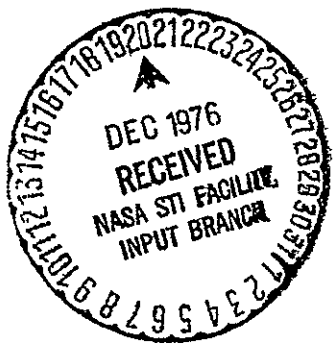
A THREE-DIMENSIONAL, TIME-DEPENDENT MODEL OF MOBILE BAY

FINAL REPORT TO THE
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
ON CONTRACT NAS8-30380

OCTOBER 1976



WRITTEN BY
F. H. PITTS
DIRECTED BY
R. C. FARMER
DEPARTMENT OF CHEMICAL ENGINEERING
LOUISIANA STATE UNIVERSITY
BATON ROUGE, LOUISIANA 70803

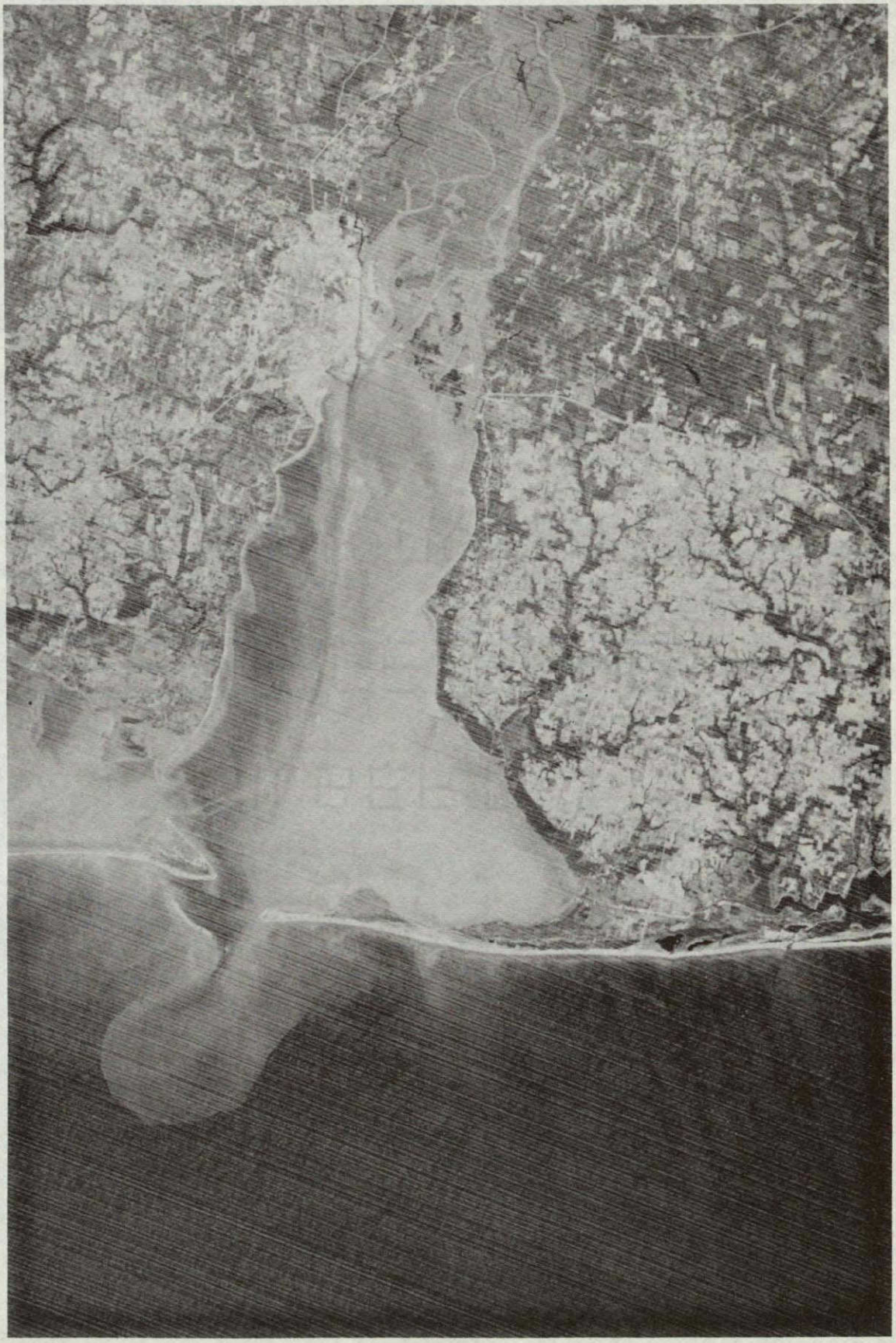
A THREE-DIMENSIONAL, TIME-DEPENDENT MODEL OF MOBILE BAY

FINAL REPORT TO THE
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
ON CONTRACT NAS8-30380

OCTOBER 1976

WRITTEN BY
F. H. PITTS
DIRECTED BY
R. C. FARMER
DEPARTMENT OF CHEMICAL ENGINEERING
LOUISIANA STATE UNIVERSITY
BATON ROUGE, LOUISIANA 70803

Mobile Bay as seen in an IR photograph from a satellite.



ACKNOWLEDGEMENTS

The authors appreciate the support which the technical monitors: Mr. Ted Paludan, Mr. Rex Morton, and Mr. H. B. Wilson of the Marshall Space Flight Center have given to this study.

Special thanks are extended to the personnel of the U.S. Army, Corps of Engineers, Mobile District who supplied much of the data discussed in this report. The outstanding cooperation of Mr. High McClelland of this office was particularly valuable.

Dr. Bill Schroeder, meterologist of the Dauphin Island Sea Laboratory, contributed much to our understanding of wind conditions on Mobile Bay.

The participation of Dr. W. R. Waldrop, TVA, Engineering Laboratory in the early phases of this research program was invaluable.

This report also constitutes the PhD Dissertation of Mr. Fred Pitts.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

A three-dimensional, time-variant mathematical model for momentum and mass transport in estuaries was developed and its solution implemented on a digital computer. The mathematical model is based on state and conservation equations applied to turbulent flow of a two-component, incompressible fluid having a free surface. Thus, bouyancy effects caused by density differences between the fresh and salt water, inertia from the river and tidal currents, and differences in hydrostatic head are taken into account. The conservation equations, which are partial differential equations, are solved numerically by an explicit, one-step finite difference scheme and the solutions displayed numerically and graphically. To test the validity of the model, a specific estuary for which scaled model and experimental field data are available, Mobile Bay, was simulated. Comparisons of velocity, salinity and water level data show that the model is valid and a viable means of simulating the hydrodynamics and mass transport in non-idealized estuaries.

CHAPTER 1

INTRODUCTION

In addition to being unique natural environments, estuaries are sources of commercially valuable seafoods, essential links in water transportation routes, and important recreational areas. Furthermore, they serve as areas to dilute and degrade urban and industrial wastes. Since the use of these essential properties is expected to increase, the effectiveness with which estuaries are utilized must improve or these natural resources will be completely consumed. Improvement in the utilization of estuaries can be achieved only through a better understanding of estuarine phenomena, both qualitatively and quantitatively. This understanding can only be achieved by a combination of computer simulations, scaled model studies, and field observations of real estuaries. To make these ideas concrete, it is expedient to select a specific estuary to study. The estuary selected here is Mobile Bay.

A. ESTUARIES

"An estuary is a semi-enclosed coastal body of water which has a free connection with the open sea and within which sea water is measurably diluted with fresh water derived from land drainage" is a widely accepted definition proposed by Cameron and Pritchard (1963). Therefore a bay, sound, inlet or small gulf into which several rivers flow, as well as the lower reaches of a single river into which salt water intrudes, may be classified as an estuary (Officer, 1976). In

any of these situations, the density difference between the fresh and salt water provides a driving force for circulation and mixing processes. In most estuaries, the density differences are a consequence of variations in salinity as opposed to variations in temperature. Other driving forces for circulation and mixing processes are tides and winds. Irrespective of the driving force or combination of forces for these turbulent momentum and mass transports, the resulting velocity and salinity distributions are complex, frequently varying in three space dimensions and time. Because river flows, tidal variations and wind conditions are seldom all constant for more than a few hours at a time, estuaries are generally in transition from one dynamic state to another. Furthermore, even the topography of an estuary may be changing rapidly if sedimentation processes are important.

The purpose of the research reported herein was to develop a generally applicable three-dimensional, time-variant mathematical model for momentum and mass transport in estuaries and to implement the solution of the model on a digital computer, that is, generate a computer model. The mathematical model is based on state and conservation equations applied to turbulent flow of a two component, incompressible fluid having a free surface. The conservation equations, which are partial differential equations, were solved numerically by an explicit finite difference method and the resulting solutions were displayed both numerically and graphically. To test the validity of the model, an estuary for which scaled model and field studies have been made, Mobile Bay, was simulated. Beyond testing the validity of

the model, Mobile Bay was selected for study because it is both technically and commercially interesting. The National Aeronautics and Space Administration has accumulated large quantities of remote sensor (i.e. satellite) data on this bay which in conjunction with an adequate hydrodynamic and salinity model should prove useful in resource and pollution studies. A point of particular interest in many of the photographs of Mobile Bay taken from satellites is that the ship channel stands out from the rest of the bay. See the photograph in front piece. Because the bay water is opaque, the bottom of the bay cannot be seen from the altitude of the satellite. Therefore, the contrast between the ship channel and the shallow waters on either side must be the consequence of a complex surface phenomenon. Again a computer model should be of value in establishing what that phenomenon is.

The Army Corps of Engineers is responsible for maintaining ship channels in the bay and for evaluating the impact of proposed new channels. A model with sediment transport incorporated in it would be valuable in scheduling dredging operations in existing channels. As it presently stands, the model developed here can be used to assess the effect of islands formed from dredging spoils on the circulation and salinity patterns, matters of critical concern to the oyster industry in the bay. A unique feature of the present work is that the bay's main ship channel has been included by representing the channel with a vertical two-dimensional model that is interfaced with the three-dimensional model of the bay proper. It is feasible to evaluate the effects of new channels in this fashion at the cost of some additional programming.

B. MOBILE BAY

The following description of Mobile Bay relies heavily on that presented by Shah and Farmer (1976). Mobile Bay, located on the northern edge of the Gulf of Mexico, is roughly pear shaped as can be seen in Fig. 1.1. The bay has an approximate length of 51 km and an average width of 20 km; the maximum width is 36 km. Its boundaries are delineated by relatively steep banks on the eastern, western and southern edges and by a causeway on the north. The overall area of the bay is 1022 km^2. The average depth is 3 m at mean low water with maximum of 18.3 m occurring near the tidal inlet to the Gulf, the Main Pass. This inlet is 4.8 km wide between Mobile Point on the east and Dauphin Island on the west. The Pass Aux Herons, the tidal inlet that connects the bay with the Mississippi Sound to the west, is 3.7 km wide and 1.1 m deep on the average. During the diurnal tide cycle, these passes contribute on the order of a billion cubic meters of flow into and out of the bay.

Aside from an average of 69 inches of rainfall evenly distributed through the year and several small streams entering the east and west sides of the bay, the fresh water sources are four interconnected rivers entering the north end of the bay. These rivers, the Mobile, Tensaw, Blakeley and Appalachie, are the distributaries of the Mobile River which results from the junction of the Tombigbee and Alabama Rivers. The respective depths of the rivers entering the bay are 9, 9, 6 and 8 m. The Mobile and Tensaw Rivers are the major contributors of fresh water and sediment to the bay.

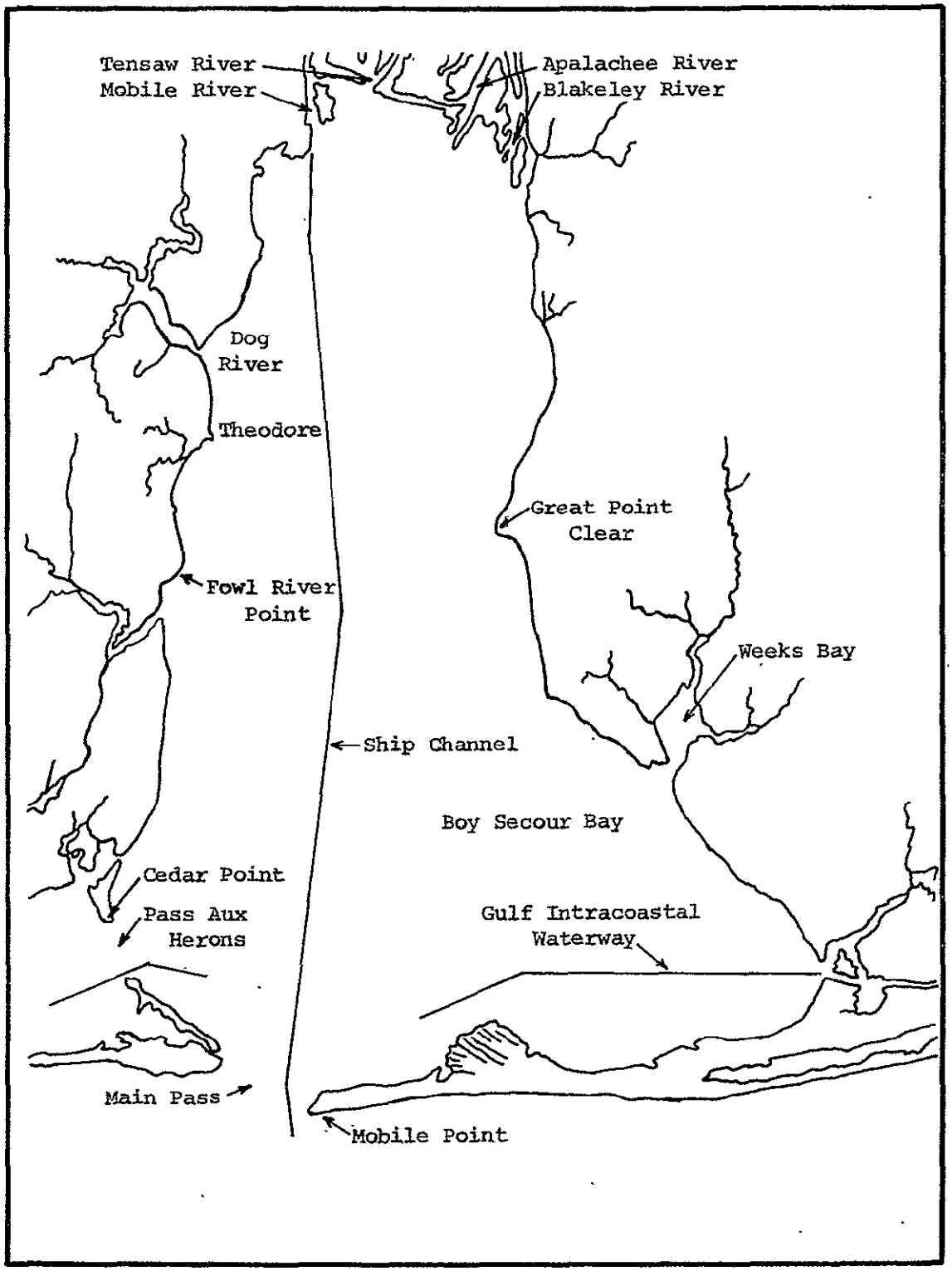


Figure 1.1 - Mobile Bay

The Mobile Harbor Ship Channel connects the lower anchorage near Mobile Point with the State Docks at the city of Mobile. This channel is maintained at a depth of 12.2 m and a width of 122 m by the Army Corps of Engineers. High salinity water intrudes almost the entire length of the channel at depths below the natural depth of the bay. The other channel of major significance in Mobile Bay is the Gulf Intracoastal Waterway, which extends from Apalachee Bay, Florida, to Brownsville, Texas. Traversing the lower end of the bay, the waterway passes through Bon Secour Bay, crosses the main ship channel approximately 4.4 km north of the Main Pass, and continues through the Pass Aux Herons into the Mississippi Sound. The depth of the waterway ranges from 3.7 m to 5.5 m. Because it is only slightly deepter than the bay itself and has little influence on salinity patterns, the Gulf Intracoastal Waterway was neglected in the present work.

As stated previously, the tides from Mobile Bay are diurnal; the difference between high and low tide normally range from 30. to 76 cm. While these tides promote mixing, a slight but measurable salinity stratification exists throughout much of the bay most of the time. Strong winds over the bay generate surface waves that can breakdown this concentration. The tide elevation is also strongly influenced by prevailing winds of several hours duration.

C. SUMMARY

This chapter discussed the concept of an estuary, presented motivations for developing an estuarine model, and stated the objective of this work. A description of Mobile Bay was also given. In the next three chapters, the equations governing the phenomena of

interest, the numerical method used to solve the set of equations and the results of applying the model to Mobile Bay will be set forth. The final chapter will present conclusions regarding the model's viability and recommendations for possible improvements and extentions. The appendices contain graphical displays of the model results and descriptions and listings of the computer model program and Varian plot program used in this research.

CHAPTER 2

THE GOVERNING EQUATIONS

The philosophy followed in developing this model of Mobile Bay was to base the model on the fundamental laws pertinent to the system and to make as few restrictive simplifications as possible. Of course, the latter part of this philosophy had to be tempered by the necessity of remaining within the capabilities of the available computing facilities. This chapter presents the general governing equations, the necessary simplifications and the specific equations used. A statement of the required initial and boundary conditions is also given.

A. STATE

The brackish waters found in Mobile Bay are a consequence of the mixing of fresh water from the river complex and gulf water from the tidal inlets. The density of this mixture at any point in space and time depends on the local composition (i.e., the salinity s which is the mass of salts in grams per kilogram of brackish water or, equivalently, in parts per thousand, PPT), temperature T, and pressure p. An equation relating these variables is referred to as an equation of state and may be represented by

$$\rho = \rho(s,T,p). \tag{2-1}$$

A theoretically derived analytical expression for Eq. (2-1) applied to brackish water over the range of values of s, T and p of interest is presently lacking. One recourse is to use a Taylor series to expand

the density about a reference value in terms of small changes in s, T and p. Thus,

$$\rho = \rho_o + \left(\frac{\partial \rho}{\partial s}\right)_{p,T} \Delta s + \left(\frac{\partial \rho}{\partial T}\right)_{s,p} \Delta T + \left(\frac{\partial \rho}{\partial p}\right)_{s,T} \Delta p \qquad (2\text{-}2)$$

A suitable reference state corresponding to typical conditions in Mobile Bay would be s = 10. PPT, T = 20.°C and p = 1. bar where $\rho = 1.0058$ gm/cm^3. For small changes in s, T and p, the partial derivatives in Eq. (2-2) are very nearly constant and may be evaluated from salinity-density, compressibility and thermal expansion data for seawater such as that given in Horne (1969). Thus,

$$\left(\frac{\partial \rho}{\partial s}\right)_{p,T} = 7.5 \times 10^{-4} \text{ gm cm}^{-3} \text{ PPT}^{-1}$$

$$\left(\frac{\partial \rho}{\partial T}\right)_{s,p} = -2.1 \times 10^{-6} \text{ gm cm}^{-3} \text{ °C}^{-1}$$

$$\left(\frac{\partial \rho}{\partial p}\right)_{s,T} = 4.5 \times 10^{-7} \text{ gm cm}^{-3} \text{ bar}^{-1}.$$

Estimates for expected differences in the variables s, T and p in the horizontal direction at a given elevation over the length of the bay are 33. PPT, 10. C deg, and 0.1 bar, respectively. Therefore, the changes in density resulting from salinity variations are three orders of magnitude larger than those from temperature variations and five orders of magnitude larger than those from pressure variations in the horizontal. Estimates for differences in s, T and p in the vertical direction in slightly stratified shallow water (i.e., depth less than 3. m) are 2. PPT, 2. C deg and 0.3 bar, respectively. Here changes in density due to salinity variations are two orders of magnitude larger than those from temperature variations and four orders larger than those from pressure variations. Deeper waters may be more stratified with appreciably larger differences in salinity but not temperature.

From the above discussion, it is apparent that the fluid in Mobile Bay is essentially incompressible and, to a good approximation, isothermal. Furthermore, density is a linear function of salinity only, with the consequence that density is a measure of the composition of the brackish water. In the rest of this work, composition will be denoted by S which is defined by

$$S = \frac{\rho - \rho_f}{\rho_g - \rho_f} \tag{2-3}$$

so that

$$\rho = \rho_f + \beta S \tag{2-4}$$

where

ρ_f = density of fresh water, gm cm^{-3}

ρ_g = density of gulf water, gm cm^{-3}

$\beta = \rho_g - \rho_f$.

