



AN IMPROVED 2-D FINITE-DIFFERENCE CIRCULATION MODEL FOR TIDE- AND WIND-INDUCED FLOWS

FERNANDO J. CAVIGLIA¹ and WALTER C. DRAGANI²

Facultad de Ciencias Naturales y Museo, UNLP, Casilla de Correo 45 (1900), La Plata, Argentina, and ²Departamento de Oceanografía, Servicio de Hidrografía Naval, Av. Montes de Oca 2124 (1271), Buenos Aires, Argentina
(e-mail: dragani@oceanar.mil.ar)

(Received 23 August 1995; revised 20 March 1996)

Abstract—An RM/FORTRAN computer program for the numerical solution of tide- and wind-driven currents is presented. The model solves the nonlinear two-dimensional hydrodynamic equations by a finite differencing method on a constructed zigzag boundary. The alternating-direction implicit (ADI) algorithm is implemented, and the model requires only the solution of block-tridiagonal systems of linear algebraic equations. In contrast to the conventional vertically integrated model, the bed shear stress is determined from the bottom currents, which are computed using the linear three-dimensional hydrodynamic equations and the Galerkin technique in the vertical. The program includes an option to run the model using the conventional representation of the bed shear stress based on the depth mean currents. Applications under idealized and field conditions are described also. Copyright © 1996 Elsevier Science Ltd

Key Words: Numerical hydrodynamic model, Tide- and wind-driven circulation, Alternating-direction implicit algorithm, Galerkin technique, Continuous vertical current profile, FORTRAN.

INTRODUCTION

The prediction of tide- and wind-driven circulation is of the utmost importance in ocean and coastal engineering. Tide level, tidal currents, and wind-generated currents usually are studied using well-established, two-dimensional, vertically-integrated models. Since these models are vertically integrated, the bed stress is computed from the depth mean current. However, this is not physically correct, and a more realistic parameterization is obtained by using the bottom current. Davies (1988) presented a model which consists in solving a nonlinear two-dimensional, vertically integrated model where the bottom stress is no longer related to the average depth currents but is computed from the bottom currents. This transformation can be performed if one supposes that the current velocity is composed of the depth mean current and its deviation from the depth mean value. Then, the linear three-dimensional hydrodynamic equations are split into two systems where only one of them is depth dependent. The application of the Galerkin method to the depth-dependent system permits a continuous vertical representation of the magnitude of the current deviation from its depth mean value at the calculation points of the two-dimensional model. The re-

lation between the vertically integrated model and the three-dimensional one is given through the bottom shear stress, which does not differ significantly in magnitude from that obtained with a full three-dimensional model (Davies, 1988). It should be noted that the model is linear only in the sense of neglecting nonlinear terms involving current structure.

This paper describes the computer implementation of a two-dimensional model in which the bottom stress is calculated from the bottom currents. The computational requirement of the model is not significantly different from that of the conventional two-dimensional model and is small compared with the requirement of a full three-dimensional model. To run the model efficiently on a desktop computer some simplifications were made which also are described here.

HYDRODYNAMIC MODEL

Vertically integrated equations

Mathematical characterization of tide- and wind-driven currents requires the simultaneous solution of the dynamic equations of motion and the unsteady continuity equations. If it is assumed that

vertical accelerations are negligible, pressure is hydrostatic over the depth, and the fluid density is homogeneous, the two-dimensional depth-averaged nonlinear equations can be written as follows:

$$\frac{\partial \eta}{\partial t} + \frac{\partial}{\partial x}(h + \eta)U + \frac{\partial}{\partial y}(h + \eta)V = 0 \quad (1)$$

$$\begin{aligned} & \frac{\partial U}{\partial t} + U \frac{\partial U}{\partial x} + V \frac{\partial U}{\partial y} - fV \\ &= -g \frac{\partial \eta}{\partial x} - \frac{1}{\rho} \frac{\partial P_a}{\partial x} + \frac{(\tau_{sx} - \tau_{bx})}{\rho(h + \eta)} + A_H \nabla^2 U \end{aligned} \quad (2)$$

$$\begin{aligned} & \frac{\partial V}{\partial t} + U \frac{\partial V}{\partial x} + V \frac{\partial V}{\partial y} + fU \\ &= -g \frac{\partial \eta}{\partial y} - \frac{1}{\rho} \frac{\partial P_a}{\partial y} + \frac{(\tau_{sy} - \tau_{by})}{\rho(h + \eta)} + A_H \nabla^2 V \end{aligned} \quad (3)$$

where t is the time, U , V are mean vertically averaged velocities in the x and y directions respectively, g the acceleration due to gravity, η the free surface elevation above mean water level, h the depth below mean water level, f the Coriolis parameter, ρ the water density, P_a the atmospheric pressure, τ_{sx} and τ_{sy} the x and y components of surface wind stress, τ_{bx} and τ_{by} the x and y components of bottom shear stress, A_H the horizontal eddy viscosity coefficient and ∇^2 the horizontal Laplacian operator.

The expressions for the x and y components of surface wind stress are given by

$$\tau_{sx} = \rho_a C_{da} |V_{10}| V_{10x} \quad \tau_{sy} = \rho_a C_{da} |V_{10}| V_{10y} \quad (4)$$

where $|V_{10}|$ is the wind velocity at 10 m above water level while V_{10x} and V_{10y} are the x and y components of wind velocity, ρ_a the density of air, and C_{da} the wind drag coefficient that in accordance with the WAMDI Group (1988) can be calculated as follows:

$$C_{da} = \begin{cases} 0.0012875 & |V_{10}| < 7.5 \text{ m/s} \\ 0.0008 + 0.000065|V_{10}| & |V_{10}| \geq 7.5 \text{ m/s} \end{cases} \quad (5)$$

In the traditional two-dimensional vertically integrated model, the bottom stress is determined from the depth mean current using either a linear or quadratic form of bed stress. In a three-dimensional model the bed stress is computed from the bottom currents. In order to enhance the two-dimensional model, the bed stress is determined from the bottom currents using a quadratic law (see Owen, 1980; Davies, 1988; Davies and Jones, 1993):

$$\tau_{bx} = \rho k u_b \sqrt{u_b^2 + v_b^2} \quad \tau_{by} = \rho k v_b \sqrt{u_b^2 + v_b^2} \quad (6)$$

where u_b and v_b are the bottom currents in the x - and y directions respectively, and k a dimensionless coefficient of bottom friction.

The bottom currents and bed stress are computed without formulating a full three-dimensional model. The method developed by Davies (1988), and Davies and Jones (1993) is used. This method considers a point model to determine the vertical current profiles and hence the bed current and bed stress.

Vertical profiles of the currents

In many tide- and wind-driven circulation problems of practical importance, the vertical velocity component and its gradient are considerably smaller than horizontal velocities and gradients, and can be neglected. Thus, flows induced by wind and tidal forces can be assumed to approximate horizontal flows. Neglecting nonlinear terms and the horizontal shear, the hydrodynamic equation of motion at a point can be written in complex notation as

$$\begin{aligned} & \frac{\partial w}{\partial t} + i f w = -g \left(\frac{\partial \eta}{\partial x} + i \frac{\partial \eta}{\partial y} \right) \\ & - \frac{1}{\rho} \left(\frac{\partial P_a}{\partial x} + i \frac{\partial P_a}{\partial y} \right) + \frac{1}{d^2} \frac{\partial}{\partial \sigma} \left(A_v \frac{\partial w}{\partial \sigma} \right) \end{aligned} \quad (7)$$

where $w = u + i v$, u and v are the velocities in the x and y directions respectively, $d = h + \eta$, A_v is the coefficient of vertical eddy viscosity, and $\sigma = (\eta - z)/(h + \eta)$, a convenient nondimensional vertical coordinate, z being the vertical coordinate measured from the mean water level and positive upwards.

