0 & 1 & ig\psi \\
0 & i\psi & 1 \\
\end{bmatrix} 
\] (3.4.21)

\[
[D] = \begin{bmatrix}
1 & 0 & -ig\alpha \\
0 & 1-R & 0 \\
-i\alpha & 0 & 1 \\
\end{bmatrix} 
\] (3.4.22)

The characteristic equation obtained from Eq. (3.4.14) is

\[
\lambda^3 + \left[ 1 - \frac{4(1 + RAB)}{(1 + R)K} \right] \lambda^2 - \left[ \frac{(1 - R)^2 K + 4(-1 + RAB)}{(1 + R)^2 K} \right] \lambda - \frac{(1 - R)^2}{(1 + R)^2} = 0 
\] (3.4.23)

where

\[
A = gh\alpha^2 \\
B = gh\psi^2 \\
K = 1 + A + B + AB
\]

Using the criteria developed in Appendix B, it can be shown that all eigenvalues are less than one, and consequently, unconditional stability may be expected according to the fourth sufficient condition.

The combined influence of the Coriolis effect and the bottom-stress terms makes the stability analysis increasingly more difficult. Generally, it would be expected that if a stability investigation of all the individual physical factors (such as the bottom stress, effect of earth rotation, or the convective term) does show numerical stability of the procedure, then the combined effect of these factors would also lead to a stable numerical procedure. Kasahara,\(^{22}\) however, indicated that this is not necessarily true. Because sufficiently detailed analyses have not been made, the computational method itself had to be tested extensively. These tests are described in Section IV.
3.5 WAVE DEFORMATION

According to the analyses in Section 2.3, it may be expected that
the velocity of propagation and possibly the amplitude of the computed
wave will differ from the physical wave.

For wave motion in one spatial dimension, a forward- and backward-
running wave was found, and the characteristic equations obtained from
the amplification matrix had two roots. In this case, for wave motion
in two spatial dimensions, the characteristic equation obtained from
the amplification matrices of the finite-difference schemes has three
roots.

If the effect of earth rotation is neglected in the characteristic
equation for wave motion in two spatial dimensions (Eq. (3.4.16)),
then this equation can be written

\[(\lambda - 1) \left( \lambda^2 + \frac{2(-1 + A)}{(1 + A)} \lambda + 1 \right) = 0 \quad (3.5.1)\]

where

\[\lambda = \exp (i\beta'\tau) \quad (3.5.2)\]

and

\[A = \frac{\tau^2}{4\ell^2} gh \left[ \sin^2 (\sigma_1 \ell) + \sin^2 (\sigma_2 \ell) + \frac{\tau^2}{4\ell^2} gh \sin^2 (\sigma_1 \ell) \sin^2 (\sigma_2 \ell) \right] \quad (3.5.3)\]

Two of these three roots give relations between the wave numbers (\(\sigma\))
and the frequency (\(\beta'\)) and represent the progressive and retrogressive
gravity waves. The third root, which does not contain these relations,
represents the steady-state flow of the whole field. \(^{(14)}\) In this case,
the flow has been set at zero for purposes of illustration.

The propagation factor was defined in Section 2.3 as the complex
ratio of the computed wave in amplitude and phase to the physical wave
after a time interval in which the physical wave propagates over its wavelength. This definition can be maintained for wave motion in two dimensions. Assuming a wave with the wave number \( \sigma \) moving in a direction which has an angle \( \gamma \) with the \( x \)-axis, then the wave motion may be described in its general form

\[
\overline{U} = \overline{U}^* \exp \left( i \beta' t + \sigma_1 x + \sigma_2 y \right) \quad (3.5.4)
\]

where

\[
\sigma_1 = \sigma \cos \gamma \quad (3.5.5)
\]

\[
\sigma_2 = \sigma \sin \gamma \quad (3.5.6)
\]

The propagation factor, which is now a function of the dimensionless parameter \( \sigma L \) and the angle direction \( \gamma \), is then expressed by

\[
T(\sigma L, \gamma) = \frac{\exp \left[ i(\beta' t + \sigma_1 x + \sigma_2 y) \right]}{\exp \left[ i(\beta t + \sigma_1 x + \sigma_2 y) \right]} \quad (3.5.7)
\]

for

\[
x = L \cos \gamma = \frac{2\pi}{\sigma} \cos \gamma
\]

\[
y = L \sin \gamma = \frac{2\pi}{\sigma} \sin \gamma
\]

\[
\tau = \frac{2\pi}{\beta}
\]

Thus, according to Eq. (2.3.14)

\[
|T(\sigma L, \gamma)| = \exp \left[ i2\pi(\beta' / \beta - 1) \right] \quad (3.5.8)
\]

From the characteristic equation (Eq. (3.5.1)) an expression is found for the relation between the computed frequency and the spatial frequencies (wave numbers)
\[ e^{iB_{2,3}^\tau} = \frac{1 - A \pm 2\sqrt{A}}{1 + A} \quad (=\lambda) \quad (3.5.9) \]

Since the modulus of the eigenvalue \( \lambda \) equals one, the imaginary parts of \( B_2 \) and \( B_3 \) are zero. The real parts of \( B_2 \) and \( B_3 \) are then obtained from

\[ \sin (B_{2,3}^\tau) = \frac{\pm 2\sqrt{A}}{1 + A} \quad (3.5.10) \]

or

\[ B_{2,3}^\tau = \pm \sin^{-1} \left[ 2\sqrt{A} / (1 + A) \right] \quad (3.5.11) \]

Introducing Eqs. (3.5.5) and (3.5.6) into Eq. (3.5.3), we find

\[
A = \frac{\tau^2}{\Delta^2} \frac{gh}{L^2} \left[ \sin^2 (\sigma t \cos \gamma) + \sin^2 (\sigma t \sin \gamma) \right. \\
+ \left. \frac{1}{2} \frac{\tau^2}{L^2} \frac{gh}{L^2} \sin^2 (\sigma t \cos \gamma) \sin^2 (\sigma t \sin \gamma) \right]
\]

The argument of the propagation factor for the progressive and retrogressive wave can now be expressed as the function of the system parameter \( \sigma t \) and the direction \( \gamma \)

\[
\arg[T(\sigma t, \gamma)] = 2\pi \left[ \frac{\sin^{-1} \left[ 2\sqrt{A} / (1 + A) \right]}{(\tau / L) \sqrt{gh} (\sigma t) \gamma} - 1 \right]
\]

Figure 3.5.1 presents this argument for the condition \( (\tau / L) \sqrt{gh} = 1 \).

It will be noted that the wave propagation in a 45-deg direction with the coordinates gives a smaller argument; thus, a better agreement results between computed velocity and the physical velocity in that direction.
Figure 3.5.1—Phase angle of the propagation factor as a function of the direction of the wave.
3.6 CONSERVATION OF MASS AND MOMENTUM

The complete partial-differential equations can be written in the following vector form:

\[
\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} + \frac{\partial \mathbf{G}(\mathbf{U})}{\partial y} + \mathbf{K}(\mathbf{U}) = \mathbf{F}(x,y,t) \quad (3.6.1)
\]

where

\[
\mathbf{F}(\mathbf{U}) = \left\{ \begin{array}{c} gU(x) + \mathbf{U}(x) \\ \mathbf{U}(x) \\ [h(x) + \zeta(x)]\mathbf{U}(x) \end{array} \right\} \quad (3.6.2)
\]

\[
\mathbf{G}(\mathbf{U}) = \left\{ \begin{array}{c} \mathbf{V}(y) \\ g\zeta(y) + \mathbf{V}(y) \\ [h(y) + \zeta(y)]\mathbf{V}(y) \end{array} \right\} \quad (3.6.3)
\]

\[
\mathbf{K}(\mathbf{U}) = \left\{ \begin{array}{c} -fV + RU \\ +fU + RV \\ 0 \end{array} \right\} \quad (3.6.4)
\]

\[
R = \frac{\sqrt{U^2 + V^2}}{C^2 (h + \zeta)} \quad (3.6.5)
\]

Omitting the forcing function, bottom stress, and the Coriolis force, a finite-difference approximation of Eq. (3.6.1) with central differences can be written

\[
\mathbf{U}^{(n+1)} = \mathbf{U}^{(n)} - \frac{\Delta t}{\Delta s} \mathbf{F}(n+\frac{1}{2}) - \frac{1}{2} \frac{\Delta t}{\Delta s} \left( \mathbf{G}^{(n+1)} + \mathbf{G}^{(n)} \right) \quad (3.6.6)
\]
and also

$$\overline{U}(n^{1/2}) = \overline{U}(n^{1/2}) - \frac{1}{2} \frac{\Delta s}{\Delta t} \left( \overline{F}(n^{1/2}) + \overline{G}(n^{1/2}) \right) - \frac{\Delta t}{\Delta s} \overline{g}^{n} \quad \quad (3.6.7)$$

Integration of Eqs. (3.6.6) and (3.6.7) over a time step in the manner described in Section 2.8 shows that mass and momentum are conserved when one accounts for the mass and momentum that enter through the boundaries during the time step.

If the convective terms, the bottom-stress terms, and the Coriolis-force terms are omitted in Eqs. (3.1.2) through (3.1.7), and if Eqs. (3.1.2), (3.1.3), and (3.1.4) are introduced into Eqs. (3.1.5), (3.1.6), and (3.1.7), respectively, Eqs. (3.6.6) is then found by taking

$$\overline{U}_{j,k} = \begin{cases} u_{j+1/2,k} \\ v_{j,k+1/2} \\ \zeta_{j,k} \end{cases} \quad \quad (3.6.8)$$

$$\overline{F}_{j,k} = \begin{cases} \frac{1}{2} \left( \frac{\Delta y}{\Delta x} \frac{\partial u_{j+1/2,k}}{\partial x} \right) \\ \frac{1}{2} \left( \frac{\Delta x}{\Delta y} \frac{\partial u_{j+1/2,k}}{\partial y} \right) \\ \frac{1}{2} \left( \frac{\partial u_{j+1/2,k}}{\partial x} \right) \\ \frac{1}{2} \left( \frac{\partial u_{j+1/2,k}}{\partial y} \right) \end{cases} \quad \quad (3.6.9)$$

$$\overline{G}_{j,k} = \begin{cases} \frac{1}{2} \left( \frac{\partial v_{j+1/2,k}}{\partial x} \right) \\ \frac{1}{2} \left( \frac{\partial v_{j+1/2,k}}{\partial y} \right) \end{cases} \quad \quad (3.6.10)$$

If Eqs. (3.1.5), (3.1.6), and (3.1.7) are introduced into Eqs. (3.1.2), (3.1.3), and (3.1.4), respectively, then Eq. (3.6.7) is formed. The nonlinear convective terms and the term expressing the effect of the bottom stress cannot be fitted into Eqs. (3.6.8) through (3.6.10) because the proper time and appropriate spatial gradients are not available. Thus,
the momentum is not exactly conserved, and it may be expected that waves will decay during computation.

Numerical experiments of long waves propagating in shallow water of the type discussed in Section 4 indicate that the contribution of the convective terms is very small. Thus, these terms are generally omitted (see Welander, Hansen, Uusitalo, Lauwerier, Harris, and Miyazaki), and the effect of the nonconservative properties of the convective terms on the computation is negligible.

The deviation of part of the velocity terms in the expression for the bottom-stress effect from the central time level can be considered unimportant, as the bottom-stress term causes waves to be dampened during computation.

Mass will be conserved if the terms marked with an asterisk in the continuity equations Eqs. (3.1.3) and (3.1.6) are taken at the later time. However, this makes the continuity equation implicit, and iteration is required for advancing the solution in time.
4. NUMERICAL EXPERIMENTS

4.0 USE OF NUMERICAL EXPERIMENTS

As described in Section 3, a large number of numerical experiments were made during the development of the finite-difference method. Also, simple one-dimensional computations were made on an on-line time-sharing computer to obtain insight into the behavior of computed-wave motion. A typical computation of this type is mentioned in Section 2 for the study of wave decay of the forward implicit method, and many of the characteristics of the finite-difference methods analyzed in Section 2 were noted during actual on-line computation.

This section discusses two-dimensional computations made for wave motions in three areas. The first area was Tokyo Bay, where the computational model used was wave motion generated by a tsunami. With this model, the multiperational method was compared to an explicit computation by Isozaki and Unoki (26) and to actual water-level observations.

The second area for which computations were made was the "Haringvliet," which is in the estuary of the Rhine River in the Netherlands. The mathematical model was used to compare measurements of tidal waves (flow) with computational data, to develop a general program for the computation of long-wave motion, and to obtain expedient ways for the presentation of computational data. This computational model was also used to illustrate the effects of different approximations of the differential equations.

Finally, a model of the tidal motion in the southern North Sea was used to test the adequacy of the program description presented in Appendix C. The following sections describe all three models together with the results of the computations.

