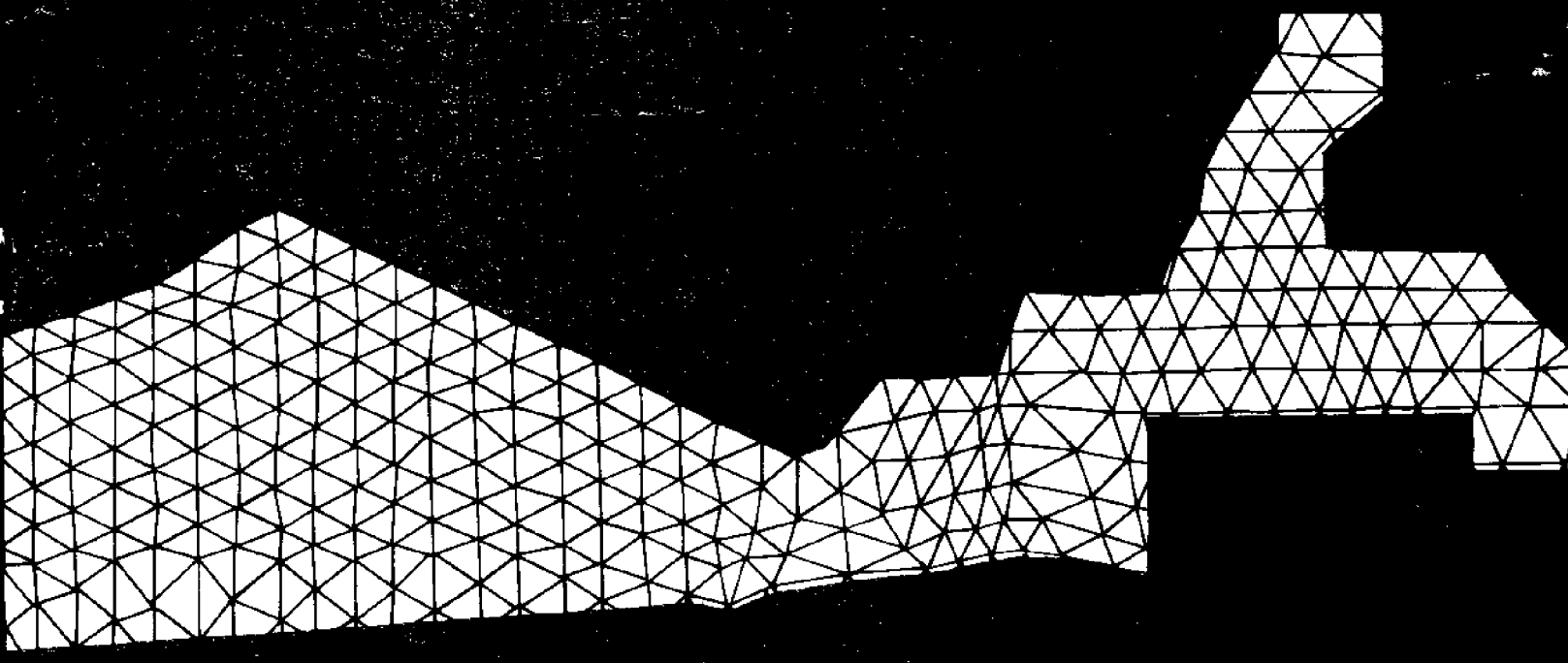
MODELING AN OCEAN POND

HSIN-PANG WANG



Mechanical Engineering and Applied Mechanics / NOAA Sea Grant
University of Rhode Island Marine Technical Report Number 40

Modeling an Ocean Pond

A Two-Dimensional, Finite Element Hydrodynamic Model of Ninigret Pond, Charlestown, Rhode Island

Hsin-Pang Wang, graduate research assistant

Department of Mechanical Engineering and Applied Mechanics

University of Rhode Island/Marine Technical Report 40/Kingston 1975

This publication is a result of research sponsored by NOAA Office of Sea Grant, Department of Commerce, under grant 04-3-158-3. The U.S. government is authorized to produce and distribute reprints for governmental purposes. Additional copies may be obtained from the Marine Advisory Service, University of Rhode Island, Narragansett Bay Campus, Narragansett, Rhode Island 02882.

CONTENTS

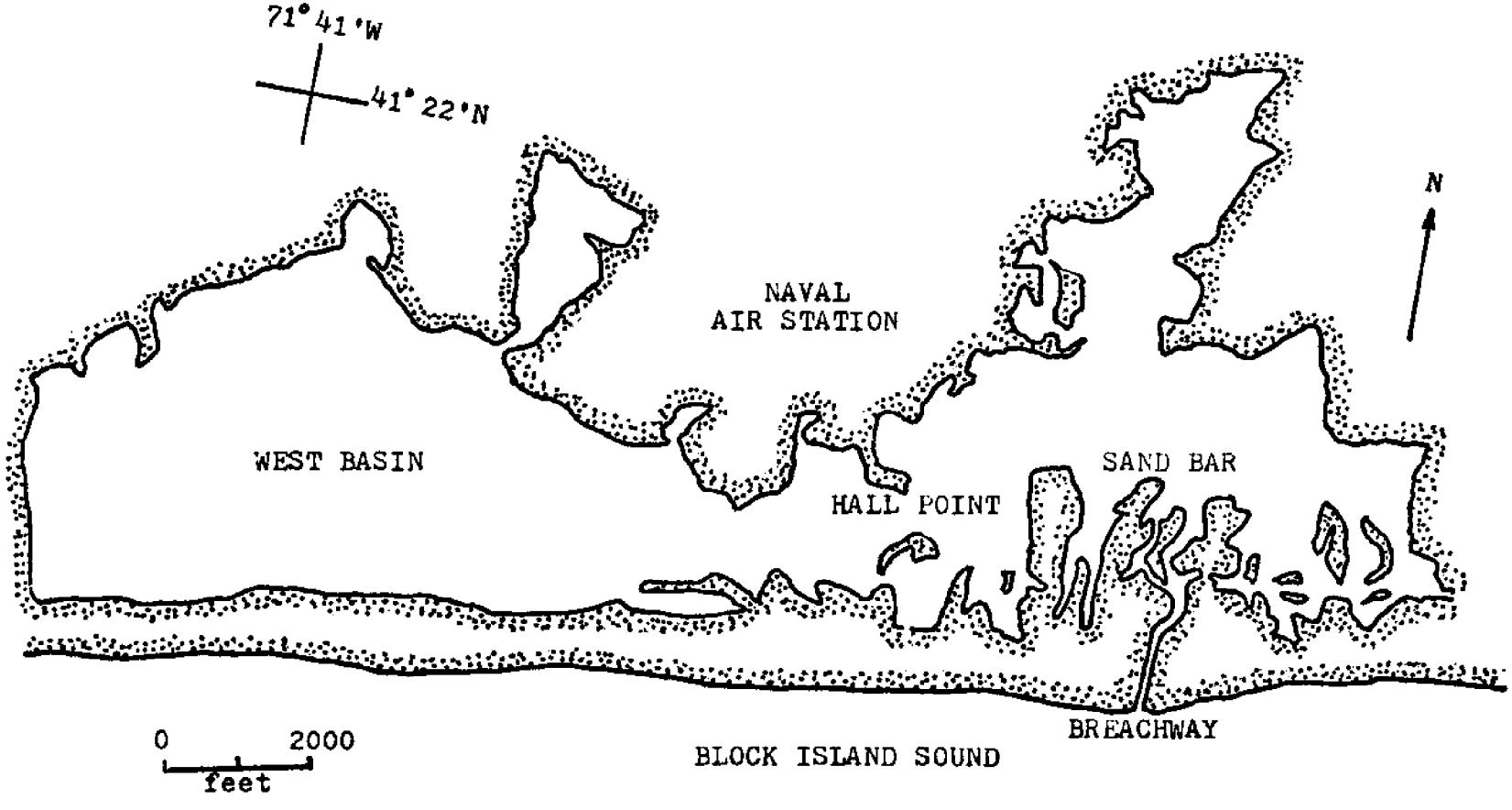


Fig. 1. Reference map of Ninigret Pond, Charlestown, Rhode Island.

CHAPTER 1. INTRODUCTION

A nuclear electric power station has been proposed for land formerly used as the Charlestown Naval Air Station in Rhode Island; extensive environmental research is required to determine the possible ecological impact of its construction.

The proposed site is on the north shore of Ninigret Pond (Fig. 1). The determination of the hydrodynamic characteristics of this salt pond is the preliminary step for subsequent investigations of parameters such as water quality, temperature and sedimentation.

A breachway connects this pond with Block Island Sound, and water circulation in the pond is caused predominantly by the tidal changes which reach the pond through this connection. During recent years, tidal motion in a shallow estuary has been modeled mathematically by vertically integrated, two-dimensional, hydrodynamic equations [1, 2, 3, 4], which present the approach to modeling in this work. But due to the complexity of this system of non-linear partial differential equations and the existence of irregular geometric boundary conditions in the pond, an analytical solution is not feasible. Instead, two numerical techniques, the finite difference and the finite element, usually are applied to real estuaries. The finite-difference method uses approximations of derivatives in terms of discrete differences and subdivides the solution domain into a finite number of grids, usually square-shaped. The finite-element method uses approximations of integrals either through minimization of a variational statement or through use of a weighted residual principle. The solution region is represented by elements of any size or shape, the most popular being triangular.

The chief advantage of the finite-element method is its convenience for use with complex configurations, such as estuaries and ponds. It includes local grid refinement at regions of steep parameter gradients, such as sharp corners or point sources of discharge.

This study used the Galerkin weighted-residual method through which the finite-element scheme can be implemented without a knowledge of the particular variational principle of the governing equation. Both Massachusetts Bay and the North Sea have been successfully modeled by this method [3, 4].

The objectives of this report are: (1.) to investigate the efficiency of different numerical integration schemes; and (2.) to develop a suitable finite-element algorithm to simulate the tidal currents in Ninigret Pond.

CHAPTER 2. GOVERNING EQUATIONS AND FINITE ELEMENT FORMULATION

2.1 GOVERNING EQUATIONS

The vertically-integrated hydrodynamic equations used in this report have been developed fully elsewhere [2, 4]. Therefore, only a brief review is presented here.

The continuity equation is

$$\frac{\partial}{\partial t}(\eta) + \frac{\partial}{\partial x}(q_x) + \frac{\partial}{\partial y}(q_y) = q_i \tag{2-1}$$

where

$H = h + \eta$ the sum of depth and tidal surface change

$$q_x = \int_{-h}^{\eta} u\,dz$$

$$q_y = \int_{-h}^{\eta} v\,dz$$

q_i = source (inflow on horizontal surface)

The source q_i could represent rainfall on the surface or a certain discharge at the bottom.

The components of momentum equations in the x and y directions are

$$\frac{\partial}{\partial t}(q_x) + \frac{\partial}{\partial x}(\bar{u}q_x) + \frac{\partial}{\partial y}(\bar{u}q_y) = fq_y - gH\frac{\partial \eta}{\partial x} + k\frac{\rho_a}{\rho}W_x|W_x| - g\frac{q_x(q_x^2 + q_y^2)^{1/2}}{C^2H^2} \tag{2-2}$$

$$\frac{\partial}{\partial t}(q_y) + \frac{\partial}{\partial x}(\bar{v}q_x) + \frac{\partial}{\partial y}(\bar{v}q_y) = -fq_x - gH\frac{\partial \eta}{\partial y} + k\frac{\rho_a}{\rho}W_y|W_y| - g\frac{q_y(q_x^2 + q_y^2)^{1/2}}{C^2H^2} \tag{2-3}$$

where

$\bar{u} = \frac{1}{H}\int_{-h}^{\eta} u\,dz = q_x/H$

$\bar{v} = \frac{1}{H}\int_{-h}^{\eta} v\,dz = q_y/H$

f = Coriolis parameter = $2\,\Omega \sin\varphi$

g = gravitational constant

k = dimensionless drag coefficient

ρ = water density

ρ_a = air density

W_x, W_y = wind velocity components

C = the Chezy coefficient

Equations (2-2) and (2-3) are convenient for the calculation of flow direction at corner point, which will be explained later. There is a second form of the basic equation. Equations (2-1, 2-2, 2-3) can be easily converted to equations (2-4, 2-5, 2-6) with no source of mass.

$$\frac{\partial \eta}{\partial t} + \frac{\partial}{\partial x}(H\bar{u}) + \frac{\partial}{\partial y}(H\bar{v}) = 0 \qquad (2\text{-}4)$$

$$\frac{\partial \bar{u}}{\partial t} + \bar{u}\frac{\partial \bar{u}}{\partial x} + \bar{v}\frac{\partial \bar{u}}{\partial y} = f\bar{v} - g\frac{\partial \eta}{\partial x} + k\frac{\rho_a}{\rho}\frac{W_x|W_x|}{H} - g\frac{\bar{u}(\bar{u}^2 + \bar{v}^2)^{1/2}}{C^2 H} \qquad (2\text{-}5)$$

$$\frac{\partial \bar{v}}{\partial t} + \bar{u}\frac{\partial \bar{v}}{\partial x} + \bar{v}\frac{\partial \bar{v}}{\partial y} = = f\bar{u} - g\frac{\partial \eta}{\partial y} + k\frac{\rho_a}{\rho}\frac{W_y|W_y|}{H} - g\frac{\bar{v}(\bar{u}^2 + \bar{v}^2)^{1/2}}{C^2 H} \qquad (2\text{-}6)$$

Both expressions give the same results only for the constant depth case and the difference will be discussed in section 2.2 and chapter four.

2.2 FINITE ELEMENT FORMULATION

In the history of the development of finite element methods, most problems were based upon minimization of a function because this method was first applied in the field of structural mechanics, which is governed by variation principles.

The variational principles in fluid mechanics have been summarized by Finlayson [5], who includes these for perfect fluids, magnetohydrodynamics, non-Newtonian fluids, and low Reynolds-number flows. He further points out that there is no known variational principle for the Navier-Stokes equations which include both inertial and viscous terms.

Similarly, it is not obvious what is an appropriate function to be minimized for the "hyperbolic" type vertically-averaged equations (2-4, 5, 6) for coastal waters. McIver [6] derived an adjoint variational principle for these equations. However, it has no direct physical usefulness due to its extra adjoint variables which double the number of unknowns and increase the computational complexity.

Use of weighted residuals is a general method for obtaining solutions to partial differential equations. The unknow solution is simulated by a set of trial functions with adjustable constants which are chosen to give the best solution.

Once selected, the trial function is substituted into the governing equations to form the residuals. The constants are chosen in such a way that the residuals, modified by weighting functions, are forced to be zero in an average sense.

There are several ways to choose the weighting function and each way represents a different method: examples are the Galerkin method, the least squares method, and the method of moments. In Galerkin's method, the weighting functions are made to be the trial functions themselves.

Recent work [3, 4] has shown that the weighted residual theory, combined with finite element concepts, can handle multi-element regions with any complex

boundary, using a digital computer performing all the calculations.

In most cases, Galerkin's process produces a simpler and more direct formulation than the construction of a function and its subsequent minimization. When the equation is self-adjoint, the variational process is identical to that of the Galerkin weighted-residual method [7].

The triangle element with linear trial functions was chosen here in consideration of both computing time and flexibility. The nodal points i, j, k represent the vertices of the triangle, and the unknown variables may be written as the following matrix products:

$$\eta = \langle N_i, N_j, N_k \rangle \begin{Bmatrix} \eta_i \\ \eta_j \\ \eta_k \end{Bmatrix} = \langle N \rangle \{\eta\} \tag{2-7}$$

$$q_x = \langle N_i, N_j, N_k \rangle \begin{Bmatrix} q_{xi} \\ q_{xj} \\ q_{xk} \end{Bmatrix} = \langle N \rangle \{q_x\} \tag{2-8}$$

$$q_y = \langle N_i, N_j, N_k \rangle \begin{Bmatrix} q_{yi} \\ q_{yj} \\ q_{yk} \end{Bmatrix} = \langle N \rangle \{q_y\} \tag{2-9}$$

where N_i are called shape functions which can be defined by the coordinates of the nodes [8]

$$N_i(x,y) = (a_i + b_i x + c_i y)/2\triangle$$

where

$$\begin{bmatrix} a_i = x_j y_k - x_k y_j \\ b_i = y_j - y_k \\ c_i = x_k - x_j \end{bmatrix} \quad \text{permutations on i, j, k}$$

and

$$\triangle = 1/2 \det \begin{vmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_k & y_k \end{vmatrix} \qquad \text{is the element area}$$

In the Galerkin Method, if r denotes the residuals and N the shape functions, the required integral condition is

$$\int_{\text{Domain}} N r \, dA = 0$$

Applying this condition to equations (2-1, 2, 3) with the given shape functions N_i results in

$$\sum_{e=1}^{n} \int_{\Delta} \langle N \rangle^T \left[\langle N \rangle \{\dot{\eta}\} + \langle N_{,x} \rangle \{q_x\} + \langle N_{,y} \rangle \{q_y\} \right]_e \, dA = 0 \qquad (2\text{-}10)$$

$$\sum_{e=1}^{n} \int_{\Delta} \langle N \rangle^T \left[\langle N \rangle \{\dot{q}_x\} + \langle N_{,x} \rangle \{q_x^2/H\} + \langle N_{,y} \rangle \{q_x q_y/H\} + \right.$$

$$g\langle N \rangle \{H\} \langle N_{,x} \rangle \{\eta\} - f\langle N \rangle \{q_y\} - K \frac{\rho_a}{\rho} W_x |W_x| +$$

$$\left. g\langle N \rangle \left\{ \frac{q_x (q_x^2 + q_y^2)^{1/2}}{C^2 H^2} \right\} \right]_e \, dA = 0 \qquad (2\text{-}11)$$

where $N_{,x}$ denotes $\partial N/\partial x$ and n the number of elements.

$$\sum_{e=1}^{n} \int_{\Delta} \langle N \rangle^T \left[\langle N \rangle \{\dot{q}_y\} + \langle N_{,x} \rangle \left\{ \frac{q_x q_y}{H} \right\} + \langle N_{,y} \rangle \left\{ \frac{q_y^2}{H} \right\} + \right.$$

$$g\langle N \rangle \{H\} \langle N_{,y} \rangle \{\eta\} + f\langle N \rangle \{q_x\} - K \frac{\rho_a}{\rho} W_y |W_y| +$$

$$\left. g\langle N \rangle \left\{ \frac{q_y (q_x^2 + q_y^2)^{1/2}}{C^2 H^2} \right\} \right]_e \, dA = 0 \qquad (2\text{-}12)$$

The integration is taken over each element and the summation over the whole domain is zero. The integration of linear shape-function products into two dimensions has been tabulated by Baker and Zelazny [9].

$$\int_{\Delta} \langle N \rangle^T \, dA = \frac{\Delta}{3} \begin{Bmatrix} 1 \\ 1 \\ 1 \end{Bmatrix}$$

$$\int_{\triangle} \langle N \rangle^T \langle N \rangle \, dA = \frac{\triangle}{12} \begin{vmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{vmatrix}$$

$$\int_{\triangle} \langle N \rangle^T \langle N \rangle^T \langle N \rangle \, dA = \frac{\triangle}{60} \begin{vmatrix} 6 & 2 & 2 \\ 2 & 2 & 1 \\ 2 & 1 & 2 \end{vmatrix} \begin{vmatrix} 2 & 2 & 1 \\ 2 & 6 & 2 \\ 1 & 2 & 2 \end{vmatrix} \begin{vmatrix} 2 & 1 & 2 \\ 1 & 2 & 2 \\ 2 & 2 & 6 \end{vmatrix}$$

In matrix form, equations 2-10,11,12 can be written as

$$[M] \{\dot{\eta}\} = \{F_\eta\}$$

$$[M] \{\dot{q}_x\} = \{F_x\}$$

$$[M] \{\dot{q}_y\} = \{F_y\}$$

where $[M] = \begin{vmatrix} m_{11} & m_{12} & \cdots & 0 \\ m_{21} & m_{22} & & \\ \vdots & & \ddots & \\ 0 & & & \end{vmatrix}$

The boundary conditions are conveniently substituted into the coupled equations (2-2, 3) relating q_x and q_y. The details will be discussed in the next section. The coupled form is

$$[M_q] \{\dot{q}\} = \{F_q\}$$

where $[M_q] = \begin{vmatrix} m_{11} & 0 & m_{12} & 0 & \cdots & 0 \\ 0 & m_{11} & 0 & m_{12} & & \\ \vdots & & \ddots & & & \\ 0 & & & & & \end{vmatrix}$

$$\{\dot{q}\} = \begin{Bmatrix} \dot{q}_{x1} \\ \dot{q}_{y1} \\ \vdots \\ \dot{q}_{xn} \\ \dot{q}_{yn} \end{Bmatrix} \qquad \{F_q\} = \begin{Bmatrix} F_{x1} \\ F_{y1} \\ \vdots \\ F_{xn} \\ F_{yn} \end{Bmatrix}$$

There are many ways to choose the approximation function for the non-linear term. For example, the convective term can be expressed in two forms.