Following the development of Davies (1988), and Davies and Jones (1993), the current velocity is written as $w = W + w'$, where $W = U + i V$ is the depth-averaged current and $w' = u' + i v'$ the current deviation from its depth mean value. Then from Equation (7) follows

$$\begin{aligned} & \frac{\partial w}{\partial t} + i f w = -g \left(\frac{\partial \eta}{\partial x} + i \frac{\partial \eta}{\partial y} \right) \\ & - \frac{1}{\rho} \left(\frac{\partial P_a}{\partial x} + i \frac{\partial P_a}{\partial y} \right) + \frac{(T_s - T_b)}{\rho d} \end{aligned} \quad (8)$$

$$\frac{\partial w'}{\partial t} + i f w' = \frac{1}{d^2} \frac{\partial}{\partial \sigma} \left(A_v \frac{\partial w'}{\partial \sigma} \right) - \frac{(T_s - T_b)}{\rho d} \quad (9)$$

where $T_s = \tau_{sx} + i \tau_{sy}$ and $T_b = \tau_{bx} + i \tau_{by}$.

Equation (8) gives the depth mean current while Equation (9) defines the current deviation from its depth mean value and gives no net transport.

In order to obtain the current profile and its associated bed stress, Equation (9) is solved using the Galerkin method in the vertical using a finite element representation. In this regard, the current velocity w' is expanded in terms of m complex coefficients $A_r(t)$ and depth-dependent real, continuous functions $g_r(\sigma)$ (the basis functions) as follows:

$$w' = \sum_{r=1}^m A_r(t) g_r(\sigma) \quad (10)$$

The coefficients $A_r(t)$ are obtained by substituting Equation (10) into Equation (9). In general, there will be a residual error R associated with this substitution because the basis functions (10) are not the exact solution of the Equation (9). The Galerkin method consists of multiplying the residual R by the basis functions and integrating over the region $\eta \geq z \geq -h$ or $0 \leq \sigma \leq 1$. Integration of Equation (9) yields

$$\int_0^1 R g_k(\sigma) d\sigma = \sum_{r=1}^m \frac{\partial A_r}{\partial t} \int_0^1 g_r g_k d\sigma + i f \sum_{r=1}^m A_r \int_0^1 g_r g_k d\sigma + \frac{1}{d^2} \sum_{r=1}^m A_r \int_0^1 \frac{\partial}{\partial \sigma} (A_v \frac{\partial g_r}{\partial \sigma}) g_k d\sigma + \frac{(T_s - T_b)}{\rho d} \int_0^1 g_k d\sigma = 0 \quad (11)$$

with $k = 1, 2, 3, \dots, m$. Integrating the term involving A_v by parts gives

$$\begin{aligned} \sum_{r=1}^m \frac{\partial A_r}{\partial t} \int_0^1 g_r g_k d\sigma + i f \sum_{r=1}^m A_r \int_0^1 g_r g_k d\sigma \\ + \frac{A_v}{d^2} \sum_{r=1}^m A_r \int_0^1 \frac{\partial g_r}{\partial \sigma} \frac{\partial g_k}{\partial \sigma} d\sigma - \frac{T_s}{\rho d} (g_k(0) - \int_0^1 g_k d\sigma) + \frac{T_b}{\rho d} (g_k(1) - \int_0^1 g_k d\sigma) = 0 \quad (12) \end{aligned}$$

where A_v was considered independent of the depth coordinate, although still varying with time, and the boundary conditions given by

$$-\rho \left[A_v \frac{\partial w}{\partial \sigma} \right]_{\sigma=0} = T_s \quad -\rho \left[A_v \frac{\partial w}{\partial \sigma} \right]_{\sigma=1} = T_b \quad (13)$$

were used. It should be noted that A_v can depend on the depth coordinate in the general Galerkin method. For the particular example in which A_v is independent of the vertical coordinate the basis functions are eigenfunctions of the eddy viscosity profile (for more details of the integration procedure see Davies, 1980a, 1980b).

Many authors have used the Galerkin method with a basis set of cosine functions (Davies and Furnes, 1980; Davies and Heaps, 1980; Pearce and Cooper, 1981; Davies, 1986):

$$g_r = \cos(\alpha_r \sigma) \quad (14)$$

Since the boundary conditions in Equation (13) are included explicitly in Equation (12), it is not necessary to choose α_r such that these conditions are satisfied by Equation (14) (Davies, 1980a, 1980b). Then, a suitable choice of α_r is

$$\alpha_r = (r - 1)\pi \quad (15)$$

Due to this choice for α_r , the basis functions (14) produce a mode ($r = 1$) which is constant through the vertical. However, by definition w' cannot contain a mean flow component which implies that A_1 must be zero.

Taking into account Equations (14) and (15), Equation (12) becomes

$$\frac{dA_r}{dt} = - \left[i f + \frac{A_v \alpha_r^2}{d^2} \right] A_r + \frac{2}{\rho d} (T_s - T_b \cos \alpha_r) \quad (16)$$

with $r = 2, 3, \dots, m$.

In order to solve Equation (16) it is necessary to specify the form of A_v . Hess (1976) has presented an expression for A_v as a function of tidal currents. From this result, it is assumed that

$$A_v = \Gamma (U^2 + V^2)^{1/2} (h + \eta) \quad (17)$$

where Γ is a dimensionless coefficient of order 0.005–0.0005. The simple eddy viscosity formulation given by Equation (17) appears to be supported by more complex turbulence energy models (Davies, 1991).

By solving Equation (16), the current velocity w_b' at the sea bed is determined from

$$w_b' = W + \sum_{r=2}^m A_r \cos \alpha_r \quad (18)$$

and the bed stress computed from Equation (6).

Davies (1988) showed that using three or four terms in Equation (10), the computed bed stress was equivalent to that determined with a full three-dimensional model. To minimize the computer memory requires the coefficients A_r to be held, therefore only three terms were retained in this paper.

NUMERICAL MODEL

Finite difference equations

The numerical scheme describes the current velocities, the undisturbed water depth, and the free surface elevation at different grid points as shown in Figure 1.

The alternating-direction implicit (ADI) algorithm was used to solve the partial differential equations (see Douglas and Gunn, 1964). The ADI method is known widely because it avoids the matrix inversion problem related to the implicit schemes. However, the method is not unconditionally stable when it is applied to the complete set of Equations (1)–(3) in an area with real bathymetry and irregular boundaries. Thus, a limitation on the time-step is necessary to ensure numerical stability. Calculations usually become unstable for Courant numbers, C_r , greater than approximately 3–5 (Weare, 1979),

$$C_r = \Delta t \sqrt{gh_{\max}} \sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}} \tag{19}$$

where Δt is the finite-difference time-step, Δx and Δy are the horizontal grid spacings in the x and y directions respectively (Fig. 1), and $h_{\max}$ the maximum water depth of the integrational area.