The variable S, which might be termed the normalized density anomaly, will range between zero and unity.

For reference, the relationships between ρ, S and two other measures of composition will be listed and examined. These measures are salinity s and mass fraction salts ω_A. They are related to ρ and S as follows:

$$\rho = \rho_f + \beta S = \rho_f + \frac{\partial \rho}{\partial s} s = \rho_f + \alpha \omega_A \tag{2-5}$$

where $\alpha = 1000. (\partial\rho/\partial s)$. The reference density ρ_o in Eq. (2-2) has been replaced by ρ_f, the reference salinity taken as 0. PPT so that $\Delta s = s$, and the pressure and temperature terms neglected. Also, the relation $\omega_A = s/1000.$ has been used. It should be noted that these relations are strictly valid only over a limited range of salinities,

say 0. to 40. PPT, which fortunately covers the range found in the problem at hand. Also, from an analytical standpoint, it is exceedingly convenient that the density is a linear function of the composition variable in each case.

B. MASS CONSERVATION

In this section, the mean continuity equation for an incompressible, variable density fluid in turbulent flow is developed. As derived in several transport phenomena texts, for instance, Bird, Stewart and Lightfoot (1960) and Slattery (1972), the equation governing mass conservation in terms of the mass density ρ and the mass average velocity $\underset{\sim}{u}$ is

$$\frac{\partial \rho}{\partial t} + \text{div } (\rho \underset{\sim}{u}) = 0. \tag{2-6}$$

The divergence term in Eq. (2-6) may be expanded and the resulting equation rearranged to

$$\frac{1}{\rho} \frac{D\rho}{Dt} + \text{div } \underset{\sim}{u} = 0, \tag{2-7}$$

where $\frac{D}{Dt}$ is the substantial derivative operator or the total derivative operator following the mass average velocity. While Eq. (2-7) is rigorously correct, its inclusion in the system of equations to be solved is undesirable because it admits sound waves to the possible solutions of the system which in turn requires very small time steps in finite difference schemes in order to achieve stability. One way to avoid this difficulty is to simplify Eq. (2-7) to

$$\text{div } \underset{\sim}{u} = 0 \tag{2-8}$$

which filters sound waves from the possible solutions (see Kamenkovich, 1975). This simplification may be justified by an analysis suggested by Batchelor (1967).

If

$$\left|\frac{1}{\rho}\frac{D\rho}{Dt}\right| = \left|\text{div } \underset{\sim}{u}\right| << \frac{V}{L} \tag{2-9}$$

where L is a length scale representing the minimum distance over which large variations in $\underset{\sim}{u}$ occur and V is a velocity scale representing the variation in $\left|\underset{\sim}{u}\right|$ in both time and space over the distance L, then Eq. (2-8) is a reasonable approximation to Eq. (2-7). Since, in general, the flow under consideration will be turbulent, the appropriate length and velocity scales are the smallest length scale occurring in the flow and its associated velocity scale. These are the Kolmogorov micro-scales (see Tennekes and Lumley, 1972) and are given by

$$L = (\nu^3/\varepsilon)^{1/4}, \quad V = (\nu\varepsilon)^{1/4}, \tag{2-10}$$

where ν is the kinematic viscosity of the fluid and ε is the dissipation rate of turbulence energy per unit mass of fluid. To test whether Eq. (2-9) is satisfied or not, one must also have an estimate for $\frac{D\rho}{Dt}$. Obtaining such an estimate requires the use of the relations in Eq. (2-5), as well as an equation governing species conservation. Presuming no chemical reaction occurs and diffusion is of the Fickian type, one then finds the equation of species conservation at the continuum level presented in transport phenomena texts as

$$\frac{\partial}{\partial t}(\rho_A) + \text{div } (\rho_A \underset{\sim}{u}) = \text{div } [\rho D_{AB} \text{grad}(\omega_A)], \tag{2-11}$$

where ρ_A is the mass concentration of the species, ω_A is the mass fraction of the species, and D_{AB} is the mass diffusivity of the species in the solution. In this case, the species is total salts. By noting that

$$\frac{\partial}{\partial t}(\rho_A) + \text{div}(\rho_A \underset{\sim}{u}) = \frac{\partial}{\partial t}(\rho_A) + \underset{\sim}{u} \cdot \text{grad}(\rho_A) + \rho_A \text{ div } \underset{\sim}{u}$$

$$= \frac{D}{Dt}(\rho_A) - \omega_A \frac{D\rho}{Dt} = \rho \frac{D}{Dt}(\omega_A),$$

Eq. (2-11) may be rewritten as

$$\frac{D}{Dt}(\omega_A) = \frac{1}{\rho} \text{div } [\rho D_{AB} \text{ grad } (\omega_A)]. \tag{2-12}$$

Applying the substantial derivative to the first and last parts of Eq. (2-5) gives

$$\frac{D\rho}{Dt} = \alpha \frac{D}{Dt}(\omega_A)$$

and combining this result with Eq. (2-12) yields

$$\frac{D\rho}{Dt} = \frac{\alpha}{\rho} \text{div } [\rho D_{AB} \text{ grad } (\omega_A)]. \tag{2-13}$$

If Ω_A is used to represent the variation in ω_A over the length scale L and the distances in the spatial derivatives are approximated by L, then Eq. (2-13) may be modified and inserted in the inequality Eq. (2-9) to give

$$\alpha\rho^{-2} D_{AB} \Omega_A L^{-2} << \frac{V}{L}$$

which may be rearranged and combined with Eqs. (2-10) to yield

$$\alpha\rho^{-2} D_{AB} \Omega_A \nu^{-1} << 1 \tag{2-14}$$

The magnitudes of the various factors appearing in the left-hand side of this equation are as follows:

$\rho = 1.0 \text{ gm cm}^{-3}$

$\alpha = 0.75 \text{ gm cm}^{-3}$

$D_{AB} = 1.3 \times 10^{-5} \text{ cm}^2 \text{ sec}^{-1}$

$\Omega_A = 4.0 \times 10^{-2}$

$\nu = 1.0 \times 10^{-2} \text{ cm}^2 \text{ sec}^{-1}$

These values substituted in Eq. (2-14) yield $4. \times 10^{-5}$ which is certainly much less than unity. Thus, for the system under consideration div $\underset{\sim}{u} = 0$, frequently called the equation of volume continuity, is an excellent approximation to the equation of continuity, Eq. (2-7).

The discussion to this point has been in terms of instantaneous values of the flow field variables. Since the flow field under consideration is turbulent, these instantaneous values fluctuate rapidly and with a certain degree of randomness. Because it is practially impossible to track these fluctuations numerically, it is necessary to express the governing equations in terms of variables with the turbulent fluctuations averaged out. Having determined that div $\underset{\sim}{u} = 0$ is the appropriate equation to describe mass conservation at the continuum level, one can proceed to an equation at the turbulence level by utilizing the ensemble or statistical averaging operation $\lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} a_i$ where a_i is the instantaneous value of an observed variable in the i^{th} realization of an ensemble of repeated experiments. The reader is referred to Leslie (1973), p. 6, or Townsend (1976), p. 4, for an introductory discussion of the concept of an ensemble mean and its relation to the more conventional time mean. Denoting the ensemble mean velocity by $\overline{\underset{\sim}{u}}$ and the fluctuation about the mean or turbulent velocity by $\underset{\sim}{u}'$, one has that $\underset{\sim}{u} = \overline{\underset{\sim}{u}} + \underset{\sim}{u}'$ and

$$\text{div } (\overline{\underset{\sim}{u}} + \underset{\sim}{u}') = 0 \qquad (2\text{-}15)$$

Averaging Eq. (2-15) over an ensemble of flow realizations, an operation indicated by $\langle\ \rangle$, one obtains

$$\langle \text{div } (\overline{\underset{\sim}{u}} + \underset{\sim}{u}') \rangle = 0$$

which reduces to

$$\text{div } \overline{\underset{\sim}{u}} = 0 \qquad (2\text{-}16)$$

since the divergence and averaging operators are commutative and $\langle \overline{\underset{\sim}{u}} + \underset{\sim}{u}' \rangle = \overline{\underset{\sim}{u}}$. Eq. (2-16), which in a rectangular cartesian coordinate system reads

$$\frac{\partial}{\partial x} (\overline{u}) + \frac{\partial}{\partial y} (\overline{v}) + \frac{\partial}{\partial z} (\overline{w}) = 0 \qquad (2\text{-}17)$$

where $\overline{u}$, $\overline{v}$, $\overline{w}$ are the components of the ensemble mean velocity in the x, y, z directions, respectively, is the mean volume continuity equation.

C. SPECIES CONSERVATION

As stated in the previous section, the species continuity equation is

$$\frac{\partial}{\partial t} (\rho_A) + \text{div } (\rho_A \underset{\sim}{u}) = \text{div } [\rho D_{AB} \text{ grad } (\omega_A)] \qquad (2\text{-}18)$$

assuming no chemical reactions occur and brackish water can be approximated by a binary mixture of salts and water. The first term on the left side of this equation represents the rate of accumulation of the salts in a differential volume of fluid fixed relative to the coordinate system while the second term represents the net efflux of salts across the boundaries of the differential volume by convection. The term on the right side models the net influx of salts resulting from ordinary diffusion. This equation governs the instantaneous values of the flow field variables at the continuum level.

To obtain a useful equation at the turbulence level it is necessary to treat the density and mass diffusivity in Eq. (2-18) as constants. Neither assumption is in serious error physically since the density varies by only 3% and the diffusivity by 6% for the range of salinities encountered in estuaries. Moving ρ inside the gradient operator gives

$$\frac{\partial}{\partial t}(\rho_A) + \text{div}\,(\rho_A \underset{\sim}{u}) = \text{div}\,[D_{AB}\,\text{grad}\,(\rho_A)]. \qquad (2\text{-}19)$$

Substituting ensemble mean plus turbulent fluctuation variables for ρ_A and $\underset{\sim}{u}$ in this equation and applying the averaging operation introduced in the previous section gives

$$\left\langle \frac{\partial}{\partial t}(\rho_A + \rho_A') + \text{div}[(\overline{\rho}_A + \rho_A')(\overline{\underset{\sim}{u}} + \underset{\sim}{u}')] - \text{div}[D_{AB}\,\text{grad}(\overline{\rho}_A + \rho_A')] \right\rangle = 0$$

which reduces to

$$\frac{\partial}{\partial t}(\overline{\rho}_A) + \text{div}\,(\overline{\rho}_A \overline{\underset{\sim}{u}}) = \text{div}\,[D_{AB}\,\text{grad}(\overline{\rho}_A)] - \text{div}\langle \rho_A' \underset{\sim}{u}' \rangle \qquad (2\text{-}20)$$

after rearrangement. The physical significance of each term in Eq. (2-20) is the same as for the corresponding term in Eq. (2-18) except that a new term representing the net influx of salts into the differential volume by turbulent diffusion now appears. The correlation $\langle \rho_A' \underset{\sim}{u}' \rangle$, frequently called the turbulent mass flux vector, is an unknown that must be modeled in terms of known flow field variables if a closed system of working equations is to be had. A common practice is to invoke an analogy with the ordinary diffusion term in the above equation and write the turbulent mass flux vector as

$$\langle \rho_A' \underset{\sim}{u} \rangle = - \underset{\approx}{D}_{(t)} \cdot \text{grad}\,\overline{\rho}_A \qquad (2\text{-}21)$$

where $\underset{\approx}{D}_{(t)}$ is a second-order tensor whose components are turbulent diffusion coefficients or eddy diffusivities and the dot $\cdot$ denotes the operation of taking an inner product in cartesian coordinates. See page 25 of Hinze (1959) or Corrsin (1974) for a discussion of the necessity of using a second-order tensor as opposed to a scalar in anything other than isotropic turbulence.

Inserting the expression for $\langle \rho_A' \underset{\sim}{u}' \rangle$ given by Eq. (2-21) into Eq. (2-20) and rearranging yields

$$\frac{\partial}{\partial t}(\bar{\rho}_A) + \text{div}\,(\bar{\rho}_A \underset{\sim}{\bar{u}}) = \text{div}\,(\underset{\approx}{D}_{(e)} \cdot \text{grad}\,\bar{\rho}_A) \tag{2-22}$$

where $\underset{\approx}{D}_{(e)}$ is the effective diffusivity tensor given by

$$\underset{\approx}{D}_{(e)} = D_{AB}\underset{\approx}{I} + \underset{\approx}{D}_{(t)}. \tag{2-23}$$

$\underset{\approx}{I}$ is the identity tensor; the off-diagonal components of $\underset{\approx}{D}_{(e)}$ are generally nonzero. It is known that for fully turbulent flows the diagonal components of $\underset{\approx}{D}_{(t)}$ are several orders of magnitude larger than D_{AB}. The flow in this work is considered to be fully turbulent everywhere except in close proximity to solid boundaries and the equations are applied away from these boundaries. Therefore, ordinary diffusion is neglected and $\underset{\approx}{D}_{(e)}$ set equal to $\underset{\approx}{D}_{(t)}$. The components of $\underset{\approx}{D}_{(e)}$ are unknowns like those of the vector $\langle \rho'_A \underset{\sim}{u}' \rangle$ and are subject to measurement and prediction. This topic will be taken up in a latter section of the present chapter. Finally, by dividing Eq. (2-22) with the density ρ taken as a constant and utilizing the relation between mass fraction salts ω_A and normalized density anomaly S given by Eq. (2-5), one can rewrite Eq. (2-22) as

$$\frac{\partial}{\partial t}(\bar{S}) + \text{div}\,(\overline{S\underset{\sim}{u}}) = \text{div}\,(\underset{\approx}{D}_{(e)} \cdot \text{grad}\,\bar{S}). \tag{2-24}$$

This is the mean species continuity equation. In terms of its components in a rectangular cartesian coordinate system it reads

$$\begin{aligned}
&\frac{\partial}{\partial t}(\bar{S}) + \frac{\partial}{\partial x}(\overline{Su}) + \frac{\partial}{\partial y}(\overline{Sv}) + \frac{\partial}{\partial z}(\overline{Sw}) = \\
&\frac{\partial}{\partial x}\left[D_{xx}\frac{\partial}{\partial x}(\bar{S}) + D_{xy}\frac{\partial}{\partial y}(\bar{S}) + D_{xz}\frac{\partial}{\partial z}(\bar{S})\right] + \\
&\frac{\partial}{\partial y}\left[D_{yx}\frac{\partial}{\partial x}(\bar{S}) + D_{yy}\frac{\partial}{\partial y}(\bar{S}) + D_{yz}\frac{\partial}{\partial z}(\bar{S})\right] + \\
&\frac{\partial}{\partial z}\left[D_{zx}\frac{\partial}{\partial x}(\bar{S}) + D_{zy}\frac{\partial}{\partial y}(\bar{S}) + D_{zz}\frac{\partial}{\partial z}(\bar{S})\right].
\end{aligned} \tag{2-25}$$

D. MOMENTUM CONSERVATION

The equation governing the conservation of linear momentum in a Newtonian fluid at the continuum level may be written

$$\frac{\partial}{\partial t}(\underset{\sim}{u}) + \text{div}(\underset{\sim}{\underset{\sim}{uu}}) = -\frac{1}{\rho}\text{grad}(p) + \nu\,\text{div grad}(\underset{\sim}{u}) + \underset{\sim}{g}. \qquad (2\text{-}26)$$

Here, p is a scalar called the mean pressure, ν is the kinematic viscosity[1], and $\underset{\sim}{g}$ is the body force vector. All other symbols retain their previous definitions. The first term on the left models the accumulation of momentum per unit mass of fluid in a differential volume fixed relative to the coordinate system; the second term represents the net efflux of specific momentum across the differential volume boundaries due to convection. Note that in cartesian coordinates $\underset{\sim}{\underset{\sim}{uu}}$ is a dyadic product and, therefore, a symmetric second-order tensor. The first term on the right models the force per unit mass resulting from the mean pressure. The second term on the right symbolizes the shearing stresses per unit mass due to the relative movement or deformation of the fluid. The last term is the body force per unit mass resulting from the gravitational force field of the earth.

An equation in terms of the ensemble mean flow field variables is developed by substituting mean plus fluctuating variables for $\underset{\sim}{u}$ and p in Eq. (2-26) and ensemble averaging the result. To pursue this course of action successfully, it is necessary to assume that density fluctuations are so small that there are no relevant correlations between the fluctuations in density and the derivatives of fluctuations in velocity and mean pressure. In essence, this assumption along with the approxi-

---

[1] Eq. (2-26) has been written assuming that ν is constant. This is a valid approximation for brackish water; see Horne (1969).

mation that div $\underset{\sim}{u} = 0$ are tantamount to invoking the Boussinesq approximation; see Kamenkovich (1975). The result is

$$\frac{\partial}{\partial t}(\overline{\underset{\sim}{u}}) + \text{div}\,(\overline{\underset{\sim}{u}\underset{\sim}{u}}) =$$

$$-\frac{1}{\rho}\,\text{grad}\,\overline{p} + \nu\,\text{div grad}\,(\overline{\underset{\sim}{u}}) - \text{div}\,\langle\underset{\sim}{u}'\underset{\sim}{u}'\rangle + \underset{\sim}{g} \qquad (2\text{-}27)$$

which contains terms analogous to those in Eq. (2-26) plus a new term, $-\,\text{div}\,\langle\underset{\sim}{u}'\underset{\sim}{u}'\rangle$, which is the net increase of specific momentum due to turbulent shear stresses. The set of nine turbulent velocity correlations $\langle\underset{\sim}{u}'\underset{\sim}{u}'\rangle$, six of which are distinct, is commonly called the Reynold's stress tensor.

The components in the Reynold's stress tensor are unknowns and, just like the components of the turbulent mass flux vector, they must be modeled in terms of known flow field variables if a closed system of equations is to be had. Again the common practice is to write by analogy with the viscous stress term

$$\langle u'u'\rangle = -\,2\underset{\approx}{N}_{(t)}\cdot\underset{\approx}{D} \qquad (2\text{-}28)$$

where $\underset{\approx}{N}_{(t)}$, the eddy viscosity tensor, is a tensor of at least second-order and possibly, fourth-order, and $\underset{\approx}{D} = \{\text{grad}\,(\overline{\underset{\sim}{u}}) + [\text{grad}\,(\overline{\underset{\sim}{u}})]^{T}\}/2$ is the deformation tensor. If the latter is the case, the inner product operation would be replaced by the double inner product operation so that the result would be a second-order tensor. See page 22 of Hinze (1959) for a limited discussion of the need for tensorial representation of eddy viscosities. Specification of the components of $\underset{\approx}{N}_{(t)}$ will be taken up in a latter section of this chapter along with those of the eddy diffusivity tensor.

Substitution of the expression for $\langle\underset{\sim}{u}'\underset{\sim}{u}'\rangle$ given above into Eq. (2-27) produces

$$\frac{\partial}{\partial t}(\underset{\sim}{\bar{u}}) + \text{div}\,(\underset{\sim}{\overline{uu}}) = -\frac{1}{\rho}\,\text{grad}\,\bar{p} + 2\,\text{div}\,(\underset{\approx}{N}_{(e)}\cdot\underset{\approx}{D}) + \underset{\sim}{g} \qquad (2\text{-}29)$$

where $\underset{\approx}{N}_{(e)}$ is the effective eddy viscosity tensor defined by

$$\underset{\approx}{N}_{(e)} = \nu\underset{\approx}{I} + \underset{\approx}{N}_{(t)}. \qquad (2\text{-}30)$$

In fully turbulent flows the diagonal components of $\underset{\approx}{N}_{(t)}$ are generally several orders of magnitude larger than the kinematic viscosity; therefore, following the argument of the previous section $\underset{\approx}{N}_{(e)}$ is set equal to $\underset{\approx}{N}_{(t)}$.