(a) $\dfrac{\partial(uq_x)}{\partial x}$: Let $uq_x = \langle N \rangle \{uq_x\}$

then $\dfrac{\partial(uq_x)}{\partial x} = \langle N_{,x} \rangle \{uq_x\}$

(b) $\dfrac{\partial(uq_x)}{\partial x} = u \dfrac{\partial q_x}{\partial x} + q_x \dfrac{\partial u}{\partial x}$

Let $u = \langle N \rangle \{u\}$, $q_x = \langle N \rangle \{q_x\}$

then

$$\frac{\partial(uq_x)}{\partial x} = \langle N \rangle \{u\} \langle N_{,x} \rangle \{q_x\} + \langle N \rangle \{q_x\} \langle N_{,x} \rangle \{u\}$$

Usually, the first method lumps all the variables into one approximation. The second one is more lengthy and involves higher order integration terms such as $\int \langle N \rangle^T \langle N \rangle$ ..... dA. The lumped-variable form (a) was chosen for the square root friction term and convective term.

Again, the only difference between the q_x, q_y expressions in equations (2-1, 2-2, 2-3) and the $\bar{u}$, $\bar{v}$ expression (2-4, 2-5, 2-6) is that q_x, q_y are the lumped product forms of $\bar{u}H$ and $\bar{v}H$.

$$q_x = \langle N \rangle \{\bar{u}H\}$$

$$q_y = \langle N \rangle \{\bar{v}H\}$$

If, instead, the q_x, q_y approximations are chosen as

$$q_x = \langle N \rangle \{\bar{u}\} \langle N \rangle \{H\}$$

$$q_y = \langle N \rangle \{\bar{v}\} \langle N \rangle \{H\}$$

the same results should be eventually obtained from both expressions, because they are entered into the same governing equations.

2.3 BOUNDARY CONDITIONS

Two types of boundaries, open and land, are used in the model. On an open boundary, the tidal surface or velocity must be specified. On the land boundary, the normal velocity at each node is zero, that is, the flow direction

should be parallel to the boundary. Its direction can be arbitrary, according to the element orientation.

This "vector" boundary condition cannot be directly inserted into the matrix form using global coordinates. Instead, a multi-coordinate local system should be used to resolve this condition. The local coordinate systems are defined as the outward normal direction of the node toward the land. Then the normal component of the velocity is easily set equal to zero. At the same time, the global coordinate is still valid for interior point calculations.

There is no such complex procedure necessary in the staggered scheme of rectangular finite difference methods [1,2]. Either the $\bar{u}$ velocity or the $\bar{v}$ velocity is zero on the boundary in that method. There is no necessity for local coordinates. The details of the present coordinate transformation are explained in Appendix A.

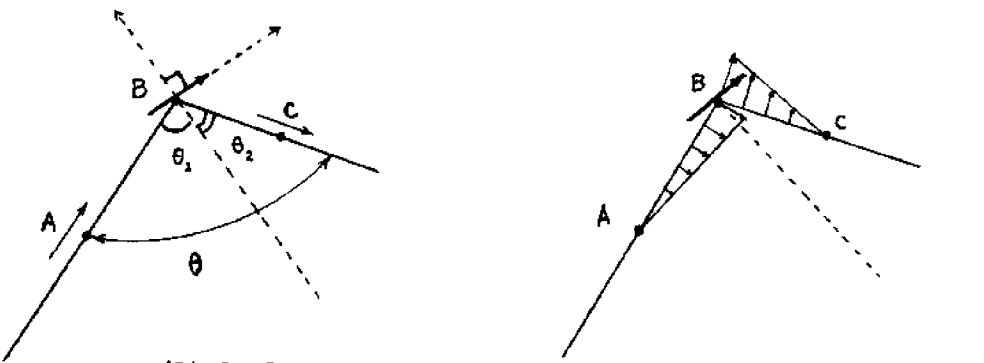
Another difficulty encountered here is the specification of the boundary condition at corners. Norton et al. [10] used zero velocity at the corner for continuity considerations. In a narrow region containing only a few nodal points, the zero velocity simplification will cause inacurracy in the global results.

A better treatment has been developed by Connor and Wang [4]. "Flow leaked out equals flow leaked in" is one way of stating the concept. Fig. 2(A) shows the relationships at a corner. The shaded volume flux areas in Fig. 2(B) are set equal for conservation of mass. The following two equations are thus solved for the flow direction.

$$\theta_1 + \theta_2 = \theta$$

$$\overline{AB} \cos \theta_1 - \overline{BC} \cos \theta_2 = 0$$

Experience shows that the condition of zero velocity is appropriate for a sharp concave corner. For a sharp convex corner, a mesh refinement in the vicinity is



(A) Angles

(B) Volume Flux Across Boundary

Fig. 2, Flow direction at the corner point.

necessary due to numerical constraints.

The unknown variables q_x, q_y, rather than $\bar{u}$, $\bar{v}$, are easier to use for the above calculations. The advantage of staggered-scheme finite difference methods, where the corner point is not in the computational field, is obvious.

2.4 STABILITY CONDITION

The precise stability condition for a specific nonlinear finite element method is not known at the present time. The finite-difference condition on the time step

$$T \leq CL/\sqrt{gh}$$

serves as a rough guide, but there are additional factors which affect finite-element stability. Some of these factors are (1.) the shape of the triangle element; (2.) the degree of refinement at sharp corners; and (3.) the area ratio of connected elements. The final time increment selected in the present study was based upon numerical experiments to ensure stability.

CHAPTER 3. THE NUMERICAL SCHEMES

After using the finite element integration in space coordinates, the original system of partial differential equations becomes a system of ordinary differential equations with respect to time. Three numerical schemes have been compared for a straight channel case where the analytic solution is known.

First, the commonly used fourth order Runge-Kutta method is applied [11]. If

$$\frac{dy}{dt} = f(t,y) \ ,$$

$$y_{n+1} - y_n = 0.17476028K_1 - 0.55148066K_2 + 1.20553560K_3 + 0.17118478K_4$$

where

$$K_1 = \Delta t \ f(t_n, y_n)$$

$$K_2 = \Delta t \ f(t_n + 0.4\,\Delta t, \ y_n + 0.4K_1)$$

$$K_3 = \Delta t \ f(t_n + 0.45573725\,\Delta t, \ y_n + 0.29697761K_1 + 0.15875964K_2)$$

$$K_4 = \Delta t \ f(t_n + \Delta t, \ y_n + 0.21810040K_1 - 3.05096516K_2 + 3.83286476K_3)$$

The second method is the fourth-order Adams-Moulton predictor-corrector method [12].

1. Compute $y_{n+1}^{(0)}$, using the Adams-Bashforth formula as a predictor:

$$y_{n+1}^{(0)} = y_n + \frac{\Delta t}{24} \ (55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3})$$

2. Compute $f_{n+1}^{(0)} = f(t_{n+1}, \ y_{n+1})$

3. Compute $y_{n+1}^{(0)}$, using the Adams-Moulton formula as a corrector:

$$y_{n+1}^{(j)} = y_n + \frac{h}{24} \left[9f(t_{n+1}, \ y_{n+1}^{(j-1)}) + 19f_n - 5f_{n-1} + f_{n-2}\right]$$

4. Iterate on j until

$$\frac{y_{n+1}^{(j)} - y_{n+1}^{(j-1)}}{y_{n+1}} < \varepsilon \text{ for a prescribed } \varepsilon$$

Since multi-step formulas are not self-starting, the predictor-corrector method is combined with the previous Runge-Kutta method to obtain necessary starting values.

The third method is the semi-implicit method. The continuity equation and the momentum equation are solved in successive order.

$$[M]\{\dot{\eta}\} = \{F_\eta\ (q_n,\ \eta_{n-1/2})\}$$

$$\{\eta\}_{n+1/2} = \{\eta\}_{n-1/2} + \Delta t\ \{\dot{\eta}\}$$

$$[M_q]\{\dot{q}\} = \{F_q\ (q_n,\ \eta_{n+1/2})\}$$

$$\{q_{n+1}\} = \{q\}_n + \Delta t\ \{\dot{q}\}$$

The popular alternating direction implicit (A.D.I.) method is not feasible here due to the irregular meshes.

The pure implicit method must solve both equations in a coupled form. The storage requirements usually exceed the memory capacity of the machine (in core). Through the use of peripheral devices (disks, tapes, etc.), this memory requirement can be reduced, and this is recommended for further study because of the favorable time increment gained.

Fig. 3 gives the dimensions and finite element network of the channel. The amplitude at the open end is 1 foot and the channel depth is 30 feet.

The analytical solution is

$$\eta = \frac{a}{\cos(\omega \frac{L}{\sqrt{gh}})} \cos\left\{\omega \frac{L}{\sqrt{gh}}\left(1 - \frac{x}{L}\right)\right\} \cos \omega t$$

$$u = -\frac{a\sqrt{gh}}{h \cos(\omega \frac{L}{\sqrt{gh}})} \sin\left\{\omega \frac{L}{\sqrt{gh}}\left(1 - \frac{x}{L}\right)\right\} \sin \omega t$$

where L, h are the length and depth.

The open boundary condition is

$$\eta = a \cos \omega t \qquad \text{at } x=0$$

Node	Time = 3/4 cycle				Time = 1 cycle			
	Numerical		Analytical		Numerical		Analytical	
	η	u	η	u	η	u	η	u
1	0.0000	0.3405	0.0	0.3389	1.0000	0.0005	1.0000	0.0
2	0.0000	0.3399	0.0	0.3389	1.0000	-0.0003	1.0000	0.0
3	0.0000	0.3405	0.0	0.3389	1.0000	0.0005	1.0000	0.0
4	0.0003	0.2740	0.0	0.2727	1.0183	0.0001	1.0187	0.0
5	0.0007	0.2750	0.0	0.2727	1.0193	0.0005	1.0187	0.0
6	0.0003	0.2740	0.0	0.2727	1.0183	0.0001	1.0187	0.0
7	0.0013	0.2073	0.0	0.2055	1.0331	0.0001	1.0333	0.0
8	0.0008	0.2066	0.0	0.2055	1.0334	0.0005	1.0333	0.0
9	0.0013	0.2073	0.0	0.2055	1.0331	0.0001	1.0333	0.0
10	0.0017	0.1382	0.0	0.1375	1.0436	0.0004	1.0438	0.0
11	0.0013	0.1381	0.0	0.1375	1.0438	0.0003	1.0438	0.0
12	0.0017	0.1382	0.0	0.1375	1.0436	0.0004	1.0438	0.0
13	0.0016	0.0694	0.0	0.0689	1.0500	0.0004	1.0501	0.0
14	0.0018	0.0695	0.0	0.0689	1.0502	0.0001	1.0501	0.0
15	0.0016	0.0694	0.0	0.0689	1.0500	0.0004	1.0501	0.0
16	0.0013	0.0	0.0	0.0	1.0530	0.0	1.0522	0.0
17	0.0021	0.0	0.0	0.0	1.0515	0.0	1.0522	0.0
18	0.0013	0.0	0.0	0.0	1.0530	0.0	1.0522	0.0

Table 1. Comparison of numerical and analytical results for the rectangular channel in Figure 3.

Table 1 shows good agreement between numerical and analytical results. The numerical error is approximately 0.5 percent.

ΔT_c in table 2 shows the largest time increment with stable results in each method. The third method is obviously the fastest.

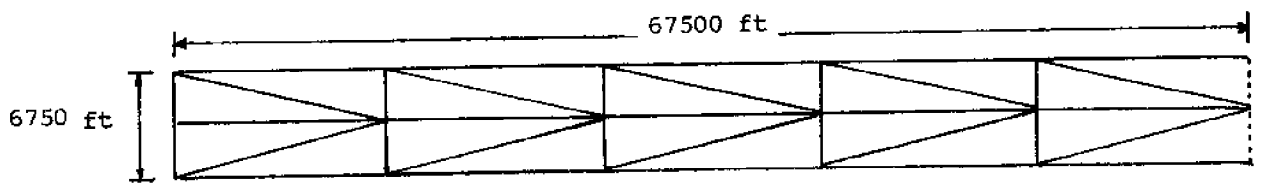


Fig. 3. The dimensions and finite element network for a straight channel.

Numerical method	ΔT_c (sec)	Computation time (one cycle)
1. Runge-Kutta	180	1.2 minutes
2. Adams-Moulton	60	1.8 minutes
3. semi-implicit	120	0.45 minutes

Table 2. Comparison of three different schemes.

CHAPTER 4. APPLICATION AND RESULTS

4.1 APPLICATION

Ninigret Pond is a small salt pond with a complex geometry. Due to its small size compared to the tidal wavelength, a relatively large amount of computer time is required to model it.

Two critical regions, the breachway and Hall Point, as shown in Fig. 1, prove to be the most difficult to incorporate into a proper layout of the elements. The breachway is a long, narrow, irregular channel through which the tidal amplitude sharply decreases. The open boundary is specified at the pond end of the breachway to exclude this complicated narrow region. Hall Point connects the west basin and the central pond. It is a very narrow channel with a deep center. An enlarged width distortion was specified here in the model to conserve computer time.

The finite-element network chosen is shown in Fig. 4. It is composed of 461 elements and 284 nodes, which makes it still a rather crude model for such complicated geometry. The depth varies from 1.5 to 8 feet, and the element area varies from 87,500 to 210,000 square feet. The smallest value of $L/\sqrt{gh}$ is 29 seconds; thus the time increment $\Delta T = 15$ seconds was selected for stable results. (3,000 steps per tidal period of 12.42 hours.)

The Chezy coefficient is calculated as

$$C = \frac{1.49}{N} (h + \eta)^{1/6}$$

where N is Manning's roughness factor.

The value of the Manning factor basically depends on bottom roughness and was taken as 0.03 for the whole pond. This value can be adjusted to fit measured data in the field.

On all nodes, zero velocity was assumed as an initial condition, and the water levels were taken at high tide. The transient starting error will be

damped out by bottom friction after a period of time. The transient period decreases with the Manning factor. The starting disturbance disappeared after approximately eight hours in the present calculations.

The tidal wave at the open boundary was prescribed for simplicity to be sinusoidal. Since there were only three nodes across the entrance, the same amplitude and time lag were specified for all three. The amplitude at the open boundary is about 0.2 feet for neap tide and 0.5 feet for spring tide. Since the tidal amplitude exceeds 10 percent of the depth in shallow regions, the convective momentum terms cannot be neglected.

4.2 RESULTS

Two sets of numerical results have been calculated, one based on the flux approximation and the other on the velocity approximation. Figs. 5 and 6 show the flood and ebb currents, respectively, in both cases. The two different approximations give nearly the same result except for the entrance region.

Winds can play an important role in the west basin currents. Gyres are usually generated in this area with various wind conditions (Figs. 7-10). The currents in the shallow region along the land are strongly influenced by the wind. Water depth is the essential factor in determining the rotational direction. In Fig. 7, the wind direction is opposite to the ebb currents, and the two gyres separate with ebb flow primarily down the center. In Fig. 8, the wind and flood currents coincide and the primary flow is along the boundaries. Both the flux and velocity approximations give similar results. Figs. 9 and 10 show results computed for a southwest wind, which prevails in the area.

Fig. 11 shows the typical time history hodograph of computed currents. An arrow from the origin to the solid line would represent the magnitude and direction of the instantaneous velocity. H denotes high water, L low water. The dotted lines represent the unrealistic transient period which occurs during

model start-up. Figs. 12-22 show the computed currents plotted for every tenth of the tidal period, with no wind and 0.4 feet amplitude at the open boundary. The measured maximum current values in Fig. 23 are only for gross comparison, because the tidal amplitude and wind conditions are not reported in that set of data [13].

The measured tidal amplitude in the west basin is about one-half of that at the entrance [14]. However, the computed tidal amplitudes are approximately equal everywhere with a 15-minute time lag between amplitude at the west basin and the entrance as listed in table 3. This discrepancy results mainly because the element size is too large for a good representation of this pond, especially in the narrow region of the breachway and Hall Point. A sand bar is located at the entrance which is too small to be included in the present model. The presence of this sand bar is indeed important to the accuracy of the global results. The enlarged width at Hall Point lets more water flow into the west basin than in the real case. A more refined finite-element mesh can decrease this discrepancy in the future.

Point	Tidal Amplitude (ft)	Phase Angle (degree)
A	0.415	7.0^{o}
B	0.402	2.2^{o}
C	0.400	0.0^{o}
D	0.412	4.5^{o}
E	0.416	7.5^{o}
F	0.405	2.4^{o}

Table 3. Computed tidal amplitudes and phase angles (see Fig. 4).

TIDE AND VELOCITY COMPUTATIONS.

Points A and B: see Figure 11a and Table 3

Points C and D: see Figure 11b and Table 3

Points E and F: see Table 3

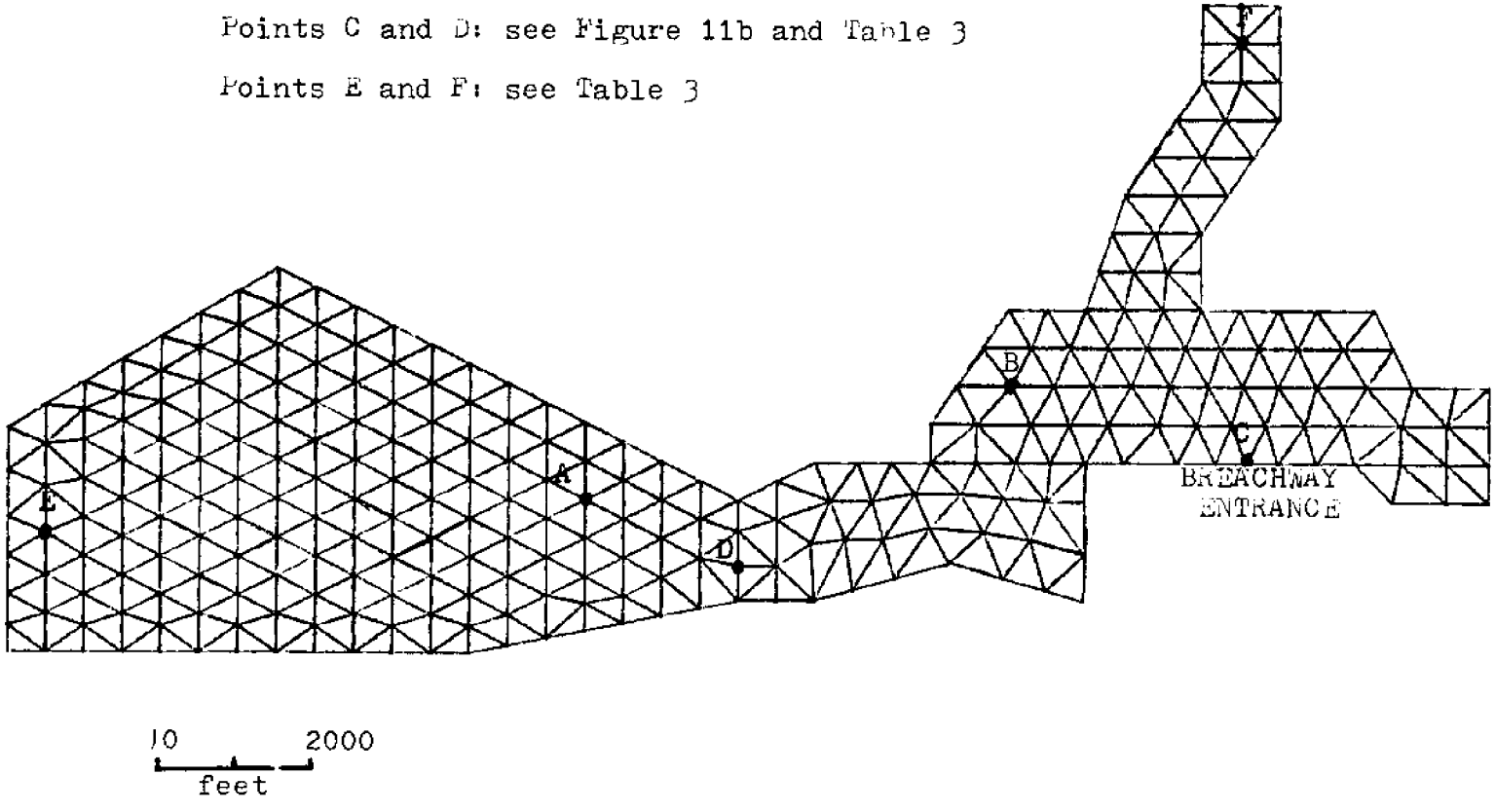


Fig. 4. Finite element network for Ninigret Pond
(see Fig. 1 for comparison map).