Before replacing the derivatives with an appropriate set of finite-difference analogs, the continuity equation (1) is used to eliminate the terms $\frac{\partial U}{\partial x}$ from the momentum equation (2) for the x component, and $\frac{\partial V}{\partial y}$ from the momentum equation (3) for the y component:

$$\begin{aligned} &\frac{\partial U}{\partial t} - \frac{U}{d} \frac{\partial \eta}{\partial t} - \frac{U^2}{d} \frac{\partial h}{\partial x} - \frac{UV}{d} \frac{\partial d}{\partial y} \\ &- U \frac{\partial V}{\partial y} + V \frac{\partial U}{\partial y} - fV = - \left[g - \frac{U^2}{d} \right] \frac{\partial \eta}{\partial x} \\ &- \frac{1}{\rho} \frac{\partial P_\alpha}{\partial x} + \frac{(\tau_{sx} - \tau_{bx})}{\rho(h + \eta)} + A_H \nabla^2 U \end{aligned} \tag{20}$$

$$\begin{aligned} &\frac{\partial V}{\partial t} - \frac{V}{d} \frac{\partial \eta}{\partial t} - \frac{V^2}{d} \frac{\partial h}{\partial y} - \frac{UV}{d} \frac{\partial d}{\partial x} \\ &- V \frac{\partial U}{\partial x} + U \frac{\partial V}{\partial x} - fU = - \left[g - \frac{V^2}{d} \right] \frac{\partial \eta}{\partial y} \\ &- \frac{1}{\rho} \frac{\partial P_\alpha}{\partial y} + \frac{(\tau_{sy} - \tau_{by})}{\rho(h + \eta)} + A_H \nabla^2 V \end{aligned} \tag{21}$$

The finite-difference equations corresponding to differential Equations (1), (20), and (21) are expressed in an ADI form, with the time-step split into two halves. For the first time-step, from time $\eta \Delta t$ to $(n + \frac{1}{2}) \Delta t$, the terms involving U and η of Equations (1) and (20) are expressed implicitly in the x direction while the terms involving V are expressed explicitly. Once the values of U and η are computed, Equation (3) is expressed implicitly in the y direction and V is determined, at the same time-step, using the values of U and η previously computed. At the next step, from time $(n + \frac{1}{2}) \Delta t$ to $(n + 1) \Delta t$, the procedure is reversed so that V and η are computed from Equations (1) and (21) which are expressed implicitly in the y direction with all terms involving U expressed explicitly. Using the computed values of V and η , the new values of U are determined from Equation (2) expressed implicitly in the x direction.

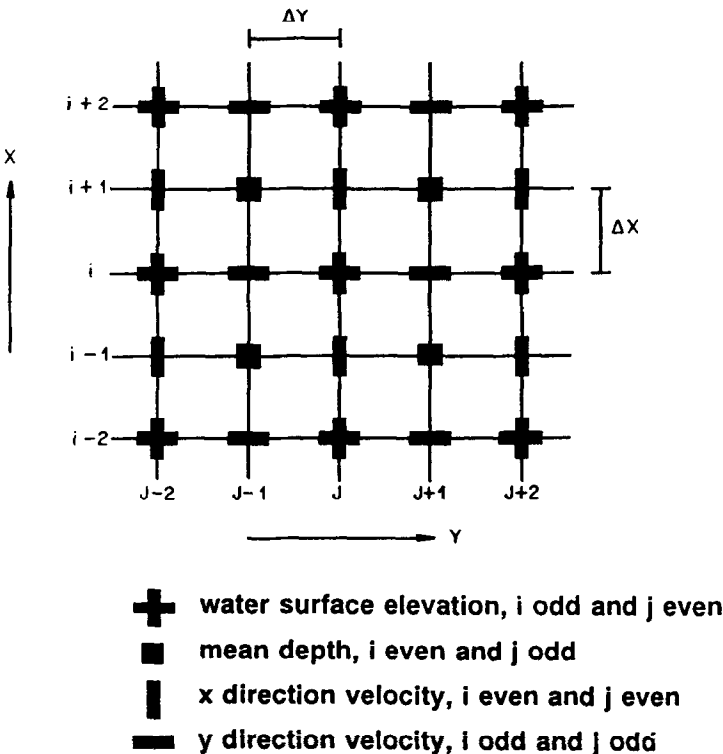


Figure 1. Computing grid.

The equations are discretized in such a way that the continuity equation is evaluated at nodes corresponding to η values and the momentum equations for the x and y components are evaluated at nodes corresponding to U and V values, respectively.

Consider an $M \times N$ matrix, with M and N being the number of grid points in the x and y directions. At the interior nodes (x, y) and denoted by the subscripts i, j , where $x = (i - 1)\Delta x$ and $y = (j - 1)\Delta y$, the Equations (1) and (20) are expressed implicitly in the x direction using the following notation:

$$\begin{aligned}
 f_{i,j}^n &= f(i\Delta x, j\Delta y, n\Delta t) \\
 f^{n+\mu/2} &= \mu f_{i,j}^{n+1/2} + (1 - \mu)f_{i,j}^n \\
 \bar{f}^x &= \frac{1}{2}(f_{i+1,j} + f_{i-1,j}) \\
 \bar{f} &= \frac{1}{4}(f_{i+1,j+1} + f_{i-1,j+1} + f_{i-1,j-1} + f_{i+1,j-1}) \\
 \text{as} \\
 \eta^{n+1/2} &= \eta^n - \frac{\Delta t}{4\Delta x} [(\bar{h}^y + \bar{\eta}^x)U]_x^{n+\mu/2} \\
 &\quad - \frac{\Delta t}{4\Delta y} [(\bar{h}^x + \bar{\eta}^y)V]_y^n \text{ at } i, j \quad (22) \\
 U^{n+1/2} &= U^n + \frac{U^{n+\mu/2}}{(\bar{h}^y + (\bar{\eta}^x)^n)} \\
 &\quad \times \left[[(\bar{\eta}^x)^{n+1/2} - (\bar{\eta}^x)^n] + \frac{\Delta t}{4\Delta x} U^{n+\mu/2} (\bar{h}^x)_x^y \right. \\
 &\quad \left. + \frac{\Delta t}{4\Delta y} (\bar{V})^n (h_y + (\bar{\eta}^y)_y^n) \right] + \frac{\Delta t}{4\Delta y} [U^{n+\mu/2} (\bar{V}^x)_y^n \\
 &\quad - [(\bar{V})(\bar{U}^y)_y]^n] \\
 &\quad + \frac{\Delta t}{2} f(\bar{V})^n - \frac{\Delta t}{4\Delta x} \left[g - \frac{(U^2)^{n+\mu/2}}{(\bar{h}^y + (\bar{\eta}^x)^n)} \right] \eta_x^{n+\mu/2} \\
 &\quad - \frac{1}{\rho} \frac{\Delta t}{4\Delta x} (P_a)_x \\
 &\quad + \frac{(\tau_{sx} - \tau_{bx}^{n+\mu/2})}{\rho(\bar{h}^y + (\bar{\eta}^x)^n)} + \frac{\Delta t}{8\Delta x^2} A_H U_{xx}^n \\
 &\quad + \frac{\Delta t}{8\Delta y^2} A_H U_{yy}^n \text{ at } i + 1, j \quad (23)
 \end{aligned}$$

The approximation is central in space and the coefficient μ takes any value between 0 and 1. A value of $\mu > \frac{1}{2}$ permits theoretically the use of higher Courant numbers.

It should be noted that Equations (22) and (23) form a system of nonlinear algebraic equations. In order to avoid iterative procedures, which are time consuming, and noting that the simulated flow constitutes a slow transient phenomena, a linearization procedure has been adopted to reduce these equations to a system of linear algebraic equations from which a direct solution can be obtained. Each term Q involving a nonlinear function of $U^{n+1/2}$ and/or $\eta^{n+1/2}$ is linearized by means of a local Taylor expansion,

$$\begin{aligned}
 Q^{n+1} &= Q^n + B^n (U^{n+1} - U^n) \\
 &\quad + C^n (\eta^{n+1} - \eta^n) + 0(\Delta t^2) \quad (24)
 \end{aligned}$$

where $B^n = [\frac{\partial Q}{\partial U}]^n$ and $C^n = [\frac{\partial Q}{\partial \eta}]^n$. Furthermore, at each time-step, the total water depth is expressed as $d^n = h + \eta^{n-1/2}$

Applying the linearization procedure to Equations (22) and (23), we obtain the following system of linear algebraic equations

$$\eta_{i,j}^{n+1/2} + G_{1,i,j}^n U_{i+1,j}^{n+1/2} + G_{2,i,j}^n U_{i-1,j}^{n+1/2} = G_{3,i,j}^n \quad (25)$$

$$G_{4,i+1,j}^n U_{i+1,j}^{n+1/2} + G_{5,i+1,j}^n \eta_{i,j}^{n+1/2} + G_{6,i+1,j}^n \eta_{i+2,j}^{n+1/2} = G_{7,i+1,j}^n \quad (26)$$

At each time-step, Equations (25) and (26) lead to a tridiagonal system of linear algebraic equations which is solved by a forward sweep and a backward substitution. The algorithm is simply Gaussian elimination.