With the addition of one more term to the right-hand side, Eq. (2-29) becomes the working momentum conservation equation. This term accounts for the fact that the governing equations are written for a coordinate system rotating relative to the inertial coordinate system attached to the fixed stars. Called the Coriolis effect, it is written $-2\underset{\sim}{\Omega} \times \underset{\sim}{\bar{u}}$ where $\underset{\sim}{\Omega}$ is the earth's angular velocity vector and the cross x denotes the vector product operation. See page 22 of Waldrop (1973) for a fuller discussion of the origin of the effect. Its consequences are that large scale flows in the northern hemisphere tend to turn to the right if not restrained by a boundary parallel to the flow direction. If such a boundary is present, slopes in the free surface result that balance this apparent force through pressure gradients. Dyer (1973) discusses the influence of the Coriolis effect on various types of estuaries. The final form of the mean momentum equation is

$$\frac{\partial}{\partial t}(\underset{\sim}{\bar{u}}) + \text{div}\,(\underset{\sim}{\overline{uu}}) =$$

$$-\frac{1}{\rho}\,\text{grad}\,(\bar{p}) + 2\,\text{div}\,(\underset{\approx}{N}_{(e)}\cdot\underset{\approx}{D}) + \underset{\sim}{g} - 2\underset{\sim}{\Omega} \times \underset{\sim}{\bar{u}} \qquad (2\text{-}31)$$

In expanded form, the x-directed component of this vector equation is

$$\frac{\partial}{\partial t}(\bar{u}) + \frac{\partial}{\partial x}(\overline{uu}) + \frac{\partial}{\partial y}(\overline{uv}) + \frac{\partial}{\partial z}(\overline{uw}) = -\frac{1}{\rho}\frac{\partial}{\partial x}(\bar{p})$$

$$+ \frac{\partial}{\partial x}\{N_{xx}[\frac{\partial}{\partial x}(\bar{u}) + \frac{\partial}{\partial x}(\bar{u})] + N_{xy}[\frac{\partial}{\partial y}(\bar{u}) + \frac{\partial}{\partial x}(\bar{v})] + N_{xz}[\frac{\partial}{\partial z}(\bar{u}) + \frac{\partial}{\partial x}(\bar{w})]\}$$

$$+ \frac{\partial}{\partial y}\{N_{xx}[\frac{\partial}{\partial x}(\bar{v}) + \frac{\partial}{\partial y}(\bar{u})] + N_{xy}[\frac{\partial}{\partial y}(\bar{v}) + \frac{\partial}{\partial y}(\bar{v})] + N_{xz}[\frac{\partial}{\partial z}(\bar{v}) + \frac{\partial}{\partial y}(\bar{w})]\}$$

$$+ \frac{\partial}{\partial z}\{N_{xx}[\frac{\partial}{\partial x}(\bar{w}) + \frac{\partial}{\partial z}(\bar{u})] + N_{xy}[\frac{\partial}{\partial y}(\bar{w}) + \frac{\partial}{\partial z}(\bar{v})] + N_{xz}[\frac{\partial}{\partial z}(\bar{w}) + \frac{\partial}{\partial z}(\bar{w})]\}$$

$$+ g_x - 2(\Omega_y \bar{w} - \Omega_z \bar{v}). \qquad (2\text{-}32)$$

Analogous equations exist for the y and z directions.

E. ENERGY CONSERVATION

For completeness an energy equation will be presented even though it is not used because the estuary under consideration is isothermal. If, as shown in Slattery (1972) or Bird, Stewart and Lightfoot (1960), a kinetic energy equation derived from the momentum equations is subtracted from the total energy conservation equation, an equation for the internal energy results. If the internal energy is expressed in terms of temperature through the proper thermodynamic relationships, the fact that div $\underset{\sim}{u} = 0$ utilized, and allowance for energy transfer by thermal radiation made, this equation reads

$$C_V\left(\frac{\partial T}{\partial t} + \text{div}\,(\underset{\sim}{u}T)\right) = \frac{1}{\rho}\,\text{div}\,(\kappa\ \text{grad}\ T) + \frac{1}{\rho}\,\text{div}\,\underset{\sim}{q}_r + \nu\Phi_v \qquad (2\text{-}33)$$

where the new symbols are as follows:

C_V = heat capacity at constant volume[1]

T = temperature

κ = thermal conductivity[1]

---

[1]Both the heat capacity and thermal conductivity of seawater may be treated as constants over the ranges of temperature and salinity encountered here. See Horne (1969).

$\underset{\sim}{q}_r$ = thermal radiation flux vector

Φ_v = positive viscous dissipation function.

The viscous dissipation function is generally neglected unless the flow approaches sonic velocity and will not be considered further. Averaging this continuum equation over an ensemble of turbulent flow realizations gives

$$\frac{\partial}{\partial t}(\overline{T}) + \text{div}(\underset{\sim}{u}\overline{T}) = (\rho C_V)^{-1}\text{div}[\underset{\approx}{K}_{(e)}\text{grad}(\overline{T})] + (\rho C_V)^{-1}\text{div}(\overline{\underset{\sim}{q}_r}) \quad (2-34)$$

where

$$\underset{\approx}{K}_{(e)} = \kappa \underset{\approx}{I} + \underset{\approx}{K}_{(t)} \quad (2-35)$$

As in the previous two sections, the eddy conductivity $\underset{\approx}{K}_{(t)}$ is a second-order tensor with diagonal components several orders of magnitude larger than κ so that the effective conductivity $\underset{\approx}{K}_{(e)}$ may be set equal to $\underset{\approx}{K}_{(t)}$. Eq. (2-34) is the mean equation describing the temperature distribution within in a turbulent flow field once the components of $\underset{\approx}{K}_{(e)}$ have been specified. Its application is no different conceptually than that of the mean equation for conservation of normalized density anomaly, Eq. (2-24).

F. EDDY TRANSPORT COEFFICIENTS

This section will be limited to a discussion of eddy diffusivities and viscosities. Eddy conductivities would be handled in a fashion very similar to that for eddy diffusivities. At the outset, it should be made clear that eddy transport coefficients are properties of the flow field and not the fluid itself. Therefore, anytime significant changes are made in the boundary conditions or the flow geometry, the possibility that different eddy transport coefficients will be required has to be considered. To add to the complexity of

the problem, these coefficients may vary with position in steady flows and with time and position in unsteady flows.

As presented in the development of the mean conservation equations, the eddy transport coefficients have tensor properties. While the physics of the problem demand this character of the transport coefficients, it is only in the simplest of flow geometries that sufficiant numbers and types of measurements have been made to enable one to specify components other than those on the main diagonal. See Corrsin (1974) for a determination of off-diagonal components in the eddy conductivity tensor when applied to turbulent heat transfer in a boundary layer and a circular jet. In estuarine flows experimental measurements and theory have advanced only to the point that it is possible to estimate the diagonal components. Consequently, the eddy diffusivity and viscosity tensors must be simplified to

$$\begin{bmatrix} D_{xx} & 0 & 0 \\ 0 & D_{yy} & 0 \\ 0 & 0 & D_{zz} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} N_{xx} & 0 & 0 \\ 0 & N_{yy} & 0 \\ 0 & 0 & N_{zz} \end{bmatrix}$$

respectively. This restricted form of the eddy transport tensors implies that they are oriented with their major axes parallel to the coordinate axes. Since the selection of the coordinate directions are somewhat arbitrary, particularly in the horizontal plane, and the magnitude of an eddy transport tensor component should depend on whether it is for a direction parallel or perpendicular to the flow or stable density stratification, the above simplified eddy transport tensors are not completely adequate for geometrically complex, time-dependent flows. To utilize what remains of the eddy diffusivity tensor, one often assumes that D_{xx} equals D_{yy} and neglects them since

mass transfer in the horizontal plane is dominated by advection or sets them equal to D_{zz} on the assumption that the turbulence is nearly isotropic. In the case of the eddy viscosity tensor, N_{xx} and N_{yy} are retained and often set equal and N_{zz} is ignored. The dependence of the remaining eddy transport coefficients on stable density stratification is formulated in terms of the gradient Richardson number; this topic is discussed below. Furthermore, when a solution to the equations is obtained by a finite difference scheme, there are apparent or artificial viscosities and diffusivities that compensate for the neglected horizontal transport and confuse the issue when it is not. These simplifications in the eddy transport tensors allow attention to be focused on the prediction and measurement of D_{zz}, N_{xx} and N_{yy}, a topic to be considered shortly.

Although eddy transport coefficients can be determined by dye or particle release experiments, they are generally calculated from field measurements of mean flow properties through the appropriate conservation equations. Since measurements of estuarine flows are expensive and prone to error, there is a dearth of quality eddy coefficient data based on short term (say, one minute) averages of the flow properties. More frequently, the eddy coefficients found in the literature are based on averages of the mean flow properties over the entire tide cycle. These tidal-average eddy coefficients are generally an order of magnitude smaller than the short-term values and are not appropriate for the problem at hand. Dyer (1973) presents tables of tidal-average coefficients for salt and momentum transfer in several different estuaries and quotes a few short-term eddy viscosity values obtained by Bowden and others. The short-term values are not directly

applicable to Mobile Bay because the bay in which they were measured is geometrically different. Consequently, the eddy transport coefficients used in the present model are estimated values obtained from theoretical and empirical considerations.

Another number associated with turbulent transport of scalar quantities is the dispersion coefficient. This coefficient appears in the diffusive terms of conservation equations that have been spatially averaged in one or two directions in order to reduce the dimensionality of a given problem. It has the two-fold task of accounting for (1) the difference between the convective transport resulting from the actual velocity distribution and that predicted by the spatially-averaged velocity and (2) the turbulent diffusive transport. See page 432 of Daily and Harleman (1966) for a rudimentary discussion of these coefficients. Levenspiel and Bischoff (1963) develop the concept at length. Since the present model is fully three dimensional, dispersion coefficients are inappropriate.

The following discussion of eddy transport theory parallels that of Bowden and Hamilton (1975). While the discussion is phrased in terms of a two-dimensional flow, it is applicable to three-dimensional flow in light of the approximations presented in the first paragraph of this section. Furthermore, the theory is based on a broad spectrum of flow problems in which estuarine flows are a representative, but complex case.

The Prandtl mixing length hypothesis relates the eddy viscosity coefficient to the magnitude of the mean velocity gradient by

$$N_{xx} = \ell^2 \left| \frac{\partial}{\partial z}(\bar{u}) \right| \qquad (2\text{-}36)$$

where ℓ, the mixing length, is to be prescribed in some manner and the flow is assumed to be parallel to the x-axis and to vary only in the z-direction. For instance, in the constant stress layer of turbulent flow parallel to a wall, the mixing length is given by

$$\ell = k_o (z + z_o) \tag{2-37}$$

where k_o is Von Karman's constant and z_o is the size of the wall roughness elements. Since the velocity profile in this layer is logarithmic in the distance from the surface, the eddy viscosity is

$$N_{xx} = k_o u_* (z + z_o) \tag{2-38}$$

where u_*, the friction velocity, is related to the wall shear stress τ_o by $u_* = \sqrt{\tau_o/\rho}$. For uniform flow in a two-dimensional, open channel of depth h and with no applied stress at the free surface, a mixing length of the form

$$\ell = k_o z (1 - z/h)^{1/2} \tag{2-39}$$

has been proposed. The corresponding velocity profile is logarithmic and the eddy viscosity coefficient given by

$$N_{xx} = k_o u_* z (1 - z/h). \tag{2-40}$$

Both ℓ and N_{xx} become zero at the free surface, approach small values at $z = z_o$ and have maxima at mid-depth. The maximum value of N_{xx} is

$$(N_{xx})_{max} = \frac{1}{4} k_o u_* h. \tag{2-41}$$

While the preceding formulations for the mixing length are derived for steady flows, they appear to be applicable to neutrally stable tidal flows during maximum flood and ebb. At other times during the tide cycle, the velocity profiles deviate significantly from the logarithmic curve. Furthermore, the velocity profile near

the free surface exhibits a weak dependence on N_{xx} so that to a good approximation the eddy viscosity can be taken as a constant above the friction layer located at the bottom. Based on measurements in neutrally stable tidal currents offshore, Bowden, Fairbairn and Hughes suggested that N_{xx} increase linearly from the bottom by Eq. (2-38) up to $z = \alpha h$ and then have the value

$$N_{xx} = k_o \alpha u_* h \tag{2-42}$$

from there to the free surface. The corresponding velocity profile is logarithmic in the lower segment and parabolic above. For their observations, $\alpha = 0.14$ gave the best fit.

Order of magnitude estimates of the eddy viscosity during maximum flood and ebb currents can be made with the formulations presented above. Utilizing the empirical relationship

$$\tau_o = \rho u_*^2 = k\rho U^2 \tag{2-43}$$

where k is a friction factor having a value in the range 0.0025-0.005 and U is the depth-mean velocity, Eqs. (2-41) and (2-42) may be rewritten as

$$(N_{xx})_{max} = 5.1 \times 10^{-3}\ Uh \tag{2-44}$$

and

$$(N_{xx})_{max} = 2.9 \times 10^{-3}\ Uh \text{ for } z > 0.14\ h \tag{2-45}$$

respectively, if Von Karman's constant is taken as 0.41 and the friction factor as 0.0025, a typical value. In addition, the eddy diffusivity D_{zz} is generally presumed to be equal to the eddy viscosity N_{xx} in neutrally stable or homogeneous waters. As pointed out by Bowden and Hamilton (1975), it is physically unrealistic to assume that N_{xx} remains constant throughout the tide cycle, and yet tidal currents

appear to be reasonably well modeled with a time-independent eddy viscosity. When nonhomogeneous waters are considered, the eddy transport coefficients may be modified to account for the reduction in vertical turbulent transport induced by stable density stratifications. A set of modifications, proposed by Munk and Anderson (1948) as a result of a study of the thermocline in the open sea, are

$$N_{xx} = (N_{xx})_{max} (1 + 10Ri)^{-0.5} \tag{2-46}$$

and

$$D_{zz} = (D_{zz})_{max} (1 + 3.33Ri)^{-1.5} \tag{2-47}$$

where Ri is the gradient Richardson number given by

$$Ri = g_z \rho^{-1} \frac{\partial \rho}{\partial z} \left[\frac{\partial}{\partial z} (\bar{u})\right]^{-2} \tag{2-48}$$

and $(N_{xx})_{max} = (D_{zz})_{max}$ are values for neutrally stable flows. Different modifications, or dampening factors, are used for the eddy viscosity and eddy diffusivity because stable density stratification influences the two coefficients differently.

This examination of eddy transport theory is somewhat cursory by intent. The reasons for the superficiality are two-fold. First, there are several formulations for eddy transport coefficients in estuarine flows, none of which are completely adequate. Bowden and Hamilton (1975) devote an entire literature article to assessing the influence of eddy transport coefficients of the following forms on the results of a two-dimensional model in the vertical:

1) $N_{xx} = 2D_{zz}$ = a constant

2) $N_{xx} = N_o + N_1 h|U|$

   $D_{zz} = D_o + D_1 h|U|$

   where $N_o = 2D_o$ and $N_1 = 2D_1$

3) $N_{xx} = N_o + N_1 h|U|(1 + m\beta Ri)^q$

$D_{zz} = D_o + D_1 h|U|(1 + \beta Ri)^P$

where $N_o = 2D_o$, $N_1 = D_1$ and

m, β, q, P are constants.

The Richardson number in the above is of the overall type and is defined by

$$Ri = g_z h \Delta\rho U^{-2}$$

where $\Delta\rho$ is the density difference between the bottom and surface. Bowden and Hamilton conclude that the third form is the best model, but that it is not entirely adequate for the range of conditions encountered in their study. The present author objects to their use of the depth-mean velocity and overall Richardson number in these formulae. These parameters are of little value when tidal effects are weak and salt water intrusion is strong. In a three-dimensional model, Leendertse and Liu (1976) utilize eddy transport coefficients of the form

$$N_{xx} = N_o EXP\{-mg_z\rho^{-1} \frac{\partial\rho}{\partial z}[\frac{\partial}{\partial z}(\bar{u})]^{-2}\} + N_1\{[\frac{\partial}{\partial z}(\bar{u})]^{-2} + [\frac{\partial}{\partial z}(\bar{v})]^2\}^{1/2}$$

$$D_{xx} = D_o EXP\{-rg_z\rho^{-1} \frac{\partial\rho}{\partial z}\{[\frac{\partial}{\partial z}(\bar{u})]^2 + [\frac{\partial}{\partial z}(\bar{v})]^2\}^{-1}\}$$

$$+ D_1\{[\frac{\partial}{\partial z}(\bar{u})]^2 + [\frac{\partial}{\partial z}(\bar{v})]^2\}^{1/2}$$

where m and r are constants. For the y-directed momentum equation N_{xx} is replaced by N_{yy} and $\bar{u}$ by $\bar{v}$ in the exponentional term. Second, there are strong arguments that eddy transport coefficients are inadequate models of turbulence. Harsha (1971), after reviewing all the current eddy viscosity models for planar mixing layers and circular jets and comparing their predictions against the available experimental data, concluded:

> It is doubtful whether methods which fail to take into account the fact that the flow is turbulent, and not laminar with some badly behaving viscosity, can ever be made to agree with more than a small range of experiments. It is clearly time that the methods of analysis of turbulent flow recognize that it is indeed turbulent.

To overcome the limitations of simple eddy transport coefficient models, considerable work with turbulent kinetic energy models and Reynold's stress closure schemes has been conducted since the mid-1960's. For an introduction and recent reviews of the state of these turbulence models, the reader is referred to Chapter 5 of the textbook by Reynolds (1974) and the articles by Reynolds (1976) and Lumley and Khajeh-Nouri (1974). In order to give an impression of the complexity of the partial differential equations involved and the degree of empiricism required, a two-equation model proposed by Spalding and coworkers (1974) for a two-dimensional boundary layer will be presented. The equations are

$$N_{xx} = C_5 \rho q^2 \varepsilon^{-1}$$

$$\rho \frac{Dq}{Dt} = \frac{\partial}{\partial z} \left[C_4 N_{xx} \frac{\partial q}{\partial z} \right] + N_{xx} \left[\frac{\partial}{\partial z} (\bar{u}) \right]^2 - \rho \varepsilon$$

$$\rho \frac{D\varepsilon}{Dt} = \frac{\partial}{\partial z} \left[C_3 N_{xx} \frac{\partial \varepsilon}{\partial z} \right] + C_1 N_{xx} \varepsilon q^{-1} \left[\frac{\partial}{\partial z} (\bar{u}) \right]^2 - C_2 \rho \varepsilon^2 q^{-1}$$

where q is the turbulent kinetic energy per unit mass of fluid, ε is the dissipation rate mass of turbulent kinetic energy per unit mass and C_1 through C_5 are constants determined by applying the equations to limiting cases. This model requires that two more partial differential equations be solved and the attendant boundary conditions be

specified, neither of which are trivial tasks. Furthermore, Corrsin (1974) presents an analysis that indicates the concept of eddy transport coefficients is strictly applicable to limited ranges of length scales, time scales and variations in the mean flow properties that are not found in many turbulent flows of interest. The turbulent kinetic energy models are suspect because they explicitly contain eddy transport coefficients. To circumvent this problem, if it is one, one must use a Reynold's stress closure scheme that contains even more equations and empirical constants. See Lumley and Khajeh-Nouri (1974) for the details of one such scheme.

Despite the complexity and empiricism of these higher order models, they have the potential advantage of a wider range of applicability than the simpler eddy transport coefficient formulations. Still, none of these schemes have advanced to the point where they can be effectively applied to three-dimensional, time-dependent free surface flows with the available numerical methods and computing facilities. In light of the above situation, constant eddy transport coefficients multiplied by gradient Richardson number dependent dampening factors, as in Eqs. (2-46) and (2-47), were used in this work.