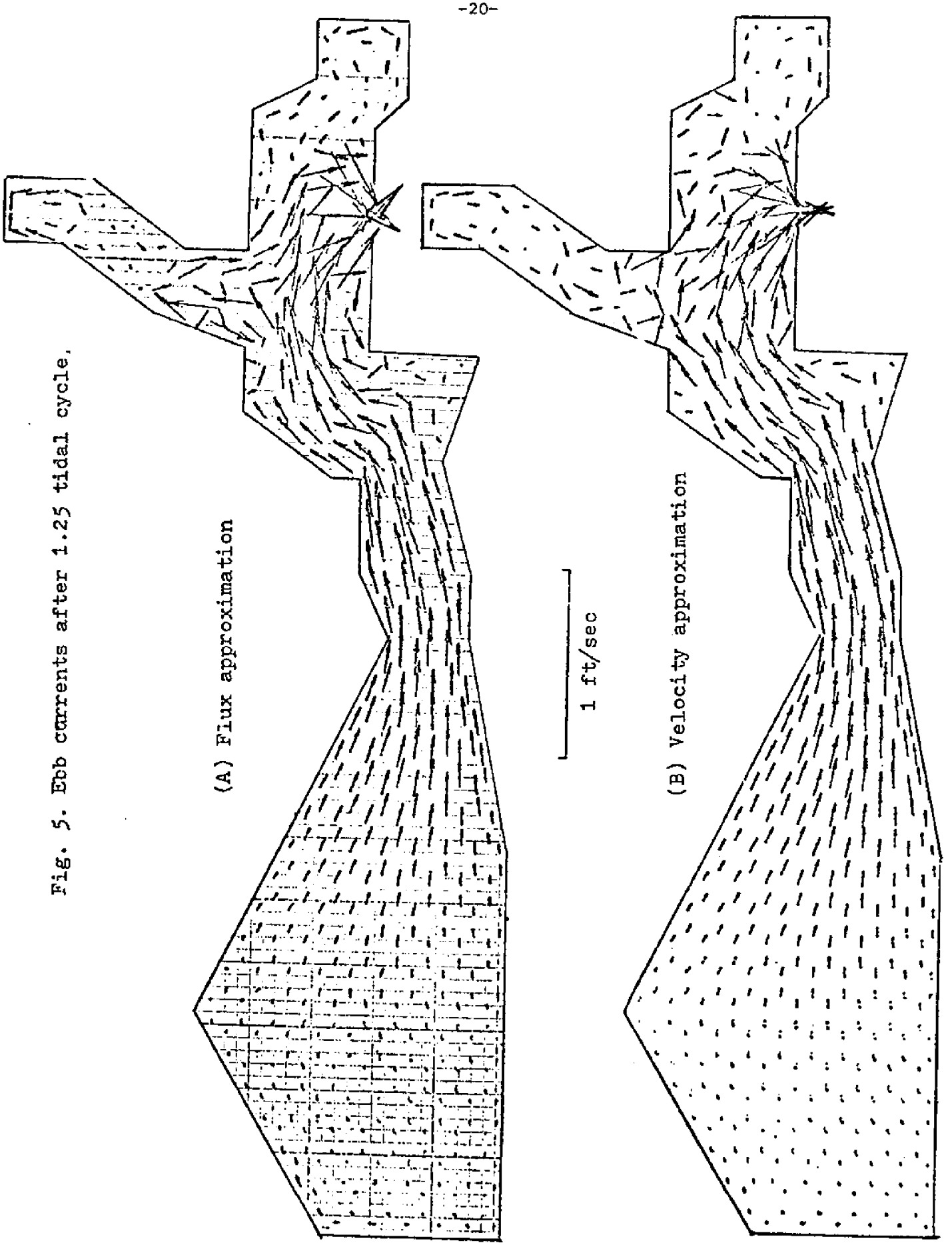


Fig. 5. Ebb currents after 1.25 tidal cycle.

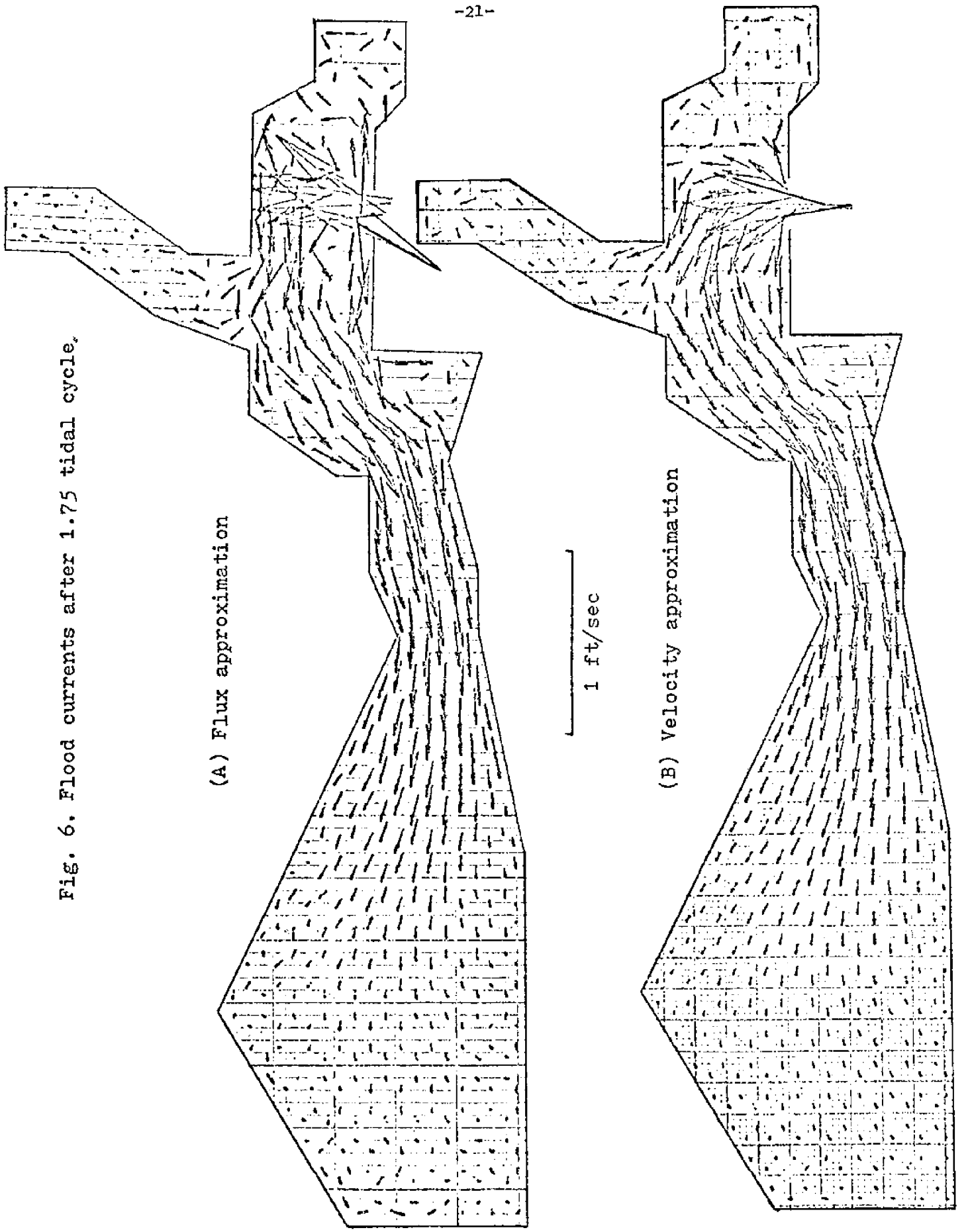


Fig. 6. Flood currents after 1.75 tidal cycle.

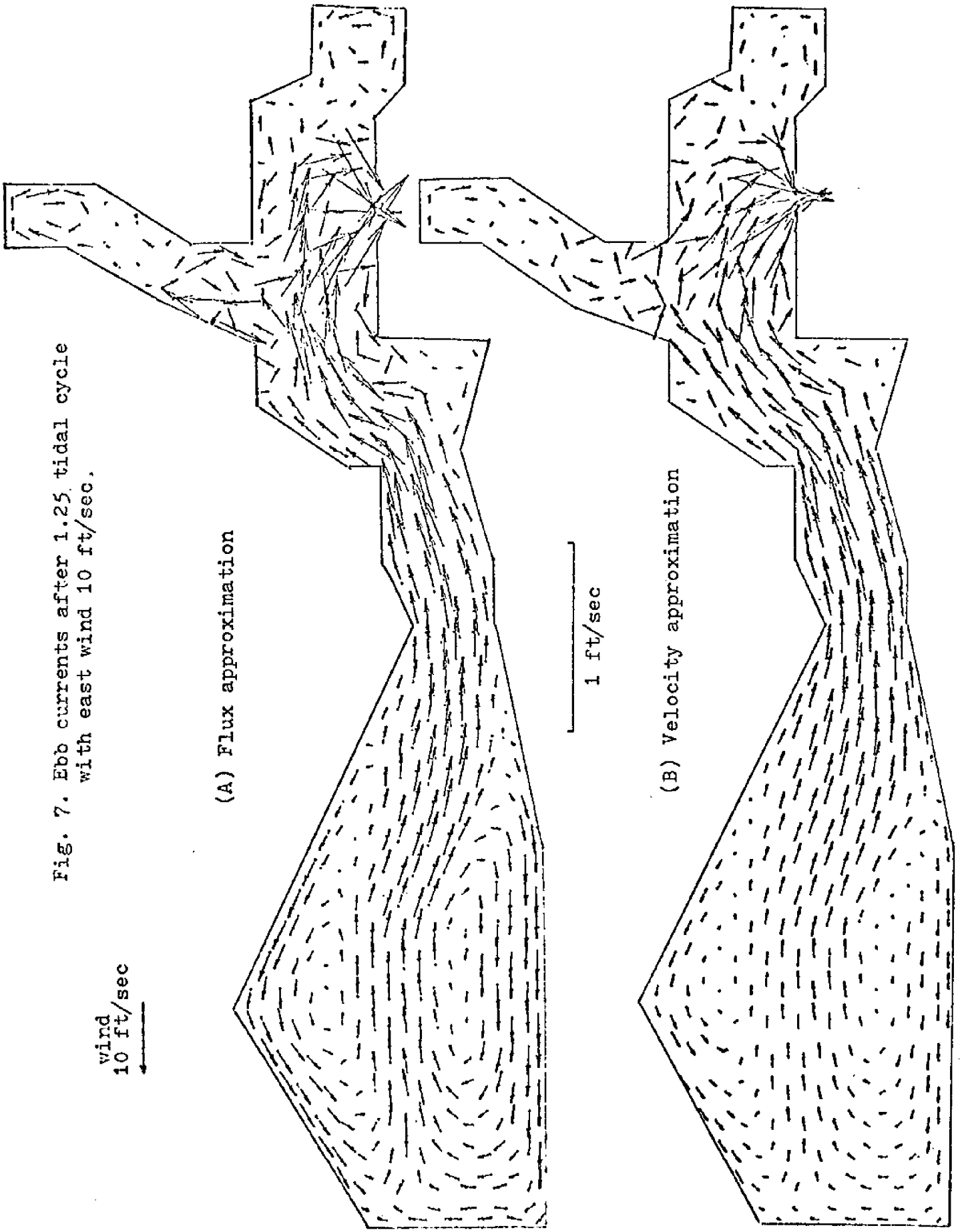


Fig. 7. Ebb currents after 1.25 tidal cycle with east wind 10 ft/sec.

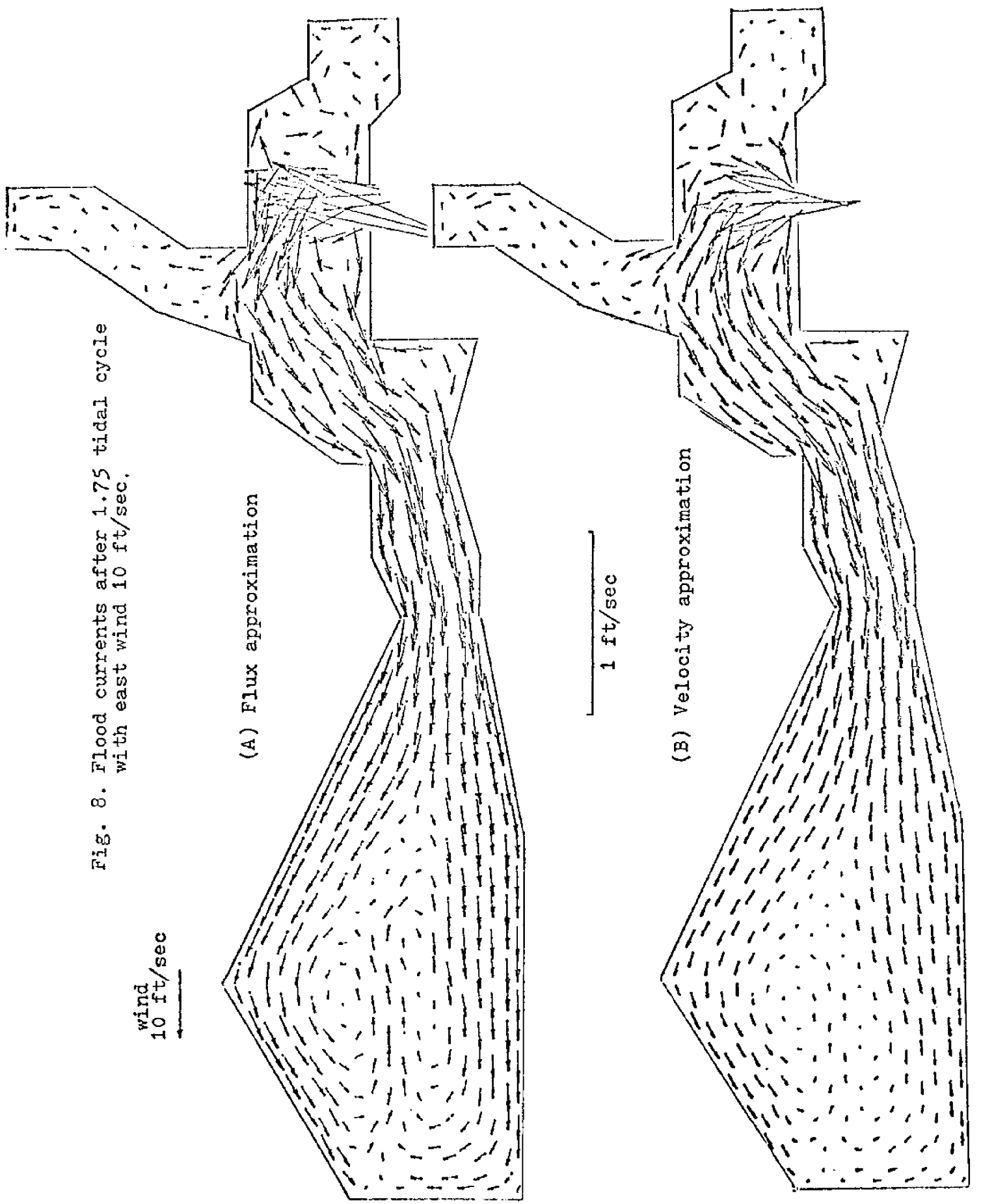


Fig. 8. Flood currents after 1.75 tidal cycle with east wind 10 ft/sec.