Once the values of $U^{n+1/2}$ and $\eta^{n+1/2}$ are computed throughout the computational domain, the values of $V^{n+1/2}$ are determined from Equation (3), which is expressed implicitly in the y direction,

$$\begin{aligned}
 V^{n+1/2} &= V^n - \frac{\Delta t}{4\Delta x} \bar{U}^{n+\mu/2} (\bar{V}^x)_x^n \\
 &\quad - \frac{\Delta t}{4\Delta y} [V(\bar{V}^y)_y]^{n+\mu/2} - \frac{\Delta t}{2} f \bar{U}^{n+\mu/2} \\
 &\quad - \frac{\Delta t}{4\Delta y} g \eta_y^{n+\mu/2} - \frac{1}{\rho} \frac{\Delta t}{4\Delta y} (P_a)_y + \frac{(\tau_{sy} - \tau_{by}^{n+\mu/2})}{\rho(\bar{h}^x + (\bar{\eta}^y)^n)} \\
 &\quad + \frac{\Delta t}{8\Delta x^2} A_H V_{xx}^n + \frac{\Delta t}{8\Delta y^2} A_H V_{yy}^{n+\mu/2} \text{ at } i, j. \quad (27)
 \end{aligned}$$

Applying the linearization procedure to each term involving a nonlinear function of $V^{n+1/2}$, we obtain the following tridiagonal system of linear algebraic equations

$$F_{1,i,j}^n V_{i,j-2}^{n+1/2} + F_{2,i,j}^n V_{i,j-1}^{n+1/2} + F_{3,i,j}^n V_{i,j+1}^{n+1/2} = F_{4,i,j}^n \quad (28)$$

Once the values of $V^{n+1/2}$ are computed, the time-step is incremented from $(n + \frac{1}{2})\Delta t$ to $(n + 1)\Delta t$

and the procedure is reversed. Equations (1), (21), and (2) are written in discrete form as

$$\begin{aligned} \eta^{n+1} &= \eta^{n+1/2} - \frac{\Delta t}{4\Delta x} [(\bar{h}^y + \bar{\eta}^x)U]_x^{n+1/2} \\ &\quad - \frac{\Delta t}{4\Delta y} [(\bar{h}^x + \bar{\eta}^y)V]_y^{n+(\mu+1)/2} \text{ at } i, j \quad (29) \\ V^{n+1} &= V^{n+1/2} + \frac{V^{n+(\mu+1)/2}}{(\bar{h}^x + (\bar{\eta}^y)^{n+1/2})} \\ &\quad \times \left[[(\bar{\eta}^y)^{n+1} - (\bar{\eta}^y)^{n+1/2}] + \frac{\Delta t}{4\Delta y} V^{n+(\mu+1)/2} (\bar{h}^y)_y^x \right. \\ &\quad \left. + \frac{\Delta t}{4\Delta x} (\bar{U})^{n+1/2} (h_x + (\bar{\eta}^x)_x^{y_{n+1/2}}) \right] \\ &\quad + \frac{\Delta t}{4\Delta x} [V^{n+(\mu+1)/2} (\bar{U}^y)_x^{n+1/2} - [(\bar{U})(\bar{V}^x)_x]^{n+1/2}] \\ &\quad + \frac{\Delta t}{2} f (\bar{U})^{n+1/2} \\ &\quad - \frac{\Delta t}{4\Delta y} \left[g - \frac{(V^2)^{n+(\mu+1)/2}}{(\bar{h}^x + (\bar{\eta}^y)^{n+1/2})} \right] \eta_y^{n+(\mu+1)/2} - \frac{1}{\rho} \frac{\Delta t}{4\Delta y} (P_a)_y \\ &\quad + \frac{(\tau_{sy} - \tau_{by}^{n+(\mu+1)/2})}{\rho(\bar{h}^x + (\bar{\eta}^y)^{n+1/2})} + \frac{\Delta t}{8\Delta x^2} A_H V_{xx}^{n+1/2} \\ &\quad + \frac{\Delta t}{8\Delta y^2} A_H V_{yy}^{n+1/2} \text{ at } i, j + 1 \quad (30) \end{aligned}$$

$$\begin{aligned} U^{n+1} &= U^{n+1/2} - \frac{\Delta t}{4\Delta x} [U(\bar{U}^x)_x]^{n+(\mu+1)/2} \\ &\quad - \frac{\Delta t}{4\Delta y} V^{n+(\mu+1)/2} (\bar{U}^y)_y^{n+1/2} \\ &\quad + \frac{\Delta t}{2} f \bar{V}^{n+(\mu+1)/2} - \frac{\Delta t}{4\Delta x} g \eta_x^{n+(\mu+1)/2} - \frac{1}{\rho} \frac{\Delta t}{4\Delta x} (P_a)_x \\ &\quad + \frac{(\tau_{sx} - \tau_{bx}^{n+(\mu+1)/2})}{\rho(\bar{h}^y + (\bar{\eta}^x)^{n+1/2})} + \frac{\Delta t}{8\Delta x^2} A_H U_{xx}^{n+(\mu+1)/2} \\ &\quad + \frac{\Delta t}{8\Delta y^2} A_H U_{yy}^{n+1/2} \text{ at } i, j \quad (31) \end{aligned}$$

where the following notation has been used

$$f_{ij}^{n+(\mu+1)/2} = \mu f_{ij}^{n+1} + (1 - \mu) f_{ij}^{n+1/2}$$

At each time-step, the vertically integrated model is coupled with the three-dimensional model through the bottom shear stress. In order to avoid an iterative scheme between the two models, the numerical solution of Equation (16) is carried out using the bed stress and eddy viscosity computed at

the previous time-step. Thus, the complex coefficients can be determined directly for each individual grid element as follows

$$\begin{aligned} A_r^{n+1/2} &= \frac{A_r^n \left[1 - \frac{\Delta t}{2} (1 - \mu) M_r^n \right] + \frac{\Delta t}{\rho d^n} [T_s - T_b^n \cos \alpha_r]}{\left[1 + \frac{\Delta t}{2} \mu M_r^n \right]} \quad (32) \end{aligned}$$

where

$$M_r^n = \left[if + \frac{A_v^n \alpha_r^2}{(d^2)^n} \right].$$

Initial and boundary conditions

The initial conditions are taken as still-water values, that is U , V , and η are zero everywhere. However, the model accepts as initial conditions for surface fluctuation and current velocities nonzero values which must be dynamically consistent.

Along closed boundaries the normal component of current is set to zero.

For tide-induced circulation within the computational domain, it is necessary to specify the tide elevation and current velocities along open boundaries. Since, generally, the current velocities are not known accurately enough to prescribe them for the model, the values of U and V on the boundary were set equal to their values at the first interior point.