G. WATER LEVEL

A realistic model of a tidal estuary requires that one account for changes in the water level (i.e., the location of the free surface or the interface between the water and the atmosphere above it) with horizontal position and time. Noting that div $\underset{\sim}{\bar{u}} = 0$ implies fluid volume is conserved, one can derive an equation governing the vertical motion of the water level by performing a fluid volume balance on the

differential element shown in Fig. 2.1. The bottom of the element is located in a horizontal reference plane denoted by z_o that is below the lowest water level. The top of the element coincides with the free surface which has an elevation denoted by $\bar{h} = \bar{h}(x,y,t)$. Sides 1 and 2 are in planes perpendicular to the x-axis and have a width of Δy; sides 3 and 4 are perpendicular to the y-axis and have a width of Δx. The flux of fluid volume through the bottom of the element is denoted by $\bar{w}_o$. Performing the volume balance yields

$$\Delta y\left(\int_{z_o}^{\bar{h}_1} \bar{u}_1 dz - \int_{z_o}^{\bar{h}_2} \bar{u}_2 dz\right) + \Delta x\left(\int_{z_o}^{\bar{h}_3} \bar{v}_3 dz - \int_{z_o}^{\bar{h}_4} \bar{v}_4 dz\right) + \Delta x \Delta y \bar{w}_o$$

$$= \frac{\partial}{\partial t}\left(\Delta x \Delta y \int_{z_o}^{\bar{h}} dz\right)$$

where the subscripts 1 through 4 associate the variables with the appropriate element faces. Dividing this equation by $\Delta x \Delta y$ and taking the limits as Δx and Δy approach zero gives

$$\frac{\partial}{\partial t}\int_{z_o}^{\bar{h}} dz = -\frac{\partial}{\partial x}\int_{z_o}^{\bar{h}} \bar{u} dz - \frac{\partial}{\partial y}\int_{z_o}^{\bar{h}} \bar{v} dz + \bar{w}_o$$

or

$$\frac{\partial \bar{h}}{\partial t} = -\frac{\partial}{\partial x}\int_{z_o}^{\bar{h}} \bar{u} dz - \frac{\partial}{\partial y}\int_{z_o}^{\bar{h}} \bar{v} dz + \bar{w}_o \qquad (2\text{-}49)$$

since z_o is independent of time. Eq. (2-49) has been derived in terms of ensemble mean values of the variables h, u and v, and is referred to as the mean water level equation where mean is not to be confused with a time average over a tide cycle. If the instantaneous values (i.e., mean plus fluctuating components) were used instead and an ensemble average of the resulting equation taken, Eq. (2-49) plus the terms

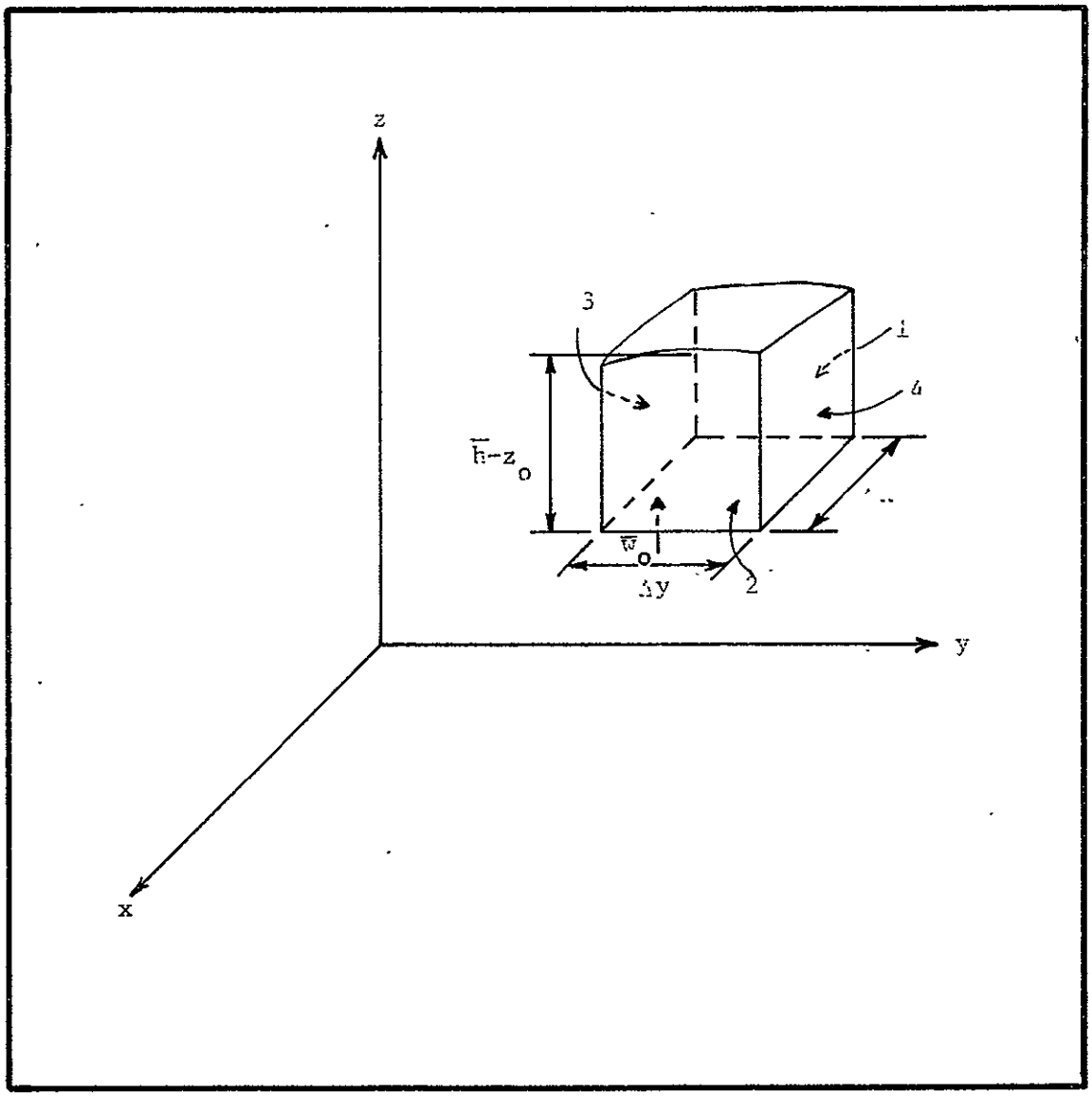


Figure 2.1 - Differential element at free surface

$$-\frac{\partial}{\partial x}\left\langle \int_{\bar{h}}^{\bar{h}+h'} (\bar{u} + u')\, dz \right\rangle - \frac{\partial}{\partial y}\left\langle \int_{\bar{h}}^{\bar{h}+h'} (\bar{v} + v')\, dz \right\rangle$$

added to the right hand side would be obtained. These terms are neglected in the subsequent work by assuming that the free surface is smooth and determined by long gravity waves (i.e., the length of the wave is large compared to the depth of the flow).

H. APPLIED EQUATIONS

The state equation, mean conservation equations for mass, species and momentum, and the mean water level equation set forth in the preceding sections are sufficient to describe the dynamics of an isothermal estuary if proper initial and boundary conditions are supplied. A general analytical solution to this set of partial differential equations and initial-boundary conditions is presently impossible because of the nonlinearities in the system. On the other hand, specific analytical solutions can be derived only after many drastic simplifications are made, and the results obtained are of limited applicability. If the phenomena modeled are restricted to those associated with long gravity waves and essentially horizontal flows, then the inertial and convective acceleration, turbulent momentum diffusion and Coriolis terms in the vertical momentum equation are small compared to the pressure gradient and gravitational acceleration terms and may be neglected. Commonly used in estuary and oceanic modeling, this simplification is called the hydrostatic approximation (Dyer, 1973; Bowden, 1967). Besides being based on sound physical ground, this approximation has the added benefit of greatly reducing the computational effort required to obtain a solution because it facilitates the pressure field

calculation. The numerical scheme used in this work utilizes this fact to full advantage. However, the approximation should not be considered a limitation on the scheme because the neglected terms in the z-directed momentum equation could have been retained.

The other simplifications made concern the models for the turbulent diffusion of mass and momentum. The model used for turbulent mass diffusion is

$$D_1 \left[\frac{\partial^2}{\partial x^2}(\bar{S}) + \frac{\partial^2}{\partial y^2}(\bar{S})\right] + D_2 \frac{\partial^2}{\partial z^2}(\bar{S})$$

where eddy diffusivity D_1 is constant and D_2 is Richardson number dependent. The model for turbulent momentum diffusion is simplified to $N \frac{\partial^2}{\partial z^2}(\bar{u})$ in the x-directed momentum equation and $N \frac{\partial^2}{\partial z^2}(\bar{v})$ in the y-directed equation. In both equations the eddy viscosity N is Richardson number dependent. Turbulent diffusion in the z-directed momentum equation is neglected in making the hydrostatic approximation. The decision to retain the horizontal turbulent diffusion terms in the species continuity equation and to neglect the corresponding terms in the momentum equations is the fact that mass generally diffuses faster than momentum. For either quantity, the turbulent diffusion rates in the vertical are significantly larger than those in the horizontal and the horizontal convective rates dominate the corresponding turbulent diffusion rates. For convenience in future discussions, the actual set of equations used in the model are presented in Table 2.1.

I. INITIAL-BOUNDARY CONDITIONS

According to the theory of partial differential equations, second-order, linear, constant coefficient equations in one dependent and two or more independent variables can be classified as being hyperbolic,

TABLE 2.1

Applied Equations

I. State

$$\rho = \rho_f + \beta\bar{S} \tag{2-50}$$

II. Volume Conservation

$$\frac{\partial}{\partial x}(\bar{u}) + \frac{\partial}{\partial y}(\bar{v}) + \frac{\partial}{\partial z}(\bar{w}) = 0 \tag{2-51}$$

III. Species Conservation

$$\frac{\partial}{\partial t}(\bar{S}) = -\frac{\partial}{\partial x}(\bar{u}\bar{S}) - \frac{\partial}{\partial y}(\bar{v}\bar{S}) - \frac{\partial}{\partial z}(\bar{w}\bar{S})$$

$$+ D_1\left[\frac{\partial^2}{\partial x^2}(\bar{S}) + \frac{\partial^2}{\partial y^2}(\bar{S})\right] + D_2\frac{\partial^2}{\partial z^2}(\bar{S}) \tag{2-52}$$

IV. x-Directed Momentum Conservation

$$\frac{\partial}{\partial t}(\bar{u}) = -\frac{\partial}{\partial x}(\bar{u}^2) - \frac{\partial}{\partial y}(\bar{u}\bar{v}) - \frac{\partial}{\partial z}(\bar{u}\bar{w}) - \frac{1}{\rho}\frac{\partial}{\partial x}(\bar{p})$$

$$+ N\frac{\partial^2}{\partial z^2}(\bar{u}) + f\bar{v} \tag{2-53}$$

V. y-Directed Momentum Conservation

$$\frac{\partial}{\partial t}(\bar{v}) = -\frac{\partial}{\partial x}(\bar{u}\bar{v}) - \frac{\partial}{\partial y}(\bar{v}^2) - \frac{\partial}{\partial z}(\bar{v}\bar{w}) - \frac{1}{\rho}\frac{\partial}{\partial y}(\bar{p})$$

$$+ N\frac{\partial^2}{\partial z^2}(\bar{v}) - f\bar{u} \tag{2-54}$$

VI. z-Directed Momentum Conservation

$$0 = -\frac{1}{\rho}\frac{\partial}{\partial z}(\bar{p}) - g \tag{2-55}$$

VII. Eddy Transport Coefficients

$$N = N_{max}(1 + 10.\ R_i)^{-0.5}$$

$$D_1 = D_{max} \tag{2-56}$$

(continued)

TABLE 2.1 (continued)

$$D_2 = D_{max} \, (1 + 3.33 \, Ri)^{-1.5}$$

$$Ri = g \, \rho^{-1} \, \frac{\partial \rho}{\partial z} \, [\frac{\partial}{\partial z} \, (\overline{u}^2 + \overline{v}^2)^{0.5}]^{-2} \qquad (2\text{-}56)$$

VIII. Water Level

$$\frac{\partial}{\partial t} \, (\overline{h}) = - \frac{\partial}{\partial x} \int_{z_o}^{\overline{h}} \overline{u} dz - \frac{\partial}{\partial y} \int_{z_o}^{\overline{h}} \overline{v} dz + \overline{w}_o \qquad (2\text{-}57)$$

parabolic or elliptic by examining the coefficients of the highest order derivatives in the equation. Once the equation has been classified, specification of initial and boundary conditions follow in a straightforward manner. For instance, if the equation is parabolic, one initial condition (the value of the dependent variable as a function of position at the initial time) is specified throughout the domain of the problem and one boundary condition (either the value of the dependent variable or its first spatial derivative normal to the boundary or a condition on a combination of both values) is specified at all points along the boundary of the problem domain. Unfortunately the problem at hand can not be reduced to a second order, linear, constant coefficient equation in one dependent variable. In fact, it consists of a system involving seven dependent variables and seven equations of either the partial differential or algebraic type. Partial differential equation theory has not developed to the stage that such a complex system can be classified as to type and the required initial boundary conditions automatically determined. Thus, one must base the specification of these auxiliary conditions on the physics of the problem and on the requirements of the numerical scheme used to solve the system of equations. A set of auxiliary conditions based on the physical aspects of the problem are presented in Table 2.2 and discussed below. These initial-boundary conditions will be discussed again in the next chapter because their implementation in the numerical scheme is not always straightforward.

Since the problem is time dependent, obviously a set of initial conditions on the dependent variables will be required. These should be physically realistic with regard to the expected behavior of the

TABLE 2.2

Auxiliary Conditions for the Applied Equations

Initial Conditions[1]

$$\bar{u} = \bar{u}_o(x,y,z) \qquad \bar{S} = \bar{S}_o(x,y,z) \qquad \bar{p} = \bar{p}_o(x,y,z)$$
$$\bar{v} = \bar{v}_o(x,y,z) \qquad \bar{h} = \bar{h}_o(x,y) \qquad \rho = \rho_o(x,y,z) \quad (2\text{-}58)$$
$$\bar{w} = \bar{w}_o(x,y,z)$$

Boundary Conditions

o Solid surface

+ No mass transfer condition[2]

$$\frac{\partial \bar{S}}{\partial n} = (\text{grad } \bar{S}) \cdot \underset{\sim}{n} = 0 \quad (2\text{-}59)$$

+ Inviscid flow velocity conditions

$$\bar{\underset{\sim}{u}} \cdot \underset{\sim}{n} = 0 \qquad \frac{\partial}{\partial n} |\bar{\underset{\sim}{u}}| = 0 \quad (2\text{-}60)$$

+ Viscous flow velocity conditions

$$\bar{\underset{\sim}{u}} = 0 \quad (2\text{-}61)$$

+ Surface wave reflection condition

$$\frac{\partial \bar{h}}{\partial n} = 0 \quad (2\text{-}62)$$

o Flow surface

+ River mouth conditions

$$\bar{\underset{\sim}{u}} = \bar{\underset{\sim}{u}}_1(x,y,z,t) \qquad \bar{S} = \bar{S}_1(x,y,z,t) \qquad \frac{\partial \bar{h}}{\partial n} = 0 \quad (2\text{-}63)$$

+ Tidal inlet conditions

$$\frac{\partial}{\partial n}(\bar{\underset{\sim}{u}}) = 0 \qquad \bar{h} = \bar{h}_1(x,y,t) \qquad \bar{S} = \bar{S}_1(x,y,z,t) \text{ or } \frac{\partial \bar{S}}{\partial n} = 0 \quad (2\text{-}64)$$

---

[1] The subscripts 0 and 1 denote initial conditions and boundary conditions, respectively.

[2] $\underset{\sim}{n}$ denotes a unit vector normal to the boundary surface and n, the distance in the direction of $\underset{\sim}{n}$. (continued)

TABLE 2.2 (continued)

- o Free surface
  - + Pressure condition

    $\bar{p} = \bar{p}_1(x,y,t)$

  - + Shear stress condition

    $\tau_{xz} = \tau_{yz} = 0$ (2-65)

  - + Wind induced shear stress condition[3]

    $\tau_{xz} = c\, v_w^{4/3} \cos\theta_w \qquad \tau_{yz} = c\, v_w^{4/3} \sin\theta_w$ (2-66)

    where $v_w = v_{w1}(x,y,t)$ and $\theta_w = \theta_{w1}(x,y,t)$

[3] c is an empirical constant.

system for the imposed boundary conditions and they must satisfy those equations not containing time derivatives such as the volume conservation, hydrostatic and state equations. The boundary conditions can be grouped according to the type surface under consideration. These are solid, flow and free surfaces where the first consist of the bottom and banks of the bay, the second the river mouths and tidal inlets, and the third the interface between the water and the overlying atmosphere. The condition on the salt concentration at a solid surface is that there is no diffusion of salt through the surface and thus that the normal gradient (i.e., the gradient in the direction normal to the surface) of the concentration is zero. The conditions on the velocity vector at solid surfaces depend on the type of flow expected in the region adjacent to the surface. If the flow is inviscid, then the normal gradient of the magnitude of the velocity is zero. If the flow is viscous, the velocity vector itself will be zero at the surface. For reasons to be explained in the next chapter, the inviscid flow will be associated with solid surfaces corresponding to the banks of the bay and the viscous flow with the bottom. In both cases, the component of the velocity vector normal to the surface is zero at the surface. Another boundary condition associated with solid surfaces, vertical banks in particular, is that the normal gradient of the water level is zero at the surface. This condition is based on the fact that the theory of small amplitude surface waves on an ideal fluid predicts that the waves are reflected exactly at vertical walls.

The most troublesome boundary conditions to specify are those at flow surfaces because the nature of these conditions depend on whether inflow or outflow occurs and on what combination of velocity, water

level and salinity distributions one is predicting as opposed to forcing. In the present work, the velocity and salinity distributions in the river mouths are forced while the water level distribution is predicted. Since both inflow (i.e., flow into the bay) and outflow occur at the river mouths in the course of a tide cycle, the forced salinity distribution may become inappropriate if high salinity water is convected from the interior of the bay across the flow surface. The normal gradients of the water level at the river mouths are set to zero due to a lack of data and the physical intuition that these gradients have little influence on the water level in the interior of the bay. At tidal inlets ideally the water level distributions are forced and the velocity distributions are predicted. The salinity distributions may be forced over a portion of the tidal inlet, say below a certain elevation, and predicted over the remaining area during ebb tide and forced over the entire area during flood tide. In those portions of the flow surface where velocity or salinity distributions are predicted, the normal gradients of the pertinent variables are set to zero. As in the case of the river mouths, the forced salinity distribution in the tidal inlet may become inappropriate if low salinity water is convected through the corresponding area. The pressure and density distributions at both solid and flow surfaces are determined by the water level and salinity distributions as a consequence of the hydrostatic and state equations. On the free surface, the pressure, which is equal to the atmospheric pressure, and the shear stress are forced boundary conditions. The shear stress is related to the wind velocity v_w at an elevation 10. m above the water level through an empirical relationship presented by Hsu (1972).

CHAPTER 3

THE NUMERICAL MODEL

A. LITERATURE SURVEY

The governing equations presented in the preceding chapter form the basis for most estuarine models appearing in the literature. Differences in the published models stem from the simplifications of the governing equations used to make the problem more manageable and from the numerical techniques employed to solve the simplified equations. A comprehensive survey of the literature on estuary models would be prohibitively long since interest in the field has grown rapidly in recent years and resulted in numerous publications. However, if one limits oneself to time-dependent models that are truly three dimensional and account for tidal phenomena and transport of a scalar such as salt or heat, the relevant literature is considerably smaller. Numerical models meeting these requirements have been developed by Leendertse and others (1973), Waldrop and Farmer (1973), and Spraggs and Street (1975). Before listing other relevant literature and discussing these models in detail, the following observations are in order. A characteristic common to the three works listed is that explicit finite difference methods were used to solve the momentum and scalar transport equations. This trait is a consequence of the fact that in three-dimensional, time-variant fluid flow problems, fully implicit finite difference methods or finite element methods require more computer storage than is currently economical to use. Furthermore,

semi-implicit finite difference methods such as the alternating-direction-implicit scheme are extremely difficult to apply in three-dimensional problems with complex geometries. The reader is referred to Roache (1976) for a discussion of implicit finite difference methods and to Oden and others (1974) and Gallagher and others (1975) for finite element methods as applied to fluid flow problems. Street (1976) reviews several models applicable to estuarine systems but not necessarily meeting the requirements outlined above.

Leendertse and others (1973) and Leendertse and Liu (1975) employed the hydrostatic approximation and conservation equations that have been averaged over horizontal layers. These layers were of constant but not necessarily equal thickness in all but the top layer. The thickness of the top layer varied in space and time with the water level. Their calculation sequence is as follows. At each new time level, horizontal velocity components, salinities, temperatures and water levels were computed through the corresponding conservation equation from flow field variables at the preceding two time levels. Vertical velocities were obtained from the new horizontal velocities through the volume continuity equation. The density field was calculated from the new salinities and temperatures through a sophisticated state equation and the horizontal pressure gradients from the new water levels and densities through the hydrostatic equation. Turbulent diffusion of momentum, salt and heat was modeled with eddy transport coefficients in the vertical and dispersion coefficients in the horizontal directions. While the dispersion coefficients were equal and constant, the eddy transport coefficients depended on the vertical gradient of the local horizontal velocities and on the gradient

Richardson number. The Richardson number dependency promoted turbulent diffusion in unstable density stratifications and inhibited it in stable situations. Examples of these relations may be found on pa. 29 of Chapter 2. Since centered difference approximations to the space and time derivatives were used, the numerical scheme may be classified as a midpoint rule or leap-frog method. This method is conditionally stable for pure convection problems but unstable for combined convection-diffusion problems if the diffusion terms are evaluated with flow field variables at the same time level as those used to evaluate the convection terms. To prevent instability the turbulent diffusion terms were computed with flow field variables from the preceding time level. The maximum stable time step was limited by the Courant-Friedricks-Lewy (CFL) criterion and a diffusion stability condition. After extensive testing on simplified as well as real estuaries, Leendertse and Liu (1975) concluded that the model was viable even though special precautions were required to prevent instabilities from originating at tidal inlet boundaries.