4.1 TOKYO BAY MODEL

The grid system used for computation in the Tokyo Bay area is presented in Fig. 4.1.1. This grid system is the same system as used by Isozaki (26). The water depths are identical to those assumed by Isozaki, with the exception of a water depth of 1 m along the coast. Isozaki used zero depth at these points. The input boundary data,
water levels computed at a few stations, and observations and computations of Isozaki are presented in Fig. 4.1.2. His results and those obtained by use of the method described in Section 3, using a C value of 50 m$^{1/2}$ sec$^{-1}$, were found to be in close agreement.

Isozaki used a quadratic resistance term with a coefficient of 0.0026, which is equivalent to a C value of about 60 m$^{1/2}$ sec$^{-1}$. In addition, a smoothing process was used which presented the effect of horizontal eddy viscosity. However, it has not been indicated by Isozaki how smoothing of computed data is obtained. This smoothing naturally has a stabilizing effect in the numerical computation and a tendency to dampen the wave.

4.2 THE HARINGVLIET MODEL: EFFECT OF STEP SIZE

The second model used for testing the computational procedure is a model of the mouth of the Haringvliet, which is shown in Fig. 4.2.1. This model was chosen because detailed measurements of tidal levels and currents were available.

The water depth varies considerably over short distances, as can be seen in Fig. 4.2.2. A grid size of 400 m with a computational array of 31 by 55 points was chosen for the computations.

As an aid in the evaluation of the results, a plot program was developed for the display of the currents and water levels at certain times during computation. Plot information from the actual computation is placed on magnetic tape, is then processed, and is finally plotted by the Stromberg Carlson SC 4020 microfilm recorder. An enlargement of such a plot is given in Fig. 4.2.3.

The computational array is shown in the bottom part of this figure, and the origin of the coordinates of the points upon which the water levels are computed is shown in the lower left-hand corner. The integer values along the vertical Y-axis are counted as N values, and the integer values along the horizontal X-axis are indicated by M values. Every tenth point is shown on the axis of the array. The locations of the water levels within the boundaries are indicated by dots, and the locations where water levels are given as a function of time are indicated by asterisks. The magnitude and direction of the computed
Fig. 4.1.2—Water-level histories in Tokyo Bay
May 29, 1960
Fig. 4.2.1—Location of Haringvliet and North Sea computational models
Fig. 4.2.2—Location of Harlingvliet model
Fig. 4.2.3—Haringvliet (estuary of the Rhine)
Situation 15.6 hr (real time) after start of computation

Bottom  Field of the computation and comparison between measured velocities and computed velocities. The measured velocities are average velocities of similar tidal cycles at a particular location, while the computed velocities are the average velocity in an 800-by-800-m area of a particular tidal cycle.

Top  Isometric sections representing computed water levels on every fifth line of the bottom graph.
velocity are plotted on 24 locations. This velocity is the average in an 800-by-800-m field. In addition, measured velocities are plotted in magnitude and direction every hour (real time) of the tidal cycle. The latter velocities are averages of measured values obtained on a particular location from similar tidal cycles. A solid line indicates a rough outline of the coast.

In the top part of Fig. 4.2.3 isometric sections of water levels computed on every fifth line of the bottom graph are shown on the base plane at the reference level. A 1-m mark is indicated on the vertical scale to the left.

In addition to these displays of water levels and currents at a certain time during computation, plots have been made of water-level histories. A typical plot of this type is shown in Fig. 4.2.4. Values computed after the first operation are indicated by (0), and values computed after the second operation are indicated by (+). If available, measured water levels were also plotted, and in incidental cases, the discharge through the inlet was plotted.

The computations were made for periods up to 15 hr (real time). On all points in the array, a zero velocity was assumed as an initial condition, and the water levels over the whole array were taken at -0.70 m. This starting condition is close to the actual situation, and the effect of the starting errors in the initial conditions disappear after a few hours of real time. Since a periodic cycle of 12.4 hr was used for the computation, the effect of starting errors can be found by comparing the water levels and velocities computed at the end of the computation with those computed 12.4 hr earlier.

All water levels at the North Sea boundary of the model were described as functions of a few measured water-level histories. The function used was a linear interpolation, according to the distance between given adjacent water levels. The time functions used are shown in Figs. 4.2.5 through 4.2.8. Figure 4.2.9 shows the water levels used at the river side of the model, and Fig. 4.2.10, which was extrapolated from Figs. 4.2.7 and 4.2.8, presents the water-level history at the extreme end (x = 31, y = 55). In the comparison of different computations which follows, only one or a few of the computed graphs of each
Fig. 4.2.4—Typical plot of a water-level history

Gage No. 7 Comparison between measured data (−) and computation (+0 +)
Fig. 4.2.5—Water levels at $n = 24$, $m = 1$

Fig. 4.2.6—Water levels at $n = 31$, $m = 1$
Fig. 4.2.7—Water levels at $n = 31$, $m = 23$

Fig. 4.2.8—Water levels at $n = 31$, $m = 46$ (solid line)
Fig. 4.2.9—Water levels at \( n = 1, m = 27 \)

Fig. 4.2.10—Water levels at \( n = 31, m = 55 \) (by extrapolation)
computation are presented. Those selected can be considered representative of all graphs obtained for the computation.

Computations were made with identical boundary conditions, depths, Chezy values, and time steps for each of the two different operations. These time steps are called half time steps and are 360 sec, 180 sec, and 90 sec, respectively. (The depths and the Chezy values used for these computations can be found in Appendix C.)

The time-step size influenced the velocities considerably, i.e., the larger the time step the lower the computed velocities. This can be seen by comparing the current vectors in Figs. 4.2.11 through 4.2.16. The influence of the time step is also very clearly shown on the plots of computed transport through the narrow channel (inlet) on n = 4 in Figs. 4.2.17 through 4.2.19.

Obviously, a considerable deviation occurs in the approximation with a half time step of 360 sec. Use of this time step gives values larger than 5 for the dimensionless parameter \( \tau \lambda^{-1} \sqrt{gh} \). From the discussion about wave deformation in Sections 2.3 and 3.5 and from Fig. 2.3.1, it will be clear that for the higher components in the tidal wave, the phase velocity becomes much lower than the physical velocity. As the model is steered by water levels at boundaries, there is the apparent effect of the larger distances involved between the boundaries, which shows in the computation as lower velocities. The difference between velocities computed with a half time step of 180 and 90 sec is approximately 10 percent. Thus one may expect that a half time step of 90 sec is adequate.

The influence of time-step size on the computed water levels is insignificant. Figures 4.2.20 through 4.2.22 show the water levels computed near the middle of the computational array at the location n = 13, m = 31 for half time steps of 360, 180 and 90 sec, respectively. In Fig. 4.2.22, only the result of the water levels computed at even time steps are plotted for clarity.

4.3 THE HARINGVLIET MODEL: EFFECT OF ITERATION

Section 3.1 indicates that the water level in the continuity equation should be taken at the time level n + 1. However, as this value
Fig. 4.2.11—Currents and water levels 5.2 hr after start of computation with $\tau = 360$ sec

Fig. 4.2.12—Currents and water levels 5.2 hr after start of computation with $\tau = 180$ sec
Fig. 4.2.13—Currents and water levels 5.2 hr after start of computation with \( \tau = 90 \text{ sec} \)

Fig. 4.2.14—Currents and water levels 8.6 hr after start of computation with \( \tau = 360 \text{ sec} \)
Fig. 4.2.15—Currents and water levels 8.6 hr after start of computation with $\tau = 180$ sec

Fig. 4.2.16—Currents and water levels 8.6 hr after start of computation with $\tau = 90$ sec
Fig. 4.2.17—Transport through line $n = 4$ computed with $\tau = 360$ sec

Fig. 4.2.18—Transport through line $n = 4$ computed with $\tau = 180$ sec
Fig. 4.2.19—Transport through line n = 4 computed with \( \tau = 90 \) sec

Fig. 4.2.20—Water levels at n = 13, m = 31 computed with \( \tau = 360 \) sec
Fig. 4.2.21—Water levels at $n = 13$, $m = 31$
computed with $\tau = 180$ sec

Fig. 4.2.22—Water levels at $n = 13$, $m = 31$
computed with $\tau = 90$ sec
is not available, the value at the time level \( n \) is used as a first approximation.

As also indicated in Section 3.1, a more accurate result can be obtained by an iteration procedure, which increases the computational time linearly with the number of iterations. Figures 4.2.20 and 4.3.1 present the results of water levels computed at \( n = 13, m = 31 \), with and without an iteration. The differences in the computed water levels are only fractions of a centimeter. The use of one iteration roughly doubles the computation time. A better approximation can be expected by using half the time step and no iteration. Figure 4.2.21 shows the result of such a computation, the agreement between computed and measured data being very close.

4.4 THE HARINGVLIET MODEL: EFFECT OF APPROXIMATION

Section 3.1 indicates that several different approximations are possible for the expression of the convective-inertia terms and the resistance terms. A series of test computations were made, and the important ones are discussed here. In each test the computational formulas were changed, but the time step, boundary condition, and Chezy values remained the same throughout. A description of each of these tests follows, and Test A is taken as a basis of comparison for the other test findings. A large time step of 360 sec is used to exaggerate local instabilities and the effects of the different approximations of the differential equations obtained by the two difference equations.

Test A. The computational formulas used are described in Sections 3.1 and 3.2 and in Appendices A and C. All depth values and Chezy values can be found in Appendix C. Figures 4.4.1 and 4.4.2 present the water levels computed on two locations; namely, at \( n = 13, m = 31 \), and at \( n = 3, m = 29 \). The first location is typical for most of the plots, and the second location is where local instabilities are most apparent.

Test E. The velocity \( v \) in the convective-inertia term \( v(\partial v/\partial y) \) in Eq. (3.1.4) and the velocity \( u \) in the convective-inertia term \( u(\partial u/\partial y) \) in Eq. (3.1.5) are taken at the lower time level, according to the preferred formula in Table 3.2.1 of Section 3.2.
Fig. 4.3.1—Water levels at \( n = 13, \ m = 31 \) (one iteration used in continuity equation)
Fig. 4.4.1—Water levels at $n = 13$, $m = 31$
Test A

Fig. 4.4.2—Water levels at $n = 3$, $m = 29$
Test A
Figures 4.4.3 and 4.4.4 present the water levels computed at the two selected locations. The computation at \( n = 3, m = 29 \) appears somewhat less stable than the computation at \( n = 13, m = 31 \).

**Test F.** Computations are made with the formulas described in Sections 3.1 and 3.2. Rather than omit the convective-inertia terms along the boundaries as in Test A, the convective-inertia terms are taken from the velocity-gradient information inside the field and are thus off-centered. As one would expect from the discussion in Section 2.7, the use of off-centered spatial gradients results in local instabilities. Figures 4.4.5 and 4.4.6 present the water levels at the two selected locations, the results being similar to Test E.

**Test G.** Here the resistance terms are changed from those in Eqs. (3.2.5) through (3.2.9) to those shown in Eqs. (4.4.1) through (4.4.4):

\[
R(x)^{(n)} = \frac{1}{4} \Delta t \ g (u^{(n)} + u^{(n+\frac{1}{2})}) \sqrt{\frac{(u^{(n)})^2 + (v^{(n)})^2}{(h^x + c^x(n)) (c^x)^2}} \quad \text{at } j + \frac{1}{2}, k
\]  
(4.4.1)

\[
R(y)^{(n+\frac{1}{2})} = \frac{1}{2} \Delta t \ g (v^{(n)} + v^{(n+\frac{1}{2})}) \sqrt{\frac{(v^{(n+\frac{1}{2})})^2 + (v^{(n)})^2}{(h^y + c^y(n+\frac{1}{2})) (c^y)^2}} \quad \text{at } j, k + \frac{1}{2}
\]  
(4.4.2)

\[
R(y)^{(n+1)} = \frac{1}{2} \Delta t \ g (u^{(n+\frac{1}{2})} + u^{(n+1)}) \sqrt{\frac{(u^{(n+\frac{1}{2})})^2 + (v^{(n+1)})^2}{(h^x + c^x(n+1)) (c^x)^2}} \quad \text{at } j + \frac{1}{2}, k
\]  
(4.4.3)

\[
R(y)^{(n+\frac{1}{2})} = \frac{1}{2} \Delta t \ g (v^{(n+\frac{1}{2})} + v^{(n+1)}) \sqrt{\frac{(v^{(n+\frac{1}{2})})^2 + (v^{(n+1)})^2}{(h^y + c^y(n+\frac{1}{2})) (c^y)^2}} \quad \text{at } j, k + \frac{1}{2}
\]  
(4.4.4)

Again, the system appears less stable, as may be seen in Figs. 4.4.7 and 4.4.8.
Fig. 4.4.3 — Water levels at $n = 13$, $m = 31$
Test E

Fig. 4.4.4 — Water levels at $n = 3$, $m = 29$
Test E
Fig. 4.4.5 — Water levels at $n = 13$, $m = 31$
Test F

Fig. 4.4.6 — Water levels at $n = 3$, $m = 29$
Test F
Fig. 4.4.7—Water levels at $n = 11$, $m = 22$
Test G

Fig. 4.4.8—Water levels at $n = 3$, $m = 29$
Test G
Test H. Here, most of the velocities in the resistance terms are now taken at the lower time level; instead of Eqs. (3.2.6) and (3.2.7), the following equations are used:

\[ R(y)^*(n+\frac{1}{2}) = \frac{1}{2} \Delta t \, g v(n) \sqrt{\frac{u(n+\frac{1}{2})^2 + v(n)^2}{h^x + c_0^{y(n+\frac{1}{2})}}} \quad \text{at } j, k + \frac{1}{2} \]

(4.4.5)

\[ R(x)^*(n+1) = \frac{1}{2} \Delta t \, g u(n+\frac{1}{2}) \sqrt{\frac{u(n+\frac{1}{2})^2 + u(n+1)^2}{h^y + c_0^{x(n+1)}}} \quad \text{at } j + \frac{1}{2}, k \]

(4.4.6)

This computational method appeared to be completely unstable. Computation was terminated after a few steps because values exceeded the computational limits of the computer and no graph could be obtained.