For flows induced by the atmospheric forcing the variables at the open boundaries are calculated using a partially clamped, explicit, radiation condition given by

$$\frac{\partial \phi}{\partial t} \pm \sqrt{gh} \frac{\partial \phi}{\partial \lambda} = -\frac{\phi}{T_f} \quad (33)$$

where $\phi = \eta$, U , or V , $\lambda = x$ or y , and T_f is the friction time scale, and whose value should be about $100 \Delta t$ (Chapman, 1985). The radiation condition (33) is applied only to those nodes with depths less than 200 m. The other nodes are approximated by clamped boundary conditions in which the open boundary variables are fixed at zero throughout the computations.

The model is spun up by increasing the tidal and atmospheric forcing from zero linearly over one inertial period.

Nonlinear aliasing effects

The model includes the horizontal eddy viscosity A_H which filters out the spurious growth of short waves. However, unreal values of A_H sometimes have to be used to avoid the large aliasing error introduced by the occurrence of nonlinear instabilities. For these cases the nine point filter of Shapiro (1970) is introduced. It consists of the periodic ap-

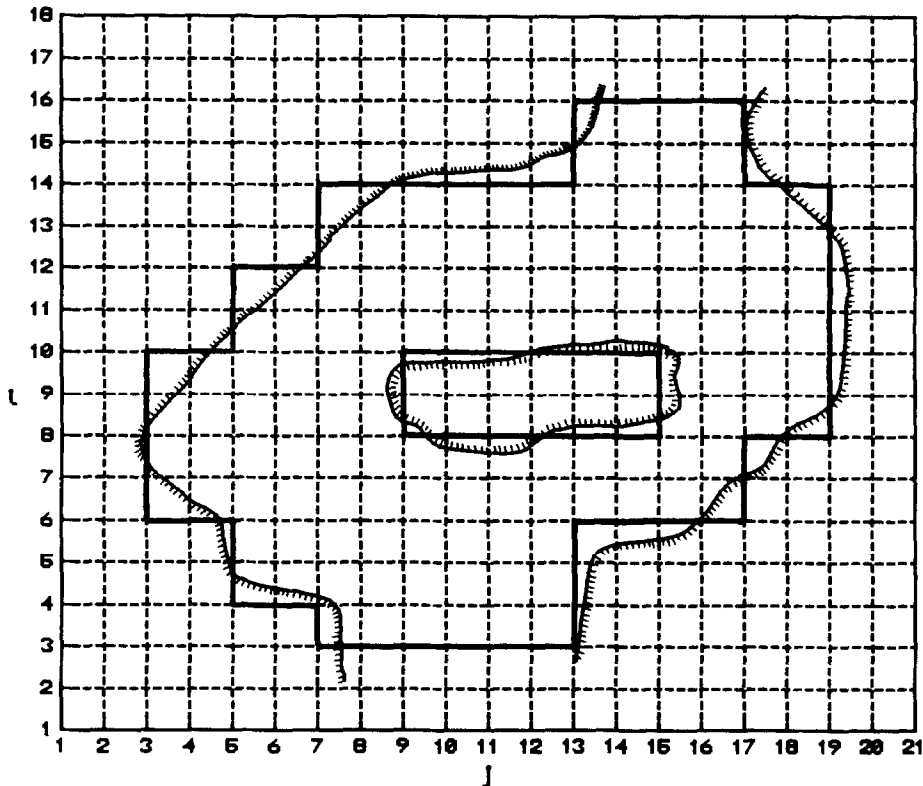


Figure 2. Idealized domain with computing grid superimposed.

plication of an operator which spatially averages the computed velocities and water surface displacement at the end of a fixed number of time-steps by using a predetermined weighting factor as follows

$$\begin{aligned} \Psi_{i,j}' &= \Psi_{i,j} + \frac{S}{2}(1-s) \\ &\times (\Psi_{i+2,j} + \Psi_{i-2,j} + \Psi_{i,j+2} + \Psi_{i,j-2} - 4\Psi_{i,j}) \\ &+ \frac{S^2}{4}(\Psi_{i-2,j+2} + \Psi_{i+2,j+2} + \Psi_{i+2,j-2} + \Psi_{i-2,j-2} - 4\Psi_{i,j}) \end{aligned} \quad (34)$$

where Ψ' is the averaged value of the variable Ψ which stands for U , V , or η , and s is the weighting factor which can take any value between 0 and 0.5.

The adopted procedure has the effect of simulating the transfer of turbulent energy to scales smaller than the double of the grid resolution.

It should be noted that the user must specify the number of time-steps at which the calculated values of U , V , and η are filtered.

COMPUTER IMPLEMENTATION

Input specifications

We first describe the input specifications and then the various subroutines of the program. The computational domain is contained in an $M \times N$ matrix

where the nodes with even values of i and j correspond to U , the nodes with odd values of i and j correspond to V , the nodes with odd values of i and even values of j correspond to η , and the nodes with even values of i and odd values of j correspond to h (Fig. 1). The input variables and parameters of the program are prescribed in five files. In order to describe these files better, an application of the model to an idealized domain, as shown in Figure 2, is presented where the computational domain and the boundary are superimposed.

The first input file is called **PARAMETR.DAT** and contains 35 data in 13 lines. Figure 3 shows the corresponding file for the example given in Figure 2.

The data contained in the **PARAMETR.DAT** file are explained in sequential form as follows:

NN: number of columns to be swept in the x direction. The first three records of the configurational file given by Figure 4 are indicating the initial and final nodes of each column. In the example, from $j = 9$ to $j = 15$, six columns must be considered because the island is not included in the computational domain;

MM: number of rows to be swept in the y direction. The fourth, fifth and sixth records of the configurational file given by Figure 4 are indicating the initial and final nodes of each row. In the example, from $i = 8$ to i

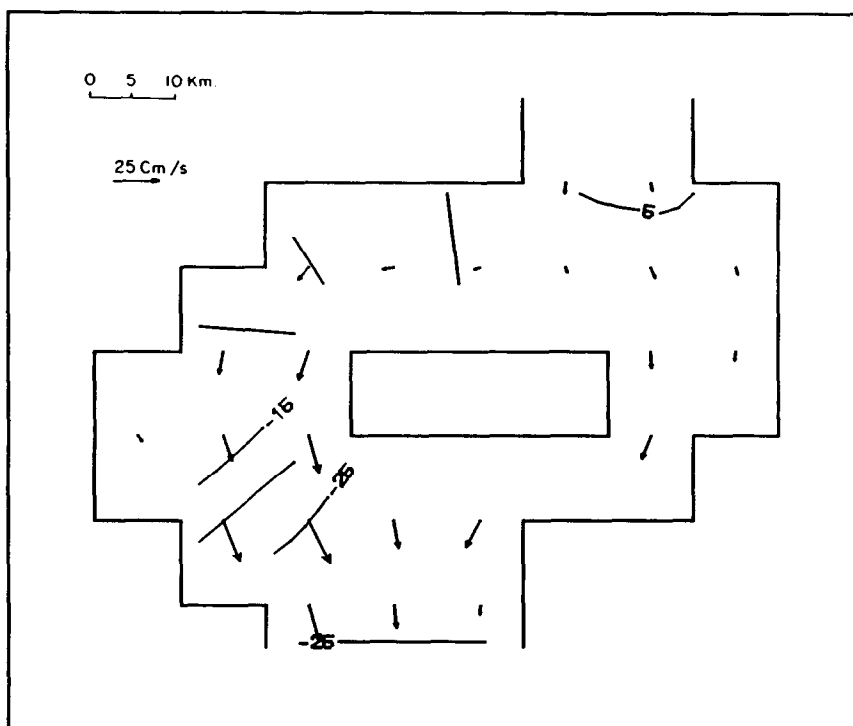


Figure 5. Resulting surface water elevation in cm and depth of mean currents for example of Figure 2