Rather than make the hydrostatic approximation, Waldrop and Farmer (1973, 1974) elected to neglect the local acceleration and Coriolis terms in the vertical momentum equation but to retain the convective acceleration and turbulent diffusion terms in addition to the pressure gradient and gravity terms. The water level calculation was based on the approximation that the rate of change in the water level is proportional to the rate of change in the vertical velocity component at the free surface. This approximation becomes exact as the gradients in water level become small; but it should be re-examined when applied to problems where significant tidal variations occur. In

the publication cited, tides were not considered. Waldrop and Farmer stretched the grid system in the horizontal directions to improve resolution in the region of greatest interest. The calculation sequence is primarily the same as that in the model of Leendertse and others except that only flow field variables at the preceding time level were used, the water level was computed after the vertical velocity components, and the pressure field as opposed to the horizontal pressure gradient field was explicitly calculated. Turbulent diffusion was modeled with eddy transport coefficients that were spatially invariant but not necessarily equal in magnitude for each direction. The numerical scheme is a forward-in-time, centered-in-space method since forward difference approximations to time derivatives and central difference approximations to space derivatives were employed. The maximum stable time step was limited to approximately one half that given by the CFL criterion.

The model developed by Spraggs and Street (1975), Roberts and Street (1975), and Street (1976) is an extension of the marker-and-cell methods applied to constant composition, nonisothermal flows. The three velocity components and temperature at a new time level were computed from the flow field variables at the preceding time level through the corresponding conservation equations in an explicit manner. Then the pressure field was computed from the flow field variables at the previous time level through a modified Poisson equation derived from the momentum and volume conservation equations. Successive-overrelaxation was employed to solve the Poisson equation at each time level. The density was then calculated from a state equation. The water level distribution was obtained from a kinetic free surface equation by a

double-sweep alternating-direction-implicit method that rendered the solution insensitive to rotations of the coordinate system about the vertical axis. The turbulent diffusion terms were modeled with temporally and spatially variant eddy transport coefficients calculated from the rate of strain tensor for the mean velocity field and the dimensions of the local grid cell. The eddy transport coefficients did not depend on a Richardson number. Forward difference approximations were used for time derivatives and centered differences for the space derivatives except for the convective terms where a windward differencing scheme was employed. Apparently the critical stability limitation for this model was the CFL criterion. Because an iterative calculation was required to obtain the pressure field at each time level, Street (1976) concluded that the model is too expensive to be used for long real time calculations.

B. DESCRIPTION OF NUMERICS

The finite difference method utilized in the present work represents an extension of the method developed by Waldrop and Farmer (1974). The method and the modifications made to it since the last published description will be discussed in detail below. The discussion will proceed from the nondimensionalization of the partial differential equations to the grid system used and the temporal and spatial discretization of the applied equations to obtain finite difference equations. Then the calculation sequence will be presented. Finally, the numerical viscosity, stability, convergence and accuracy of the method will be considered.

Nondimensionalization of Applied Equations

It is generally accepted that proper scaling or normalization of the variables in a problem will result in a more accurate numerical solution. For this reason, the applied equations of Table 2.1 were rendered dimensionless with the following reference variables:

ρ_f = density of fresh water

H = length characteristic of the depth of the flow

g = gravitational acceleration

The nondimensionalization was accomplished by dividing the variables in the applied equations by combinations of the reference variables as shown in Table 3.1 and then canceling common factors. Note that the normalized density anomaly $\overline{S}$ does not appear in the table because it is already dimensionless. If the asterisks are dropped, equations identical to those of Table 2.1 are recovered except that ρ_f and g are replaced by unity; therefore, that table will not be repeated here. Note that dimensionless variables and equations will be used in the remainder of this work except where it is noted otherwise. To emphasize this fact a period will be substitued for the hyphen in equation numbers when the equation is nondimensional. Furthermore, the bar over ensemble mean variables will be dropped since it is understood that the mean conservation equations are being used.

Configuration of Grid Systems

The implementation of a finite difference method requires that the partial differential equations be discretized, that is, the partial derivatives in the equations must be replaced by appropriate finite difference operators. The differences for these operators are defined

TABLE 3.1

Dimensionless Variables

1.	Density	$\rho^* = \rho/\rho_f$
2.	State equation constant	$\beta^* = \beta/\rho_f$
3.	Position vector	$\underset{\sim}{x}^* = \underset{\sim}{x}/H$
4.	Pressure	$p^* = \overline{p}/(\rho_f Hg)$
5.	Time	$t^* = t/\sqrt{H/g}$
6.	Velocity	$\underset{\sim}{u}^* = \overline{\underset{\sim}{u}}/\sqrt{Hg}$
7.	Eddy viscosity	$N^* = N/(H\sqrt{Hg})$
8.	Eddy diffusivity	$D^* = D/(H\sqrt{Hg})$
9.	Coriolis parameter	$f^* = f/(\sqrt{g/H})$

in terms of a grid system superimposed on the time and space domain of the problem. Modeling Mobile Bay required the use of two spatial grids of different dimensions, a three-dimensional grid for the bay proper and a two-dimensional grid for the ship channel. A two-dimensional vertical grid was judged adequate for the ship channel because it is straight, deep and narrow relative to the bay dimensions. Furthermore, attempting to use a three-dimensional grid for the bay and channel combined would result in the waste of many grid points because they would be located in the ground. The three-dimensional spatial grid employed in the bay proper is indicated schematically in Fig. 3.1. The grid system for the space and time domain is generated by repeating the spatial grid system at increments of Δt along the t-axis.

The discretization process in effect replaces the problem variables defined at an infinity of points in time and space by a set of discrete variables defined at a finite number of points in time and space generated by the intersections of the grid planes. Thus it becomes convenient to number or index the grid points with integers and to specify grid points with their indices. Thus, the space and time coordinates of a general grid point are denoted by $t = t\{n\}$, $x = x\{i\}$, $y = y\{j\}$ and $z = z\{k\}$ where the integers (n,i,j,k) are specifically associated with the coordinates (t,x,y,z), respectively. The value of any time and space dependent variable, say Q, at a general grid point is denoted by $Q\{n,i,j,k\}$.

A horizontal view of the three-dimensional grid with the boundaries of Mobile Bay indicated is presented in Figure 3.2. The origin of the coordinate system for the grid is located 5. m below mean sea level at the State Docks below the Mobile River. The positive x-axis

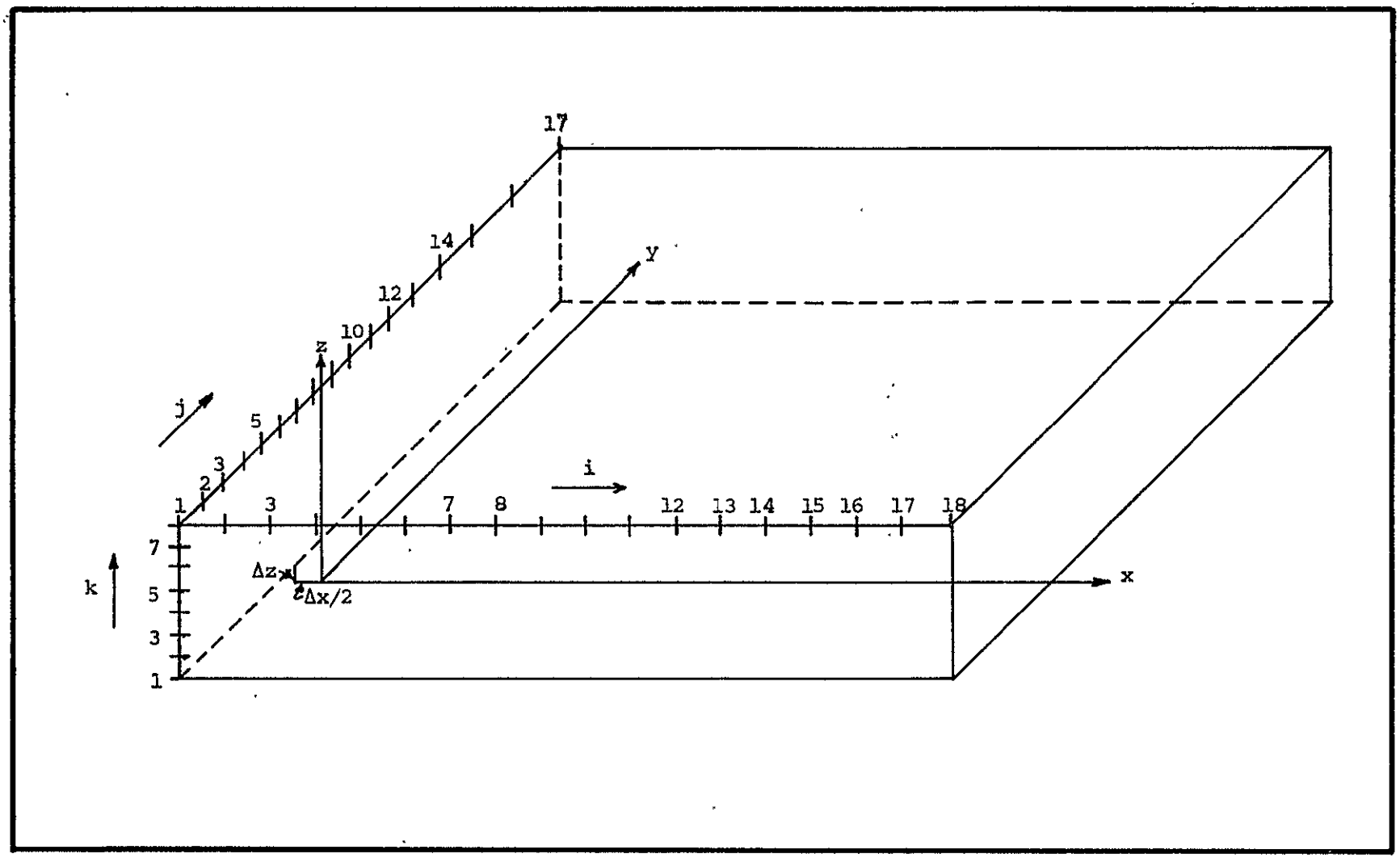


Figure 3.1 - Three dimensional grid system for bay

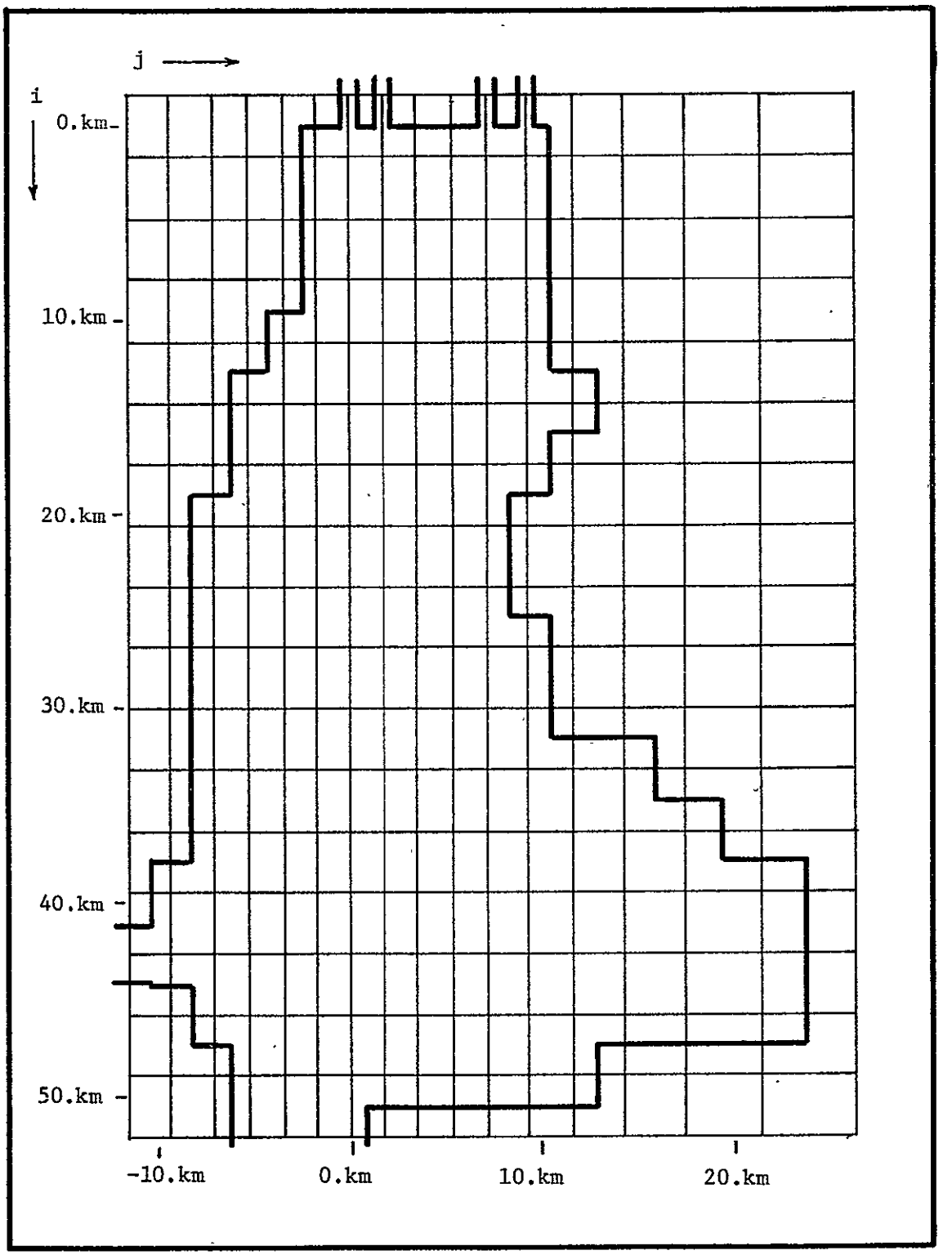


Figure 3.2 - Top view of three-dimensional grid and bay boundaries

is directed due south, the positive y-axis due east, and the positive z-axis upward. The ship channel is approximately aligned with the x-axis in reality and was taken as being exactly aligned with it in the model. The y-axis coincides with the causeway across the northern end of the bay. The grid planes intersecting the x- and z-axes are spaced at constant increments Δx and Δz, respectively, while the increments between the grid planes intersecting the y-axis vary with distance along that axis. The variation in the y-axis grid increments is such that resolution is increased in the vicinity of the ship channel at the expense of reduced resolution near the east and west banks. The motive for generating such a grid system was that large horizontal gradients in the flow variables were expected to occur along the channel and high resolution in the area of steep gradients generally results in more accurate solutions overall.

Programming an efficient computer code is facilitated if the grid increments for a given coordinate direction are equal in size. Equal increments in the independent variable can be obtained by defining a coordinate transformation or stretching function, say $Y = Y(y)$, such that equal increments in Y give the desired distribution of increments in y, and then rewriting the partial differential equations in terms of the new independent variable Y. This last step is accomplished by applying the chain rule to the derivatives involving the coordinate to be transformed. For instance, working with the general dependent variable Q, one has that

$$\frac{\partial Q}{\partial y} = \frac{\partial Q}{\partial Y} Y' \tag{3.1}$$

and

$$\frac{\partial^2 Q}{\partial y^2} = \frac{\partial^2 Q}{\partial Y^2} (Y')^2 + \frac{\partial Q}{\partial Y} Y'' \tag{3.2}$$

where

$$Y' = \frac{dY}{dy} \text{ and } Y'' = \frac{d^2 Y}{dy^2} \tag{3.3}$$

Eqs. (3.1) and (3.2) are used to express the derivatives with respect to y in the equations of Table 2.1 in terms of derivatives with respect to Y so that y appears as an independent variable in derivatives of Y only. The particular stretching function employed to transform the y-axis in the Mobile Bay model is

$$Y = k_2^{-1} \tan^{-1} (k_1^{-1} y) \tag{3.4}$$

or $$y = k_1 \tan (k_2 Y) \tag{3.5}$$

where Y ranges from -1. to 1. and k_1 and k_2 are constants. The constant k_2 controls the distribution of increments in y. The smaller the value of k_2, the more evenly the grid planes tend to be distributed; the larger the value of k_2, the more the grid planes tend to be concentrated about the x-axis. In any case, k_2 should be between 0. and $\pi/2$. The constant k_1 is fixed by the maximum absolute value of the variable y.

The two-dimensional grid employed in the ship channel is shown schematically in Fig. 3.3. Having the same origin and grid increments in the x- and z-directions as the three-dimensional grid for the bay, the two-dimensional grid coincides with the latter for positive values of z.

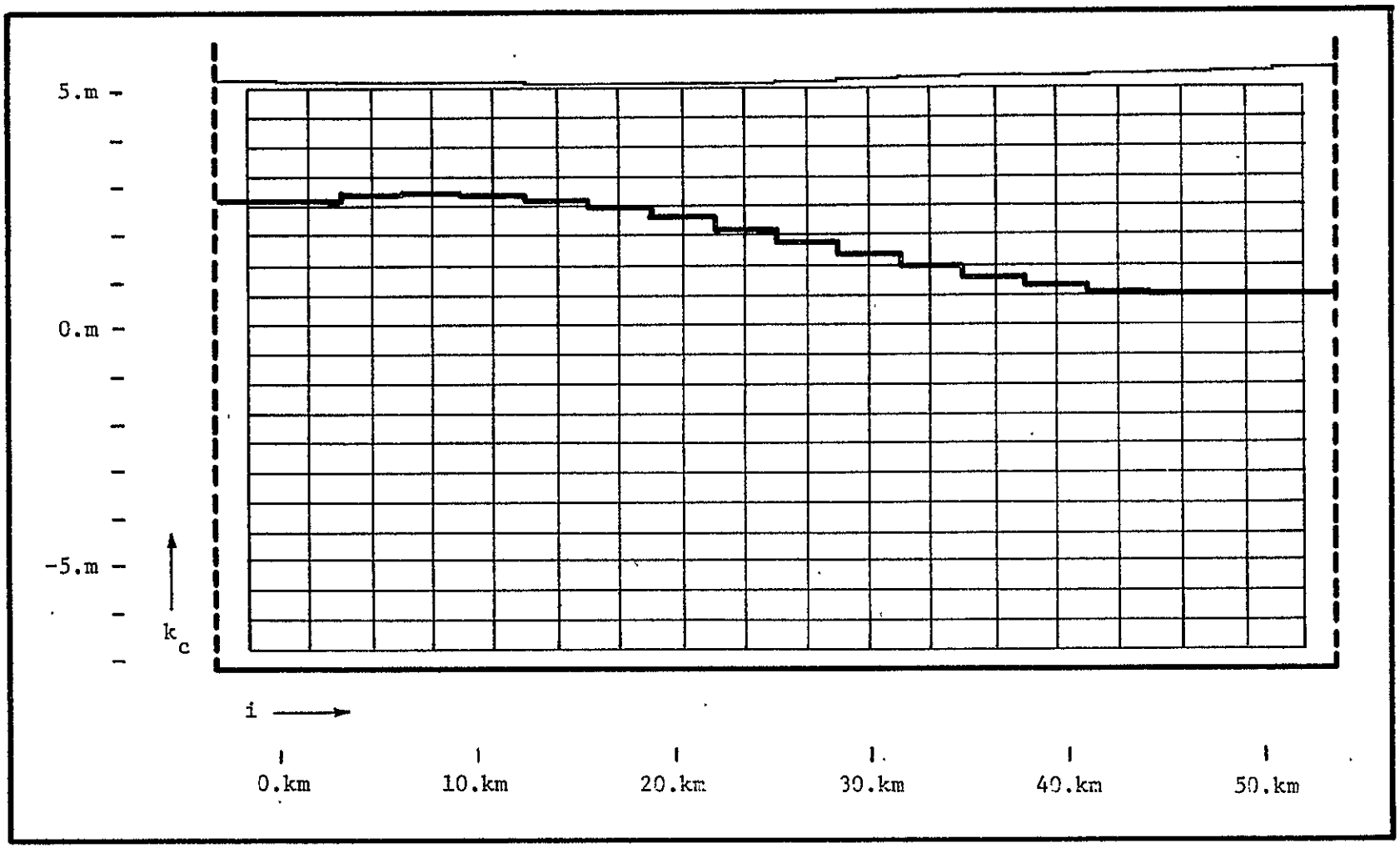


Figure 3.3 - Side view of two-dimensional grid system and bay-channel boundaries

Model Boundaries

The solid and flow surfaces in the Mobile Bay model were established by approximating the prototype boundaries with a set of rectangular, flat surfaces oriented normal to the coordinate axes. All vertical solid surfaces in the model are located midway between the grid planes that bound them. Thus, a vertical solid surface in the model oriented perpendicular to the x-axis and falling between x{i} and x{i+1} is located at x{i} + Δx/2, while one oriented perpendicular to the stretched y-axis and falling between y{i} and y{j+1} is located at $y\{j\} + (\Delta Y/2)[(Y'\{j\} + Y'\{j+1\}/2]^{-1}$. All flow surfaces are vertical and coincide with grid planes. Horizontal solid surfaces are positioned at any elevation desired within the grid below $z\{k_{max} - 2\}$. This restriction was imposed so that at least one general interior point exists in each grid column. The reason for the additional flexibility in positioning the horizontal surfaces will become apparent when the boundary conditions are discussed.