4.5 NORTH SEA MODEL

As a final check of the general computational procedures developed during the experiments with the Haringvliet area, a tidal computation was made of the southern North Sea (see Fig. 4.2.1). The grid used for the computation is shown on Fig. 4.5.1, and the grid size is 5,600 m. Water levels are computed in the middle of each square, and velocities are computed in the middle of each side of the squares.

The northern boundary of the model was described at four locations as a time function of the water levels. Other points of this boundary were computed by linear interpolation or extrapolation, as indicated in Fig. 4.5.1. The southern boundary in the English Channel was described at the coasts as a time function of the water levels, and the intermediate points were computed by interpolation.

In the southwestern part of the Netherlands, currents in the different parts of the estuaries of the Rhine and Schelde were used for the boundary. Tidal data for the period from 0.00 hr (Middle European Time), September 12, 1958, to 0.50 hr on September 13, 1958, were used
- Tide gage
- Land bound
- Open boundary with water level as input
- Open boundary with current as input
- Direct input from tables
- Linearly interpolated
- Linearly extrapolated

Grid size: 5.6 x 5.6 km
Water levels are computed in the middle of each square
for computation (see Fig. 4.5.2). The tidal curves for the four points on the northern bound and the curves describing the water levels on the Channel are shown in Fig. 4.5.2. Some adjustments were made at the beginning and end of this period in order to make all tidal curves a complete cycle over this period. The half time step of each operation was taken at 5 min. The C values were computed every half hour (real time) on the location of the water level as a function of the average depth at that moment. The following expression which was obtained experimentally from computations of the Haringvliet was used:

$$C = 19.4 \ln \left[0.9(h + \zeta)^7\right]$$  \hspace{1cm} (4.5.1)

Computations were started with all water levels and velocities taken at zero. The water levels and currents at the open boundaries were increased from zero with five steps into the given tidal curve. The starting disturbance disappeared after approximately 14 hr. Results of the computation of the second 24.8-hr cycle are shown in Figs. 4.5.3 through 4.5.9, where the water levels and velocity vectors are represented at 2-hr intervals in the afternoon of September 12, 1958.

The locations of equal high-tide levels (co-range lines) obtained from the results of this computation are shown in Fig. 4.5.10, together with the arrival times of the highest water levels (co-tidal lines). Figure 4.5.11 shows the same for the low tides. In these two figures a counterclockwise rotation of the vertical tide may be seen. The water-level information shown in Figs. 4.5.10 and 4.5.11 concurs with tidal information presented by Defant (27) as to phases and amplitudes of the tides.

The maximum magnitude of the computed currents shown in Figs. 4.5.3 through 4.5.9 agrees with information given on tidal charts, but no detailed analysis has been made of the phase and amplitude of these currents with respect to such information. Water levels and currents along the coast are not accurate as the grid size is too large for a good representation, and also more accurate input data is needed for these computations.
Fig. 4.5.2—Tidal data for September 12, 1958
Fig. 4.5.3—North Sea computed water levels and currents on September 12, 1958 at 12.00 hr

Fig. 4.5.4—North Sea computed water levels and currents on September 12, 1958 at 14.00 hr
Fig. 4.5.5—North Sea computed water levels and currents on September 12, 1958 at 16.00 hr

Fig. 4.5.6—North Sea computed water levels and currents on September 12, 1958 at 18.00 hr
Fig. 4.5.7—North Sea computed water levels and currents on September 12, 1958 at 20.00 hr

Fig. 4.5.8—North Sea computed water levels and currents on September 12, 1958 at 22.00 hr
Fig. 4.5.9—North Sea computed water levels and currents on September 12, 1958 at 24.00 hr
Fig. 4.5.10—Iso high-tide levels and times on September 12, 1958

Fig. 4.5.11—Iso low-tide levels and times on September 12, 1958
5. CONCLUDING REMARKS

It has been shown that the propagation of long waves in coastal waters can be studied successfully by use of a numerical integration method. The multioperation method developed is characteristic of implicit methods; namely, there is no upper limit on the time step for stability reasons, as is the case with explicit methods. The multioperation method allows a direct and rapid solution of all velocities and water levels on each time level.

The multioperation method described is particularly suitable for long-wave computation in coastal waters, where water movements are introduced by changes in the water level (or currents) along the sides of the model and where the effect of bottom friction is larger than the effects of lateral eddy viscosity, which is neglected.

The contribution of the convective-inertia terms in the equation of motion is assumed to be small compared to that of other terms. These terms are represented with a lower order of accuracy.

The detailed description of computational procedures permits an expedient introduction of geographic features such as water depth, boundaries, and characteristics of bottom roughness for the modeling of wave propagation in hydraulic engineering research.

For the design of numerical experiments, the discussions concerning wave deformation and the expressions for the propagation factors gives an indication as to the accuracy that might be expected from the computation.

Generally, information concerning the magnitude of the effect of bottom roughness is inadequate. In some cases, like the Haringvliet model, the water movements are influenced by bottom roughness to a considerable extent. In such a case, the parameter C has to be found in an iterative manner by comparing computed results with actual field measurements. The rate at which the model can be adjusted to resemble the prototype depends on the extent of available field data and on the experience of the engineer making the investigation and his insight into the physics of the wave problem and into the behavior of the method of numerical solution.
Appendix A

COMPUTATION FORMULAS

In this appendix the formulas developed in Section 3 are presented in a different manner. In this formulation, a separate coordinate system is adopted for each of the velocity components ($U, V$), the water level ($\zeta$), and the depth ($h$). These four coordinate systems allow an efficient use of the computer memory, as all points of each of the four arrays can be indicated by two integer values. The nomenclature is shown in Fig. A.1.

The velocities $U$ and $V$ and the water level $\zeta$ are indicated at the beginning of a whole time step. At the end of the first operation their values are indicated by $U', V'$, and $\zeta'$, while at the end of the second operation their values are indicated by double-prime values. The coefficients in the recursion formulas (Eqs. (3.1.24) and (3.1.25)) in the $x$-direction become

\[
P_m = \frac{\tau}{2L} \left( h_{n,m} + h_{n-1,m} + \zeta_{n,m} + \zeta_{n,m+1} \right) \frac{1}{1 + \frac{\tau}{2L} \left( h_{n,m-1} + h_{n-1,m-1} + \zeta_{n,m-1} + \zeta_{n,m} \right) R_{m-1}}
\]

(A.1)

\[
Q_m = \frac{A_m + \frac{\tau}{2L} \left( h_{n,m-1} + h_{n-1,m-1} + \zeta_{n,m-1} + \zeta_{n,m} \right) S_{m-1}}{1 + \frac{\tau}{2L} \left( h_{n,m-1} + h_{n-1,m-1} + \zeta_{n,m-1} + \zeta_{n,m} \right) R_{m-1}}
\]

(A.2)

\[
R_m = \frac{\frac{\tau}{L} g}{1 + \frac{\tau}{L} \left[ gP_m + (1 - \alpha_{n,m}) (U_{n,m+1} - U_{n,m}) + \alpha_{n,m} (U_{n,m} - U_{n,m-1}) \right]}
\]

(A.3)

\[
S_m = \frac{P_m + \frac{\tau}{L} gQ_m}{1 + \frac{\tau}{L} \left[ gP_m + (1 - \alpha_{n,m}) (U_{n,m+1} - U_{n,m}) + \alpha_{n,m} (U_{n,m} - U_{n,m-1}) \right]}
\]

(A.4)
All points in this square have same subscript $n, m$

All points on these rows have same subscript $m$

All points on these columns have same subscript $n$

- $U$ velocity
- $V$ velocity

$\bullet$ Depth
+ Water level

Fig. A.1—Computational grid
where \( a = 0.5 \) and\( b = 0.5. \)

\[
A_{n,m} = \zeta_{n,m} - \frac{\tau}{2L} \left( h_{n,m} + h_{n,m-1} + \zeta_{n,m} + \zeta_{n+1,m} \right) v_{n,m} + \frac{\tau}{L} \left( h_{n-1,m} + h_{n-1,m-1} + \zeta_{n,m} + \zeta_{n-1,m} \right) v_{n-1,m} \tag{A.5}
\]

\[
B_{n,m} = U_{n,m} + \frac{1}{4} \left[ (1 - \gamma_{n,m}) (v_{n+1,m} - v_{n,m}) - \gamma_{n,m} \frac{1}{4} (v_{n,m} - v_{n-1,m}) \right] (v_{n,m} + v_{n,m+1} + v_{n-1,m} + v_{n-1,m+1})
\]

\[
\frac{\delta_{n,m}}{2} \left[ \frac{\left( \frac{v_{n,m}^2}{v_{n,m}} + \frac{1}{16} \left( \frac{v_{n,m} + v_{n,m+1} + v_{n-1,m} + v_{n-1,m+1}}{v_{n,m} + v_{n,m+1} + v_{n-1,m} + v_{n-1,m+1}} \right)^2 \right)}{\left( \zeta_{n,m} + \zeta_{n,m+1} + h_{n,m} + h_{n-1,m} + \zeta_{n,m} + \zeta_{n,m+1} \right)^2} \right] \tag{A.6}
\]

The explicit operation on the velocity \( v \) in Eq. (3.2.15) becomes

\[
v'_{n,m} = \frac{V_{n,m} - \frac{1}{4} \left[ \frac{\tau}{2L} \left( h_{n,m} + h_{n,m+1} + \zeta_{n,m} + \zeta_{n+1,m} \right) (v_{n,m} + v_{n,m+1} + v_{n-1,m} + v_{n-1,m+1}) - \frac{\tau}{L} \left( \zeta_{n+1,m} - \zeta_{n,m} \right) \right]}{1 + \left( \frac{1}{16} \left( \frac{v_{n+1,m} + v_{n,m+1} + v_{n-1,m} + v_{n-1,m+1}}{v_{n,m} + v_{n,m+1} + v_{n-1,m} + v_{n-1,m+1}} \right)^2 \right)} + \frac{\tau}{2} \left( 1 - \beta_{n,m} \right) \left( v_{n+1,m} - v_{n,m} \right) + \beta_{n,m} \left( v_{n,m} - v_{n-1,m} \right) \tag{A.7}\]

where \( \delta = 0.5 \) and \( b = 0.5. \) The terms for the recursion formulas in y-direction become

\[
\phi_n = \frac{\left( h_{n,m} + h_{n,m-1} + \zeta_{n,m} + \zeta_{n+1,m} \right)}{1 + \frac{\tau}{2L} \left( h_{n-1,m} + h_{n-1,m-1} + \zeta_{n-1,m} + \zeta_{n,m} \right) v_{n-1,m}} \tag{A.8}
\]

\[
q_n = \frac{a_n + \frac{\tau}{2L} \left( h_{n-1,m} + h_{n-1,m-1} + \zeta_{n-1,m} + \zeta_{n,m} \right)}{1 + \frac{\tau}{2L} \left( h_{n-1,m} + h_{n-1,m-1} + \zeta_{n-1,m} + \zeta_{n,m} \right) v_{n-1,m}} \tag{A.9}
\]

\[
\tau_n = \frac{\left( \phi_n + \left( 1 - \beta_{n,m} \right) (v'_{n+1,m} - v'_{n,m}) + \beta_{n,m} (v'_{n,m} - v'_{n-1,m}) \right)}{1 + \frac{\tau}{L} \left( \phi_n + \left( 1 - \beta_{n,m} \right) (v'_{n+1,m} - v'_{n,m}) + \beta_{n,m} (v'_{n,m} - v'_{n-1,m}) \right)} \tag{A.10}
\]
\[ s_n = \frac{b_n + \frac{\pi}{2} \theta_n}{1 + \frac{\pi}{2} \left[ \theta_n + \frac{1}{2} (\theta_{n+1} + \theta_{n-1}) \right] (v'_{n+1,m} - v'_{n-1,m}) + \theta_n (v'_{n,m} - v'_{n-1,m})} \]  

where

\[ a_n = c_n' - \frac{\pi}{2} \theta_n \left( h_{n,m} + h_{n-1,m} + c_n' + c_{n+1}' \right) u_n', \frac{\pi}{2} \left( h_{n,m+1} + h_{n-1,m+1} + c_n' + c_{n+1}' \right) u_{n,m+1} \]

\[ b_n = \frac{1}{4} \left[ \theta_n + \frac{\pi}{2} \left( h_{n,m} + h_{n-1,m} + c_n' + c_{n+1}' \right) u_n', \frac{\pi}{2} \left( h_{n,m+1} + h_{n-1,m+1} + c_n' + c_{n+1}' \right) u_{n,m+1} \right] \]

\[ - \frac{\theta_n}{\sqrt{2}} \left[ \frac{v'_{n,m} + v'_{n,m+1} + v'_{n+1,m} + v'_{n+1,m+1}}{2} \right] ^{1/2} \]

\[ c_n = c_n' + c_{n+1}' + h_{n,m} \]

\[ \frac{v''_{n,m+1} + v''_{n,m+1} + v''_{n+1,m} + v''_{n+1,m+1}}{2} \]

\[ \left[ 1 - \alpha_{n,m} (v'_{n,m+1} - v'_{n,m}) + \alpha_{n,m} (v'_{n,m} - v'_{n,m-1}) \right] \]

\[ \frac{\pi}{2} \left( c_{n+1}' - c_n' \right) \]