Γ : dimensionless coefficient of proportionality of the vertical eddy viscosity;

IBAND: this parameter takes the value 0 or 1. If *IBAND* = 0 the velocity vertical profile is not computed and the bed shear stress is calculated using the depth average currents;

IFILT: number of time-steps at which the calculated values of U , V , and η are filtered;

ϕ : mean latitude of modelled region, in degrees. Positive in the Northern hemisphere;

ICORI: this parameter takes the value 0 or 1. If *ICORI* = 0 the Coriolis force is not considered;

IMETEO: this parameter takes the value 0 or 1. If *IMETEO* = 0 the atmospheric forcing is not considered;

File name 1: name of the file that contains the positions i , j of the computational domain boundaries;

File name 2: name of the file that contains the $M \times N$ matrix with the initial conditions and depths;

File name 3: name of the file that contains the $M \times N$ matrix with the atmospheric forcing;

NE: node numbers of the open boundaries where the U , V , or η values must be specified as boundary conditions or calculated with the gravity radiation condition. When the computational domain does not have open boundaries *NE* must be set equal to 1.

File name 4: name of the file that contains the U , V , and/or η values which are used as forcing in the open boundaries for tide-induced circulation;

TX1, *TX2*, *TX3*, and *TX4*: times at which model outputs are desired, in h.

In general, the computational domain will have an irregular geometry so that the nodes i , j of the boundary must be given in a configuration file whose name is given in the *PARAMETR.DAT* file. The file corresponding to the example given in Figure 2 is shown in Figure 4.

This configuration file consists of 21 lines which are explained in sequential form as follows:

I—initial i of each one of the NN columns to be swept in the x direction;

II—final i of each one of the NN columns to be swept in the x direction;

III— j of the NN columns given by I and II. For the example given in Figure 2, the first column to be swept is given by $j = 4$ from $i = 6$ to $i = 10$ (see also Fig. 4);

IV— i of the MM rows given below by V and VI;

V—initial j of each one of the MM rows to be swept in the y direction;

VI—final j of each one of the MM rows to be swept in the y direction. For the example given in Figure 2, the first row to be swept is given by $i = 5$ from $j = 5$ to $j = 13$ (see also

Fig. 4);

- VII—*i* of the previous nodes to the initial nodes of each of the *NNN* columns of the computational domain;
- VIII—*i* of the subsequent nodes to the final nodes of each of the *NNN* columns of the computational domain;
- IX—*j* of the *NNN* columns given by VII and VIII. For the example given in Figure 2, the first column is given by *j* = 3 from *i* = 5 to *i* = 11 (see also Fig. 4);
- X—*i* of the *MMM* rows given below by XI and XII;
- XI—*j* of the previous nodes to the initial nodes of each of the *MMM* rows of the computational domain;
- XII—*j* of the subsequent nodes to the final nodes of each of the *MMM* rows of the computational domain. For the example given in Figure 2, the first row is given by *i* = 3 from *j* = 6 to *j* = 14 (see also Fig. 4);
- XIII—*i* of each one of the *NE* nodes of the open boundaries. If open boundaries do not exist then *i* = 1;
- XIV—*j* of each one of the *NE* nodes of the open boundaries. If open boundaries do not exist then *j* = 1;
- XV—the nodes of the open boundaries can be placed on the right, left, top and/or bottom margins of the computational domain, and they are recognized using the numbers 2, -2, 1 and -1, respectively. This is necessary for solving the radiation condition. Zero must be used if there are no open boundaries. For the example given in Figure 2, the first node is given by *i* = 3 and *j* = 8, and the corresponding number is -1 because it is placed in a bottom margin (see also Fig. 4);

- XVI—initial *i* of the *NNNN* columns between columns to be swept in the *x* direction;
- XVII—final *i* of the *NNNN* columns between columns to be swept in the *x* direction;
- XVIII—*j* of the *NNNN* columns given by XVI and XVII. For the example given in Figure 2, the first column is given by *j* = 5 from *i* = 6 to *i* = 10 (see also Fig. 4);
- XIX—initial *j* of the *MMMM* rows between rows to be swept in the *y* direction;
- XX—final *j* of the *MMMM* rows between rows to be swept in the *y* direction;
- XXI—*i* of the *MMMM* rows given by XIX and XX. For the example given in Figure 2, the first row is given by *i* = 4 from *j* = 7 to *j* = 13 (see also Fig. 4).

The bathymetry and the initial conditions are specified in a file whose name is given in the PARAMETR.DAT file. This file contains *M* lines with *N* data each. The *h* and *η* values must be given in centimeters, and the *U* and *V* values in cm/sec. It should be noted that if the initial conditions are taken as still-water values, then *U*, *V*, and *η* are zero everywhere. Table 1 shows the corresponding file for the example of Figure 2.

The atmospheric forcing is specified in a file whose name is given in the PARAMETR.DAT file. The wind velocity (in knots) in the *x* and *y* directions must be given in the nodes with even and odd *i,j* values, respectively. The atmospheric pressure (in HPa) is given in the nodes with odd *i* and even *j*.

The open boundary conditions are specified in a file whose name is given in the PARAMETR.DAT file. This file contains *L* lines, where

$$L = \frac{KEND\ 3600}{(\Delta t/2)} + 1 \tag{35}$$

with *NE* data each.

Table 1. Example of bathymetry and initial condition data file for computational domain given in Figure 2

18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	200	0	500	0	200	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	200	0	200	0	200	0	200	0	500	0	200	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	200	0	200	0	300	0	300	0	300	0	300	0	200	0	200	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	200	0	200	0	300	0	100	0	100	0	100	0	100	0	300	0	200	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	200	0	500	0	300	0	100	0	100	0	100	0	100	0	300	0	200	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	200	0	200	0	300	0	300	0	300	0	200	0	200	0	200	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	200	0	500	0	500	0	500	0	200	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
i \ j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

Table 2. Program outputs at specified times

Times	Free surface elevation	Depth averaged currents	Surface currents ($\sigma = 0$)	Half depth currents ($\sigma = 0.5$)	Bottom currents ($\sigma = 1$)
TX1	ETAOUT01.GRD	VELOA01.DAT	VELOS01.DAT	VELOM01.DAT	VELOB01.DAT
TX2	ETAOUT02.GRD	VELOA02.DAT	VELOS02.DAT	VELOM02.DAT	VELOB02.DAT
TX3	ETAOUT03.GRD	VELOA03.DAT	VELOS03.DAT	VELOM03.DAT	VELOB03.DAT
TX4	ETAOUT04.GRD	VELOA04.DAT	VELOS04.DAT	VELOM04.DAT	VELOB04.DAT

Main program

The main program TAWIC reads the first data from the subroutines PARAM, INITI, and METEO and after some preliminary calculations calls the subroutines BOUND, SWEEPX, SWEEPY, FILTER, ENERGY, and OUTPUT. The program then loops through the main subroutines SWEEPX and SWEEPY which are concerned with the ADI algorithm for each time-step. Before calling these subroutines all the boundary conditions of the computational domain are updated on the subroutine BOUND. After a predetermined number of time-steps the subroutine FILTER is called and the calculated values of U , V , and η are filtered. At the end of each loop the total mechanical energy of the system is computed in the subroutine ENERGY and stored in the ENERGY.DAT file. Optionally, the user can create a file with the time history of U , V , or η at any desired node. This output must be achieved by modifying the program just below where the subroutine ENERGY is called.

Subroutines

SWEEPX (SWEEPY): this subroutine performs the bulk of the work involved in applying the ADI algorithm. The U (V) and η values are computed by forward sweep and backward substitution in the x direction (y direction).