Using the approximations described above, the sides of Mobile Bay are treated as steep banks which is reasonably accurate. The bottom of the bay is fitted with a patchwork of horizontal rectangles located at elevations corresponding to the average local depth at each vertical grid column and vertical rectangles to seal the gaps. The resulting model boundaries are illustrated in the top and side views of the grid system in Figs. 3.2 and 3.3, respectively.

The degree of approximation used in fitting the location of the free surface depends on the particular applied equation involved. For the horizontal momentum equations applied at grid points just below the free surface, a crude approximation is sufficient with the surface being

fixed at an elevation midway between the top grid plane and a fictitious grid plane one vertical grid increment above the top plane. An exception to this statement is that the gradient of the shear stress induced by wind is dependent on the actual free surface location at each vertical grid column. A better approximation is required for the vertical momentum and water level equations so the free surface is located a distance $h\{n,i,j\} - z\{k_{max}\}$ above the top grid plane at each vertical column. The latter fit to the free surface is shown in Fig. 3.3.

Finite Difference Equations

The appropriate finite difference operators referred to in the preceding subsection will now be presented. The presentation will be in terms of the fictitious dependent variable Q at an arbitrary grid point internal to the flow field. To facilitate the presentation of the finite difference equations, several of the operators will be assigned symbols.

First-order, spatial derivatives in the linear terms of the conservation equations are approximated by central difference operators. These operators are denoted by Δ_x^c where the subscript x indicates the coordinate direction with respect to which the difference is taken. The point about which the operator is applied and the size of the spatial increment used is indicated by the indices of the dependent variable. For instance, the first derivative of Q with respect to x evaluated at the grid point (n,i,j,k) is approximated by

$$\frac{\partial}{\partial x} Q\{n,i,j,k\} \approx \Delta_x^c Q\{n,i,j,k\}$$

$$= (2\Delta x)^{-1}(Q\{n,i+1,j,k\} - Q\{n,i-1,j,k\}) \qquad (3.6)$$

or if the coordinate has been stretched as y in the present work, by

$$\frac{\partial}{\partial y} Q\{n,i,j,k\} \approx \Delta_y^c Q\{n,i,j,k\}$$

$$= (2\Delta Y)^{-1}(Y'\{j\})(Q\{n,i,j+1,k\} - Q\{n,i,j-1,k\}) \qquad (3.7)$$

where $Y'\{j\}$ is given in Eqs. (3.3). The first-order derivative with respect to x evaluated at a point midway between the grid points (n,i,j,k) and (n,i+1,j,k) is approximated by

$$\frac{\partial}{\partial x} Q\{n,i+1/2,j,k\} \approx \Delta_x^c Q\{n,i+1/2,j,k\}$$

$$= (\Delta x)^{-1}(Q\{n,i+1,j,k\} - Q\{n,i,j,k\}) \qquad (3.8)$$

Second-order, spatial derivatives in the conservation equations are also evaluated with central difference operators. The fact that the operator represents a second order derivative is indicated by repeating the subscripts on the symbol for a central difference operator twice. Thus the second derivative of Q with respect to x evaluated at the grid point (n,i,j,k) is approximated by

$$\frac{\partial^2}{\partial x^2} Q\{n,i,j,k\} \approx \Delta_{xx}^c Q\{n,i,j,k\}$$

$$= (\Delta x)^{-2}(Q\{n,i+1,j,k\} - 2Q\{n,i,j,k\} + Q\{n,i-1,j,k\}) \qquad (3.9)$$

or for the stretched coordinate y by

$$\frac{\partial^2}{\partial y^2} Q\{n,i,j,k\} \approx \Delta_{yy}^c Q\{n,i,j,k\}$$

$$= (\Delta Y)^{-2}(Y'\{j\})^2(Q\{n,i,j+1,k\} - 2Q\{n,i,j,k\} + Q\{n,i,j-1,k\})$$

$$+(2\Delta Y)^{-1}(Y''\{j\})(Q\{n,i,j+1,k\} - Q\{n,i,j-1,k\}) \qquad (3.10)$$

The second-order, central difference operators are applied only at defined grid points and not at points midway between grid points in the present work. The truncation error for a finite difference operator is the error made in approximating a derivative by the operator. This

error can be assessed by deriving the operator from Taylor series expansions of the dependent variable about the points where the operator is applied. The truncation errors for the central difference operators presented above are second order in the grid increment; that is, the error is proportional to the square of the grid increment and the operators are said to be second-order accurate.

The first-order derivatives in the convective terms of the momentum and species conservation equations are nonlinear and represent a special case in finite differencing when they are important to the problem solution. Apparently no single difference operator can duplicate all the properties associated with the convective derivatives. Roache (1976), Spraggs and Street (1975) and Codell (1973) discuss several difference operators in terms of such concepts as phase error, numerical viscosity, the conservative property, the transportive property and truncation error. Truncation error has already been defined and numerical viscosity will be discussed later in this chapter. Phase error is related to the rates at which the various wavelengths in a Fourier series representation of the dependent variable are transported. If some wavelengths move at the proper rates while others do not, phase errors are said to exist. A difference operator has the conservative property if it is capable of identically satisfying integral formulations of the conservation laws disregarding round-off errors. It has the transportive property if it convects a local disturbance (i.e., a disturbance at a single grid point) in the direction of the local velocity only. While the consensus is not unanimous, the following upwind difference operator presented by Roache (1976) appears to be superior in terms of the above mentioned properties. Let Q

represent a conserved quantity such as specific momentum or a chemical species and u and v represent velocity components in the x and y directions, respectively. Then the convective derivative is approximated by

$$\frac{\partial}{\partial x}(uQ)\{n,i,j,k\} \approx \Delta_x^w(uQ)\{n,i,j,k\} = (\Delta x)^{-1}(u_F Q_F - u_B Q_B) \qquad (3.11)$$

where $u_F = 0.5\ (u\{n,i+1,j,k\} + u\{n,i,j,k\})$

$$u_B = 0.5\ (u\{n,i,j,k\} + u\{n,i-1,j,k\})$$

$$Q_F = \begin{cases} Q\{n,i,j,k\} & \text{if } u_F \geq 0 \\ Q\{n,i+1,j,k\} & \text{if } u_F < 0 \end{cases}$$

$$Q_B = \begin{cases} Q\{n,i-1,j,k\} & \text{if } u_B \geq 0 \\ Q\{n,i,j,k\} & \text{if } u_B < 0 \end{cases}$$

and the subscripts F and B are intended to suggest forward and backward, respectively. For the stretched coordinate y, the operator is given by

$$\frac{\partial}{\partial y}(vQ)\{n,i,j,k\} \approx \Delta_y^w(vQ)\{n,i,j,k\}$$

$$= (\Delta Y)^{-1}(Y'\{j\})(v_F Q_F - v_B Q_B) \qquad (3.12)$$

where v_F, v_B, Q_F and Q_B are defined in a manner analogous to the corresponding variables in Eq. (3.11). The truncation error for this upwind difference operator is formally first-order in the grid increment, but the error approaches being second order when the conserved quantity is distributed uniformly (i.e., $Q_F = Q_B$).

All the terms involving spatial derivatives in the applied equations of Table 2.1 may be approximated by selecting an appropriate finite difference operator from those introduced above with one exception, the integro-differential term of the water level equation. The difference operator for this term and a symbol to denote the operator are now defined. The approximation is, letting m = n + 1,

$$\left(\frac{\partial}{\partial x}\int_{z\{k_{max}-1\}}^{h} u\,dz\right)\{n,i,j\} \approx \Delta_x^{w_I h\{n,i,j\}}{}_{z\{k_{max}-1\}} u\{m,i,j,k\}$$

$$= (8\Delta x)^{-1}[(u\{m,i+1,j,k_{max}\} + u\{m,i+1,j,k_{max}-1\} + u\{m,i,j,k_{max}\}$$

$$+ u\{m,i,j,k_{max}-1\})(h\{n,i+1,j\} + h\{n,i,j\} + 2\Delta z)$$

$$- (u\{m,i,j,k_{max}\} + u\{m,i,j,k_{max}-1\} + u\{m,i-1,j,k_{max}\}$$

$$+ u\{m,i-1,j,k_{max}-1\})(h\{n,i,j\} + h\{n,i,j+1\} + 2\Delta z)] \qquad (3.13)$$

or for a stretched coordinate such as y, it is

$$\left(\frac{\partial}{\partial y}\int_{z\{k_{max}-1\}}^{h} v\,dz\right)\{n,i,j\} \approx \Delta_y^{w_I h\{n,i,j\}}{}_{z\{k_{max}-1\}} v\{m,i,j,k\}$$

$$= [4(y\{j+1\}-y\{j-1\})]^{-1}[(v\{m,i,j+1,k_{max}\}+v\{n,i,j+1,k_{max}-1\}$$

$$+ v\{m,i,j,k_{max}\}+v\{m,i,j,k_{max}-1\})(h\{n,i,j+1\}+h\{n,i,j\}+2\Delta z)$$

$$- (v\{m,i,j,k_{max}\}+v\{m,i,j,k_{max}-1\}+v\{m,i,j-1,k_{max}\}$$

$$+ v\{m,i,j,k_{max}-1\})(h\{n,i,j\}+h\{n,i,j-1\}+2\Delta z)] \qquad (3.14)$$

Note that $(\Delta Y)^{-1}Y'\{j\}$ has been replaced by the equivalent $[(y\{j+1\} - y\{j-1\})/2]^{-1}$ and that $z\{k_{max}-1\}$ corresponds to the z_o of Eq. (2-5). The I in the symbol for the operator is intended to suggest the integral. Since the operator is centered about the point of its application, it should be second-order accurate.

Now all the terms involving spatial derivatives in the applied equations of Table 2.1 can be represented by finite differences. The results of doing so are given in Table 3.2. There is a one-to-one correspondence in the terms for each equation in the two tables with four exceptions. Remembering that the variables have been made dimensionless so that ρ_f and g are replaced by unity, one finds the exceptions to be as follows: (1) In the volume conservation equation, the gradients in the horizontal velocity components are evaluated at (n,i,j,k+1/2) by averaging central difference operators applied at

(n,i,j,k) and (n,i,j,k+1). The motive for using this particular approximation will become apparent when the calculation sequence is discussed below. (2) The z-directed momentum equation (or hydrostatic equation) has been modified before differencing it. The applied equation is

$$\frac{\partial p}{\partial z} = -\rho$$

which upon integration from elevation z to the free surface gives

$$p = \int_{z}^{h} \rho dz' \tag{3.16}$$

if atmospheric gage pressure is taken to be zero. If Eq. (3.16) is applied to freshwater in static equilibrium and having a water level located at $z = z\{k_{max}\}$ which corresponds to mean sea level in the present work, one has that

$$p_{eq} = \int_{z}^{z\{k_{max}\}} dz \quad \text{for } z \leq z\{k_{max}\}$$

$$p_{eq} = 0 \qquad \text{for } z > z\{k_{max}\} \tag{3.17}$$

Subtracting Eqs. (3.17) from Eq. (3.16), rearranging and applying $\rho = 1. + \beta S$ gives

$$p' = p - p_{eq} = \int_{z\{k_{max}\}}^{h} \rho dz' + \int_{z}^{z\{k_{max}\}} \beta S dz' \quad \text{for } z \leq z\{k_{max}\}$$

$$p' = p - p_{eq} = \int_{z}^{h} \rho dz' \qquad \text{for } z > z\{k_{max}\} \tag{3.18}$$

The pressure difference p' represents the deviation of actual pressure from a reference pressure caused by changes in water level and density. The advantage of using p' instead of p in calculations with numbers of finite length is primarily realized when pressure gradients are induced by density variations or small deviations in the water level from the

TABLE 3.2

Finite Difference Equations

I. State

$$\rho = 1. + \beta S\{n,i,j,k\}, \quad \rho^{-1} = 1. - \beta S\{n,i,j,k\} \tag{3.19}$$

II. Volume Conservation

$$\Delta_z^c w\{n,i,j,k+1/2\} = -0.5(\Delta_x^c u\{n,i,j,k+1\} + \Delta_x^c u\{n,i,j,k\} + \Delta_y^c v\{n,i,j,k+1\} + \Delta_y^c v\{n,i,j,k\}) \tag{3.20}$$

III. Species Conservation

$$\frac{\partial}{\partial t} S\{n,i,j,k\} = -\Delta_x^w(uS)\{n,i,j,k\} - \Delta_y^w(vS)\{n,i,j,k\} - \Delta_z^w(wS)\{n,i,j,k\}$$
$$+ D_1(\Delta_{xx}^c S\{n,i,j,k\} + \Delta_{yy}^c S\{n,i,j,k\}) + D_2\Delta_{zz}^c S\{n,i,j,k\} \tag{3.21}$$

$$S\{n+1,i,j,k\} = S\{n,i,j,k\} + (0.5)(\frac{\partial}{\partial t} S\{n,i,j,k\} + \frac{\partial}{\partial t} S\{n-1,i,j,k\})(\Delta t) \tag{3.22}$$

IV. x-Directed Momentum Conservation

$$\frac{\partial}{\partial t} u\{n,i,j,k\} = -\Delta_x^w(uu)\{n,i,j,k\} - \Delta_y^w(vu)\{n,i,j,k\} - \Delta_z^w(wu)\{n,i,j,k\}$$
$$- \rho^{-1}\Delta_x^c p'\{n,i,j,k\} + N\Delta_{zz}^c u\{n,i,j,k\} + fv\{n,i,j,k\} \tag{3.23}$$

$$u\{n+1,i,j,k\} = u\{n,i,j,k\} + (0.5)(\frac{\partial}{\partial t} u\{n,i,j,k\} + \frac{\partial}{\partial t} u\{n-1,i,j,k\})(\Delta t) \tag{3.24}$$

TABLE 3.2 (continued)

V. y-Directed Momentum Conservation

$$\frac{\partial}{\partial t} v\{n,i,j,k\} = -\Delta_x^w(uv)\{n,i,j,k\} - \Delta_y^w(vv)\{n,i,j,k\} - \Delta_z^w(wv)\{n,i,j,k\}$$
$$- \rho^{-1}\Delta_y^c p'\{n,i,j,k\} + N\Delta_{zz}^c v\{n,i,j,k\} - fu\{n,i,j,k\} \qquad (3.25)$$

$$v\{n+1,i,j,k\} = v\{n,i,j,k\} + (0.5)\left(\frac{\partial}{\partial t} v\{n,i,j,k\} + \frac{\partial}{\partial t} v\{n-1,i,j,k\}\right)(\Delta t) \qquad (3.26)$$

VI. z-Directed Momentum Conservation

$$p'\{n,i,j,k_{max}\} = \rho h\{n,i,j\} \qquad \text{for } k = k_{max}$$

$$\Delta_z^c p'\{n,i,j,k+1/2\} = -0.5\beta(S\{n,i,j,k\} + S\{n,i,j,k+1\}) \quad \text{for } k < k_{max} \qquad (3.27)$$

VII. Water Level

$$\frac{\partial}{\partial t} h\{n,i,j\} = -(\Delta_x^c I_{z\{k_{max}-1\}} h\{n,i,j\}) u\{n+1,i,j,k\}) - (\Delta_y^c I_{z\{k_{max}-1\}} h\{n,i,j\}) v\{n+1,i,j,k\})$$
$$+ \gamma(\Delta_{xx}^c h\{n,i,j\} + \Delta_{yy}^c h\{n,i,j\}) \qquad (3.28)$$

$$h\{n+1,i,j\} = h\{n,i,j\} + (0.5)\left(\frac{\partial}{\partial t} h\{n,i,j\} + \frac{\partial}{\partial t} h\{n-1,i,j\}\right)(\Delta t) \qquad (3.29)$$

VIII. Eddy Transport Coefficients

$$N = N_{max}(1. + 10.\ Ri)^{-0.5}$$
$$D_2 = D_{max}(1. + 3.333\ Ri)^{-1.5}$$
$$Ri = \beta\Delta_z^c S\{n,i,j,k\}[\Delta_z^c(u^2 + v^2)^{0.5}\{n,i,j,k\}]^{-2} \qquad (3.30)$$

reference level. In this situation, several significant figures of accuracy are retained that would be otherwise lost when horizontal pressure gradients are computed near the bottom of the flow field. Of course, this advantage diminishes as the deviations of the water level from the reference level increase. (3) An arbitrary term is added to the difference equation for the water level. This diffusion-like term has a stabilizing effect on the calculation. In fact, without the term the calculation would be unconditionally unstable. Determining the magnitude of γ, referred to as an artificial viscosity, will be discussed later in this chapter when stability is considered in detail. (4) With little error, the density in the Richardson number calculation is taken to be unity. Thereby, an operation in an expression that is evaluated thousands of times in the course of a simulation is eliminated. The density gradient in the Richardson number is simply β times the gradient of S.

With consistent flow field data at time level n, the volume and z-directed momentum conservation difference equations of Table 3.1 are identically satisfied and the right-hand sides of the difference equations containing time derivatives can be evaluated. To proceed to time level n+1, finite difference operators for the time derivatives are required. The forward difference operator,

$$\frac{\partial}{\partial t} Q\{n,i,j,k\} \approx (\Delta t)^{-1}(Q\{n+1,i,j,k\} - Q\{n,i,j,k\}) \tag{3.31}$$

which is first-order accurate in Δt, is commonly used. The use of this operator results in difference equations that can be solved explicitly for the dependent variable at time level n+1. Thus,

$$Q\{n+1,i,j,k\} = Q\{n,i,j,k\} + (\Delta t)(\frac{\partial}{\partial t} Q\{n,i,j,k\}) \tag{3.32}$$

where $\frac{\partial}{\partial t} Q\{n,i,j,k\}$ represents the results of evaluating the right-hand side of a conservation or water level equation in Table 3.2. The limitations of the forward difference operator are the small allowable time step required for stable calculations and the fact that solutions generated with it tend to oscillate from one time level to the next. A simple modification reported by Waldrop and others (1974) that improves stability and removes the oscillations is to replace

$$\frac{\partial}{\partial t} Q\{n,i,j,k\} \text{ with } \frac{1}{2}\left(\frac{\partial}{\partial t} Q\{n,i,j,k\} + \frac{\partial}{\partial t} Q\{n-1,i,j,k\}\right)$$

to obtain

$$Q\{n+1,i,j,k\} = Q\{n,i,j,k\} + (0.5)\left(\frac{\partial}{\partial t} Q\{n,i,j,k\} + \frac{\partial}{\partial t} Q\{n-1,i,j,k\}\right)(\Delta t) \qquad (3.33)$$

This modified forward difference operator is formally first-order accurate in Δt. It should be noted that when large time steps are as important as accurate solutions, there may be an advantage to using a first-order accurate finite difference operator for the time derivative. The advantage lies in the fact that the truncation error is directly proportional to Δt for a first-order accurate operator while it is proportional to a higher power of Δt for a higher order accurate operator. Thus the truncation error for the latter may be greater than that for a first order accurate operator at large time steps. Also note that the scheme represented by Eq. (3.33) is simple to implement and requires little additional computer storage or operations over that of the normal forward difference operator.

Boundary Conditions

The finite difference equations developed in the preceding paragraphs apply at general interior grid points, that is, points whose closest neighbors are immersed in fluid, also. To those grid points located adjacent to or in solid, flow or free surfaces, special consideration must be given in order that boundary conditions applying at the surfaces are satisfied. In many cases, this is accomplished by assigning certain values to dependent variables at grid points located outside the flow field. To avoid conflicts that arise at convex corners of the flow field boundary, these assignments are made at the time the finite difference equations are evaluated. No data are stored in or retrieved from the computer storage locations associated with grid points outside the flow field. Some boundary conditions require that the finite difference formulations of the applied equations be modified. The details of applying the boundary conditions will be presented at this point. The reader is referred to Table 2.2 for analytical statements of these auxiliary conditions.