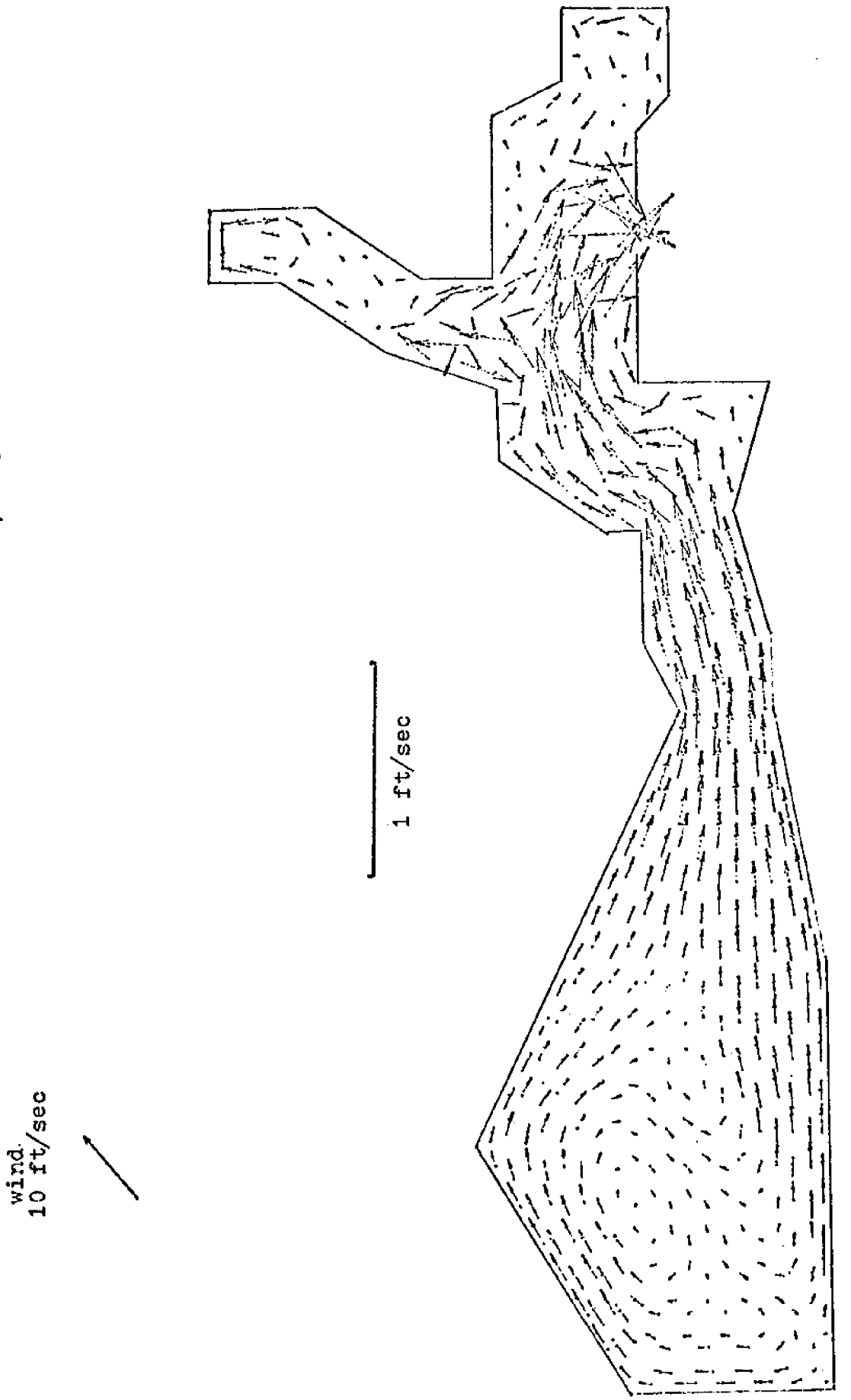
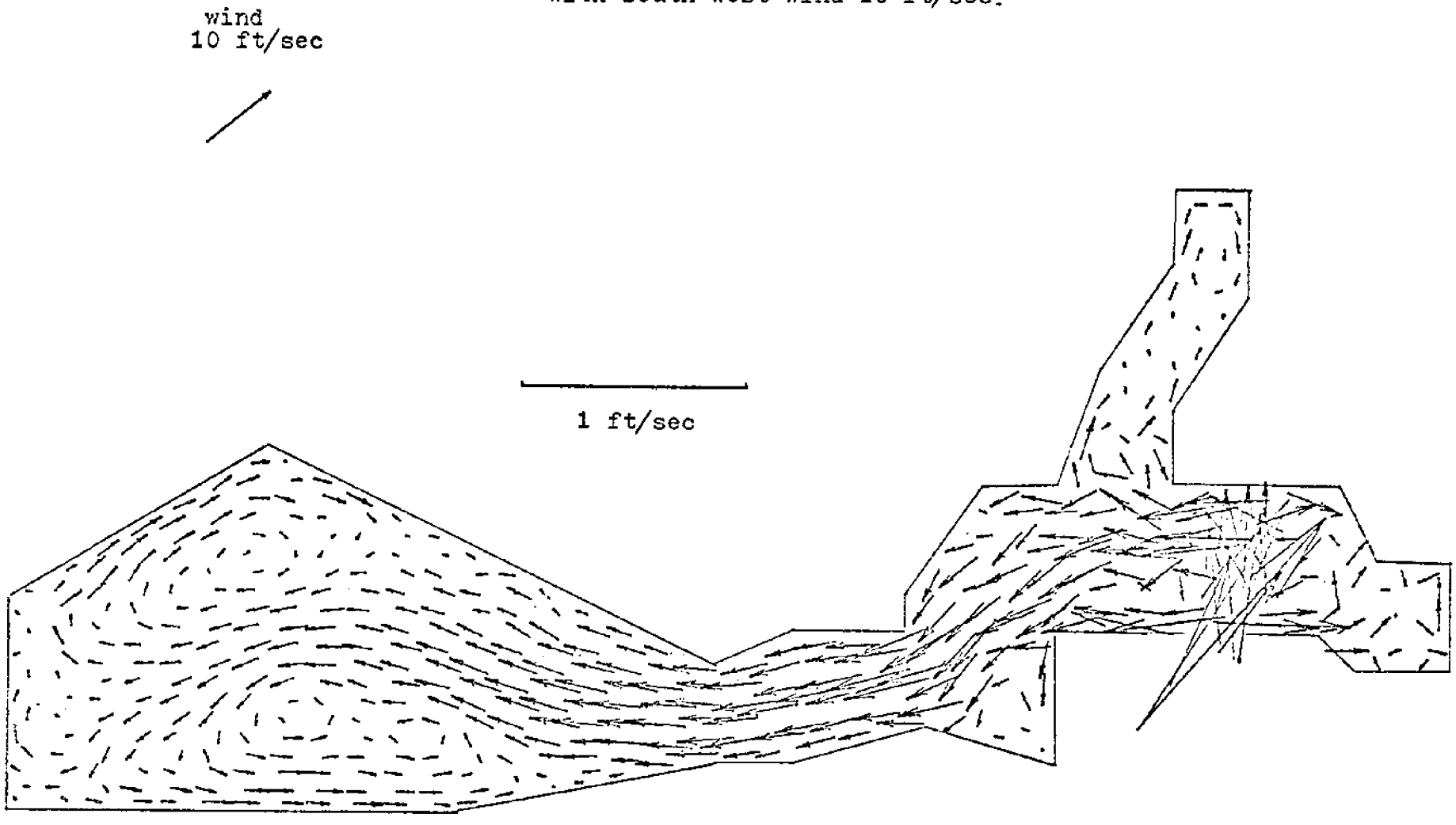


Fig. 9. Ebb currents after 1.25 tidal cycle with south-west wind 10 ft/sec.

Fig. 10. Flood currents after 1.75 tidal cycle
with south-west wind 10 ft/sec.



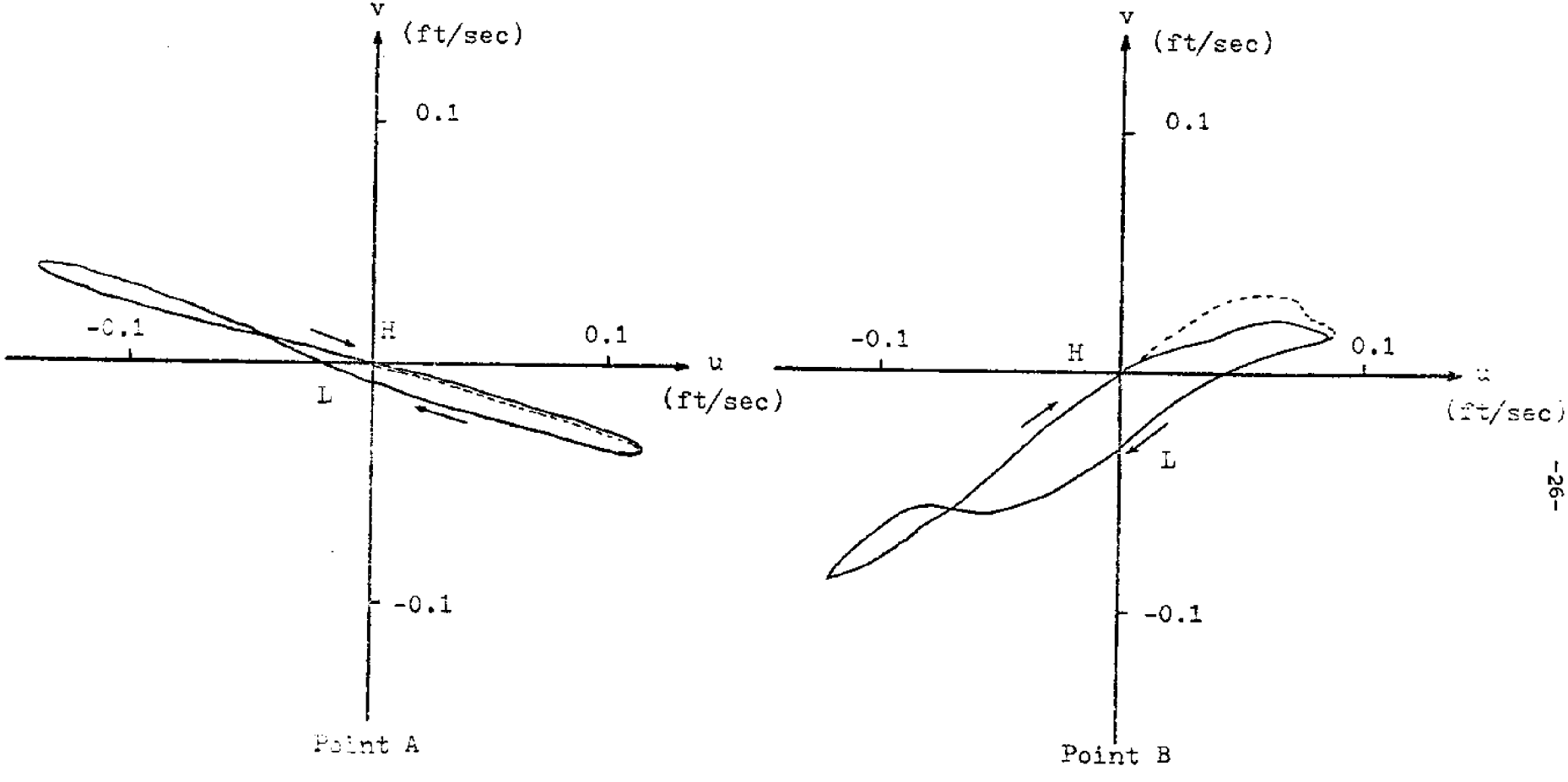


Fig. 11a Computed currents at Points A and B

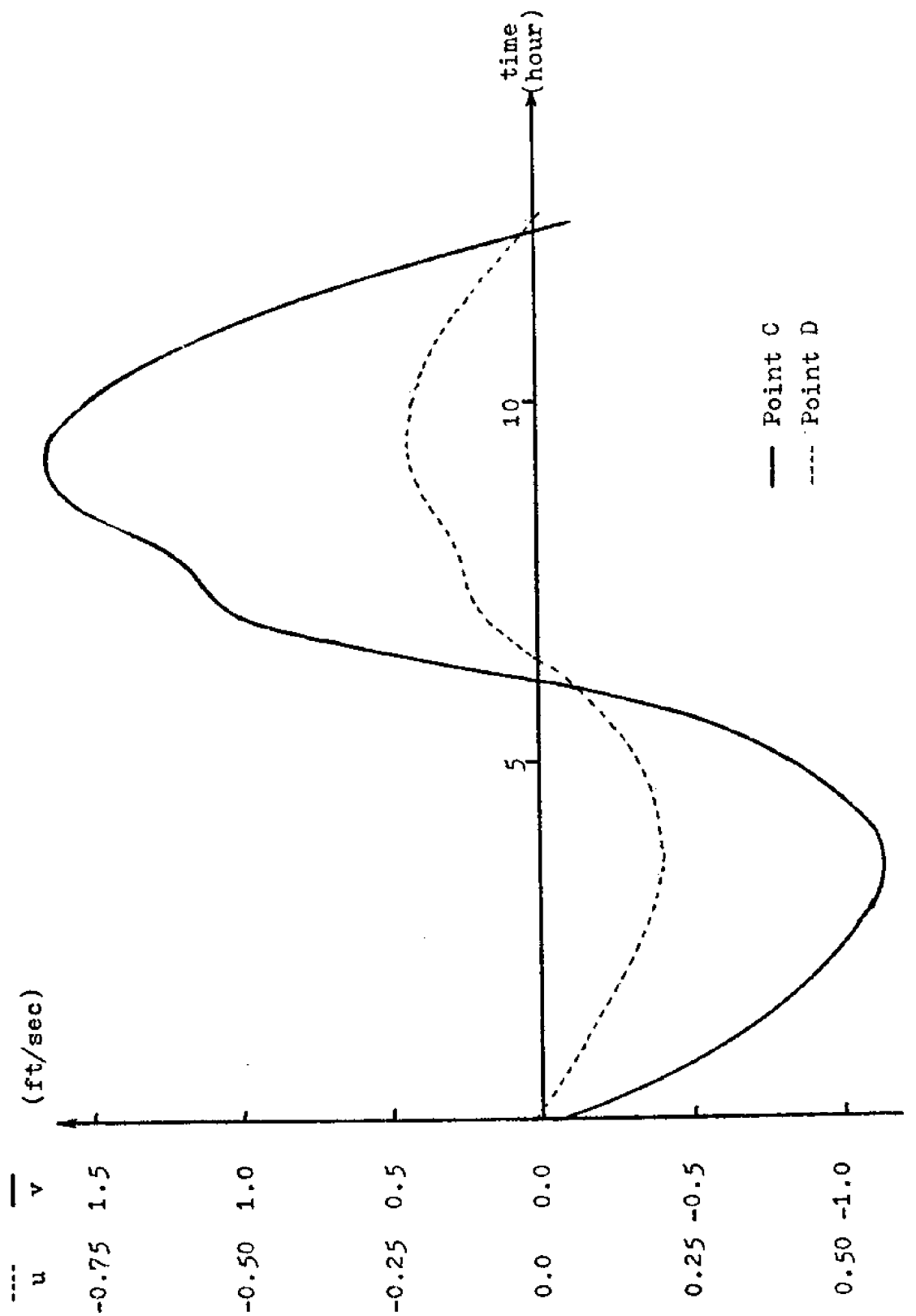


Fig. 11b, Computed currents at Points C and D.