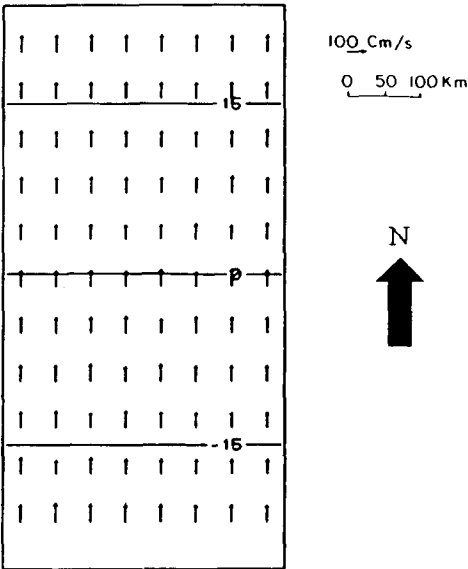
GEGEX (GEGEY): this subroutine called from SWEEPX (SWEEPY) computes the elements of the tridiagonal matrix when the computational domain is swept in the x direction (y direction).

CALV (CALU): this subroutine called from SWEEPX (SWEEPY) computes the V (U) values by forward sweep and backward substitution in the y direction (x direction) after the U (V) and η values were computed in SWEEPX (SWEEPY) at the same time-step,

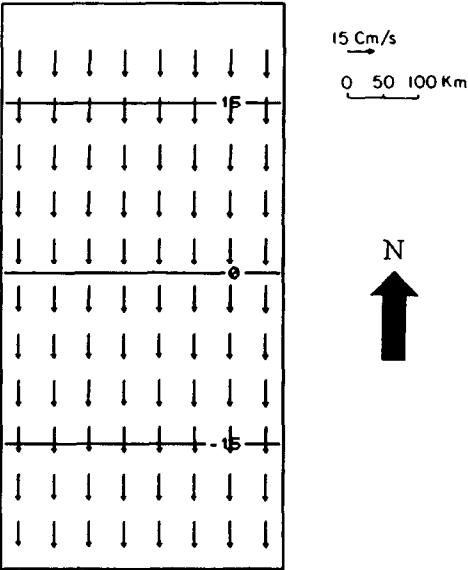
BOTTOM: this subroutine (called from GEGEX, GEGEY, CALV, and CALU) performs a point calculation of the bottom currents applying the Galerkin technique in the vertical. This subroutine is called only when the depths are less than 200 m. If the depths exceed 200 m the bed stress is computed from

$$\tau_{bx} = \rho k U \sqrt{U^2 + V^2} \quad \tau_{by} = \rho k V \sqrt{U^2 + V^2} \quad (36)$$

OUTPUT: this subroutine performs the outputs of the program at those times specified in TX1,

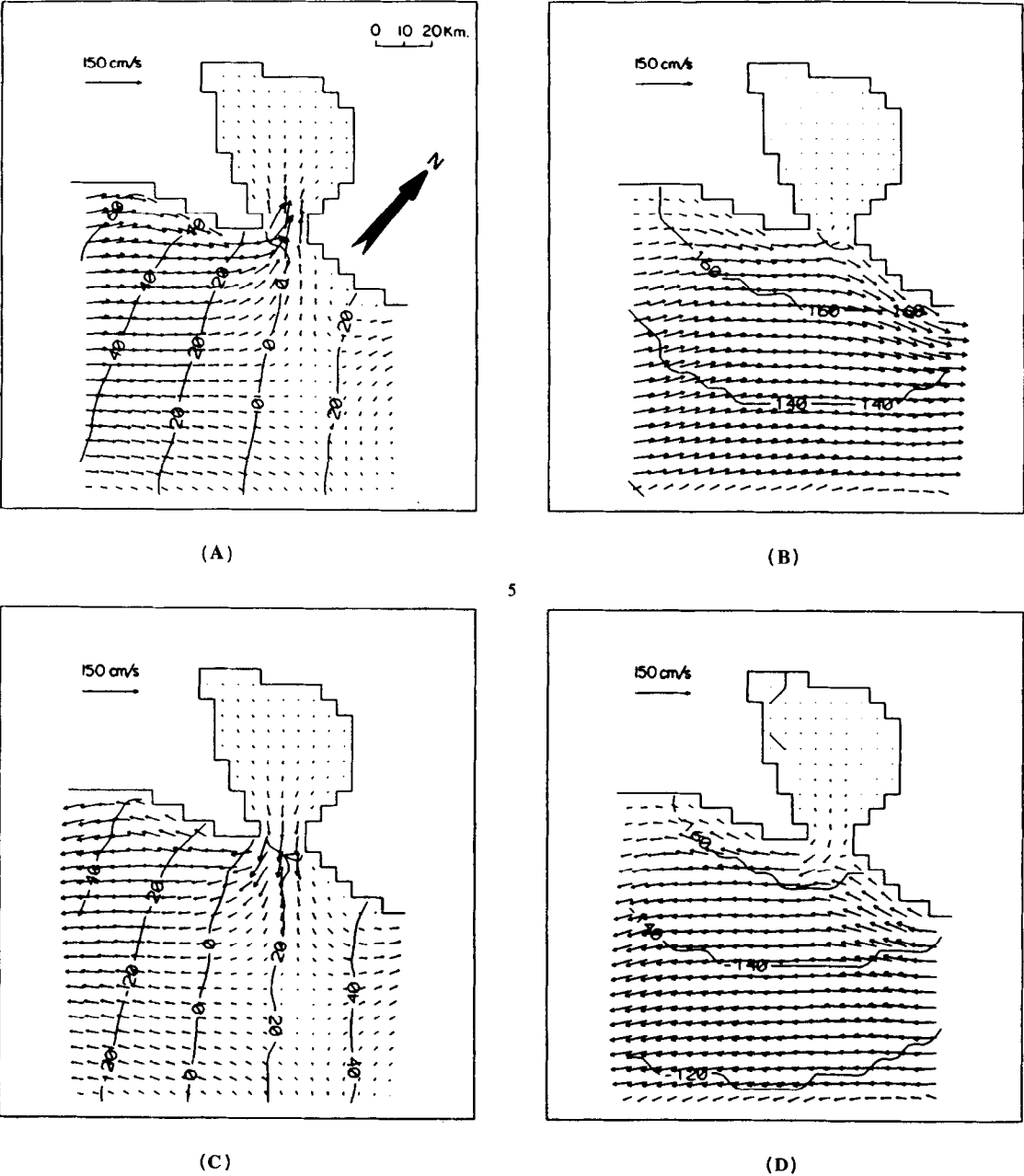


(A)



(B)

Figure 6. Steady-state surface-water elevation in cm and surface (A) and bottom (B) currents produced by a southerly wind of 12.5 m/sec in a closed rectangular basin.



5

Figure 7. Tidal amplitude in cm and tidal currents for semidiurnal lunar M_2 tide at cotidal hours 3 (A), 6 (B), 9 (C), and 12 (D) corresponding times of flood, high, ebb, and low tide at Golfo Nuevo (Lat: 43°S , Long: 64°W), Argentina.

TX2, TX3, and TX4. The outputs consist of five files (Table 2) in a format readable by the Golden Software SURFER version 4. The files with the GRD extension contain the free surface elevation for each specified time. The other files are SURFER post files with the depth average currents, and the surface ($\sigma = 0$), half depth ($\sigma = 0.5$), and bottom ($\sigma = 1$) currents when depths on the computational domain are less than 200 m.