The explicit operation upon the velocity \( U \) becomes

\[ U_{n+1,m} = \frac{\left( \frac{\pi}{2} (1 - v_{n,m})(v'_{n+1,m} - v'_{n,m}) - \frac{\pi}{2} (v'_{n,m} - v'_{n-1,m}) \right) \left( v''_{n,m+1} + v''_{n,m+1} + v''_{n+1,m} + v''_{n+1,m+1} \right) - \frac{\pi}{2} (c'_{n+1}' - c_n')}{1 + \left( c_{n,m} + c_{n+1,m} + h_{n,m} \right)^2} \]

The recursion formulas (Eqs. (3.1.24) and (3.1.25)) for each row \( n \) are now written

\[ c_n' = -p_n v''_{n,m} + q_n \]

\[ u_{n,m+1} = -R_{n+1} c_{n,m} + s_{n+1} \]

while for each implicit operation in the second half time step the recursion formulas (Eqs. (3.2.22) and (3.2.23)) become

\[ c_n' = -p_n v''_{n,m} + q_n \]

\[ u_{n+1,m} = -c_{n+1} c_{n,m} + s_{n+1} \]
In the FORTRAN program presented in Appendix C, only two time levels are used. The values computed in the first half time step (designated UP, VP, SEP) are inserted in the locations of the lower time levels (U, V, SE) in the second half time step. The computations are then made, resulting in values for U", V", and φ", which are again designated (UP, VP, SEP). Thus, for each of the two operations, the computed values at the lower level are indicated in the formulas by U, V, and SE, while the computed values at the higher level are termed UP, VP, SEP.
Appendix B

DETERMINATION OF ROOTS WITHIN THE UNIT CIRCLE OF A CUBIC EQUATION WITH REAL COEFFICIENTS

By investigating the numerical stability of wave motion in two dimensions, the limits of the roots of a cubic equation may be found, and the stability of this method is assured if all roots are within the unit circle.

Assume that a characteristic cubic equation $g(\lambda)$ with real coefficients is obtained from the amplification matrix. This equation would be written

$$g(\lambda) = \lambda^3 + a_1 \lambda^2 + a_2 \lambda + a_3 \quad (B.1)$$

This polynomial can then be transformed by

$$\lambda = \frac{1}{u} \quad (B.2)$$

and Eq. (B.1) becomes

$$1 + a_1 u + a_2 u^2 + a_3 u^3 = 0 \quad (B.3)$$

Subsequently, Eq. (B.3) is again transformed, using

$$u = \frac{1 - v}{1 + v} \quad (B.4)$$

Thus

$$v = \frac{1 - u}{1 + u} \quad (B.5)$$
Therefore, if an inverse root of Eq. (B.1) is a complex number, then

\[ u_1 = a + ib \]  \hspace{1cm} (B.6)

and

\[ v = \frac{1 - a^2 - b^2 - 2ib}{1 + a^2 + b^2 + 2a} \]  \hspace{1cm} (B.7)

It may be concluded that if the modulus \( |u_1| > 1 \) (thus \( |\lambda_1| < 1 \)), then the corresponding root \( v_1 \) is in the negative half plane \( \text{Re}(v_1) < 0 \).

Thus, if conditions can be found for the roots of a polynomial which have parts in the negative half plane, these conditions can also be used to determine the limit of the roots of the characteristic equation.

Conditions for polynomials having roots with real negative parts can be found by use of Routh-Hurwitz criteria or Liénard-Chipart criteria (see Gantmacher(28)).

The transformation of Eq. (B.3) by use of Eq. (B.4) results in

\[ (1 - a_1 + a_2 - a_3)v^3 + (3 - a_1 - a_2 + 3a_3)v^2 \]

\[ + (3 + a_1 - a_2 - 3a_3)v + (1 + a_1 + a_2 + a_3) = 0 \]  \hspace{1cm} (B.8)

if we set

\[ (1 - a_1 + a_2 - a_3) = A_0 \]  \hspace{1cm} (B.9)

\[ (3 - a_1 - a_2 + 3a_3) = A_1 \]  \hspace{1cm} (B.10)

\[ (3 + a_1 - a_2 - 3a_3) = A_2 \]  \hspace{1cm} (B.11)

\[ (1 + a_1 + a_2 + a_3) = A_3 \]  \hspace{1cm} (B.12)
The roots of Eq. (B.8) have real negative parts if the following Routh-Hurwitz conditions are satisfied. Assuming that $A_o$ is positive, then

\[
\Delta_1 > 0 \quad \text{(B.13)}
\]

\[
\Delta_2 > 0 \quad \text{(B.14)}
\]

\[
\Delta_3 > 0 \quad \text{(B.15)}
\]

where

\[
\Delta_i = \begin{vmatrix}
A_1 & A_3 & A_5 & \cdots \\
A_o & A_2 & A_4 & \cdots \\
0 & A_1 & A_3 & \cdots \\
0 & A_o & A_2 & A_4 & \cdots \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
& & & & A_i
\end{vmatrix}
\quad (A_m = 0 \text{ for } m > n) \quad \text{(B.16)}
\]

and $n$ is the order of the polynomial. Note that $\Delta_n = A_n \Delta_{n-1}$; thus the inequality equation, Eq. (B.15), may be written as $\lambda_2 > 0$, $\lambda_3 > 0$. The conditions in Eqs. (B.13) through (B.15) represent the stability condition, and may be written as follows: if $|\lambda_i| \leq 1$

\[
1 - a_1 + a_2 - a_3 > 0 (A_0 > 0) \quad \text{(B.17)}
\]

then

\[
3 - a_1 - a_2 + 3a_3 > 0 \quad (A_1 > 0 \text{ or } \Delta_1 > 0) \quad \text{(B.18)}
\]

\[
1 - a_2 + a_1a_3 - a_3^2 > 0 \quad (\Delta_2 > 0) \quad \text{(B.19)}
\]
1 + a_1 + a_2 + a_3 > 0 \quad (A_3 > 0 \text{ or } \Delta_3 > 0) \quad (B.20)

and if the first term of the transformed equation, Eq. (B.8), is negative

1 - a_1 + a_2 - a_3 < 0 \quad (B.21)

then

3 - a_1 - a_2 + 3a_3 < 0 \quad (B.22)

1 - a_2 + a_1a_3 - a_3^2 < 0 \quad (B.23)

1 + a_1 + a_2 + a_3 < 0 \quad (B.24)

Using the Liénard-Chipart criteria\(^{(27)}\) rather than the Routh-Hurwitz conditions, and again assuming \(A_0 > 0\), the roots of Eq. (B.8) are in the negative half plane if the following conditions are met:

\[ A_n > 0, A_{n-2} > 0, \ldots; \Delta_1 > 0, \Delta_3 > 0, \ldots \quad (B.25) \]

\[ A_n > 0, A_{n-2} > 0, \ldots; \Delta_2 > 0, \Delta_4 > 0, \ldots \quad (B.26) \]

\[ A_n > 0, A_{n-1} > 0, A_{n-3} > 0, \ldots; \Delta_1 > 0, \Delta_3 > 0, \ldots \quad (B.27) \]

\[ A_n > 0, A_{n-1} > 0, A_{n-3} > 0, \ldots; \Delta_2 > 0, \Delta_4 > 0, \ldots \quad (B.28) \]

where for the third-power polynomial under consideration, \(n = 3\).

The Liénard-Chipart condition in Eq. (B.28) gives the following rather than Eq. (B.18):
\[ (3 + a_1 - a_2 - 3a_3) > 0 \quad (A_2 > 0) \]  \hspace{1cm} (B.29)

Or, in the case of the condition in Eq. (B.21), then the condition in Eq. (B.28) gives the following rather than Eq. (B.22):

\[ (3 + a_1 - a_2 - 3a_3) < 0 \quad (A_2 < 0) \]  \hspace{1cm} (B.30)
Appendix C

PROGRAM DESCRIPTION

INTRODUCTION

This appendix, which describes a FORTRAN IV program for computation of long-wave propagation in two dimensions, contains three parts. The first part is intended for those who are only interested in the results and who want to use the program the way a hydraulic model is used. The present capabilities of the program as given here and the data required for computation are described. On the basis of this data (or lack of it), boundary conditions are determined and computations performed. No detailed knowledge of programming is necessary, as standard forms are used to insert the data. However, the user of this program should have insight into hydraulic engineering, as realistic quantities for computation have to be selected, and realistic assumptions often have to be made if all conditions are not known. Also, knowledge of the basic computational process is required to select compatible computation quantities like grid size and time step.

The second part of this appendix describes the flow of the computation in more detail and is intended for those who want to know precisely how certain quantities are computed. Information for making changes in the computation and in the procedure is included. For example, if the standard data presentation is not satisfactory and the data have to be prescribed in plot form, those who want to change the present maximum size of the grid (approximately 32 by 60 for the IBM 7040/7044) will need to acquaint themselves with this part.

The third part gives the complete listing of the program, its subroutines in FORTRAN IV, and the coding forms.

PART I

Program Capabilities

The program in its present form can be used to approximate the flow in a certain region if the following conditions hold:
1. The long-wave equations represent the fluid flow.
2. Geographic boundaries of the area are fixed. (The program in its present form does not allow for flooding.)
3. Information about the bottom roughness is available.
4. Water depths are known with respect to a mean level or a reference plane.
5. Water levels at open boundaries (with unknown flow condition) are known as a function of time.
6. The Coriolis parameter can be considered constant over the whole area.

General Description

The computations are made by use of finite differences; consequently, a choice must be made concerning the size of these differences.

The area in which the flow is simulated is represented by a two-dimensional grid in a horizontal plane on the earth's surface, and the curvature of the earth's surface is neglected. The different properties of the body of water, such as the land-water boundaries, depth, and bottom roughness, are described on this grid system. Velocities and water levels are determined on the points of the grid as a function of time. These characteristics of the geographic area can be described in coding forms from which standard FORTRAN cards are made. The use of these forms are described in this part.

In addition to the geographical characteristics, system dimensions must be set. Information such as the size of the grid and the number of operations to be performed in time is needed.

Grid Selection

Values of the wave heights (water levels) and the flow are computed on a Cartesian grid system with equal distances between the points in x- and y-directions. The basic points in this grid system are the water levels ($\zeta$), which are designated $SE(N,M)$ and $SEP(N,M)$ in the program. The first value indicates a known water level, and the second value indicates the water level to be computed. The first subscript ($N$) refers to the grid point number along the Y-axis (in the direction of the V velocities). The second subscript ($M$) refers to the grid
point number in the x-direction (in the direction of the U velocities). For the 7040/7044 computer combination with a memory of 32,000, the maximum grid size is approximately 1800 points (N x M). For purposes of data representation, the dimension in the N-direction is limited to 32.

To set up a computation in a certain area, geographic information has to be transferred onto the grid points. The procedure can best be illustrated by a simple example, as shown in Fig. C.1. Consider a bay area for which computations are required. On two sides, the south and east sides, water levels are available as functions of time. This area is overlaid by a grid of proper grid size (Δs).

Fig. C.1—Example of grid for a bay area
In the computation procedure no reference to water level or velocity, whose subscript is zero, is allowed. As a consequence, certain rules must be applied to the lower bounds of the grid. Considering the lower bound of \( N \), if the first water level encountered is an open bound and water levels are given on this bound, this bound may be set \( N = 1 \). However, if the lower bound is a land-water boundary, then the first grid point should be \( N = 2 \). The same requirement is necessary in the \( M \)-direction.

A similar rule is valid for the maximum dimensions in \( N \) and \( M \), which are called \( N_{\text{MAX}} \) and \( M_{\text{MAX}} \) in the procedure. If the upper bound (water level with the highest subscript, either \( N \) or \( M \)) is a given water-level boundary, then the \( N \) or \( M \) number may be taken as \( N_{\text{MAX}} \) or \( M_{\text{MAX}} \), respectively. However, if the upper bound is a closed boundary, \( N_{\text{MAX}} \) or \( M_{\text{MAX}} \) is one number larger. In Fig. C.1 \( N_{\text{MAX}} = 9 \), while \( M_{\text{MAX}} = 13 \).