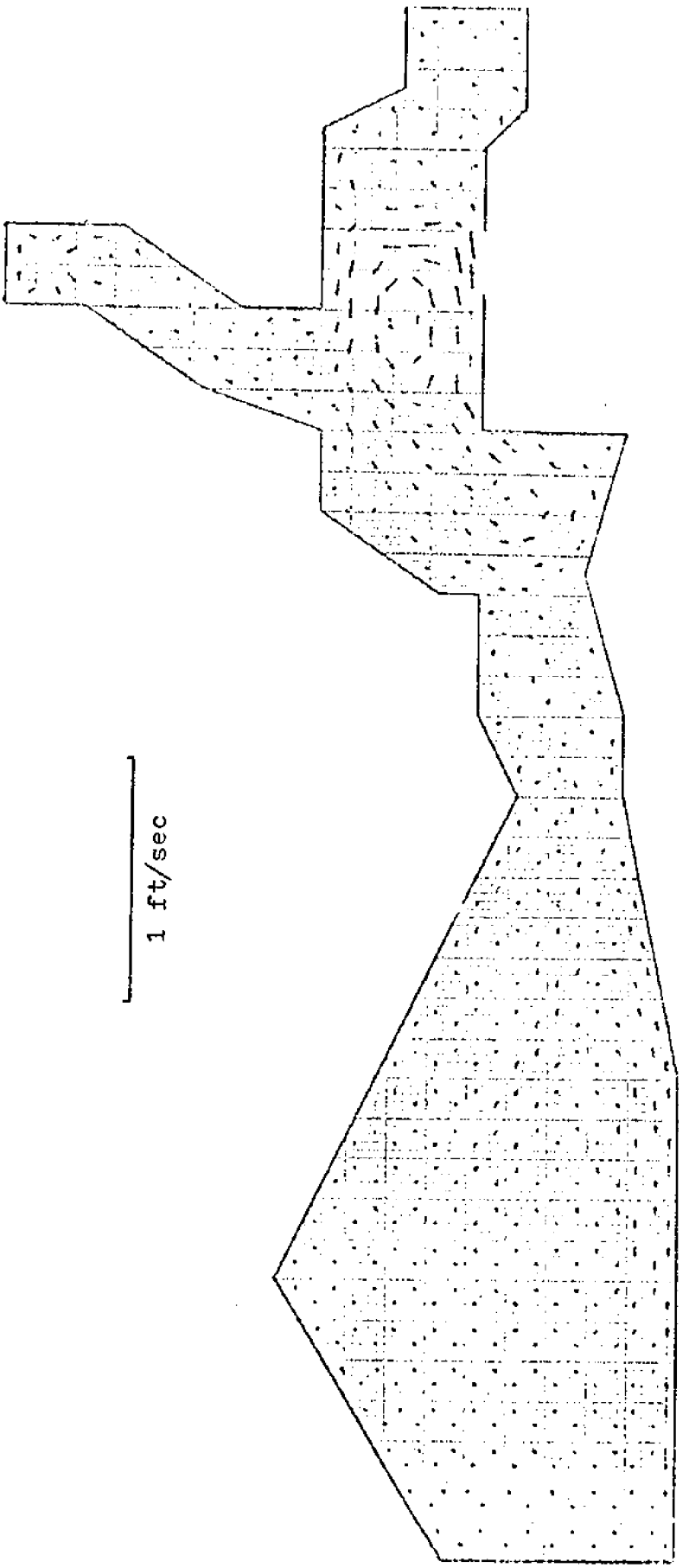


Fig. 12. Computed currents after 1 tidal cycle

Fig. 13. Computed currents after 1.1 tidal cycle

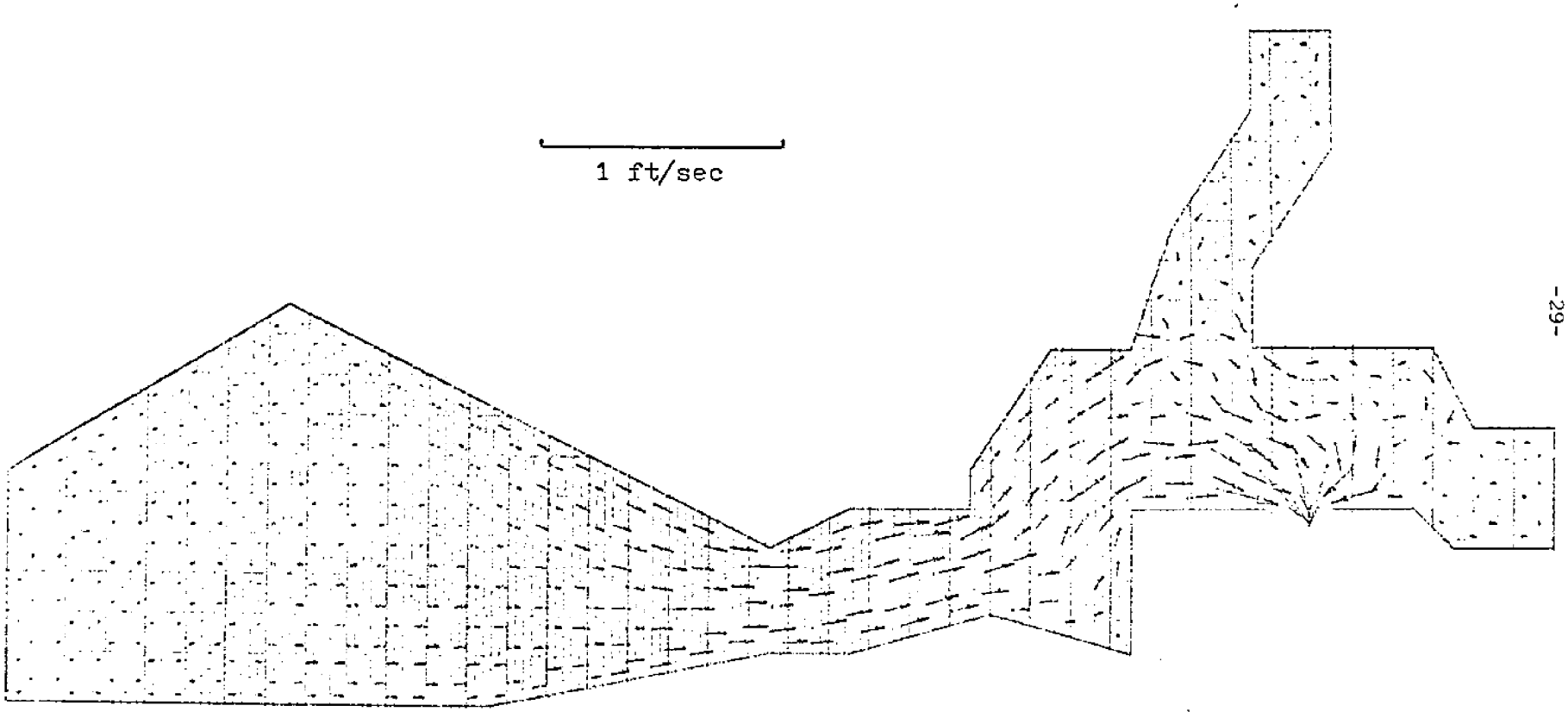


Fig. 14 Computed currents after 1.2 tidal cycle

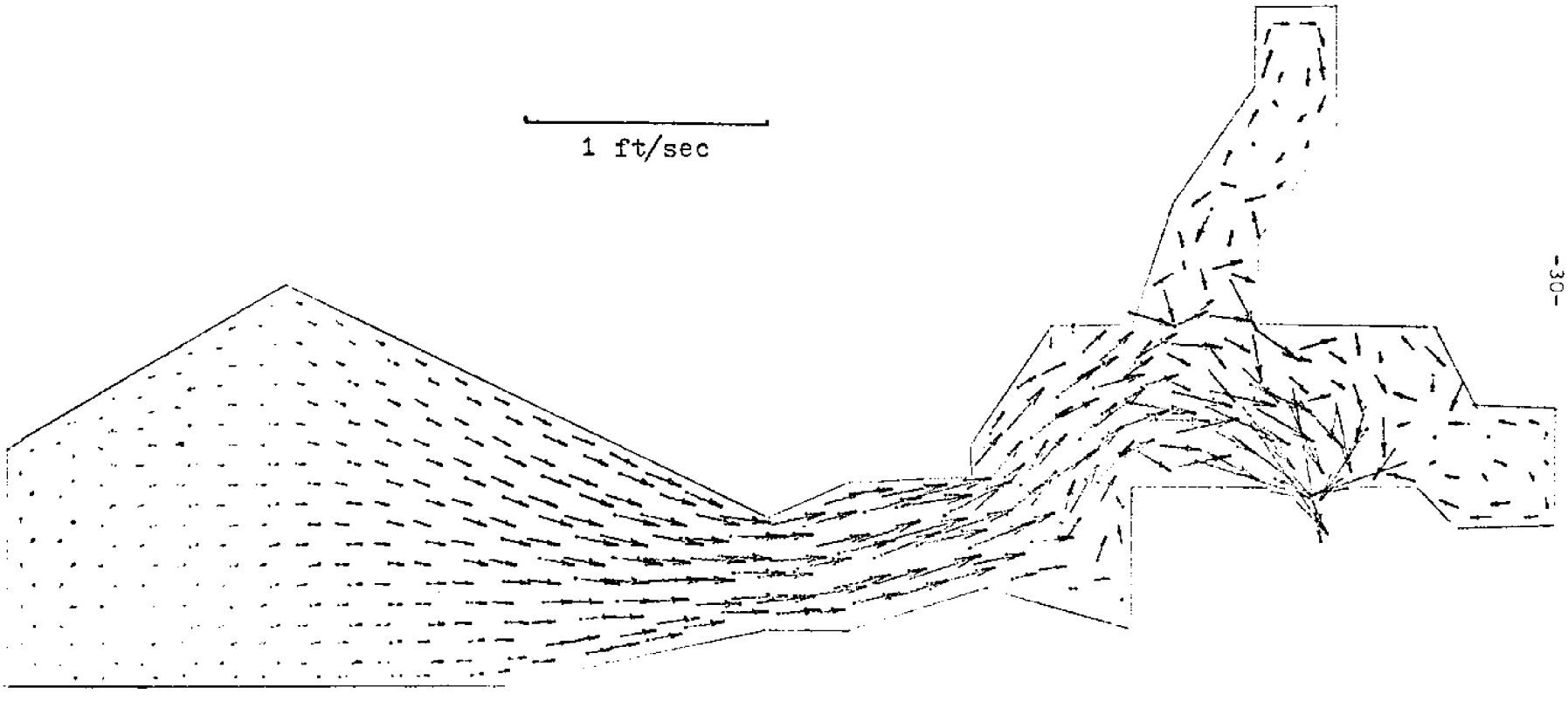
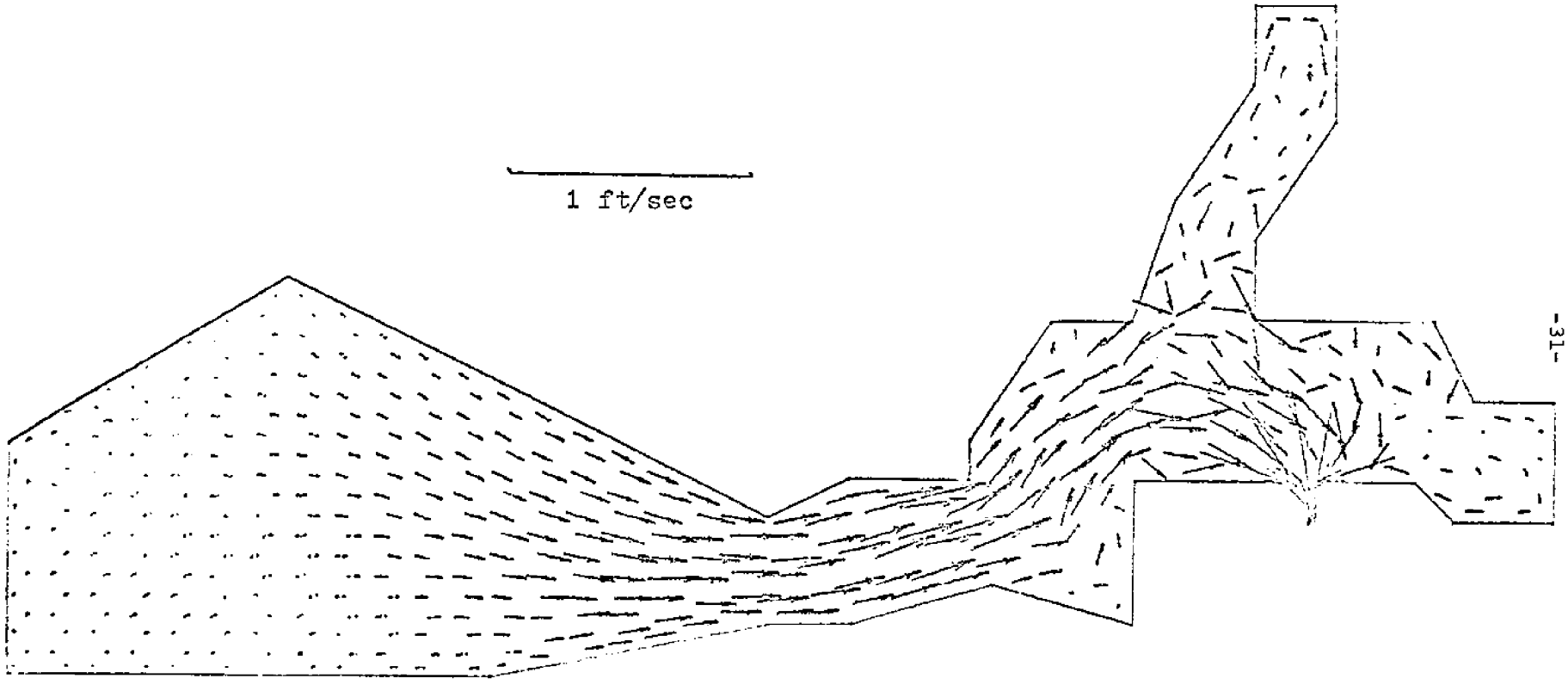


Fig. 15. Computed currents after 1.3 tidal cycle



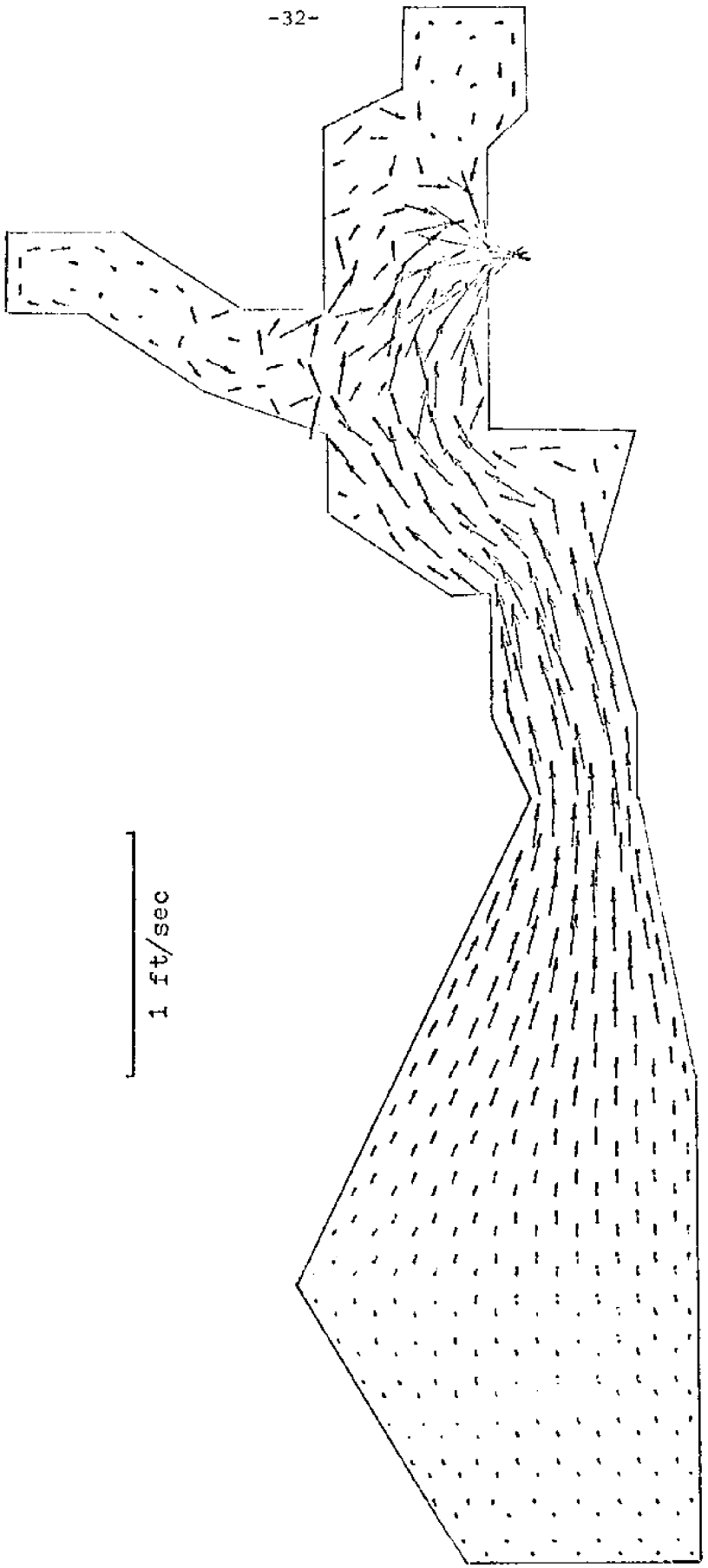
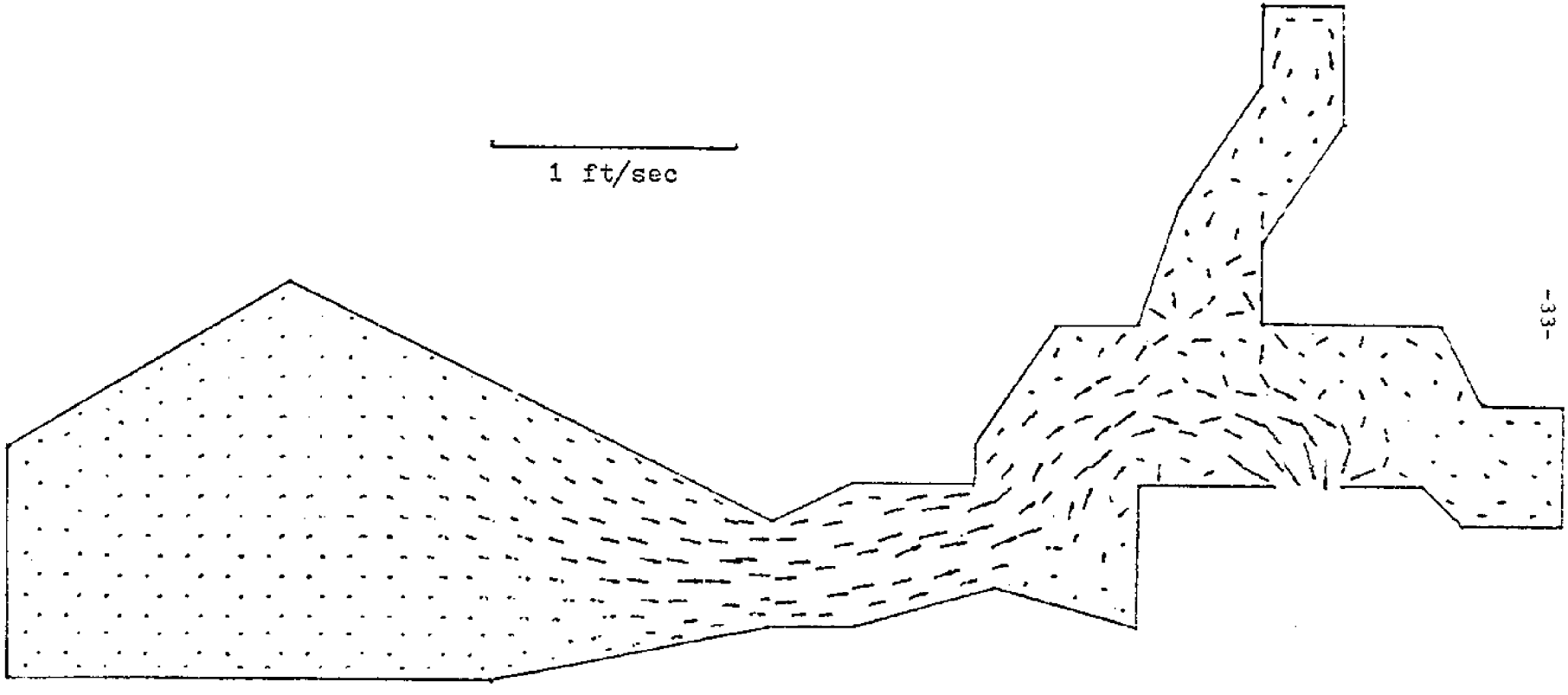