EXAMPLES OF OUTPUT

Examples of output are provided to demonstrate the program capabilities. All simulations were made on a PC 486 DX2 running at 66 MHz. Figure 5 shows a result of the model implementation for the idealized domain of Figure 2. The free surface elevation and depth mean currents are shown after 37 h of simulation using a time-step of $\Delta t = 180$ sec and a resolution of $\Delta x = \Delta y = 5$ km (Fig. 3). The

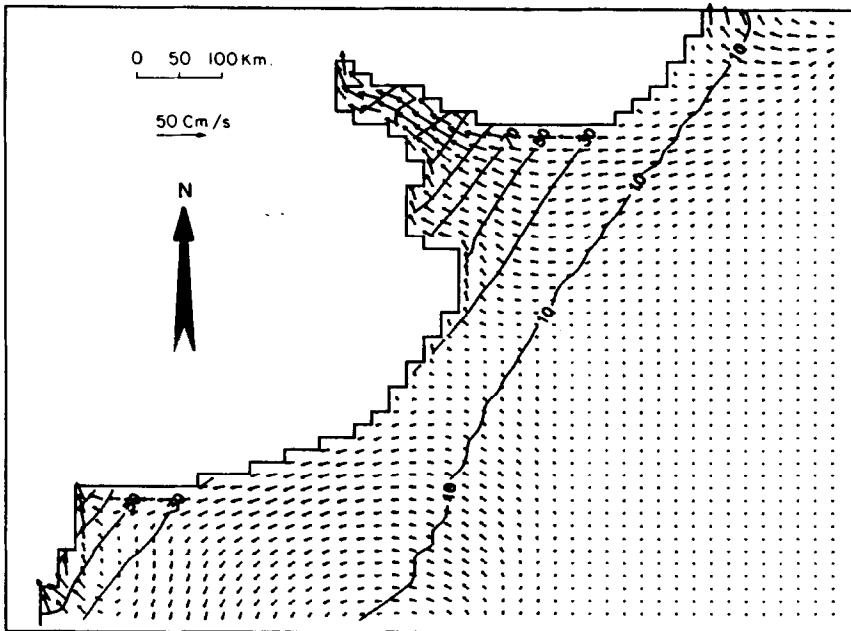


Figure 8. Steady-state sea level in cm and currents produced by southeasterly wind of 25 m/sec in Río de la Plata (Lat: 37°S, Long: 55°W), Argentina.

open boundary conditions were $\eta = 30 \cos \omega t$ for the nodes $i = 3, j = 8, 10$, and 12 , and $U = 10 \sin \omega t$ for the nodes $i = 16, j = 14$ and 16 , with $\omega = 2\pi/12$ h.

Figure 6A,B shows a closed rectangular domain with the free surface elevation, and the surface (A) and bottom (B) currents, respectively, which result from forcing the model with a constant wind of 12.5 m/sec from the south. The basin has dimension of 800 km (north–south) and 400 km (west–east), with 65 m depth. This simulation was made without considering the Coriolis force. The bottom currents have south directions as is expected.

In order to apply the model under field conditions it was implemented for Golfo Nuevo (Lat: 43°S, Long: 64°W) and the Río de la Plata (Lat: 37°S, Long: 55°W), both on the Argentine coast, to simulate tide-driven currents and wind-induced circulation, respectively. Figure 7A,B,C,D shows the sea level and currents for the cotidal hours 3, 6, 9, and 12, respectively, and corresponds to the times of flood, high, ebb, and low tide within Golfo Nuevo. This run was performed using a time-step of $\Delta t = 180$ sec and a horizontal grid spacing of $\Delta x = \Delta y = 2.75$ km.

Figure 8 shows the steady-state free-surface elevation and the depth mean currents for the Río de la Plata after two days of wind-induced circulation simulation for a southeasterly wind of 25 m/sec. This run was made using a time-step of $\Delta t = 80$ s with $\Delta x = 7.94$ km and $\Delta y = 10.87$ km. The domain has depth values of 1 m in the northwest corner and greater than 5,000 in the southeast corner.

CONCLUSIONS

Using the two-dimensional depth-averaged non-linear equations and three-dimensional linear equations, a quasi-three-dimensional numerical model was devised by coupling two modules computing the depth averaged current and vertical distribution of the velocities. In contrast to conventional two-dimensional models, the bottom stress is no longer related to the average depth currents but is computed from the bottom currents. The two modules are coupled through the bottom shear stress. The program includes the option of running the model using the conventional representation of the bed shear stress based on the depth mean currents, which requires less computational time.

The model described yields reliable knowledge of the two- and three-dimensional characteristics of the flow field for a wide range of shallow-water flows, and can be adapted easily to any domain of irregular boundaries.

The program and test datasets are available by anonymous FTP from the server IAMG.ORG.

REFERENCES

- Chapman, D. C., 1985, Numerical treatment of cross-shelf open boundaries in a barotropic coastal ocean model: *Jour. Physical Oceanography*, v. 15, no. 8, p. 1060–1075.
- Davies, A. M., 1980a, On formulating a three-dimensional hydrodynamic sea model with an arbitrary variation of vertical eddy viscosity: *Computer Method. Appl. Mech. Eng.*, v. 22, no. 2, p. 187–211.

- Davies, A. M., 1980b, Application of the Galerkin method to the formulation of a three-dimensional nonlinear hydrodynamic numerical sea model: *Appl. Math. Modelling*, v. 4, no. 4, p. 245–256.
- Davies, A. M., 1986, A three-dimensional model of the northwest european continental shelf, with application to the M_4 tide: *Jour. Physical Oceanography*, v. 16, no. 5, p. 797–813.
- Davies, A. M., 1988, On formulating two-dimensional vertically integrated hydrodynamic numerical models with an enhanced representation of bed stress: *Jour. Geophys. Res.*, v. 93, no. C2, p. 1241–1263.
- Davies, A. M., 1991, On using turbulence energy models to develop spectral viscosity models: *Continental Shelf Research*, v. 11, no. 11, p. 1313–1353.
- Davies, A. M., and Furnes, G. K., 1980, Observed and computed M_2 tidal currents in the North Sea: *Jour. Physical Oceanography*, v. 10, no. 2, p. 237–257.
- Davies, A. M., and Heaps, N. S., 1980, Influence of the Norwegian trench on the wind-driven circulation of the North Sea: *Tellus*, v. 32, no. 21, p. 164–175.
- Davies, A. M., and Jones, J. E., 1993, On improving the bed stress formulation in storm surge models: *Jour. Geophysical Res.*, v. 98, no. C4, p. 7023–7038.
- Douglas, J. Jr., and Gunn, J. E., 1964, A general formulation of alternating-direction methods: *Numer. Math.*, v. 6, no. 6, p. 428–453.
- Hess, K. W., 1976, A three-dimensional numerical model of the estuary circulation and salinity in Narragansett Bay: *Estuarine Coastal Marine Science*, v. 4, no. 3, p. 325–338.
- Owen, A., 1980, A three-dimensional model of the Bristol Channel: *Jour. Physical Oceanography*, v. 10, no. 8, p. 1290–1302.
- Pearce, B. R., and Cooper, C. K., 1981, Numerical circulation model for wind induced flow: *Jour. Hydraulics Division, Am. Soc. Civil Engineers*, v. 107, no. HY3, p. 286–302.
- Shapiro, R., 1970, Smoothing, filtering, and boundary effects: *Rev. Geophys. Space Phys.*, v. 8, no. 2, p. 359–387.
- WAMDI Group, 1988, The WAM model—A third generation ocean wave prediction model: *Jour. Physical Oceanography*, v. 18, no. 12, p. 1775–1810.
- Weare, T. J., 1979, Errors arising from irregular boundaries in ADI solutions of the shallow-water equations: *Intern. Jour. Numerical Methods Engineering*, v. 14, no. 6, p. 921–931.