Either the metric or the foot-pound system may be used in the computations. However, certain limits are set on the values of the water levels and velocities in connection with the standard data representation of the computation. The maximum deviation in the water level or the velocity from the zero reference is 9.99, which is printed as ±999.

The space location of the velocity \( V \) (in \( y \)-direction), indicated by \( V(N,M) \) in the output, is located in the middle between the points \( N \) and \( N + 1 \), on \( X = M \). The space location of the velocity \( U \) (in \( x \)-direction), indicated by \( U(N,M) \), is located in the middle between the points \( M \) and \( M + 1 \), on \( Y = N \).

The space location of the water depths \( (H) \), indicated by \( (N,M) \), is actually on \( X = M + 1/2 \) and \( Y = N + 1/2 \). The Chezy coefficients \( C(N,M) \) are located on \( N,M \) as are the water levels. In filling out the forms and reading depth information from maps, these actual locations should be used.

**Data Preparation for Computation**

**General.** Here it is assumed that the (WAV) program is available in FORTRAN IV source statements on IBM cards. In this program, or better, "series of computation instructions and fixed data," data pertaining to the particular case must be inserted. To facilitate insertion
of the case instructions and data, special instructions are inserted in a group at the beginning of the program. * Data like water depth, bottom roughness, and water-level histories are inserted in groups at the end of the program.

Special coding sheets are prepared for the presentation of the case data and are presented in Part III. Each line of the coding sheet is then used to prepare a punched card. Instructions for filling out the coding sheets are given in Part II, and the proper location of the cards obtained from the coding forms in the main program is indicated in Part III.

**System Dimensions (Form I).** The main dimensions and characteristic values are set on the coding sheet. In filling out numbers on this form, integer values must be distinguished from floating-point values, as different types of arithmetical calculations are carried out with these values.

Floating-point values carry a decimal point, even if no decimal fractions are necessary. The integer values do not carry decimal points and are typically used for grid location, grid size, and time-step number. If integer values are to be inserted in the forms, the indicator (I) is used. The values for the form are

\[
\begin{align*}
NMAX &= (I) \text{ Maximum grid size in } N\text{-direction not to exceed 32, as determined in the discussion on grid selection.} \\
MMAX &= (I) \text{ Maximum grid size in } M\text{-direction not to exceed 99, and } NMAX \times NMAX \text{ not to exceed 2000.} \\
DIMENSION A(\cdot), B(\cdot), P(\cdot), Q(\cdot), R(\cdot), S(\cdot), F(\cdot) \\
\end{align*}
\]

Values of A,B,P,Q,R,S are velocity and water-level relation values, while F is a value representing earth rotational effects. The value of MMAX should be used in (I) for the maximal dimension of these seven parameters. The continuation of the above dimension statement contains conversion values KONVRT(\cdot) and NH(\cdot), both to be dimensioned (I) NMAX.

* A few cards must also be inserted in subroutines, which are discussed later.
The next cards contain so-called COMMON statements. Again, these dimension several parameters: \( SE(\cdot,\cdot) \), \( SEP(\cdot,\cdot) \), \( V(\cdot,\cdot) \), \( VP(\cdot,\cdot) \), \( U(\cdot,\cdot) \), \( UP(\cdot,\cdot) \). These parameters are the water levels and the velocities \( V \) and \( U \) for two time levels. All dimensions \( (I) \) are \( NMAX, MMAX \). Continuing the COMMON statement we have the Chezy terms \( C(\cdot,\cdot) \), dimensioned \( (I) \) with \( NMAX, MMAX \), like \( SE(\cdot,\cdot) \).

The next two parameters \( NBD(\cdot) \), \( MBD(\cdot) \) are computational control numbers which are generated by the program from data inserted in tables. These refer to the total number of sections which contain water in the \( N- \) and \( M- \) directions, respectively. The example in Fig. C.1 has an \( NBD \) dimension of 12, as two sections in the \( M- \) direction appear in \( N = 2 \), \( N = 6 - 8 \). The \( MBD \) dimension is 12, containing one section on lines \( M = 2, \ldots, 13 \). Both the \( NBD \) and \( MBD \) dimensions should be about one and a half times the maximum value of \( NMAX \) or \( MMAX \), whichever is the largest.

The \( M\)\( NBD(\cdot) \) and \( M\)\( MBD(\cdot) \) values refer to open boundaries, but filling out these dimensions will be deferred. The value \( H(\cdot,\cdot) \) refers to the water depth and is also dimensioned \( (I) NMAX, MMAX \).

The next subscripted parameters \( XLA(\cdot) \), \( XIB(\cdot) \), etc., refer to tables of water levels which are inserted in the computation as known information. Each value in these tables refers to a water level at a particular time, and a discussion follows later.

If, for example, 200 time steps are used in the computations (each time step consisting of the two operations of the multioperation method), 400 table values are to be given. In that case, the (minimum) dimension of \( XLA(\cdot) \) is 400. The program allows up to 11 tables to be used simultaneously. All tables should be dimensioned.

The next parameters are as follows:

\[
\text{ANGLAT} = \text{Latitude in degrees (and decimal fractions thereof) of the center of the area of computation. If the area is on the Southern Hemisphere, a negative angle should be used.}
\]

\[
\text{AL} = \text{Grid size, the distance between water-level points in \( N- \) and \( M- \) directions.}
\]
AG = Acceleration of gravity.
AT = Half time step in seconds and thus the time interval used for each of the two operations.
MAXST = Maximum number of time steps in which the computation will be performed (I). (Thus the total real time of computation is $2 \times AT \times MAXST$.) This value can be effectively used as the time control of the procedure in the initial check-out phase.
NI = Number of computations of the nonlinear water level in the continuity equations (I). Generally, NI is taken as 1, and numerical experiments indicate that (for the same computation time) a better approximation is obtained by decreasing the time step.
SEINV = Initial value of the water level of the whole water-level field.

The NÖBD values (I) indicate a computational control number for the open boundary. One value is obtained for each open boundary section on a line M, and NÖBD subscripts are sequenced up to the total number of boundaries on the M line.

The first number to be set on the right side of the equal sign is the M number of the open boundary. The second two numbers indicate the lower N number of the given water level, the third two numbers indicate the upper bound of the N number of the given water level, and the last number is an indication for the side of the bound in respect to the computation field. If the open bound is on the upper (or right) side of M, then a value of one is used; if the open bound is on the left, zero is used. All sections of the open bounds are indicated in sequence. All values of NÖBD(...) which are not needed should be omitted. In the example (Fig. C.1) only one NÖBD value exists, and the coding sheet contains the following information:

$NÖBD(1) = 13|06|08|1$  (upper bound)

The NÖBD values are set in a similar manner. In Fig. C.1

$NÖBD(1) = 1|09|10|0$  (lower bound)
If one of the corner points appears in the open bounds, they may be included in either the \texttt{N\textsc{obd}} value or the \texttt{M\textsc{obd}} value, or both, or excluded. Such corner points do not play any active part in determining boundaries for computation. If a diagonal line is used for an open bound, the water level points on this diagonal will be an open bound for both directions. Consequently, these points have to be included in the \texttt{N\textsc{obd}} and \texttt{M\textsc{obd}} tables.

The values \texttt{MIND\textsc{G}} and \texttt{NIND\textsc{G}} are connected with these tables. The first number (I) is the total number of \texttt{M\textsc{obd}} values plus one, and the second number is the total number of \texttt{N\textsc{obd}} values plus one. Thus, in the example

\begin{align*}
\text{MIND\textsc{G}} &= 2 \\
\text{NIND\textsc{G}} &= 2
\end{align*}

Returning now to the \texttt{COMMON} statement, the \texttt{MIND\textsc{G}} value (I) is used for the \texttt{M\textsc{obd}(...)} dimension, and the \texttt{NIND\textsc{G}} value (I) is used for the \texttt{N\textsc{obd}(...)} dimension.

The last parameters in this form are

\begin{align*}
\text{NSECT} &= \text{the value (I) of the dimension of NBD and MBD.} \\
\text{NCARD} &= \text{the number of entries in the table NCARD \geq 2 \times \text{MAXST}.}
\end{align*}

As an aid in the manipulation of instruction cards, sequence numbers can be used in the last columns.

\textbf{Open Boundaries (Form 2).} On this form each point on the open boundaries must be described as a function of time. The subscript (K) is used for the time indicator, where K represents the number of the time level and describes the total number of individual operations performed. At the start of the computation, K = 0. In the description, reference can be made to the table values XIA(K), etc., described in the section about system dimensions. Table value XIA(K), etc., refer to water levels at known points. In the listing of the program in this appendix, XIA(K) refers to the known water level at point 24,1 and XID(K) refers to the water level at 31,1. The intermediate points are described by a linear interpolation.

\textbf{Table Values (Form 3).} The table values of the water levels are
represented on Form 3. (The FORTRAN format used is 11F6.3.) As many forms may be used as are necessary. The sequence numbers can be placed in the last columns and are only an aid in data control. The total number of entries should equal the NCARD number in Form 1. If no table values are used, no values are filled in.

After the last line of table values (if any), one line must be filled out with a title that will be used in data presentation. Sixty-six characters can be used.

**Water-Level Field Representation (Form 4).** On this form all those points of the grid system where water levels are to be computed should be indicated by L(I) in the right column of each box. Locations where water levels are to be given should not be included. If no water levels are to be computed on a line, one zero (I) should be entered. In this manner it is assured that the first line is not skipped during the preparation of the instructions. A blank card may also be used. (Note that this is always the case for the first and last line.) Sequence numbers may be used in the last column, and MMAX entries are to be made.

From this presentation, the water-land boundaries will be determined.

**Water Depth (Form 5).** Form 5a is used for the water depth for N values in the range of 1 through 16, starting with \( M = 1 \) to MMAX. (The FORTRAN format used is 16F4.1.) More than one form should be used if required. (M numbers may be filled out in the last column for data control.) Use floating point values, with a decimal point in the third place of each box. Thus, values up to 99.9 can be used, with 0.1 accuracy. The total number of entries is MMAX for each range; for the last entry in each range of \( M = \text{MMAX} \), a 0.0 must be used in one of the locations. (This information is never used.)

If \( 16 < \text{MMAX} \leq 32 \), Form 5b is used in addition to Form 5a. As the water depths are determined on a different grid system than the water levels of Form 4, the general appearance of Form 5 differs from Form 4.

**Chezy Coefficients (Form 5).** Forms 5a and 5b are also used for the bottom-roughness coefficients. The FORTRAN format used is 16F4.0.) Form 5a is used for N values in the range of 1 through 16, and Form 5b
is used for N values in the range of 16 through 32. The total number of entries is MMAX for each range of N. Floating point values with a decimal point are used in the last column of each box. Thus C = 999 is the maximum value that can be used.

If there is an open boundary point on M = MMAX, the bottom-roughness coefficients are filled out. Otherwise a zero must be entered in one of the locations to insure that the proper number (MMAX) of instruction cards are available.

Print (After Last Entry of Form 5). In addition to the Chezy coefficients of the last entries of Form 5, the indication for printout is given at the time steps of the given numbers. This data output will be printed. Use integer values, filling in as much to the right as possible in each of the four locations. (The FORTRAN format used is 16I4.) Printout numbers are to be sequenced with increasing values, with 16 numbers on one line. Five lines (cards) should be filled out. To insure that five instruction cards are available, each blank card should carry a unity value of one in the first block. A maximum of 80 printout instructions are allowed for this standardized program. This printout request produces the average values of the water levels and velocities over the second operation of the time step requested.

Order of Program Cards

After all coding sheets are punched, the COMMON cards of Form 1 should be duplicated five times.

The cards of Forms 1 and 2 are inserted in the beginning of the deck. Then one set of the COMMON statement cards are inserted in the beginning of the subroutines (FIND, DEPTH, DIVE, KURIH, and CHEZY). After $ENTRY, all cards of Forms 3, 4, and 5 are added in the order that they were written.

Part III gives the listing of a problem, and the locations of data provided by the forms are marked.
PART II

Outline of the Program

It will be seen that the use of the computation formulas is only a small part of the program. The parameters in the program follow the nomenclature described previously. The main portion of the statements concerns controlling the computations and finding the special conditions along the boundaries.

In the beginning of the program the dimensions and open-bound values are set, and the initial conditions are established. Thereafter, the following four operations are performed:

1. Computation of new values of the U velocity and the water level.
2. Computation of the new value of the V velocity.
3. Computation of the new values of the V velocity and the water level.
4. Computation of the new value of the U velocity.

The first two are operations of the first half time step of the procedure, and the last two are operations of the second half time step. The beginnings of each of these four sections are indicated by comment cards, and the printout instructions can be found between the second and third operation. Five subroutines handle all data input:

KURIH Handles the reading of all table values of the open boundaries into storage.
DIVE Reads in the locations of water-level computation.
FIND Processes the DIVE data and the MBD and NBD values and produces two tables (NBD and MBD) which are used to control the computation.
DEPTH Reads the water depth.
CHEZY Reads all Chezy coefficients, which represent an average value for the bottom roughness.