Fig. 16. Computed currents after 1.4 tidal cycle

Fig. 17. Computed currents after 1.5 tidal cycle



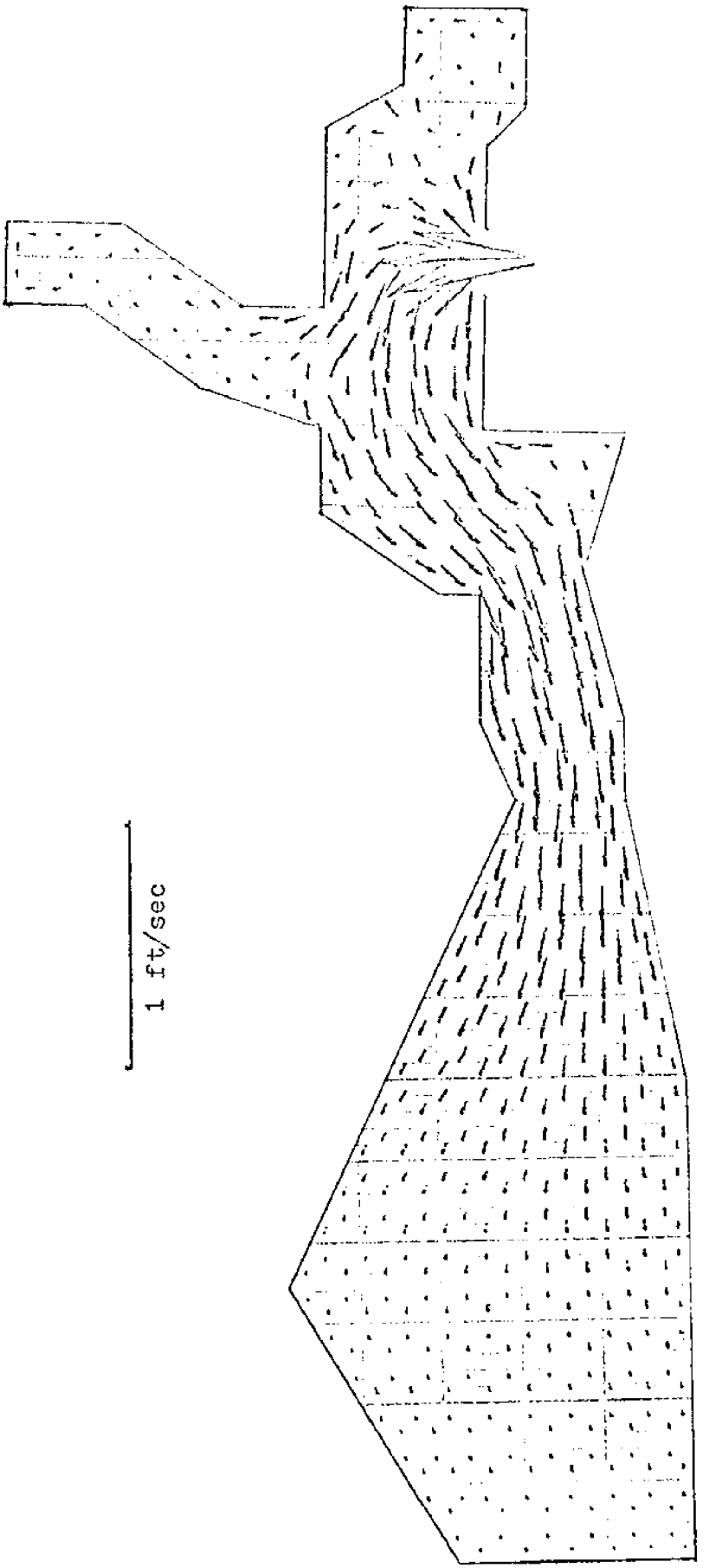


Fig. 18. Computed currents after 1.6 tidal cycle

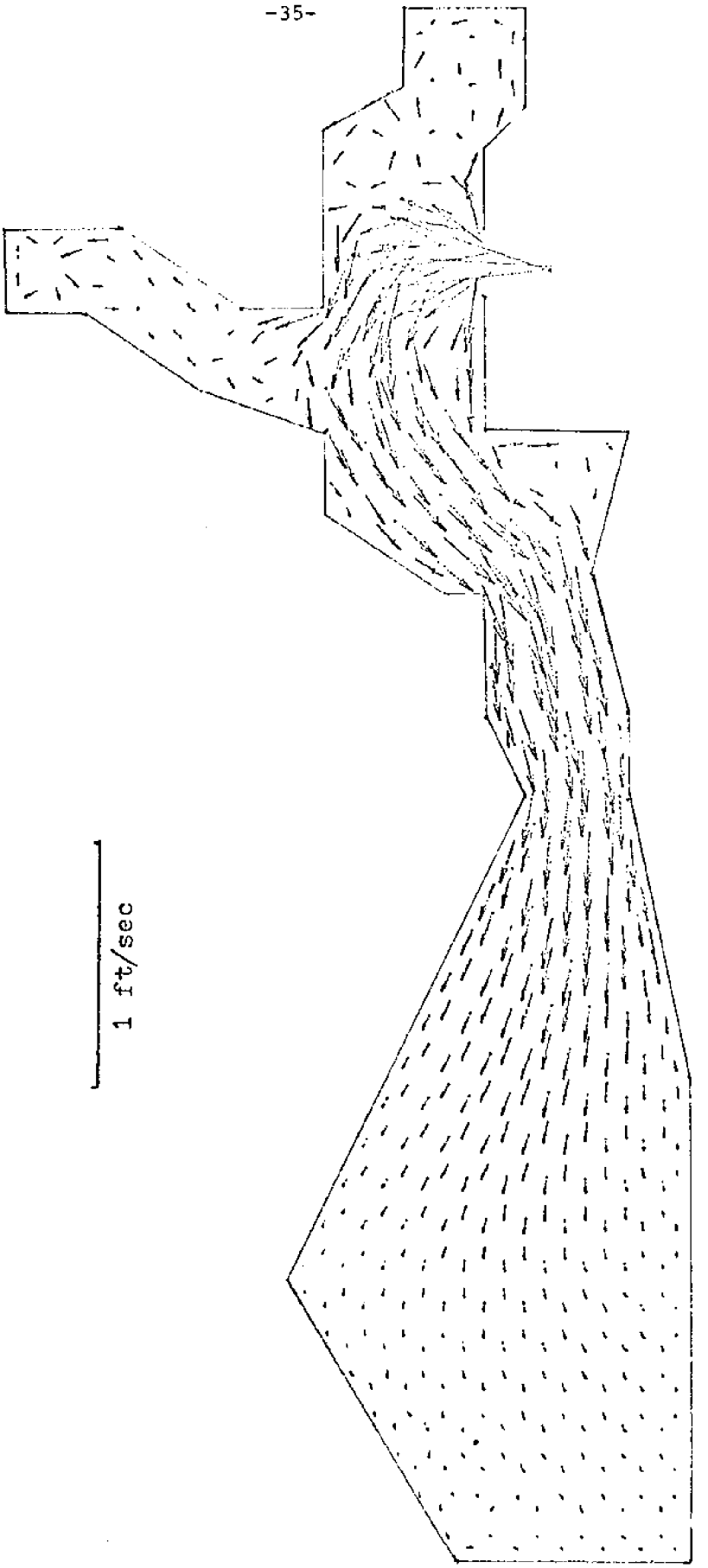


Fig. 19. Computed currents after 1.7 tidal cycle

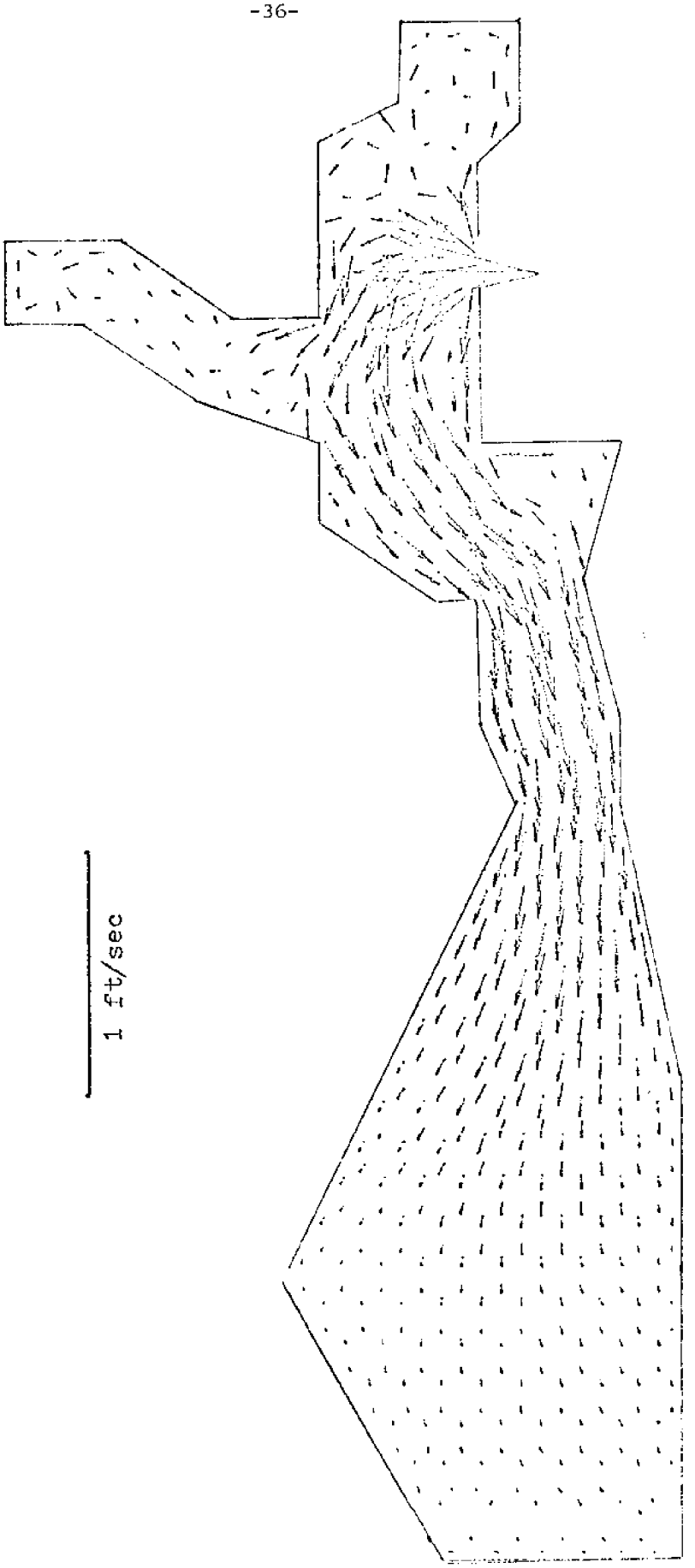


Fig. 20. Computed currents after 1.8 tidal cycle

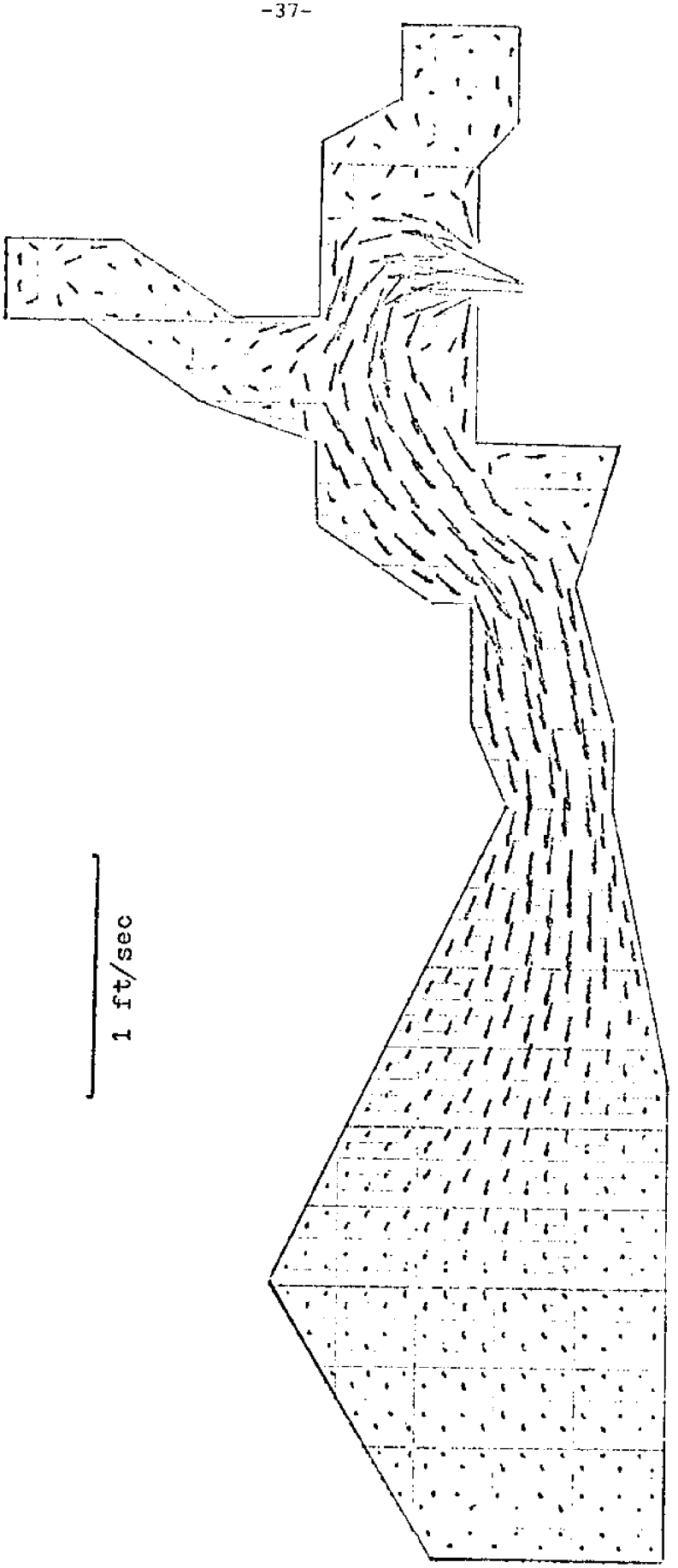
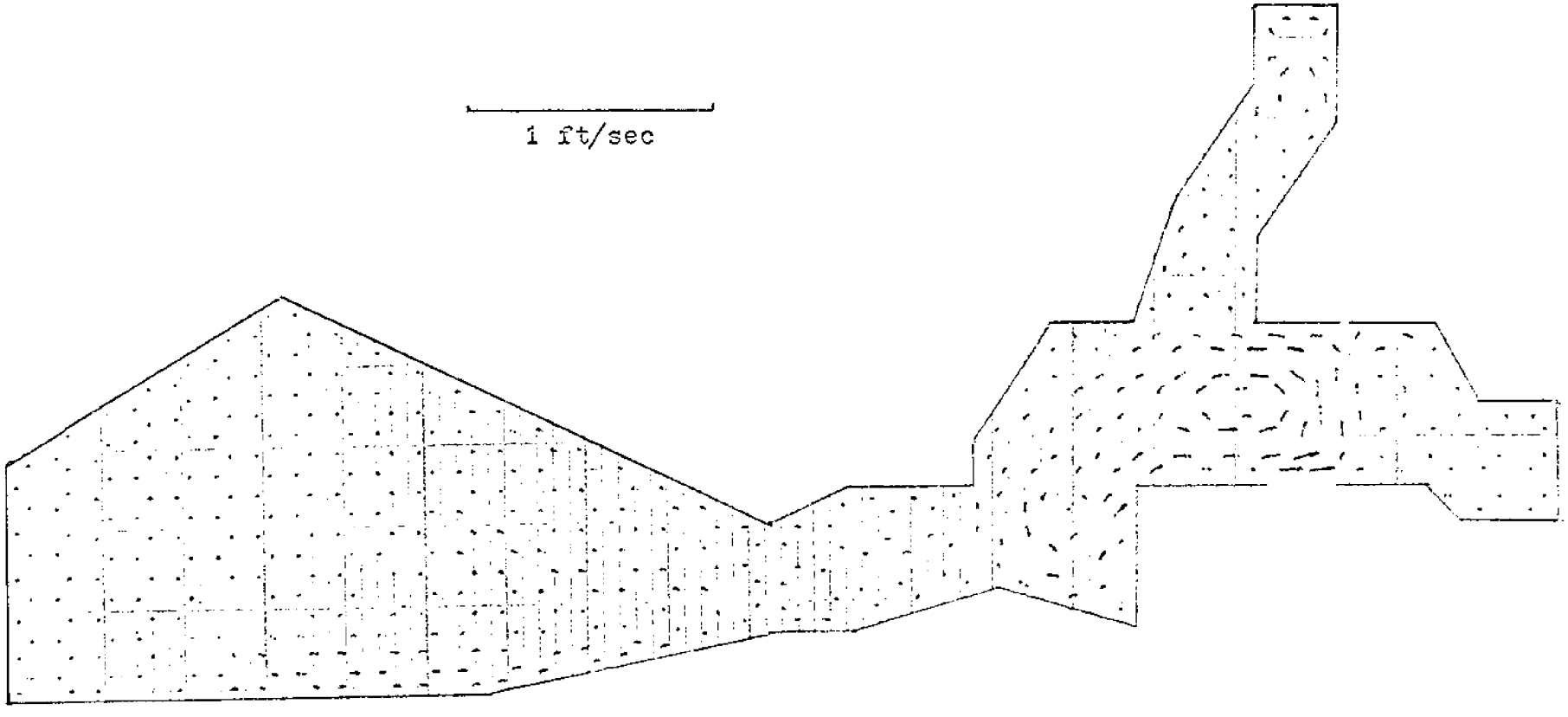


Fig. 21. Computed currents after 1.9 tidal cycle

Fig. 22. Computed currents after 2 tidal cycle



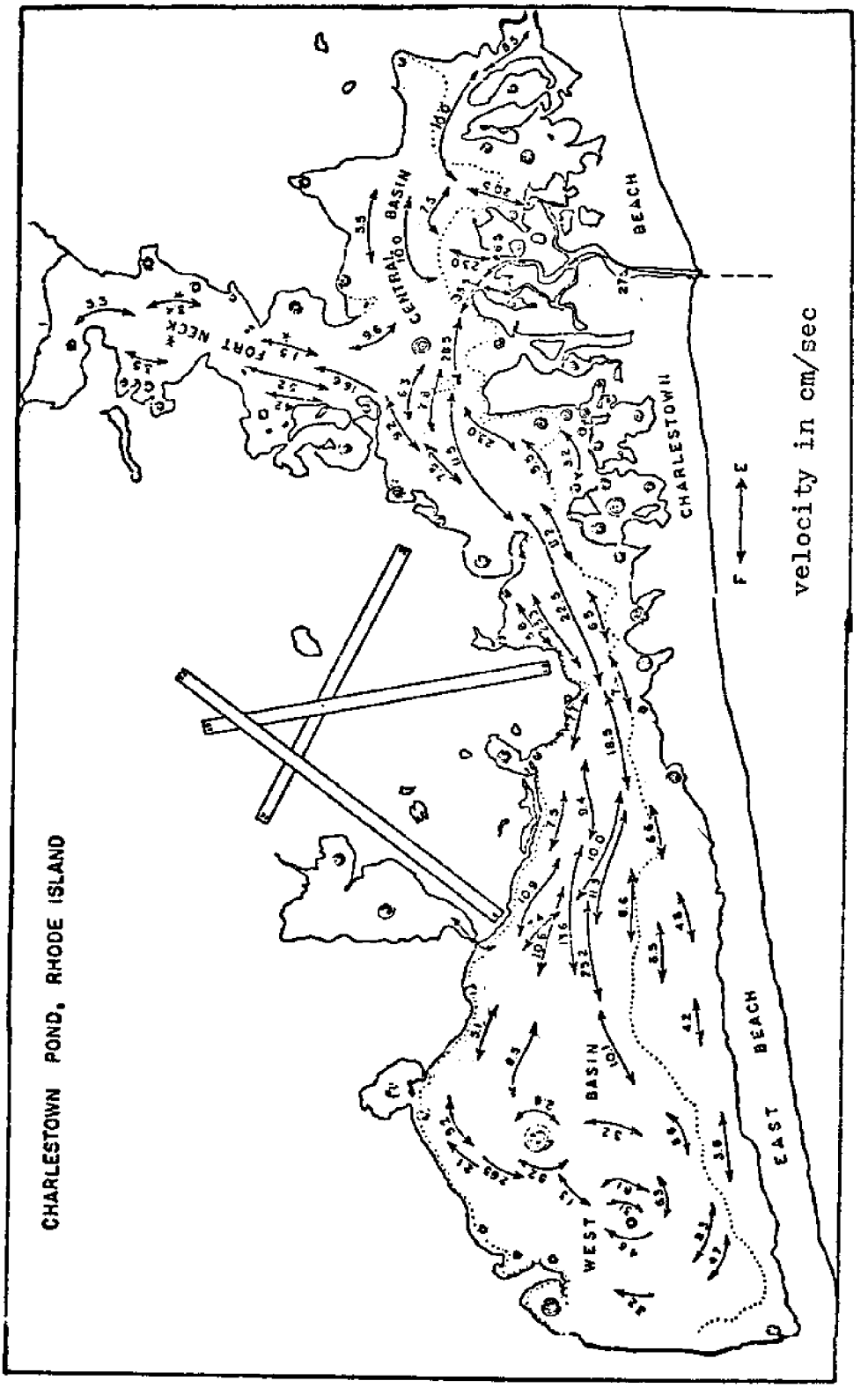


Fig. 23 Measured maximum currents reported by reference (13).

CHAPTER 5. PROGRAM DESCRIPTION AND USER'S GUIDE

5.1 PROGRAM DESCRIPTION

The computer program presented here is composed of three stages: data input, matrix formation, and numerical integration. The flow chart is given in Fig. 24.

First, five sets of data input are required to be read in:

1. computational parameters
2. nodal coordinates and depths
3. nodal connections in each element
4. land boundary conditions
5. open boundary conditions

Then, the global matrix [M] is assembled by adding each element coefficient matrix. The typical matrix contains many zeroes beyond the band width. A compact rectangular array storing only those banded elements is economical for both storage requirements and arithmetic calculations.

The systems of equations are solved by Cholesky's method, which is very efficient for banded matrices. Let AX = B be the system of equations. The matrix should be decomposed into the product A = LU where L is a lower triangular matrix, and U is a upper triangular matrix, then AX = B becomes LUX = B. The system can be solved as follows: Let UX = Y, LUX = B becomes LY = B. Thus, there are two systems to be solved. First LY = B, second UX = Y.

The advantage of this method is that the LU form retains the banded structure while other methods need full matrix storage. The LU factorization is needed only once in a given unsteady problem.