For a solid surface oriented normal to the x-axis as indicated in Fig. 3.4, the no mass transfer condition by diffusion is satisfied by assigning the value of the species concentration at the point (i,j,k) inside the flow field to that of the point (i+1,j,k) outside the flow field at each time level. Thus,

$$S\{n,i+1,j,k\} = S\{n,i,j,k\}. \tag{3.34}$$

For solid surfaces oriented normal to the other coordinate axes analogous assignments apply.

The geometry of Mobile Bay is such that the grid increments in the horizontal directions must be on the order of kilometers in magnitude

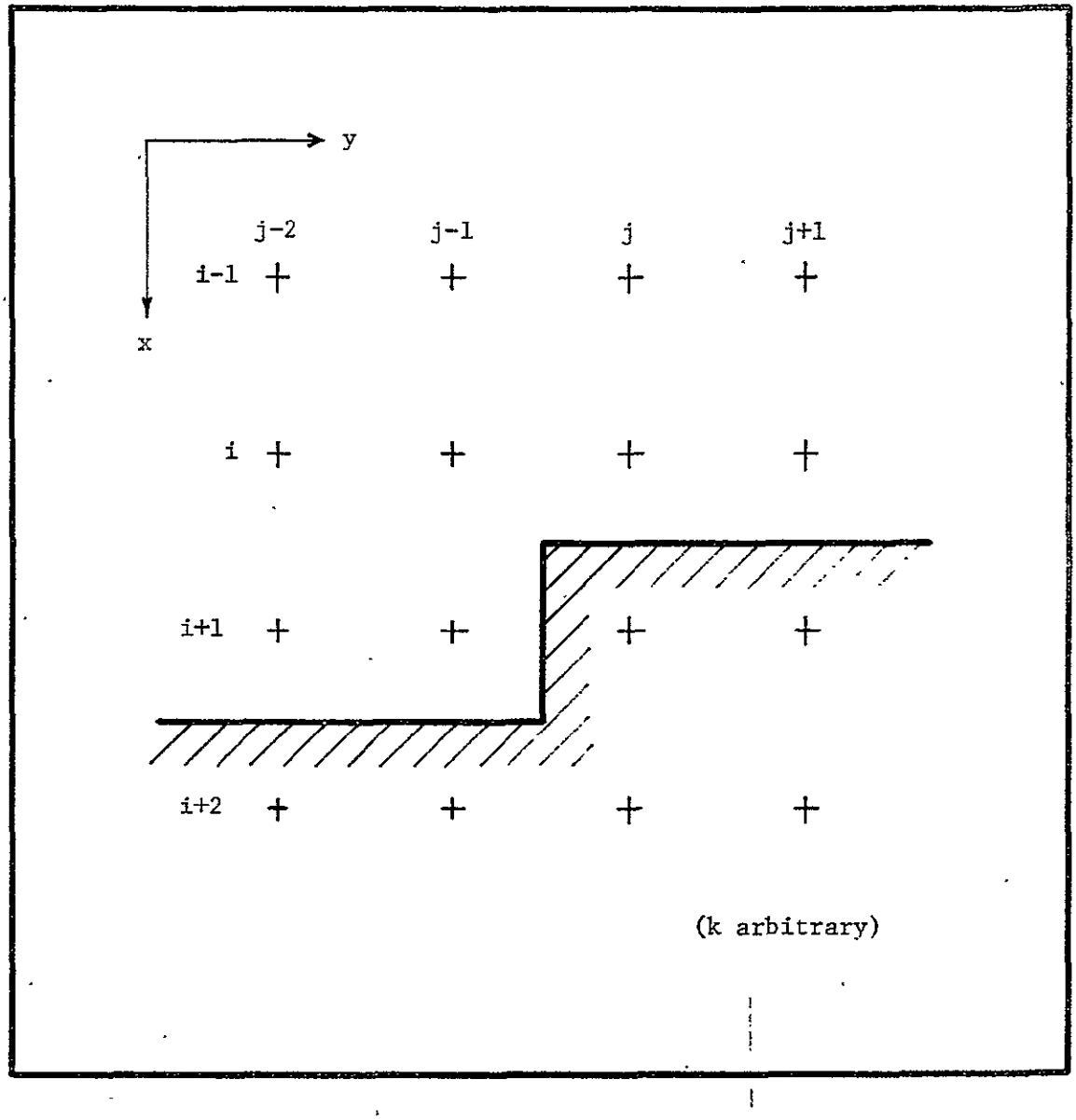


Figure 3.4 - Typical grid point located adjacent model banks

if the number of grid points is to be manageable. Since the model banks are located midway between the bounding grid planes, the normal distance from a bank to the nearest grid point is still large. In fact, the distance is so large that the solid surface has negligible viscous effect on the velocity at nearby grid points. Thus, the appropriate boundary conditions on the velocity field at the model banks are those for inviscid flow. The conditions of zero normal velocity and zero normal gradient in tangential velocity are implemented with the reflection technique of Richtmyer and Morton (1967). Consider a grid point adjacent to a bank, such as point (i,j,k) in Figure 3.4 and the point (i+1,j,k) on the opposite side of the bank in the solid. The zero normal velocity condition at the surface is achieved by making the assignment

$$u\{n,i+1,j,k\} = -u\{n,i,j,k\}. \tag{3.35}$$

When the windward difference operator of Eq. (3.11) is applied at point (i,j,k) with respect to the x-direction, Eq. (3.35) will result in zero normal flow at the bank, the desired result. The zero normal gradient in tangential velocity is implemented with the assignment

$$v\{n,i+1,j,k\} = v\{n,i,j,k\}. \tag{3.36}$$

In contrast to the large grid increments required for the horizontal directions, the vertical grid increment is small, on the order of a meter. Thus, the bay bottom is relatively close to the adjacent grid points, and the viscous effects of the horizontal solid surface on the velocity at these points should not be neglected. Therefore, the viscous flow condition of zero velocity at a solid surface is applied to the bay bottom. On the other hand, the resolution of the grid in the vertical direction is not sufficient to accurately resolve the

steep gradients in velocity associated with turbulent flow near the bottom. To overcome this obstacle, the following artifice is employed. Consider the typical bottom grid point (i,j,k_{bot}) indicated schematically in Fig. 3.5; note that the value of k_{bot} depends on the location of the vertical grid column (i,j). By assuming that the logarithmic velocity profile for a fully developed turbulent boundary layer applies from the bottom up to elevation $z\{k_{bot} + 1\}$, a horizontal velocity at (i,j,k_{bot}) can be obtained by several interpolation procedures. The procedure used here matches the slope of the model velocity profile with that of the logarithmic velocity profile at $z\{k_{bot}\} + \Delta z/2$. The interpolation formula that accomplishes this match is

$$u\{n,i,j,k_{bot}\} = u\{n,i,j,k_{bot} + 1\}\cdot[1-\Delta z(z_1)^{-1}(\ln(z_2/k_s) + 8.5k_o)^{-1}] \qquad (3.37)$$

where

z_1 = $z\{k_{bot}\} - z_{bot}\{i,j\} + \Delta z/2$

z_2 = $z\{k_{bot} + 1\} - z_{bot}\{i,j\}$

$z_{bot}\{i,j\}$ = bottom elevation at grid column (i,j)

k_s = length characteristic of bottom roughness elements

k_o = 0.4, von Karman's constant.

An analogous formula may be written for $v\{n,i,j,k_{bot}\}$. This interpolation formula is flexible in that it allows the model bottom to be positioned anywhere between grid points and that it accounts bottom roughness. Its greatest shortcoming is that it presumes that the flow at $z\{k_{bot}\}$ is in the same direction as the flow at $z\{k_{bot} + 1\}$ which is not necessarily true. The boundary condition on the vertical velocity component at the bottom is that the component is zero. This condition is imposed on the volume conservation equation by evaluating its finite

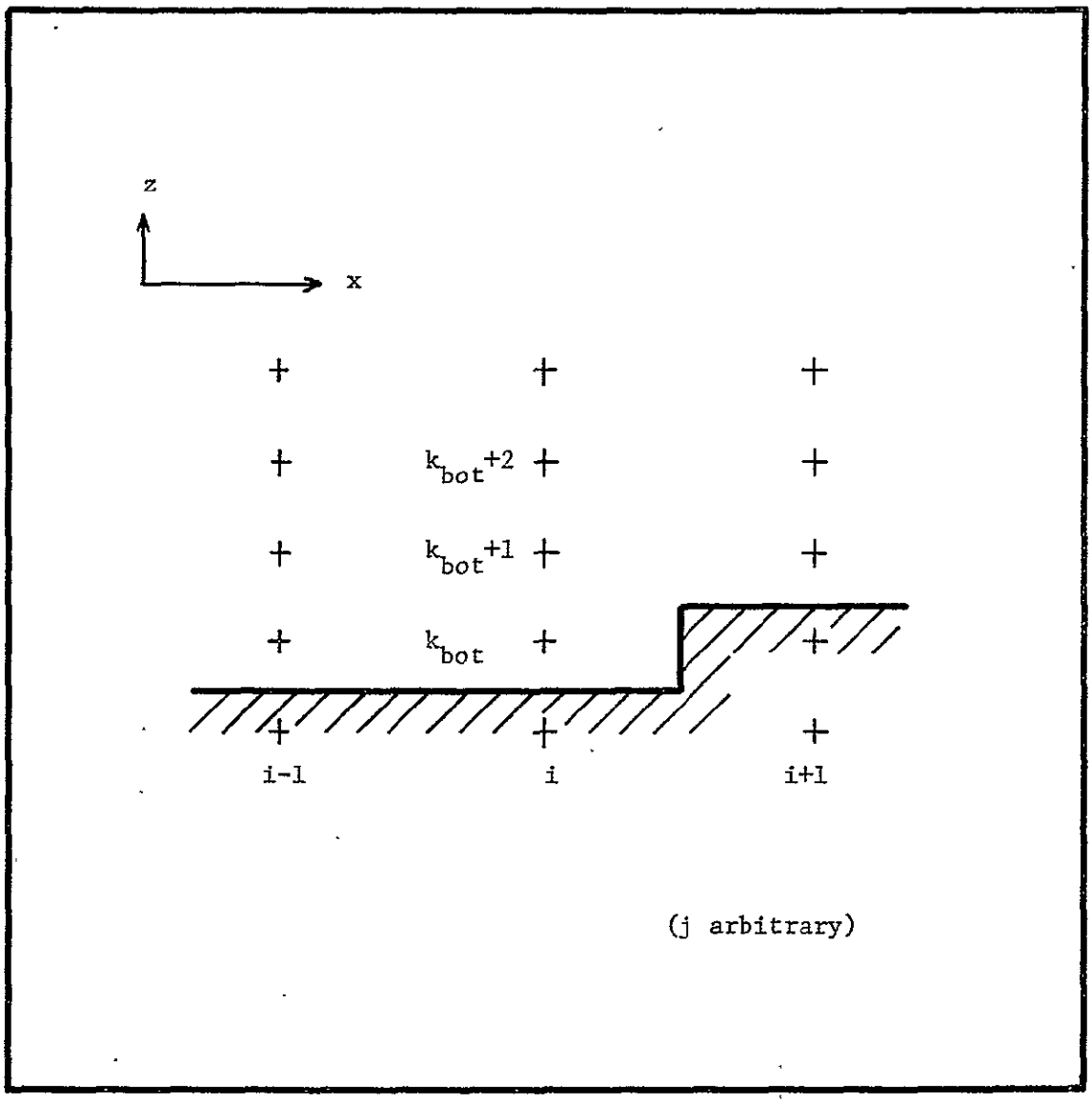


Figure 3.5 - Typical grid point adjacent to model bottom

difference form over a vertical height $z\{k_{bot}\} - z_{bot}\{i,j\}$ instead of Δz and setting the horizontal and vertical velocity components to zero at $z_{bot}\{i,j\}$ in equation (3.20).

The boundary condition on the water level at the banks of the bay is the surface wave reflection condition. It is implemented by setting the water level at points outside the flow field equal to the level at the corresponding point inside the flow field. For the typical horizontal grid location (i,j) indicated in Fig. 3.4, this assignment is accomplished with

$$h\{n,i+1,j\} = h\{n,i,j\} \tag{3.38}$$

A top view of the grid configuration at a typical river mouth is given in Fig. 3.6. As the figure indicates, the width of the river mouth is generally smaller than the horizontal grid increment normal to the direction of the discharge. The forced flow condition at the grid column located in the flow surface in the river mouth was implemented with

$$u\{n,i,j,k\} = u_{max}[1.-(1.-(z_1/z_2)^7)] \tag{3.39}$$

$$u_{max} = C(u_{avg} + u_{var1}\cos(2\pi(t\{n\}-\phi_1)/T) + u_{var2}\cos(4\pi(t\{n\}-\phi_2)/T) \tag{3.40}$$

where

$z_1 = z\{k\} - z_{bot}\{i,j\}$

$z_2 = h\{n,i,j\} - z_{bot}\{i,j\}$

u_{avg} = tidal average river discharge velocity

u_{var1}, u_{var2} = amplitudes of first and second variations in discharge velocity

ϕ_1, ϕ_2 = phase lags for first and second variations

T = period of tide cycle

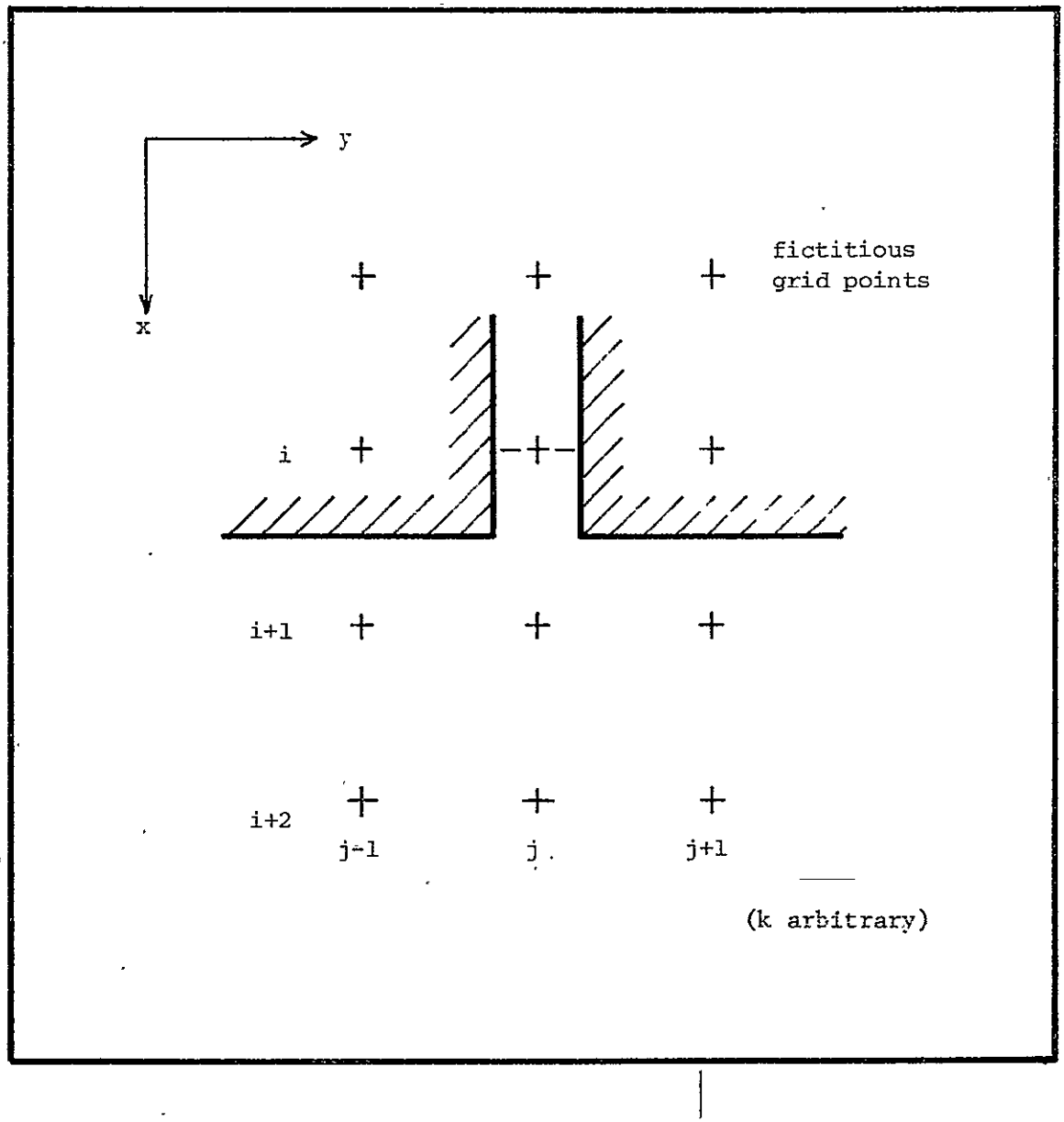


Figure 3.6 - Typical grid configuration at a forced flow surface

u_{max} = velocity at free surface

c = conversion factor relating average and maximum velocities for the profile used.

Eq. (3.39) provides a flat velocity profile that rapidly drops to zero at the bottom of the river. Eq. (3.40) accounts for the time variation of the river flows and was obtained by fitting the trigonometric series to average river velocities-time data derived from field measurements. The velocity specifications presented above are for the actual cross sections of the river mouths. Before these velocities are used in the volume conservation equation or in calculating the transport velocity for the convective acceleration term in the directed momentum equation, they are multiplied by the ratio of the width of the river mouth to $\Delta Y(Y'\{j\})^{-1}$, the horizontal grid increment normal to the direction of the river flow, in order to conserve fluid volume. The salinity distribution in the river mouths was given by

$$S\{n,i,j,k\} = 0 . \tag{3.41}$$

As commented in the previous chapter, this boundary condition would be inappropriate if saltwater intrusion or reversed flow occurred at the river mouths to a significant extent. The concentration of the species at the flow surface during outflow would have to be obtained by extrapolation from grid points in the interior of the flow field or by a simplified formulation of the species conservation equation. The boundary condition on the water level at the river mouths was implemented with

$$h\{n,i,j\} = h\{n,i+1,j\} \tag{3.42}$$

which implied that the water levels in the river mouths are essentially

controlled by the water levels in the bay. This approximation is valid for the range of flows considered in the present work.

The boundary conditions on the velocities in flow surfaces at tidal inlets are based on the assumption that the flow field does not change appreciably by the time it reaches a fictitious plane of grid points located one grid increment outside the flow field. Thus, in terms of the typical tidal inlet grid point (i,j,k) indicated in Fig. 3.7, the conditions on the horizontal velocity components are

$$u\{n,i+1,j,k\} = u\{n,i,j,k\}$$

$$v\{n,i+1,j,k\} = v\{n,i,j,k\}. \tag{3.42}$$

The vertical velocity component in the flow surface is calculated with the volume conservation equation and the assignments of Eqs. (3.42). The boundary conditions on the species concentration are

$$\begin{aligned} &S\{n,i,j,k\} = 1 \quad \text{for } z\{k\} \leq z\{k_{cut}\} \\ &\left.\begin{aligned} S\{n,i+1,j,k\} &= S\{n,i,j,k\} \text{ during ebb} \\ S\{n,i+1,j,k\} &= 1 \qquad\quad \text{during flood} \end{aligned}\right\} \text{ for } z\{k\} > z\{k_{cut}\} \end{aligned} \tag{3.43}$$

This first boundary condition is based on field data indicating that gulf water persists below a level $z\{k_{cut}\}$ throughout the tide cycle. The second condition allows bay water of low salinity to exit through the tidal inlet during ebb tide and gulf water to enter the bay during flood tide. The boundary condition for flood tide above $z\{k_{cut}\}$ presumes that the discharged bay water from the previous ebb tide is swept away by currents in the gulf or mixed with a large volume of gulf water so its salinity becomes essentially that of gulf water. The water level in the tidal inlet is a forced function of time. The function employed in this work is

$$h\{n,i+1,j\} = h_{avg} + h_{var} \cos\left(2\pi(t\{n\}-\phi_3)/T\right) \tag{3.44}$$

where

h_{avg} = tide average water level

h_{var} = amplitude of the tide variation

ϕ_3 = phase lag in the tidal variation.