Control of the Computations

The starting and stopping of the successive computations over the
rows and columns are controlled by the NBD and MBD tables which are set up in the FIND routine from the water levels in the field information and the description of the open boundary (MBD and NBD values). Values in these tables contain up to eight digits. If we assign letters to the digits, the significance can be described in an easy manner. For example, a table value of the NBD table is SR|NO|MF|LU. This indicates that on row N = NO, the lower value of the M values is MF, and the upper value is LU. The first two letters (SR) give information about the type of boundary; the first letter describes the lower bound, and the second letter describes the upper bound. If S = 0, then a land-water boundary is described on this row at the lower bound (minimum M value); if S = 1, then an open boundary is described at the lower bound (thus, a boundary where a water level is given). Also, if R = 0, the upper bound is a land-water bound; if R = 1, the upper bound is an open bound.

In this program, only land-water bounds and water bounds described by a water level are programmed. If necessary, other boundary conditions can be controlled by another number. For example, if a discharge is given, the boundary could be described by R = 2 or S = 2.

In each table the number of NBD and MBD values equals the number of separate sections that exist in N- and M-directions, respectively. Each table value is numbered and is called for in sequence. These tables are presented before the start of the actual computation.

Computation of UP and SEP on the Rows N

This section refers to the implicit computation of the mentioned values. The NBD table is used for control, and the NBD number controls one section from bound to bound on a row N. In breaking down the NBD number into its components, as described in a previous section, we first establish the type of lower bound. If the lower bound is a land-water bound, the velocity at this bound is zero, and the starting R and S are zero; if the lower bound is an open-water bound, these values are given as functions of the given water level. The type of bound is determined by an IF statement operating on NSRCH. Once this is established, A, P, Q and B, R, S are computed in succession in a DO loop until the upper
bound (L) is found. Exit from this loop is made after A. P, Q by testing the M number against the upper bound IF (M.EQ.L). Next, the conditions of the land-water and open bounds are handled by IF statements again operating on NSRCH. In the returning sweeps (DO loop), the SEP and UP are found.

It will be seen that changes in numerical procedure for the convective-inertia term are possible. The $\sigma$ controls $u(\partial u/\partial x)$, which is centered in space for $\sigma = 0.5$. At the open bounds at either end, this term contains only the part of $\partial u/\partial x$ in the field. By using $\sigma$ values of one (1.) or zero (0.), these off-centered spatial derivatives may or may not be introduced. As presented, this convective-inertia term is omitted in the last field of open boundaries, as numerical experiments indicated a more stable computation.

The IF statements operating on TEMP10 and TEMP11 control the term $v(\partial u/\partial y)$ along the bounds. In the listing, the term is omitted along all bounds. If it is desired that this term be used, then the spatial derivative can be obtained (off-centered) by extrapolation. For example, in the middle row

$$\text{TEMP10} = 2.*U(N,M) + U(NN,M)$$

The IF statement operating on IT is used in the iteration of the water level in the nonlinear term of the continuity equation. It replaces the water level of the lower time level with the water level computed in the previous cycle.

**Computation of VP on the Column N**

This part represents the explicit operation on the V velocity. Control is by the MBD table, and table values are broken down in the column value and lower and upper bounds.

The actual computation has three sections: The first section computes all values on the middle of the column, the second computes the upper-bound values, and the third computes the lower-bound values.

The term $v(\partial v/\partial y)$ is controlled by a BETA. When BETA = 0.5, it represents a centered spatial derivative. Using zero (0.) or one (1.)
in the computation of boundary velocities, this term may or may not be included. As presented in the program, the term is omitted at the open bounds. The spatial derivative \( \partial v / \partial y \) is computed by TEMP4, and the velocity \( V \) of this convective-inertia term is taken completely implicitly. Control of \( u(\partial v / \partial x) \) is again made by extrapolation of TEMP10 and TEMP11 in the manner described previously.

Print Instructions

Print instructions are entered after each half time step and after a test showing whether or not the time-step number (NST) is equal to NPRINT(IP), which is one of the numbers for which time-step print-out is made.

At the end of this section all values of the higher time level (SEP, UP, VP) are inserted in the locations of the lower time levels (SE, U, V), and the ISTEP number directs the flow of the computation to the first or second half time-step operation.

Computation of VP and SEP on Column N

This computation follows exactly the order described for the computation of UP and SEP on the rows N, only in the other direction.

Computation of UP

This computation follows the procedure described for the computation of VP on column N, only in the other direction.
PART III

5746, WAV, L 4100, 5, 80, C
SIP JOE MAP, NODECK
S1AFTC WAV NODECK
C THE RAND CORPORATION, SANTA MONICA, CALIFORNIA. JAN J. LEENDERTSE.
C PROGRAM FOR THE COMPUTATION OF LONG WAVE WAVES.
C
C SET DIMENSIONS OF THE SYSTEM

FORM 1

NMAX=31
MMAX=55
DIMENSION A(55), R(55), P(55), Q(55), S(55), F(55),
1, KONVRT(31), NH(31), T(12), NPRINT(80)
COMMON SE(31,55), SEP(31,55), VP(31,55), V(31,55), U(31,55), UP(31,55),
1, IC(31,55), NH(70), QH(70), MODI(3), MOD(3), H(31,55),
2, XI(320), XIR(320), XIG(320), XIC(320), XID(320), XIE(320), XIF(320),
3, XI(320), XIR(320), XIG(320), XIC(320), XID(320), XIE(320), XIF(320)
ANGLAI = 51.917
AL=400.
AG = 9.81
AT=100.
MAXST=2
NI=1
SEINV=-.70
MODI(1) = 124310
MOD(2) = 5529311
MODI(1) = 126280
MOD(2) = 3101551
MINDL = 3
MINDU = 3
NSEC = 70
NCARC = 320

FORM 2

SET OPEN ROUND AS FUNCTIONS OF TABLEVALUES (XIA(K), ETC.)
OR AS FUNCTIONS OF HALFTIME STEP NUMBER (K).

GO TO: 87
87 CONTINUE
SEP(24,1)=XIA(K)
SEP(25,1)=(XIF(K)+6.XIA(K))/7.
SEP(26,1)=(2.XIF(K)+5.XIA(K))/7.
SEP(27,1)=(3.XIF(K)+4.XIA(K))/7.
SEP(28,1)=(4.XIF(K)+3.XIA(K))/7.
SEP(29,1)=(5.XIF(K)+XIA(K)*2.)/7.
SEP(30,1)=(6.*XIF(K)+XIA(K))/7.
SEP(1,76)=XID(K)
SEP(1,77)=XID(K)
SEP(1,78)=XID(K)
DO 90 M=1,23
XM=M
SEP(31,4)=XIF(K)*(XM-1.)*(XIB(K)-XIE(K))/22.5
90 CONTINUE
DO 91 M=24,55
XM=M
SEP(31,4)=XIG(K)+(46.-XM)*(XIB(K)-XIC(K))/22.5
91 CONTINUE
SEP(29,55)=SEP(31,55)
SEP(30,55)=SEP(31,55)
IF (ISTEP.EQ.1) GO TO 96
GO TO 301

CONTINUE

FF= 3.1415927* SIN(ANGLAT*3.1415927/180./1/21600.
CALL KURID(NCARD)
RFAD(5,4)(TITL(J),J=1,12)

FORMAT(12A6)

1 FORMAT(11H1,12A6)
NST = 0
C1 = AT*AC/AL
C2 = AT/AL
C3 = AT/4.
C4 = 8.*AT*AC
DO 6 N=1,NMAX
DO 6 M=1,MMAX
UP(N,M)=0.0
UP(N,M)=0.0
V(N,M)=0.
SE(N,M)=0.0
SEP(N,M)=0.0
U(N,M) = 0.
C(N,M) = 0.
H(N,M) = 0.
F(N) = FF
CALL DIVN(NMAX,MMAX)
CALL FIND(MIND,NIND,MMAX,NMAX,MINDO,NINDO,NSECTION)
CALL DEPTH(NMAX,MMAX)
CALL CHEZY(Y(NMAX,MMAX)
READ(5,25) (VPRINT(N,N=1,16)
READ(5,25) (NPINIT(N),N=17,32)
READ(5,25) (NPINIT(N),N=33,43)
READ(5,25) (NPINIT(N),N=49,64)
READ(5,25) (NPINIT(N),N=65,80)

25 FORMAT(16I4)
NUM = 1

2 IF (NUM.EQ.NIND) GO TO 3
NSRCH =NBD(NUM)/1000000
N =NBD(NUM)/100000 - NSRCH*100
MF =NBD(NUM)/100 -NSRCH*10000 - N*100
L =NBD(NUM) - NSRCH*1000000 - N*10000 -MF*100
NN = N - 1
K = MF
DO 2 W = K,L
SEP(N,M) = SEINV
2 SE(N,M) = SEINV
NUM = NUM + 1
GO TO 7

3 CONTINUE

NA=1

5 IF (NA.EQ.MIND) GO TO 31

M =MORD(NA)/100000
NBOT =MORD(NA)/1000 -M*100
NTOP =MORD(NA)/100 -M*10000 - NBOT*100
DO 32 N=NBR,NTOP
SEP(N,M) = SEINV
32 SEIN,M) = SEINV
NA=NA+1
GO TO 5
31 AA=1
33 IF(NA.EQ.AINCC) GC TC 34
   NA=NCPC(NA)/1CCCCC
   MLEF =NCPC(NA)/1CC -N*1CC
   MICH =NCPC(NA)/1C -N*1CCCC -MLEF*1CC
   CC 35 M=MICH,MLF
   SEIN =SEINV
35 SEIN=SEINV
36 NA=NA+1
37 GC TC 32
34 CONTINUE

WRITE INITIAL VALUES

WRITE(6,1) (TITL(J),J=1,12)
WRITE(6,12)
12 FORMAT(/1X,24F12.1)
   CC 9 P=1,PMAX
   CC 4 C N=1,NMAX
   NF(N)=F(N)*1C. +.C1
9 WRITE(6,6CC1) P,NF(N),N=1,NMAX)
   WRITE(6,1) (TITL(J),J=1,12)
   WRITE(6,15)
15 FORMAT(/1X,2A+C VALUES UNTIL NEXT PRINTOUT)
10 FORMAT(1H ,I2,1X,32F4.4)
   CC 16 JA=1,PMAX
16 WRITE(6,1C) JA,(C(N,J),N=1,NMAX)

C
C ISTEP=2
C GC TC 5CC
C
C COMPLETE UP ANB SEP C1N RCW A ( FIRST HALF Timestep)
C
88 ISTEP=1
89 NST =NST +1
90 K=2*NST-1
   IF(NST.GT.MAXST) CALL EXIT
C
C SET CLEN PCLNC
C GC TC 85
96 NLM =1
100 IF(NLM.EQ.AINCC) GC TC 19C
   NSRCH =NBC(NLM)/1CCCCC
   N =NBC(NLM)/1CCC - NSRCH*1CC
   MF =NBC(NLM)/1CC -NSRCH*1CCCC -N*1CC
   L =NBC(NLM) - NSRCH*1CCCCC -N*1CCCC -MF*1CC
   MFF =MF-1
   NNN=N+1
69 NN = N -1
70 IT=1
   M(MFF)= C.C
   S(MFF)= C.C
   CAMP=CD
   IF(NSRCH.LE.1C.OR.NSRCH.GT.111) GC TC 99
   PFF =MF-1
   TEMPIC=U(ANN,MFF)
   IF(TEMP10.EQ.C.) TEMPIC= U(ANN,MFF)
   TEMP11=U(ANN,MFF)
IF(TMP11.EQ.C.)  TEMP11= L(M,M,F,F)
ALPHA=1.
R(MF)=C1/(1. +C2*(L(N,MF)- U(N,MF)))
S(MF)=(U(N,MF)+ C1* SEP(N,MF))
1 - U(N,MF)*SQRT(L(N,MF)**2+(((V(N,MF)+ V(NN,MF))**2)/16.))/
2*(SE(N,MFF)+ SE(N,MF)+ H(N,MFF)+ H(NN,MFF))*(C(N,MFF)+ C(N,F,F))
3*(2)**2)*C4 + (V(N,MF) + V(NN,MF))* -25*(AT* F(N) -(1.-GAMMA)* C2 *
4*(TEMPIC -L(N,MFF)) -GAMMA*C2*(U(N,MFF) - TEMP11 ))/)
5(1. + C2*(L(N,MF)- U(N,MF)))* (1.-ALPHA))
95 CONTINUE