The computational speed of LU decomposition and back substitution is quite

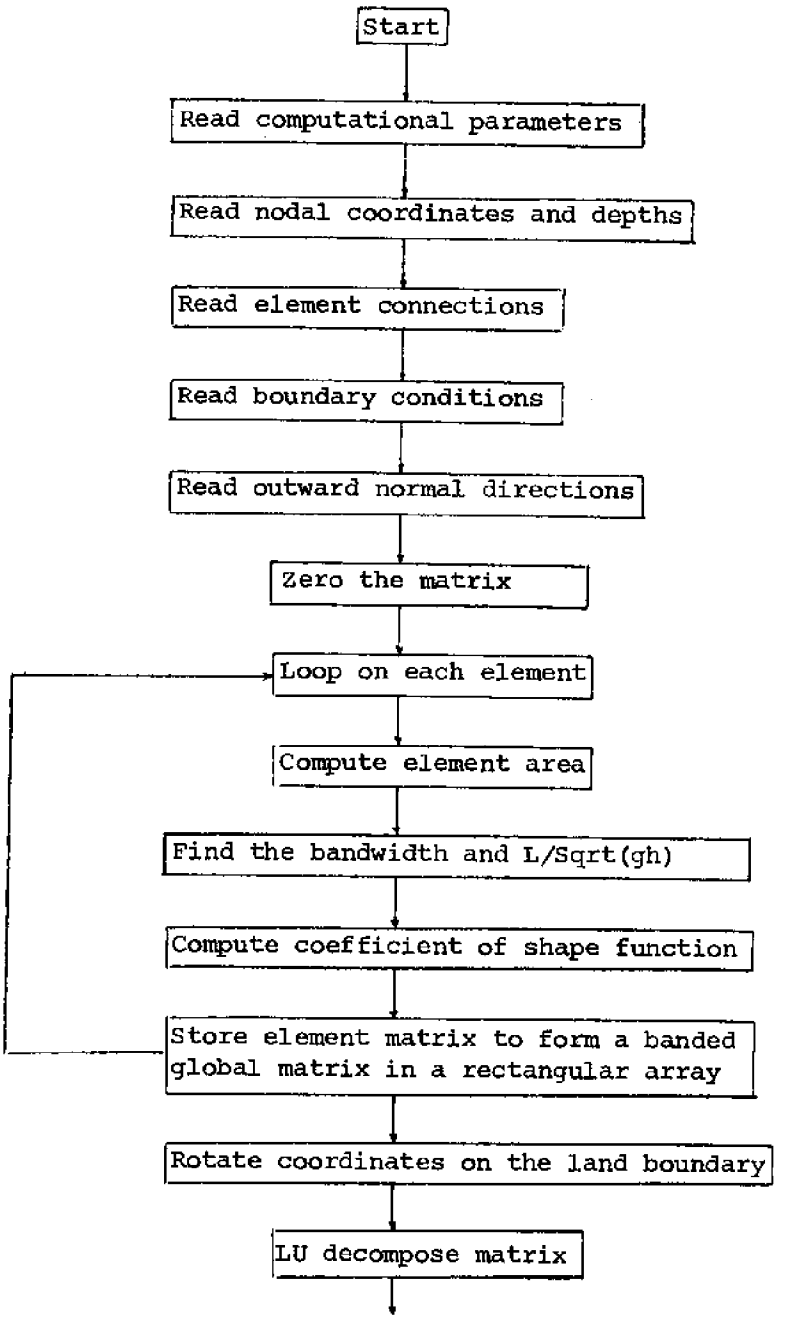


Fig. 24 Flow Chart of Computational Procedure

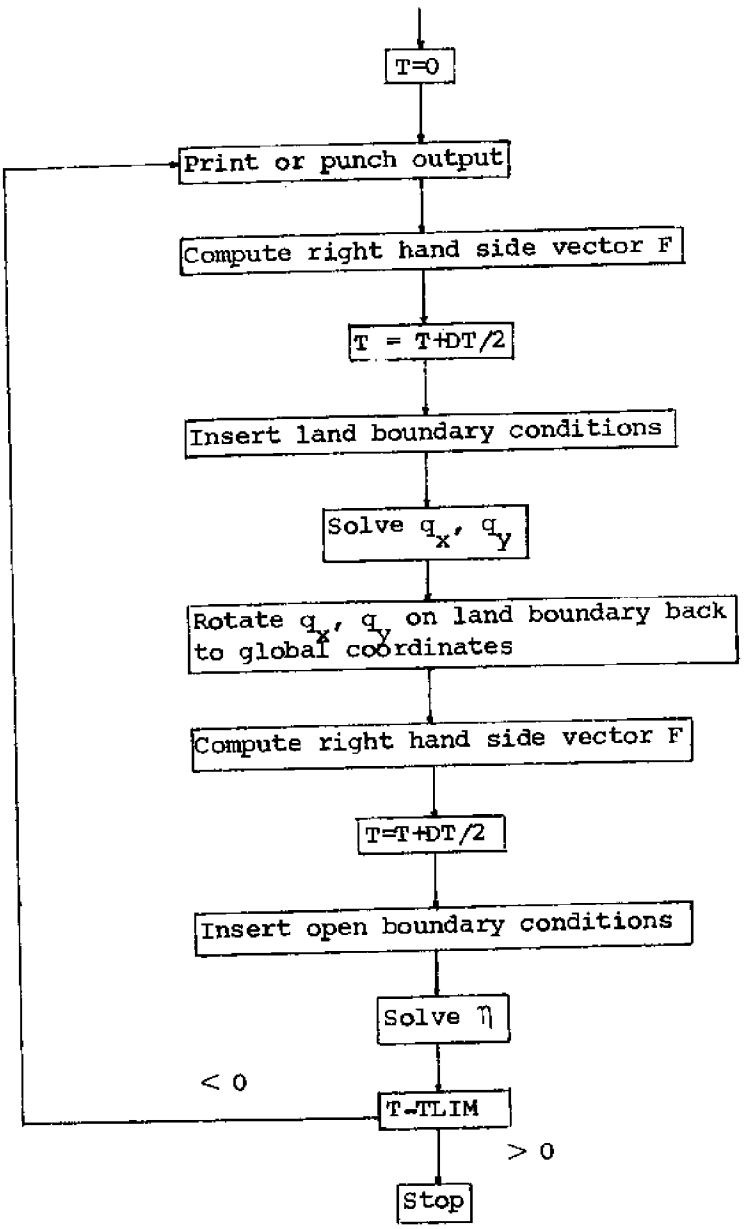


Fig. 24 Flow Chart of Computational Procedure (cont.)

fast. Presently, a 461-element system with 284 nodes and 3 unknowns at each node, requires 1.4 seconds for every time step on an IBM 370-155 computer. The H compiler generally needs more storage but less computing time.

5.2 USER'S GUIDE

There are ten sets of read statements (input) in the program. Each parameter in the statement is defined as follows:

STATEMENT

1. (3I10) NN, NM, NW

   NN: number of nodes

   NM: number of elements

   NW: bandwidth of matrix M

2. (4F10.0) T, TLIM, DT, WW

   T: start time (sec)

   TLIM: end time (sec)

   DT: time increment (sec)

   WW: period of tide (sec)

3. (4F10.1) ATTD, HA, WX, WY

   ATTD: latitude (degree)

   HA: amplitude at open boundary (ft)

   WX: x-component of wind (ft/sec)

   WY: y-component of wind (ft/sec)

4. (2F10.0) STEPO, PUNCH

   STEPO: print output for every STEPO step

   PUNCH: punch output for every PUNCH step

5. (2F10.0) UNIT, UNAR

   UNIT: unit of length

   UNAR: unit of area

6. (I4, I1, 2F10.5, F3.1) MEXT(I), MB(I), X(I), Y(I), H(I)

MEXT(I): external node number

MB(I): boundary condition type

= 0 interior node

= 1 land boundary

= 2 open boundary

= 3 connecting node of land and open boundary

= 4 zero velocity

X(I): X-coordinate

Y(I): Y-coordinate

H(I): depth

7. (3I3) N1(I), N2(I), N3(I)

3 node numbers in each element

8. (16I5) MHNO, (MH(I), I = 1, MHNO)

MHNO: number of nodes on open boundary

MH(I): node number

9. (16I5) MLNO, (MLE(I), I = 1, MLNO)

MLNO: number of nodes on land boundary

MLH(I): node number

10. (8F10.4) (ANG(I), I = 1, MLNO)

ANG(I): rotation angle

The complete computer program is given in Appendix B. Two separate programs, one for the outward normal angle at a corner point and the other for plotting the results, are listed in Appendices C and D. The program for corner point angle is based on the bisection method. The plotting program consists of two subroutines, ARROW and VECTOR, which are available in the University of Rhode Island computer system for general plotting services.

CHAPTER 6. CONCLUSIONS AND RECOMMENDATIONS

The materials presented in the preceding chapters has shown that the hydrodynamic characteristics of coastal waters can be modeled successfully by the finite element method. From the results of the application at Ninigret Pond, the following conclusions are drawn:

1. A refined finite element mesh in areas of complex geometry is extremely important to the accuracy of the global results.
2. Wind generated gyres can be clearly visualized in the computed currents. Wind conditions in shallow water areas cannot be neglected.

It is recommended that:

1. A detailed investigation of stability conditions be conducted to give information for constructing the finite element network properly.
2. An automatic renumbering subroutine to decrease the bandwidth of the matrix be included in the main program.
3. A further and more extensive comparison between the finite element method and the finite difference method is necessary to justify relative advantages and disadvantages.

REFERENCES

1. Leendertse, J.J., Aspects of a Computational Model for Long-Period Water-Wave Propagation. Rand Corporation, Santa Monica, California, Memorandum RM 5294-PR, May, 1967.

2. Hess, K.W. and White, F.M., A Numerical Tidal Model of Narragansett Bay. University of Rhode Island, Marine Technical Report No. 20, 1974.

3. Grotkop, G., "Finite Element Analysis of Long-Period Water Waves," Computer Methods in Applied Mechanics and Engineering. 2, P. 147-157, 1973.

4. Connor, J.J. and Wang, J.D., Mathematical Models of the Massachusetts Bay. Part I. Massachusetts Institute of Technology, Report No. MITSG 74-4, October, 1973.

5. Finlayson, B. A., The Method of Weighted Residuals and Variational Principles. Academic Press, 1972.

6. McIver, D.B., The Prediction of Pollutant Transport in Estuaries: Variational Principles. Unpublished report, Department of Naval Architecture, University of Strathclyde, October, 1973.

7. Oden, J.T. and Oliveira, E.R.A., Lectures on Finite Element Methods in Continuum Mechanics. The University of Alabama Press, 1973.

8. Zienkiewicz, O.C., The Finite Element Method in Engineering Science. McGraw-Hill Book Co., London, 1971.

9. Baker, A.J. and Zelaxyn, S.W., Predictions in Environmental Hydrodynamics Using the Finite Element Method. AIAA 12th Aerospace Sciences Meetings, Paper No. 74-7, 1974.

10. Norton, W.R., King, I.P. and Orlob, G.T., A Finite Element Model for Lower Granite Reservoir. Water Resources Engineers, Walnut Creek, California, Report DACW 68-71-C-0047, 1973.

11. Ralston, A., A First Course in Numerical Analysis. McGraw-Hill Book Co., New York, 1965.

12. Conte, S.D. and Carl de Boor, Elementary Numerical Analysis, second edition. McGraw-Hill Book Co., New York, 1972.

13. Conover, J.T., Environmental Relationships of Benthos in Salt Ponds, Vol. II. University of Rhode Island Graduate School of Oceanography, Technical Report No. 3, 1964.

14. Short, F.T., Nixon, S.W. and Oviatt, C.A., "Field Studies and Simulations with a Fine Grid Hydrodynamic Model." In An Environmental Study of a Nuclear Power Plant at Charlestown, Rhode Island, University of Rhode Island Marine Technical Report 33, September, 1974.

APPENDIX A. ROTATION OF THE COORDINATES

The coordinate transformation matrix $[T]$ is

$$[T] = \begin{vmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{vmatrix}$$

and the transpose form $[T]^T$ is

$$[T]^T = \begin{vmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{vmatrix} \qquad [T]\,[T]^T = [I]$$

where θ is the rotation angle which is between the outward normal of the land boundary node and the global x-coordinate.

The momentum equation $[M_q]\,\{\dot{q}\} = \{F_q\}$ is in global coordinates originally. After multiplying $[T]$ and $[T]^T$ on the rows and columns of boundary nodes, the multicoordinate system is formed.

$$[T]^T \begin{vmatrix} [\ m_l\ ] & & \begin{bmatrix} [T] \\ [m_l] \end{bmatrix}^T & & \text{O} \\ & [m] & \\ \text{O} & & \begin{bmatrix} [m_r] \end{bmatrix}^T \end{vmatrix} \begin{vmatrix} [\ m_l\ ] & & \text{O} \\ & [m] & \\ \text{O} & & [\ m_r\ ] \end{vmatrix} [T]^T \begin{Bmatrix} \dot{q}_{up} \\ \dot{q}_{xb} \\ \dot{q}_{yb} \\ \dot{q}_{lo} \end{Bmatrix} = [T]^T \begin{Bmatrix} F_{up} \\ F_{xb} \\ F_{yb} \\ F_{lo} \end{Bmatrix} \qquad (1)$$

The equation representing the particular boundary point illustrated in equation (1) becomes

$$[T]^T [\ m_l\ ] \{\dot{q}_{up}\} + [T]^T [m] [T] [T]^T \begin{Bmatrix} \dot{q}_{xb} \\ \dot{q}_{yb} \end{Bmatrix}$$

$$+ [T]^T [\ m_r\ ] \{\dot{q}_{lo}\} = [T]^T \begin{Bmatrix} F_{xb} \\ F_{yb} \end{Bmatrix} \qquad (2)$$

The boundary condition is substituted into the above equation (2) for local coordinates. The rotation procedure does not change the physical structure of the original problem. In fact, the calculation of $[T]^T [m] [T]$ in Eqn. (2) is not necessary, because if

$$[m] = \begin{vmatrix} m_{bb} & 0 \\ 0 & m_{bb} \end{vmatrix} = m_{bb} [\ ]$$

then

$$[T]^T [m] [T] = m_{bb} [T]^T [I] [T] = m_{bb} [I]$$

It is observed that equation (2) is identical to the global equation after $[T]^T$ is cancelled in every term :

$$\text{Let} \quad \begin{Bmatrix} \dot{q}'_{xb} \\ \dot{q}'_{yb} \end{Bmatrix} = [T]^T \begin{Bmatrix} \dot{q}_{xb} \\ \dot{q}_{yb} \end{Bmatrix}$$