K= MF
IC1 CC IC2 M = K,M
MMP = M+1
TEMP5=SE(N,M)
IF( IT. GT. 1) TEMP9=SEP(N,M)
TEMP1 = SE(NNN,M)
IF( TEMP1.EQ.C.) TEMP1 = 2.*SE(N,M)-SE(NNN,M)
TEMP2 = SE(NNP,M)
IF( TEMP2.EQ.C.) TEMP2 = 2.*SF(N,M) - SE(NNN,M)
TEMP3 = SE(N,MMP)
IF( TEMP3.EQ.C.) TEMP3 = 2.*SE(N,M) - SE(N,NM)
IF( IT. GT. 1) TEMP3=SEP(N,MMP)
IF( IT. GT. 1) TEMP3=SEP(N,MMP)
(TEMP5=SEP(N,MMP)
IF( TEMP4.EQ.C.) TEMP4 = 2.*SE(N,M) - SE(N,MMP)
IF( IT. GT. 1) TEMP4=SEP(N,MMP)
IF( IT. GT. 1. ANB TEMP4.EQ.C.) TEMP4= 2.*SEP(N,M)-SEP(N,MMP)
A(N) = SE(N,M) - .5*C2*(H(N,M)+ H(N,NM)+SEP(N,M)+ TEMP1 )
1/2*(V(N,M) + .5*C2*(H(N,NM)+H(N,M)+SE(N,M)+TEMP2 )- V(N,NM))
B(N) = .5*C2*(H(N,M)+ H(N,NM)+ IFM9 + TEMPO + TEMP3 )/( 1. + .5*C2 *
1/(F(N,M)+ T(N,NM)) + SEP(N,MMP) + TEMP4 + TEMP9 ) + S(MM)
1/(1. + .5*C2*(H(N,NM) + H(N,NM) + TEMP4 + TEMP9 ) + R(MM))
IFM.EQ.LC) GC TC IC2
GAMMA = C5
TEMP1C=U(NNN,M)
IF(TEMP1C.EQ.C.) TEMP1C = L(NN,M)
TEMP11=U(NN,M)
IF(TEMP11.EQ.C.) TEMP11 = L(NN,M)
TEMP6 = AT*F(N) -(1.-GAMMA)*C2*(TEMPIC -U(N,M)) - GAMMA*C2 *
1/(L(N,M) - TEMP11)
TEMP6 = .25*TEMP6
E(N) = L(N,M) + TEMP6 *(V(N,M)+ V(N,NM)+ V(NNN,M)+ V(NN,MMP))
1 - L(N,M)*SQRT(L(N,NM)**2+(((V(N,M)+ V(NN,MMP)+ V(NNN,M)+ V(NNN,MMP))
2)**2)/16.)+)/((SE(N,M)+ SF(N,NM)+ H(N,M) + T(N,M))**2*(C(N,M)+ C(N,NM)
3*M)**2)**2)*C4
ALPHA = C5
TEMP1 = L+ C2*(AG*P(N)+(1.-ALPHA)*(L(N,NM)- L(N,M)) +
1ALPHA*L(N,M)- L(N,M)))
R(M)= CI/TEMP1
S(M)=E(M)+ C1*C(M)/TEMP1
IC2 CONTINUE
L(N,M)=C.
IF(NSRC.EQ.1. CR. NSRC.EQ.11) GC TC IC3
CC TC IC4
IC3 CONTINUE
IF (NLW.EQ.1) GO TO 35C

**SRCH =**SRCh(NLW) / ICC

**M =**MUC(NLW) / ICC - MSrCh*1CC

**N =**MUC(NLW) / ICC - MSrCh*1CC

**L =**MUC(NLW) / ICC - MSrCh*1CC

**M =**M - 1

**N =**N + 1

**FF =**FF - 1

**C =**C - 1

IF (MSrCh.LT.1.0 .OR. MSrCh.GT.1.1) GO TO 319

TEMP1 = V(NFF, MM)

IF (TEMP1 .LE. C) TEMP1 = V(NFF, MM)

TEMP11 = V(NFF, MM)

ELsE STOP 1

IF (NFF) = 1 / (1.0 + C2*(V(NFF, MM) - V(NFF, MM))*(1.0 - BETA))

V(NFF) = V(NFF, MM) + C1*SEP(NFF, MM)

- (V(NFF, MM) - V(NFF, MM))**2 + (((L(NF, MM) + L(NF, MM))**2) / 16.))/

((SF(NFF, MM) + SF(NF, MM) + t(NFF, MM) + (NF, MM) + (NF, MM))**2)*((C(NFF, MM) + C(NF, MM))**2)

+ C2*P**2*(1.0 - BETA) * C2*(TEMP1 = V(NFF, MM))

+ C2*P**2*(1.0 - BETA) * C2*(TEMP11 - V(NFF, MM))

CONTINUE

**NF =** NF

**i =** i + 1

IF (NC.GE.1) GO TO 30C

NF = NF - 1

**NN =** NN + 1

**NCUG = (1.0 / NN)
\[ TEMP1 = SE(n, m) \]
\[ IF(TEMP1.EQ.C.) TEMP1 = 2.*SE(n, m) - SE(n, m) \]
\[ TEMP2 = SE(n, m) \]
\[ IF(TEMP2.EQ.C.) TEMP2 = 2.*SE(n, m) - SE(n, m) \]
\[ TEMP3 = SE(n, m) \]
\[ IF(TEMP3.EQ.C.) TEMP3 = 2.*SE(n, m) - SE(n, m) \]
\[ IF(IX.GT.1) TEMP3 = SEP(2*temp3) - SEP(2*temp3) \]
\[ TEMP4 = SE(n, m) \]
\[ IF(TEMP4.EQ.C.) TEMP4 = 2.*SE(n, m) - SE(n, m) \]
\[ IF(IX.GT.1) TEMP4 = SEP(2*temp4) - SEP(2*temp4) \]
\[ AN = SE(n, m) - 5*C2*(H(n, m) + H(n, m)) + SEP(2*temp1) \]
\[ 1P = + C2*(H(n, m) + H(n, m)) + TEMP2 + SEP(2*temp2) \]
\[ LN = SE(n, m) + H(n, m) + TEMP9 + TEMP3 \]
\[ LN = (A(N) + 5*C2*(H(n, m) + H(n, m)) + TEMP4 + TEMP9) + 5*S(N) \]
\[ IF(LX.EQ.L) GC TC 3C2 \]
\[ CELTA = C.*5 \]
\[ IF(TEMP11.EQ.C.) TEMP11 = V(L, m) \]
\[ IF(TEMP11.EQ.C.) TEMP11 = V(L, m) \]
\[ TEMP6 = ATF(N) + (1.-DELTA)*C2*(TEMP11-TEMP11) \]
\[ TEMP7 = 0.25* TEMP6 \]
\[ IF(NM = V(L, m) - TEMP6 ) \]
\[ LN = V(L, m) + L(N, m) + L(N, m) + L(N, m) + L(N, m) \]
\[ LN = 5*L(N, m) + SCRT(V(L, m) + 2*(L(N, m) + L(N, m)) + L(N, m) + L(N, m) + L(N, m)) \]
\[ IF(MSRC.EQ.1) CR.MSRC.EQ.11) GC TC 3C7 \]
\[ GC TC 3C5 \]
\[ CONTINUE \]
\[ LLL = L+1 \]
\[ V(L, m) = C.C \]
\[ IF(MSRC.EQ.1) CR.MSRC.EQ.11) GC TC 3C7 \]
\[ CONTINUE \]
\[ TEMP11 = V(L, m) \]
\[ IF(TEMP11.EQ.C.) TEMP11 = V(L, m) \]
\[ LLL = L+1 \]
\[ LL = L-1 \]
\[ CELTA = C.5 \]
\[ IF(TEMP11.EQ.C.) TEMP11 = V(L, m) \]
\[ IF(TEMP11.EQ.C.) TEMP11 = V(L, m) \]
\[ TEMP11 = TEMP11 \]
\[ CONTINUE \]
\[ 3C2 CONTINUE \]
\[ LLL = L+1 \]
\[ V(L, m) = C.C \]
\[ IF(MSRC.EQ.1) CR.MSRC.EQ.11) GC TC 3C7 \]
\[ CONTINUE \]
\[ TEMP11 = V(L, m) \]
\[ IF(TEMP11.EQ.C.) TEMP11 = V(L, m) \]
\[ LLL = L+1 \]
\[ LL = L-1 \]
\[ CELTA = C.5 \]
\[ IF(TEMP11.EQ.C.) TEMP11 = V(L, m) \]
\[ IF(TEMP11.EQ.C.) TEMP11 = V(L, m) \]
\[ TEMP11 = TEMP11 \]
\[ CONTINUE \]
\[ 3C5 CONTINUE \]
\[ 3C2 CONTINUE \]
\[ 3C5 CONTINUE \]
\[ 3C7 CONTINUE \]
AA = AA - 1
SEP(N,M) = -P(N)*VP(N,M) + C(N)
VP(NA,M) = -R(NA)*SEP(N,M) + S(NA)

3C6

AA = AA - 1
IT = IT + 1
IF (IT <= NI) GC TC 3C2
NLM = NLM + 1
GC TC 3C1

C

COMPLETE UP CA RC* N (SECOND HALF TIMESTEP)

C

35C

NLM = 1

34C

IF (NLM >= NC*(NI+1)) GC TC 4C2
NSRCH = NC(NLM)/ICCCCCC
AA = NC(NLM)/ICCCCCC - NSRCH*1CC
FF = NC(NLM)/ICCCCCC - NSRCH*1CC + AA*1CC
L = NC(NLM)*ICCCCCC - NSRCH*ICCCCCC - AA*ICCCCCC - FF*1CC

AA = AA - 1

3A1 = AA + 1
LL = LL - 1
LLL = L + 1

4C4 M = MF, LL

* * *

ALPHA = C.5

TEMP4 = C2*(!1. - ALPHA)*(L(N,M) - U(N,M)) + ALPHA*(L(N,M) - U(N,M))

TEMP1 = L(N,M)*2 + (((V(N,M) + V(N,M))*V(N,M) + V(N,M))*1.6)

TEMP2 = (SEP(N,M) + SEP(N,M) + (N,M) + U(N,M) + V(N,M))

1)**2

TEMP3 = 1. + C4*SCR(TEMP1)/TEMP2 + TEMPP4

TEMP3 = 1./TEMP3

CAMMA = 0.5

TEMP1C = L(NA,M)

IF (TEMP1C >= EC.C) TEMP1C = U(NA,M)

TEMP1 - L(N,M)

IF (TEMPL - EC.C) TEMPL = L(NA,M)

TEMP1 = AT(F(N) - (1. - GAMMA) * C2*(TEMP1C - L(N,M))

1 - CAMMA*C2*(L(N,M) - TEMPL1)

TEMP1 = .25*TEMP1

4C4

LP(N,M) = TEMP3*

1 - L(N,M)*TEMP1*(VP(N,M) + VP(N,M) + VP(N,M) + VP(N,M) + VP(N,M))

2 - C1*(SE(N,M) - SE(N,M))

IF (NSRCH*EC.L + + NC*NSRCH*EC.L) GC TC 4C5

GC TC 4C6

4C5

TEMP1C = L(NA,L)

IF (TEMP1C >= EC.C) TEMP1C = U(NA,L)

TEMP1 = L(N,L)

IF (TEMPL - EC.C) TEMPL = L(NA,L)

ALPHA = C.

TEMP4 = C2*ALPHA*(L(N,L) - L(N,L))

TEMP1 = L(N,L) + 2 + (((V(N,L) + V(N,L))*1.6)

TEMP2 = (SEP(N,L) + SEP(N,L) + H(N,L) + S(N,L))*C(N,L) + C(N,L) + C(N,L))

1)**2

TEMP3 = 1. + C4*SCR(TEMP1)/TEMP2 + TEMPP4

TEMP3 = 1./TEMP3

CAMMA = C.5

TEMP1 = .25*(AT(F(N) - (1. - GAMMA) * C2*(TEMP1O - L(N,L)) - GAMMA*C2*
LP(N,L) = TEMP3*(L(N,L)+TEMP1*(VP(N,L)+VP(N,N,L)))

1-C1*(SE(N,L)-SE(N,L))

4C6 IF(NSRCH.EQ.1.GE.11) GC TC 407

4C7 TEMP1 = U(N,N,FF)

IF (TEMP1.EQ.1.) TEMP1 = U(N,N,FF)

IF (TEMP1.EQ.1.) TEMP11 = U(N,N,FF)

ALPHA = 1.

TEMP4 = C2*(1.-ALPHA)*(U(N,MF)-U(N,FF))

TEMP1 = U(N,MF)*2+((V(N,MF)+V(N,MF))*2)/16.