The solved value of $\dot{q}'_{xb}$ and $\dot{q}'_{yb}$ should be rotated back to global coordinates for the next step calculation.

```
C
C
C
C  APPENDIX B.  MAIN COMPUTER PROGRAM
C
C
      DIMENSION X(284),Y(284),H(284),B1(568,51),B2(284,25),
     *      U(284),V(284),ETA(284),UVP(568),EP(284),MEXT(284),
     *      MINT(286),R1(568),R2(284),A(9,461),Q(568),MB(284),
     *      CMAN(461),CHEZ(461),AREA(461),N1(461),N2(461),N3(461),
     *      MH(3),MLE(104),ANG(104)
C
C           READ COMPUTATIONAL PARAMETERS
C
      READ(5,280) NN,NM,NW
  280 FORMAT(3I10)
      READ(5,281) T,TLIM,DT,WW
  281 FORMAT(4F10.0)
      READ(5,282) ATTD,HA,WX,WY
  282 FORMAT(4F10.1)
      READ(5,283) STEPO,PUNCH
  283 FORMAT(2F10.0)
      READ(5,284) UNIT,UNAR
  284 FORMAT(2F10.0)
C
C            READ GLOBAL COORDINATE OF EACH NODE
C
      WRITE(6,298)
  298 FORMAT(1H ,'  NODE NUMBER',5X,'X',9X,'Y',9X,'DEPTH',/)
      DO 205 I=1,NN
      READ(5,300)MEXT(I),MB(I),X(I),Y(I),H(I)
  300 FORMAT(I4,I1,2F10.5,F3.1)
  205 MINT(MEXT(I))=I
      WRITE(6,301)(MEXT(I),I,MB(I),X(I),Y(I),H(I),I=1,NN)
  301 FORMAT(2(3I5,3F10.3))
C
C            READ ELEMENT DATA
C
      READ(5,302)(N1(I),N2(I),N3(I),I=1,NM)
  302 FORMAT(3I3)
      WRITE(6,303)
  303 FORMAT(1H ,'THE ELEMENT CONNECTIONS ',/)
      WRITE(6,304) (I,N1(I),N2(I),N3(I),I=1,NM)
  304 FORMAT(5(I7,2X,3I4))
C
C           READ OPEN AND LAND BOUNDARY
C
      READ(5,306) MHNO,(MH(I),I=1,MHNO)
      READ(5,306) MLNO,(MLE(I),I=1,MLNO)
  306 FORMAT(16I5)
      WRITE(6,307) MHNO
  307 FORMAT(1H ,' THERE ARE',I5,'  NODES ON THE OPEN BOUNDARY')
      WRITE(6,309)(MH(I),I=1,MHNO)
      WRITE(6,308) MLNO
  308 FORMAT(1H ,' THERE ARE',I5,'  NODES ON THE LAND BOUNDARY')
      WRITE(6,309) (MLE(I),I=1,MLNO)
  309 FORMAT(3X,15I5)
      READ(5,310)(ANG(I),I=1,MLNO)
  310 FORMAT(8F10.4)
```

```
      WRITE(6,311)
  311 FORMAT(1H ,' THE OUTWARD NORMAL DIRECTIONS ',/)
      WRITE(6,312)(MLE(I),ANG(I),I=1,MLNO)
  312 FORMAT(5(I8,F10.4))
      STEBC=1.E10
      NBAND=0
      CRHO=0.00114
      G=32.174
      CDRAG=0.0025
      F=3.141592/21600. *SIN(ATTD/180.*3.141592)
      NWN=2*NW-1
      NN1=NN*2
      NW1=NW*2
      NWN1=NWN*2+1
      CCX=CDRAG*CRHO*ABS(WX)*WX
      CCY=CDRAG*CRHO*ABS(WY)*WY
      C1=1./3.
      C2=1./3.
      C3=1./3.
      C4=1./12.
      C5=1./12.
      C6=1./12.
      C7=1./6.
      C8=1./6.
      C9=1./6.
      DO 201 I=1,NN
      X(I)=X(I)*UNIT
  201 Y(I)=Y(I)*UNIT
      DO 202 I=1,MLNO
  202 ANG(I)=ANG(I)*DATAN(C.1001)/45.
      DO 204 I=1,NN1
      DO 204 J=1,NWN1
  204 B1(I,J)=0.
      DO 203 I=1,NN
      DO 203 J=1,NWN
  203 B2(I,J)=0.
C
C
      DO 10 I=1,NM
      N1(I)=MINT(N1(I))
      N2(I)=MINT(N2(I))
      N3(I)=MINT(N3(I))
      L1=IABS(N1(I)-N2(I))+1
      L2=IABS(N2(I)-N3(I))+1
      L3=IABS(N3(I)-N1(I))+1
      IF(NBAND.LT.L1) NBAND=L1
      IF(NBAND.LT.L2) NBAND=L2
      IF(NBAND.LT.L3) NBAND=L3
C
C           TRANSFER TO LOCAL COORDINATE
C
      XP1   =(2.*X(N1(I))-X(N2(I))-X(N3(I)))/3.
      XP2   =(2.*X(N2(I))-X(N1(I))-X(N3(I)))/3.
      XP3   =(2.*X(N3(I))-X(N1(I))-X(N2(I)))/3.
      YP1   =(2.*Y(N1(I))-Y(N2(I))-Y(N3(I)))/3.
      YP2   =(2.*Y(N2(I))-Y(N1(I))-Y(N3(I)))/3.
      YP3   =(2.*Y(N3(I))-Y(N1(I))-Y(N2(I)))/3.
C
C           CALCULATE ELEMENT AREA
C
```

```
      AREA(I)= 0.5*(XP2*YP3+XP1*YP2+XP3*YP1
     *        -XP2*YP1-XP3*YP2-XP1*YP3)
C
C           CALCULATE MANNING FACTOR
C
      CMAN(I)=0.03
C
C           CALCULATE COEFFICIENTS OF SHAPE FUNCTION
C
      A(1,I)=(XP2*YP3-XP3*YP2)/(2.*AREA(I))
      A(2,I)=(YP2-YP3)/(2.*AREA(I))
      A(3,I)=(XP3-XP2)/(2.*AREA(I))
      A(4,I)=(XP3*YP1-XP1*YP3)/(2.*AREA(I))
      A(5,I)=(YP3-YP1)/(2.*AREA(I))
      A(6,I)=(XP1-XP3)/(2.*AREA(I))
      A(7,I)=(XP1*YP2-XP2*YP1)/(2.*AREA(I))
      A(8,I)=(YP1-YP2)/(2.*AREA(I))
      A(9,I)=(XP2-XP1)/(2.*AREA(I))
C
C
      AREA(I)=AREA(I)/UNAR
      DIS1=SQRT((X(N1(I))-X(N2(I)))**2+(Y(N1(I))-Y(N2(I)))**2)
      DIS2=SQRT((X(N2(I))-X(N3(I)))**2+(Y(N2(I))-Y(N3(I)))**2)
      DIS3=SQRT((X(N3(I))-X(N1(I)))**2+(Y(N3(I))-Y(N1(I)))**2)
      DH1A=DIS1/SQRT(G*H(N1(I)))
      DH1B=DIS1/SQRT(G*H(N2(I)))
      DH2A=DIS2/SQRT(G*H(N2(I)))
      DH2B=DIS2/SQRT(G*H(N3(I)))
      DH3A=DIS3/SQRT(G*H(N3(I)))
      DH3B=DIS3/SQRT(G*H(N1(I)))
      IF(STEBC.GT.DH1A) STEBC=DH1A
      IF(STEBC.GT.DH1B) STEBC=DH1B
      IF(STEBC.GT.DH2A) STEBC=DH2A
      IF(STEBC.GT.DH2B) STEBC=DH2B
      IF(STEBC.GT.DH3A) STEBC=DH3A
      IF(STEBC.GT.DH3B) STEBC=DH3B
C
C                   FORM THE GLOBAL MATRIX
C
      II=N1(I)
      JJ=N2(I)
      KK=N3(I)
      III=NW
      JJ1=JJ-(II-NW)
      KK1=KK-(II-NW)
      B2(II,III)=B2(II,III)+C7*AREA(I)
      B2(II,JJ1)=B2(II,JJ1)+C4*AREA(I)
      B2(II,KK1)=B2(II,KK1)+C6*AREA(I)
      III=II-(JJ-NW)
      JJ1=NW
      KK1=KK-(JJ-NW)
      B2(JJ,III)=B2(JJ,III)+C4*AREA(I)
      B2(JJ,JJ1)=B2(JJ,JJ1)+C8*AREA(I)
      B2(JJ,KK1)=B2(JJ,KK1)+C5*AREA(I)
      III=II-(KK-NW)
      JJ1=JJ-(KK-NW)
      KK1=NW
      B2(KK,III)=B2(KK,III)+C6*AREA(I)
      B2(KK,JJ1)=B2(KK,JJ1)+C5*AREA(I)
   10 B2(KK,KK1)=B2(KK,KK1)+C9*AREA(I)
```

```
      WRITE(6,320) NBAND,STERC,UNAR
  320 FORMAT(1H0,' THE BANDWIDTH =  ',I5,'THE SMALLEST L/SQRT(GH) ='
     *    ,F10.3,//,1H ,'  ELEMENT AREA (UNIT',F10.2,' SQRT FEET)')
      WRITE(6,321)(I,AREA(I),I=1,NM)
  321 FORMAT(5(I6,F12.4))
      DO 12 I=1,NN
      DO 12 J=1,NWN
      B1(2*I-1,2*J  )=B2(I,J)
   12 B1(2*I,2*J  )=B1(2*I-1,2*J  )
C
C           ROTATE COORDINATE ON THE BOUNDARY
C
      LB1=NWN-1
      LB2=(NN1-NW1+2)/2
      DO 13 NI=1,MLNO
      I=MINT(MLE(NI))
      NMB=MB(I)
      GO TO (14,13,14,13),NMB
   14 CALL ROTB1(B1,ANG(NI),NN1,NW1,NWN1,NW,I,LB1,LB2)
   13 CONTINUE
      DO 23 NI=1,MLNO
      I=MINT(MLE(NI))
      NMB=MB(I)
      GO TO (15,23,15,21),NMB
   15 DO 16 J=1,NWN1
   16 B1(2*I-1,J)=0.
      B1(2*I-1,NW1)=1.
      GO TO 23
   21 DO 22 J=1,NWN1
      B1(2*I-1,J)=0.
   22 B1(2*I,J)=0.
      B1(2*I,NW1)=1.
      B1(2*I-1,NW1)=1.
   23 CONTINUE
C
C           L U DECOMPOSE MATRIX
C
      CALL MATRIX(B1,NN1,NW1,NWN1)
      DO 17 NI=1,MHNO
      I=MINT(MH(NI))
      DO 18 J=1,NWN
   18 B2(I,J)=0.
      B2(I,NW)=1.
   17 CONTINUE
      CALL MATRIX(B2,NN,NW,NWN)
C
C
C            SET INITIAL VALUES
C
      READ(5,318)(I,ETA(I),U(I),V(I),I=1,NN)
      DO 9 I=1,NN
      Q(2*I-1)=U(I)*(ETA(I)+H(I))
    9 Q(2*I)=V(I)*(ETA(I)+H(I))
C
      KL=0
      ML=0
      MLL=1
C
C           START SEMI-IMPLICIT METHOD
C
```

```
 25 IF(KL.EQ.ML*STEPO) GO TO 27
    GO TO 28
 27 ML=ML+1
    TIM=T/60.
    WRITE(6,315) TIM
315 FORMAT(/,'  TIME=',F10.2)
    WRITE(6,316)
316 FORMAT(2X,'ELEMENT  TIDE     U VELOCITY   V  VELOCITY')
    WRITE(6,317)(I,MEXT(I),ETA(I),U(I),V(I),I=1,NN)
317 FORMAT(3(2I4,3F10.3))
 28 IF(KL.EQ.MLL*PUNCH) GO TO 34
    GO TO 36
 34 MLL=MLL+1
    WRITE(7,318)(I,ETA(I),U(I),V(I),I=1,NN)
318 FORMAT(I5,3F11.4,I5,3F11.4)
 36 CALL WH(T,HH,WW,HA)
    DO 38 NI=1,MHNO
    I=MINT(MH(NI))
    ETA(I)=HH
 38 CONTINUE
 39 DO 40 I=1,NN
    R1(2*I-1)=0.
 40 R1(2*I)=0.
    DO 42 I=1,NM
    UU=Q(2*N1(I)-1)**2+Q(2*N2(I)-1)**2+Q(2*N3(I)-1)**2
    VV=Q(2*N1(I))**2+Q(2*N2(I))**2+Q(2*N3(I))**2
    AVEG=((UU+VV)/3.)**0.5
    HE1=H(N1(I))+ETA(N1(I))
    HE2=H(N2(I))+ETA(N2(I))
    HE3=H(N3(I))+ETA(N3(I))
    AVEGH=((HE1+HE2+HE3)/3.)**2
    CHEZ(I)=1.49/CMAN(I)*((HE1+HE2+HE3)/3.)**(1./6.)
    C746E=C7*ETA(N1(I))+C4*ETA(N2(I))+C6*ETA(N3(I))
    C485E=C4*ETA(N1(I))+C8*ETA(N2(I))+C5*ETA(N3(I))
    C659E=C6*ETA(N1(I))+C5*ETA(N2(I))+C9*ETA(N3(I))
    C746H=C7*H(N1(I))+C4*H(N2(I))+C6*H(N3(I))
    C485H=C4*H(N1(I))+C8*H(N2(I))+C5*H(N3(I))
    C659H=C6*H(N1(I))+C5*H(N2(I))+C9*H(N3(I))
    A258E=A(2,I)*ETA(N1(I))+A(5,I)*ETA(N2(I))+A(8,I)*ETA(N3(I))
    A369E=A(3,I)*ETA(N1(I))+A(6,I)*ETA(N2(I))+A(9,I)*ETA(N3(I))
    PREX1=G*(C746H+C746E)*A258E
    PREX2=G*(C485H+C485E)*A258E
    PREX3=G*(C659H+C659E)*A258E
    PREY1=G*(C746H+C746E)*A369E
    PREY2=G*(C485H+C485E)*A369E
    PREY3=G*(C659H+C659E)*A369E
    C746U=C7*Q(2*N1(I)-1)+C4*Q(2*N2(I)-1)+C6*Q(2*N3(I)-1)
    C485U=C4*Q(2*N1(I)-1)+C8*Q(2*N2(I)-1)+C5*Q(2*N3(I)-1)
    C659U=C6*Q(2*N1(I)-1)+C5*Q(2*N2(I)-1)+C9*Q(2*N3(I)-1)
    C746V=C7*Q(2*N1(I))+C4*Q(2*N2(I))+C6*Q(2*N3(I))
    C485V=C4*Q(2*N1(I))+C8*Q(2*N2(I))+C5*Q(2*N3(I))
    C659V=C6*Q(2*N1(I))+C5*Q(2*N2(I))+C9*Q(2*N3(I))
    QXX1=Q(2*N1(I)-1)**2/HE1
    QXX2=Q(2*N2(I)-1)**2/HE2
    QXX3=Q(2*N3(I)-1)**2/HE3
    QYY1=Q(2*N1(I))**2/HE1
    QYY2=Q(2*N2(I))**2/HE2
    QYY3=Q(2*N3(I))**2/HE3
    QXY1=Q(2*N1(I)-1)*Q(2*N1(I))/HE1
    QXY2=Q(2*N2(I)-1)*Q(2*N2(I))/HE2
```

```
      QXY3=Q(2*N3(I)-1)*Q(2*N3(I))/HE3
      CONU=(A(2,I)*QXX1+A(5,I)*QXX2+A(8,I)*QXX3
     *     +A(3,I)*QXY1+A(6,I)*QXY2+A(9,I)*QXY3)/3.
      CONV=(A(2,I)*QXY1+A(5,I)*QXY2+A(8,I)*QXY3
     *     +A(3,I)*QYY1+A(6,I)*QYY2+A(9,I)*QYY3)/3.
      GACA=G*AVEG/(CHEZ(I)**2)/AVEGH
      FRU1=GACA*C746U
      FRV1=GACA*C746V
      FRU2=GACA*C485U
      FRV2=GACA*C485V
      FRU3=GACA*C659U
      FRV3=GACA*C659V
      UK1=  (CONU+PREX1-F*C746V-CCX+FRU1)*AREA(I)
      UK2=  (CONU+PREX2-F*C485V-CCX+FRU2)*AREA(I)
      UK3=  (CONU+PREX3-F*C659V-CCX+FRU3)*AREA(I)
      VK1=(CONV+PREY1+F*C746U-CCY+FRV1)*AREA(I)
      VK2=(CONV+PREY2+F*C485U-CCY+FRV2)*AREA(I)
      VK3=(CONV+PREY3+F*C659U-CCY+FRV3)*AREA(I)
      II=N1(I)
      JJ=N2(I)
      KK=N3(I)
      R1(II*2-1)=R1(II*2-1)-UK1
      R1(JJ*2-1)=R1(JJ*2-1)-UK2
      R1(KK*2-1)=R1(KK*2-1)-UK3
      R1(II*2)=R1(II*2)-VK1
      R1(JJ*2)=R1(JJ*2)-VK2
   42 R1(KK*2)=R1(KK*2)-VK3
      T=T+DT/2.
      DO 50 NI=1,MLNO
      I=MINT(MLE(NI))
      NMB=MB(I)
      GO TO (49,50,49,52),NMB
   52 R1(2*I-1)=0.
      R1(2*I)=0.
      GO TO 50
   49 CALL ROTUV(R1(2*I-1),R1(2*I),ANG(NI))
      R1(2*I-1)=0.
   50 CONTINUE
      CALL SOLVE(B1,R1,UVP,NN1,NW1,NWN1)
      DO 53 NI=1,MLNO
      I=MINT(MLE(NI))
      NMB=MB(I)
      GO TO (54,53,54,53),NMB
   54 CALL ROTUV(UVP(2*I-1),UVP(2*I),-ANG(NI))
   53 CONTINUE
      DO 56 I=1,NN
      Q(2*I-1)=Q(2*I-1)+DT*UVP(2*I-1)
   56 Q(2*I)=Q(2*I)+DT*UVP(2*I)
      DO 62 I=1,NN
      HE=H(I)+ETA(I)
      U(I)=Q(2*I-1)/HE
      V(I)=Q(2*I)/HE
   62 R2(I)=0.
      DO 64 I=1,NM
      EK=(A(2,I)*Q(2*(N1(I))-1)+A(5,I)*Q(2*(N2(I))-1)+
     *     A(8,I)*Q(2*(N3(I))-1)+A(3,I)*Q(2*(N1(I)))+
     *     A(6,I)*Q(2*(N2(I)))+A(9,I)*Q(2*(N3(I))))/3.*AREA(I)
      II=N1(I)
      JJ=N2(I)
      KK=N3(I)
```

```
      R2(II)=R2(II)-EK
      R2(JJ)=R2(JJ)-EK
   64 R2(KK)=R2(KK)-EK
      T=T+DT/2.
      CALL WHP(T,HP,WW,HA)
      DO 66 NI=1,MHNO
      I=MINT(MH(NI))
      EP(I)=HP
      R2(I)=EP(I)
   66 CONTINUE
      CALL SOLVE(B2,R2,EP,NN,NW,NWN)
      DO 68 I=1,NN
   68 ETA(I)=ETA(I)+DT*EP(I)
      KL=KL+1
      IF(T-TLIM) 25,25,70
   70 STOP
      END
      SUBROUTINE WH(T,HH,WW,HA)
      HH=COS(2.*3.141592*(T/WW))*HA
       RETURN
      END
      SUBROUTINE WHP(T,HP,WW,HA)
      HP=-2.*3.141592/WW*SIN(2.*3.141592*(T/WW))*HA
      RETURN
      END
C
C          LU DECOMPOSITION
C
      SUBROUTINE MATRIX(A,N,NW,NWN)
      DIMENSION A(N,NWN)
      M=N-1
      DO 30 K=1,M
      I1=K+1
      NW1=NW+K-1
      IF(NW1.LE.N) GO TO 20
      NW1=N
   20 DO 30 I=I1,NW1
      NI=NW-I+I1
      Y=A(I,NI-1)/A(K,NW)
      A(I,NI-1)=Y
      NW11=NI+NW-2
      IF(NW1.LE.N) GO TO 45
      NW11=NW-I
   45 DO 30 J=NI,NW11
      A(I,J)=A(I,J)-Y*A(K,J+I-K)
   30 CONTINUE
      RETURN
      END
      SUBROUTINE SOLVE(A,B,X,N,NW,NWN)
      DIMENSION A(N,NWN),B(N),X(N)
      X(1)=B(1)
      K1=NW-1
      DO 60 K=2,N
      SUM=0.
      NW2=NW-K+1
      IF(K.LE.NW) GO TO 54
      NW2=1
   54 DO 55 J=NW2,K1
   55 SUM=SUM+A(K,J)*X(J-NW+K)
   60 X(K)=B(K)-SUM
```

```
      X(N)=X(N)/A(N,NW)
      K=N
      NW4=NW+1
   62 SUM=0.
      K=K-1
      NW3=NW+N-K
      IF(NW3.LT.NWN) GO TO 64
      NW3=NWN
   64 DO 65 J=NW4,NW3
   65 SUM=SUM+A(K,J)*X(J-NW+K)
      X(K)=(X(K)-SUM)/A(K,NW)
      IF(K.EQ.1) GO TO 80
      GO TO 62
   80 RETURN
      END
C
C           COORDINATE ROTATION
C
      SUBROUTINE    ROTBL(B1,ANGLE,NN1,NW1,NWN1,NW,I,LB1,LB2)
      DIMENSION B1(NN1,NWN1)
      NWM=NW-1
      IF(I.LE.NW) GO TO 20
      LB=LB1
    9 DO 10 J=NW,LB
      II=I-J+NW-1
      CALL ROTSM(B1(2*II-1,2*J+2),B1(2*II-1,2*J+3),B1(2*II,2*J+1),
     *     B1(2*II,2*J+2),ANGLE)
      NWJ=4*NW-2*J
      B1(2*I,NWJ-2)=B1(2*II-1,2*J+2)
      B1(2*I,NWJ-3)=B1(2*II-1,2*J+3)
      B1(2*I-1,NWJ-1)=B1(2*II,2*J+1)
      B1(2*I-1,NWJ-2)=B1(2*II,2*J+2)
   10 CONTINUE
      GO TO 30
   20 IF(I.EQ.1) GO TO 30
      LB=I+NW-2
      GO TO 9
   30 IF(I.GT.LB2) GO TO 60
      LB=1
   50 DO 70 J=LB,NWM
      II=I-J+NW
      CALL ROTSM(B1(2*II-1,2*J),B1(2*II-1,2*J+1),B1(2*II,2*J-1),
     *     B1(2*II,2*J),ANGLE)
      NWJ=4*NW-2*J
      B1(2*I,NWJ)=B1(2*II-1,2*J)
      B1(2*I,NWJ-1)=B1(2*II-1,2*J+1)
      B1(2*I-1,NWJ+1)=B1(2*II,2*J-1)
      B1(2*I-1,NWJ)=B1(2*II,2*J)
   70 CONTINUE
      GO TO 90
   60 IF(I.EQ.NN1/2) GO TO 90
      LB=1+(I-LB2)
      GO TO 50
   90 RETURN
      END
      SUBROUTINE  ROTSM(SM1,SM2,SM3,SM4,ANGLE)
      SM1P=SM1*COS(ANGLE)+SM2*SIN(ANGLE)
      SM2=-SM1*SIN(ANGLE)+SM2*COS(ANGLE)
      SM3P=SM3*COS(ANGLE)+SM4*SIN(ANGLE)
      SM4=-SM3*SIN(ANGLE)+SM4*COS(ANGLE)
```

```
SM1=SM1P
SM3=SM3P
RETURN
END
SUBROUTINE ROTUV(A,B,ANGLE)
AP=A*COS(ANGLE)+B*SIN(ANGLE)
B=-A*SIN(ANGLE)+B*COS(ANGLE)
A=AP
RETURN
END
```

APPENDIX C. COMPUTER PROGRAM FOR CORNER POINT

```
C
C    BISECTION METHOD
C
     READ(5,100) ANG,X1,Y1,X2,Y2,X3,Y3
100  FORMAT(7F10.2)
     ANG=ANG*3.141592/180.
     THTA1=0.
     THTA2=ANG
     DIS1=SQRT((X2-X1)**2+(Y2-Y1)**2)
     DIS2=SQRT((X3-X2)**2+(Y3-Y2)**2)
     FX1=DIS1/DIS2*COS(ANG-THTA1)-COS(THTA1)
  10 THTA3=(THTA1+THTA2)/2.
     FX3=DIS1/DIS2*COS(ANG-THTA3)-COS(THTA3)
     IF(FX1*FX3) 20,60,40
  20 THTA2=THTA3
     GO TO 50
  40 THTA1=THTA3
  50 IF(ABS(THTA2-THTA1)-0.5D-05) 60,60,10
  60 THTA3=THTA3*180./3.141592
     WRITE(6,70) THTA3
  70 FORMAT(2X,F10.4)
     END
```

APPENDIX D. COMPUTER PROGRAM FOR PLOTTING FIGURES

```
      CALL NAME('INO100','CHARLESTOWN POND        ')
      DIMENSION X(294),Y(294),MEXT(294),MINT(300),MB(294),MLE(110),
     *     XL(110),YL(110),X1(474),X2(474),Y1(474),Y2(474),
     *     U(294),V(294),UE(474),VE(474),N1(474),N2(474),N3(474)
      NN=284
      NM=461
      A=1.
      B=1.
      DO 205 I=1,NN
      READ(5,300)MEXT(I),MB(I),X(I),Y(I),H
  300 FORMAT(I4,I1,2F10.5,F3.1)
  205 MINT(MEXT(I))=I
      WRITE(6,301)(MEXT(I),I,MB(I),X(I),Y(I),I=1,NN)
  301 FORMAT(3X,3I5,2F10.3)
      READ(5,302) (N1(I),N2(I),N3(I),I=1,NM)
  302 FORMAT(3I3)
      WRITE(6,303)(I,N1(I),N2(I),N3(I),I=1,NM)
  303 FORMAT(5(4I5))
      READ(5,306) MLNO,(MLE(I),I=1,MLNO)
  306 FORMAT(16I5)
      WRITE(6,304)MLNO,(MLE(I),I=1,MLNO)
  304 FORMAT(16I5)
      READ(5,200)(I,ETA,U(I),V(I),I=1,NN)
  200 FORMAT(I5,3F11.4,I5,3F11.4)
      WRITE(6,307)(I,U(I),V(I),I=1,NN)
  307 FORMAT(4(I5,2F10.4))
      DO 10 I=1,NM
      N1(I)=MINT(N1(I))
      N2(I)=MINT(N2(I))
      N3(I)=MINT(N3(I))
      UE(I)=(U(N1(I))+U(N2(I))+U(N3(I)))/6.
      VE(I)=(V(N1(I))+V(N2(I))+V(N3(I)))/6.
      X1(I)=(X(N1(I))+X(N2(I))+X(N3(I)))/3./20.-UE(I)+A
      X2(I)=(X(N1(I))+X(N2(I))+X(N3(I)))/3./20.+UE(I)+A
      Y1(I)=(Y(N1(I))+Y(N2(I))+Y(N3(I)))/3./20.-VE(I)+B
      Y2(I)=(Y(N1(I))+Y(N2(I))+Y(N3(I)))/3./20.+VE(I)+B
      AL=SQRT(UE(I)**2+VE(I)**2)/1.5
      CALL ARROW(X1(I),Y1(I),X2(I),Y2(I),AL,3)
   10 CONTINUE
      WRITE(6,330)(I,UE(I),VE(I),I=1,NM)
  330 FORMAT(5(I6,2F10.4))
      WRITE(6,305)(I,X1(I),Y1(I),X2(I),Y2(I),I=1,NM)
  305 FORMAT(3X,I5,4F10.4)
      DO 16 NI=1,MLNO
      I=MINT(MLE(NI))
      XL(NI)=X(I)/20.+A
   16 YL(NI)=Y(I)/20.+B
      CALL VECTOR (XL,YL,MLNO,1,0,0)
      CALL ENPLT(0.,0.)
      STOP
      END
```