TEMP2 = (SEP(N,MF)+SEP(N,MF)+SEP(N,MF)+SEP(N,MF))*C(N,MF)+C(N,MF)

1)**2

TEMP3 = 1. + C4*SQRT(TEMP1)/TEMP2 + TEMP4

4C8 CONTINUE

NLM = NLM + 1

4C2 CONTINUE

CC TL 5CC

END

$IPFTC KURIH

SUBROUTINE KURIH(NCARD)

COMMON SE(31,55),SEP(31,55),V(31,55),VP(31,55),U(31,55),UP(31,55),

LC(31,55),NBD(70),MBD(70),MBOC(3),NBO(3),H(31,55),

2X1G(320),X1H(320),X1C(320),X1D(320),X1E(320),X1F(320),

3X1G(320),X1H(320),X1I(320),X1J(320),X1K(320)

DO 3 K = 1, NCARD

WRITE(5,7) XI(K), X1B(K), X1C(K), X1D(K), X1E(K), X1F(K),

1X1G(K), X1H(K), X1I(K), X1J(K), X1K(K)

3 CONTINUE

WRITE(6,9)

WRITE(6,11)

DO 8 K = 1, NCARD

WRITE(6,12) K, XI(K), X1B(K), X1C(K), X1D(K), X1E(K), X1F(K),

1X1G(K), X1H(K), X1I(K), X1J(K), X1K(K)

RETURN

7 FORMAT(I1, F6.2)

9 FORMAT(I1, I12X, 35HWATER LEVELS AT STATIONS A THROUGH K)

11 FORMAT(I10, I10, I10, I10, I10, I10, I10, I10)

12 FORMAT(I10, I10, I10, I10, I10, I10, I10, I10)

END
SUBROUTINE FIND(MIND, NIND, MMAX, MMAX, MINDU, NINDO, NSECT)
LOGICAL START
COMMON SE(31, 55), SEP(31, 55), V(31, 55), VP(31, 55), U(31, 55), UP(31, 55),
LC(31, 55), NMD(70), MBD(70), MOBD(3), NORD(3), HI(31, 55),
XIA(320), XIM(320), XIC(320), XIH(320), XIE(320), XIF(320),
XIG(320), XIH(320), XIJ(320), XIK(320)
C0 1 J = 1, NSECT
NAD(J) = 0
*HD(J) = 0
MIND = 1
NIND = 1
GO TO 2
= ? MMAX
START = .TRUE.
GO TO 3
= ? MMAX
IF(.NOT.START) GO TO 4
IF(H(N,M) .EQ. 0.) GO TO 3
NAD(MIND) = M*100 + NAD(NIND)
START = .FALSE.
GO TO 3
IF(H(N,M) .NE. 0.) GO TO 5
NAD(MIND) = M - 1 + NAD(NIND) + 1000*N
GO TO 6
IF(M.NE. MMAX) GO TO 3
NAD(NIND) = M + NAD(MIND) + 1000*N
NIND = NIND + 1
START = .TRUE.
CONTINUE
2 CONTINUE
GO 12 M = 2, MMAX
START = .TRUE.
GO 13 M = 2, MMAX
IF(.NOT.START) GO TO 14
IF(H(N,M) .EQ. 0.) GO TO 13
MHD(MIND) = M*100 + MBD(MIND)
START = .FALSE.
GO TO 13
IF(H(N,M) .NE. 0.) GO TO 15
MHD(MIND) = M - 1 + MBD(MIND) + 1000*N
GO TO 16
IF(N.NE. MMAX) GO TO 13
MBD(MIND) = M + MHD(MIND) + 1000*N
NIND = MIND + 1
START = .TRUE.
CONTINUE
12 CONTINUE
NUM = 1
100 IF(NUM .EQ. NIND) GO TO 300
N = NAD(NUM) / 10000
MF = NAD(NUM) / 100 - N*100
L = NAD(NUM) - N*10000 - MF*100
MFLE = MF - 1
LRIG = L + 1
NA = 1
200 IF(NA .EQ. NINDC) GO TO 210
M = MBD(NA) / 100000
NAD1 = MBD(NA) / 1000 - M*100
NTP = MBD(NA) / 10 - M*10000 - NHOT*100
-152-

\[
\begin{align*}
\text{NBERN} &= \text{MOBD(NA) - M*100000 - NBOT*1000 - NTOP*10} \\
\text{IF}((\text{IN.GE.NBOT}) \text{AND} (\text{IN.LE.NTOP}) \text{AND} (\text{MFLEF.EQ.M})) \quad \text{NBD(NUM)} = \\
\text{INBD(NUM) + 1000000} \\
\text{IF}((\text{IN.GE.NBOT}) \text{AND} (\text{IN.LE.NTOP}) \text{AND} (\text{LRIG.EQ.M})) \quad \text{NBD(NUM)} = \\
\text{INBD(NUM) + 1000000} \\
\text{NA} &= \text{NA} + 1 \\
\text{GO TO 200}
\end{align*}
\]

\[
\begin{align*}
210 
\text{NUM} &= \text{NUM} + 1 \\
\text{GO TO 100}
\end{align*}
\]

\[
\begin{align*}
300 \text{ CONTINUE} \\
\text{NUM} &= 1 \\
1 \cdot 1 
\text{IF}(\text{NUM.EQ.MIND}) \quad \text{GO TO 301} \\
\text{M} &= \text{MBD(NUM)/10000} \\
\text{NF} &= \text{MBD(NUM)/100} - \text{M*100} \\
\text{L} &= \text{MBD(NUM) - M*10000 - NF*100} \\
\text{NFBOT} &= \text{NF} - 1 \\
\text{LTOP} &= \text{L} + 1 \\
\text{NA} &= 1
\end{align*}
\]

\[
\begin{align*}
201 \text{ IF}(\text{NA.EQ.NINDO}) \quad \text{GO TO 211} \\
\text{N} &= \text{NBD(INA)/100000} \\
\text{MLEF} &= \text{NBD(INA)/1000} - \text{N*100} \\
\text{MRIG} &= \text{NBD(INA)/10} - \text{N*10000 - MLEF*100} \\
\text{MBER} &= \text{NBD(INA) - N*100000 - MLEF*1000 - MRIG*10} \\
\text{IF}(\text{M.GE.MLEF AND M.LE.MRIG AND NFBOT.EQ.N}) \quad \text{MBD(NUM)} = \text{MBD(NUM)} \\
\text{1} + \quad \text{1000000} \\
\text{IF}(\text{M.GE.MLEF AND M.LE.MRIG AND LTOP.EQ.N}) \quad \text{MBD(NUM)} = \text{MBD(NUM)} \\
\text{1} + \quad \text{1000000} \\
\text{NA} &= \text{NA} + 1 \\
\text{GO TO 201}
\end{align*}
\]

\[
\begin{align*}
211 
\text{NUM} &= \text{NUM} + 1 \\
\text{GO TO 201}
\end{align*}
\]

\[
\begin{align*}
301 \text{ CONTINUE} \\
\text{WRITE(6,20)} \\
\text{DO 22 J = 1*NSECT} \\
\text{WRITE(6,21) J,NBD(J),MBD(J)} \\
22 \text{ CONTINUE} \\
\text{RETURN}
\end{align*}
\]

\[
\begin{align*}
20 \text{ FORMAT(IH1,3X,3HNUM,6X,3HNBD,7X,3HMBD)} \\
21 \text{ FORMAT(IH,2X,14,2X,19,1X,19)} \\
\text{FND}
\end{align*}
\]
SUBROUTINE DIVE(NMAX,MMAX)
COMMON SF(31,55),SEP(31,55),V(31,55),VP(31,55),U(31,55),UP(31,55),
IC(31,55),NHD(70),MBD(70),MORD(3),NORD(3),H(31,55),
XIA(320),XIA(320),XIC(320),XID(320),XIF(320),XIF(320),
XI6(320),XI6(320),XI7(320),XI7(320),XI7(320),XI7(320),
TIME(31),NHD(6)
NJ = 6
DO 1 N = 1,NMAX
WRITE (6,6) (NHD(N),N = 1,NMAX)
IF (N .LT. 1) GO TO 1
WRITE (4,4) M,NHD(N),N = 1,NMAX
DO 1 M = 1,NMAX
H(N,M) = FLOAT(NHD(N))
RETURN
1 FORMAT (3I12)
2 FORMAT (1H1,12,5X,3I12)
3 FORMAT (1H1,10X,2I12,2I12)
4 FORMAT (1H1,2I12)
5 FORMAT (35X)

SUBROUTINE DEPTH(NMAX,MMAX)
COMMON SF(31,55),SEP(31,55),V(31,55),VP(31,55),U(31,55),UP(31,55),
IC(31,55),NHD(70),MBD(70),MORD(3),NORD(3),H(31,55),
XIA(320),XIA(320),XIC(320),XID(320),XIF(320),XIF(320),
XI6(320),XI6(320),XI7(320),XI7(320),XI7(320),XI7(320),
NC = 16
IF (NMAX .LT. 16) NO = NMAX
DO 10 M = 1,NMAX
READ (5,3) (H(N,M),N = 1,NO)
CONTINUE
10 CONTINUE
IF (NMAX .LE. 16) GO TO 12
DO 11 M = 1,NMAX
READ (5,3) (H(N,M),N = 17,NMAX)
CONTINUE
11 CONTINUE
12 CONTINUE
RETURN
6 FORMAT (16F4.1)
7 FORMAT (35X)
SUBROUTINE CHEZ (NMAX, MMAX)
COMMON SE(31,55), SEP(31,55), V(31,55), VP(31,55), U(31,55), UP(31,55),
IC(31,55), NBD(70), MB(70), MOBD(3), NOBD(3), H(31,55),
XIA(320), XIB(320), XIC(320), XID(320), XIE(320), XIF(320),
XIG(320), XIH(320), XIJ(320), XI(320), XIK(320)
NO = 16
IF (NMAX .LT. 16) GO TO NMAX
DO 10 M = 1, MMAX
READ (5, 3) (C(N,M), N = 1, NO)
10 CONTINUE
IF (NMAX .LE. 16) GO TO 12
DO 11 M = 1, MMAX
READ (5, 3) (C(N,M), N = 17, NMAX)
11 CONTINUE
12 CONTINUE
RETURN
FORMAT (16F4.0)
END
ENTRY

-0.700 -0.700 -0.700 -0.700 -0.700
-0.703 -0.711 -0.705 -0.705 -0.701
-0.705 -0.721 -0.711 -0.711 -0.702
-0.705 -0.731 -0.716 -0.719 -0.703
-0.706 -0.742 -0.722 -0.726 -0.703
-0.707 -0.749 -0.727 -0.734 -0.704
-0.708 -0.755 -0.733 -0.742 -0.704
-0.709 -0.762 -0.739 -0.749 -0.704
-0.710 -0.768 -0.744 -0.756 -0.705
-0.711 -0.770 -0.751 -0.765 -0.705
-0.711 -0.772 -0.757 -0.774 -0.705
-0.711 -0.775 -0.763 -0.781 -0.704
-0.710 -0.779 -0.770 -0.788 -0.703
-0.708 -0.778 -0.774 -0.795 -0.700
-0.708 -0.778 -0.779 -0.802 -0.698

0.886 0.752 0.658 0.356 0.922
0.922 0.791 0.694 0.400 0.958
0.956 0.826 0.727 0.441 0.990
0.991 0.861 0.761 0.483 1.023
1.025 0.895 0.796 0.523 1.056

FORM 3

FORINGDLIFT
<table>
<thead>
<tr>
<th>Depth</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4</td>
</tr>
<tr>
<td>1.6</td>
</tr>
<tr>
<td>1.8</td>
</tr>
<tr>
<td>2.0</td>
</tr>
<tr>
<td>2.2</td>
</tr>
<tr>
<td>2.4</td>
</tr>
<tr>
<td>2.6</td>
</tr>
<tr>
<td>2.8</td>
</tr>
<tr>
<td>3.0</td>
</tr>
<tr>
<td>3.2</td>
</tr>
<tr>
<td>3.4</td>
</tr>
<tr>
<td>3.6</td>
</tr>
<tr>
<td>3.8</td>
</tr>
<tr>
<td>4.0</td>
</tr>
<tr>
<td>4.2</td>
</tr>
<tr>
<td>4.4</td>
</tr>
<tr>
<td>4.6</td>
</tr>
<tr>
<td>4.8</td>
</tr>
<tr>
<td>5.0</td>
</tr>
<tr>
<td>5.2</td>
</tr>
<tr>
<td>5.4</td>
</tr>
<tr>
<td>5.6</td>
</tr>
<tr>
<td>5.8</td>
</tr>
<tr>
<td>6.0</td>
</tr>
<tr>
<td>6.2</td>
</tr>
<tr>
<td>6.4</td>
</tr>
<tr>
<td>6.6</td>
</tr>
<tr>
<td>6.8</td>
</tr>
<tr>
<td>7.0</td>
</tr>
<tr>
<td>7.2</td>
</tr>
<tr>
<td>7.4</td>
</tr>
<tr>
<td>7.6</td>
</tr>
<tr>
<td>7.8</td>
</tr>
<tr>
<td>8.0</td>
</tr>
</tbody>
</table>

**FORM 5b**
<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>32</th>
<th>36</th>
<th>40</th>
<th>44</th>
<th>48</th>
<th>52</th>
<th>56</th>
<th>60</th>
<th>64</th>
<th>68</th>
<th>72</th>
<th>76</th>
<th>80</th>
<th>84</th>
<th>88</th>
</tr>
</thead>
<tbody>
<tr>
<td>92</td>
<td>46</td>
<td>100</td>
<td>104</td>
<td>108</td>
<td>112</td>
<td>116</td>
<td>120</td>
<td>124</td>
<td>128</td>
<td>132</td>
<td>136</td>
<td>140</td>
<td>144</td>
<td>148</td>
<td>152</td>
<td></td>
</tr>
<tr>
<td>155</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SIPSY  
SIPSY  
ENDJOB  
TOTAL NUMBER OF CARDS IN YOUR INPUT DECK
REFERENCES