The forced water level boundary condition was applied at the fictitious grid plane at x(i+1) in Fig. 3.7 because it was expected that flow geometry and Coriolis effects would generate gradients in the water level across the tidal inlets. Presently there are no published field data to substantiate this expectation or to generate water level forcing functions that are dependent on location across the inlets. However, one would expect there to be smaller lateral gradients in the water level than longitudinal ones at positions removed from the inlets. The original intent was to treat both the Main Pass and Pass Aux Herons in the same way with regard to boundary conditions. As the development proceeded, it became necessary to treat the Pass Aux Herons with the river velocity boundary conditions for velocity with u_{max} being determined as a percentage of the average velocity through the Main Pass. The treatment of the Pass Aux Herons boundary conditions will be discussed in more detail in Chapter 4.

If application of boundary conditions to solid and flow surfaces is complex and fraught with approximations, application of boundary conditions to free surfaces is doubly so because the surface is free to move and not necessarily bound between any two grid points for all time. In the present work, the approach to application of boundary conditions to the free surface is very simplistic but the particular implementations used are sufficiently accurate to obtain reasonable solutions. The implementations used may be divided into two categories:

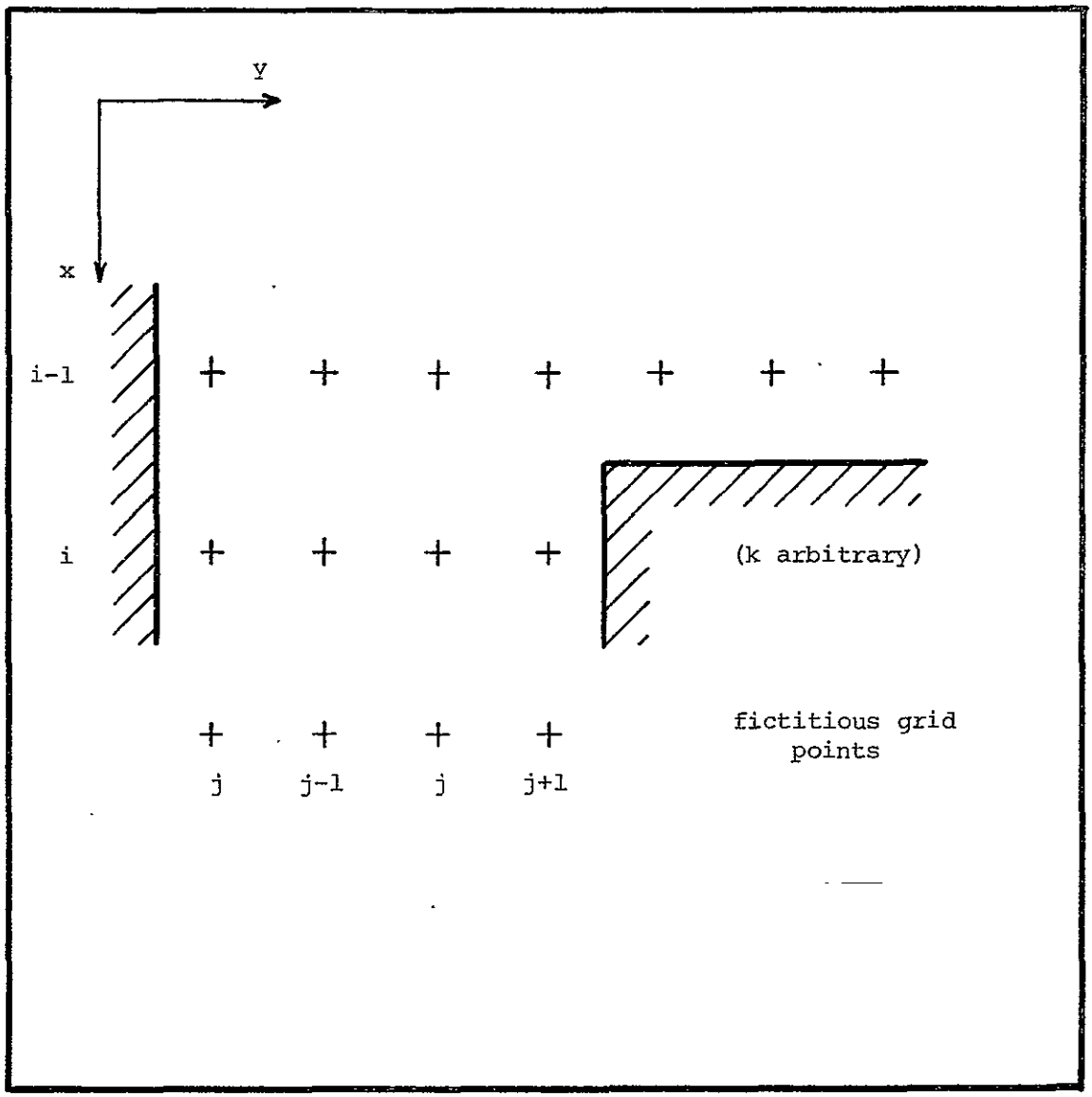


Figure 3.7 - Typical grid configuration at a forced water level boundary

1) those whose application of necessity depend on the exact location of the free surface, and 2) those whose application do not require such exactness. An approximation inherent in all the implementations used here is that the gradient of a quantity normal to the free surface is equal to the gradient in the vertical direction.

The boundary conditions falling in the first category in the present work is the pressure condition, and the wind shear stress condition. The pressure condition was implemented through the first equation of Eqn. (3.27) where $p'_{atm} = 0$. The wind shear stress condition influences the horizontal velocity components at the elevation $z\{k_{max}\}$ through the vertical gradients of the horizontally directed shear stress. This influence is incorporated in the finite difference formulations of the horizontal momentum equations by adding extra terms of the form

$$\tau^w_{xz}/(h\{n,i,j\}-z\{k_{max}\} + \Delta z/2) \text{ and } \tau^w_{yz}/(h\{n,i,j\}-z\{k_{max}\} + \Delta z/2)$$

to the x- and y-directed equations, respectively when they are evaluated at $z\{k_{max}\}$. These implementations, which account for the effect of the free surface location on the magnitudes of the gradients in a natural way, presume that the vertical diffusion of momentum at $z\{k_{max}\}-\Delta z/2$ without wind is negligible in comparison to that induced by the wind.

The boundary conditions falling in the second category are the no mass transfer condition and zero shear stress condition in the absence of wind. In terms of the typical grid point (i,j,k_{max}) near the free surface in Fig. 3.8, these conditions were implemented with

$$S\{n,i,j,k_{max} + 1\} = S\{n,i,j,k_{max}\} \tag{3.45}$$

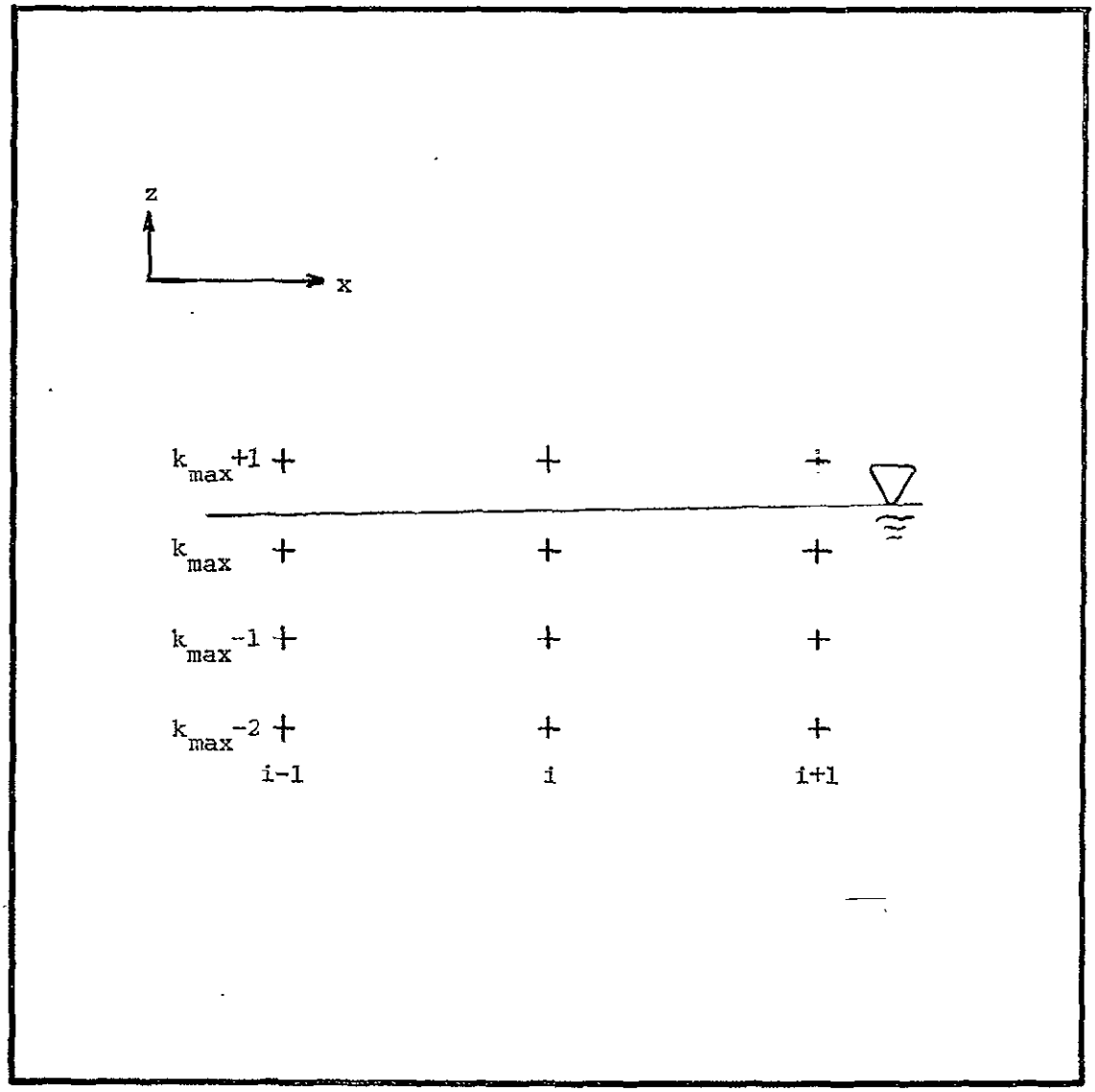


Figure 3.8 - Typical grid configuration at free surface

$$u\{n,i,j,k_{max}+1\} = u\{n,i,j,k_{max}\}$$

$$v\{n,i,j,k_{max}+1\} = v\{n,i,j,k_{max}\}$$

$$w\{n,i,j,k_{max}+1\} = w\{n,i,j,k_{max}\} \qquad (3.46)$$

where $(k_{max}+1)$ denotes a fictitious grid plane located a distance Δz above $z\{k_{max}\}$. The primary justification for using the course approximations of Eqns. (3.45) and (3.46) is that the implementation of more accurate specifications generally results in a slow computer code. Now that it has been demonstrated that the present model is operative, more accurate specification of free surface boundary conditions is a logical next step in improving the model.

As the model is presently programmed, the water level may not move below the elevation $z\{k_{max}-1\}$. This restriction is imposed in order that in shallow water areas, at least one general interior grid point will remain submerged in water. While there are no programmed restrictions on how high the water level may rise, the solution would be suspect if the level moved above $z\{k_{max}\} + (2)(\Delta z)$.

Calculation Sequence

Assuming that values of u, v, w, S, p and h at time level n and of $\frac{\partial u}{\partial t}$, $\frac{\partial v}{\partial t}$, $\frac{\partial S}{\partial t}$ and $\frac{\partial h}{\partial t}$ at time levels n and n-1 are available for the relevant grid points in the flow field, the sequence for computing the values of these variables at time level n+1 is as follows:

1. On the first sweep of the flow field, values of S, u and v are calculated with Eqns. (3.21) through (3.26) using the pertinent boundary conditions at the solid and free surfaces. Although it is immaterial to the results, the sweep of the flow field proceeds first in

the positive z-direction, then in the positive y-direction and finally in the positive x-direction.

2. On the second sweep of the flow field, values of w are calculated with a spatial integration of Eqn. (3.20) using the new horizontal velocity components. The sweep of the flow field on this step must proceed in the positive z-direction first in order that the zero vertical velocity boundary condition may be applied at the bay bottom. The sweep then proceeds arbitrarily first in the positive y-direction and then the positive x-direction.

3. In addition to the spatial integration of step 2, on the second sweep values of h are computed with Eqns. (3.28) and (3.29).

4. On the third sweep of the flow field, values of p' are calculated using the new salinities and water levels. In contrast to step 2, the sweep must proceed in the negative z-direction first so that the zero atmospheric gage pressure can be satisfied at the free surface.

5. Values of u, v, S and h associated with flow surfaces that are forced functions of time are calculated.

6. Instantaneous and net volumetric flows through flow surfaces and accumulations within the bay are computed with the appropriate surface, volume and time integrations.

This sequence is repeated as many times as are required to complete the simulation. The combination of the above calculation sequence and the finite difference equations of Table 3.2 yields an explicit method that requires the retention of arrays of u, v, S and h at two consecutive time levels and arrays of w, p, $\frac{\partial u}{\partial t}$, $\frac{\partial v}{\partial t}$, $\frac{\partial S}{\partial t}$ and $\frac{\partial h}{\partial t}$ at one time level. The retention of ρ is not required because the density and its reciprocal are calculated from Eqn. (3.19) as needed.

Numerical Viscosity, Stability, Convergence and Accuracy

According to Roache (1976), implicit numerical or artificial viscosity is a particular kind of truncation error produced by some finite difference approximations to the convective terms in conservation equations. Furthermore, explicit numerical viscosity can be generated by adding to the finite difference equation the product of a non-physical coefficient times a difference approximation to the second-order spatial derivative of the conserved quantity. Explicit numerical viscosity is frequently employed to achieve stability. Both implicit and explicit numerical viscosity are present in the system of finite difference equations used in the present work and are considered below.

An analysis for the implicit numerical viscosity in a finite difference equation is possible only if the equation is linear with constant coefficients. In addition the analysis becomes prohibitively complex if coupled systems of difference equations are considered. Within these limitations the most physically meaningful equation left subject to analysis is the convection-diffusion equation

$$\frac{\partial Q}{\partial t} = -V \frac{\partial Q}{\partial x} + \alpha \frac{\partial^2 Q}{\partial x^2} \tag{3.47}$$

or a multidimensional variation therefore, where Q is a conserved quantity, V is a constant convective velocity, and α is viscosity or diffusion coefficient. Assuming that V is positive and applying the modified forward difference operator for the time derivative, the windward difference operator for the convective term and the centered difference operator for the diffusion term as described earlier in this chapter, one obtains

$$\frac{\partial Q}{\partial t}\{n,i\} = -V\,\frac{Q\{n,i\}-Q\{n,i-1\}}{\Delta x} + \alpha\,\frac{Q\{n,i+1\}-2Q\{n,i\}+Q\{n,i-1\}}{(\Delta x)^2}$$

$$Q\{n,i+1\} = Q\{n,i\}+(0.5)(\Delta t)\left[\frac{\partial Q}{\partial t}\{n,i\} + \frac{\partial Q}{\partial t}\{n-1,i\}\right]. \qquad (3.48)$$

By expanding the discretized variables of Eqn. (3.48) in terms of a double Taylor series expansion in Δx and Δt about the grid point (n,i) and collecting like terms, one obtains

$$\frac{\partial Q}{\partial t}\{n,i\} = -V\,\frac{\partial Q}{\partial x}\{n,i\} + (\alpha+\alpha_n)\,\frac{\partial^2 Q}{\partial x^2}\{n,i\} + \text{H.O.D.} \qquad (3.49)$$

where $\alpha_n = \frac{V(\Delta x)}{2}\left[1-2\,\frac{V(\Delta t)}{(\Delta x)}\right]$

H.O.D. = higher order derivatives.

If the indices (i,n) are dropped and the higher order derivatives neglected on the assumption they are small compared to the retained terms, one recovers the original convection-diffusion equation plus an extra term that resembles a viscosity term if the conserved quantity is momentum or a diffusion term if the quantity is a chemical species. This numerical viscosity has nothing to do with the physical system and is an error that originates in the discretization process. Furthermore, there is a good probability that the results of the above analysis can not be extended to the system of equations used in the present problem because interactions that provide other sources for truncation error exist between the various equations. In light of the circumstance, any estimate of the implicit numerical viscosity in the Mobile Bay model based on Eqn. (3.49) would be of dubious value. Possibly, with more work a rigorous analysis of the model for implicit numerical viscosity can be obtained in the future.

The explicit numerical viscosity present in the finite difference equations for the model appears in the water level equation, Eqn. (3.28),

in the form $\gamma\ (\Delta^c_{xx} h\{n,i,j\} + \Delta^c_{yy} h\{n,i,j\})$. This term, which ostensibly cancels negative implicit numerical viscosity, is required in order to obtain stable solutions. The same effect can be obtained by averaging water level values at (i,j) and the four surrounding grid columns in some fashion, but the explicit numerical viscosity form is simpler to manipulate. Since γ is nonphysical, selection of the correct value to use in calculations represents a problem. Nichols and Hirt (1973) suggest that

$$\gamma \geq \frac{\Delta t}{2} \max\ (u^2,\ v^2) \tag{3.50}$$

is necessary for stability. Preliminary experiments by this author indicated that a γ value based on Eqn. (3.50) was inadequate. Subsequent experimentation revealed that

$$\gamma > \frac{\Delta t}{2}[Hg + \max(u^2,v^2)] \tag{3.51}$$

where $\sqrt{Hg}$ is the wave celerity, is required for stability in the present model. For the Δt = 60.s, H = 5. m and max (v^2,v^2) = 1.3 m^2s^{-2} Eqn. (3.51) gives a minimum γ of 1500. m^2s^{-1}; a value of 2400 m^2s^{-1} was actually used in the computations. The reason for the difference in the restriction on γ may stem from the fact that Nichols and Hirt used a marker-and-cell technique in which vertical velocities are calculated from the z-directed momentum equation while in the present model vertical velocities are calculated indirectly from the horizontal momentum equations through the volume conservation equation.

A numerical method for solving partial differential equations is said to be stable if errors, either from round-off generated when working with numbers of finite length or from inaccuracies in the initial conditions, remain bounded as the calculation proceeds. Most

explicit methods for solving partial differential equations are conditionally stable in the sense that there are restrictions on the magnitudes of certain parameters in the finite difference model. The condition on γ in Eqn. (3.51) is one such restriction. To date, most analyses of finite difference methods for stability conditions have been limited to problems where the equations are linear or to linearized versions of nonlinear equations. The reader is referred to Roache (1976) for a discussion of various methods of analyzing for stability and to Schumann (1975) for an analysis of linearized conservation equations. Because of the complexity of the system of coupled nonlinear equations in the present problem such analyses have not been fruitful. Instead, stability conditions have been determined by extending the results of analysis for limiting cases and testing the validity of these proposed stability conditions by numerical experiment. In addition to the condition given in Eqn. (3.51), two other restrictions appear to apply to the numerical method used here. One is the CFL criterion

$$\Delta t < \frac{\min(\Delta x, \Delta y)}{\sqrt{Hg}} \tag{3.52}$$

which originates in the theory of characteristics for small amplitude surface waves on inviscid flows. See Stoker (1957) for a treatise on the mathematics of this theory. For the grid system used in the Mobile Bay model

$$\Delta t < \frac{1.7 \times 10^3 \text{m}}{\sqrt{(5.\text{m})(9.8\text{ms}^{-2})}} = 240.\text{s}$$

Another restriction applicable to the present problem is the stability condition related to diffusion of momentum given by

$$\Delta t < \frac{(\Delta z)^2}{2N} \tag{3.53}$$

where N is the maximum eddy viscosity for turbulent diffusion of momentum in the vertical direction. An analogous condition related the diffusion of salt is

$$\Delta t < \frac{(\Delta z)^2}{2D_2} \tag{3.54}$$

where D_2 is the eddy diffusivity.

This stability condition may be derived from a stability analysis of the finite difference formulation of the simple diffusion equation. The values of Δz and maximum N used in the present work are 0.625 m and 0.0030 $m^2 s^{-1}$ which give

$$\Delta t \leq \frac{(0.625m)^2}{2(0.0030 m^2 s^{-1})} = 65.\ s. \tag{3.55}$$

The stability condition related to diffusion gives the most restrictive time step for the Mobile Bay model with the present grid configuration. The time step actually employed in the calculations was 60. s.

A concept closely related to stability is convergence. A numerical scheme for solving partial differential equations is said to be convergent if the solution of the finite difference equations approaches the solution of the governing differential equations as the grid system is successively refined. The convergence of the numerical method used here with some slight variations in the windward difference operator and the technique for introducing explicit numerical viscosity into the water level equation has been demonstrated by Waldrop (Farmer, 1976). In the cited work, the same problem was run twice, once on a coarse grid and again on a refined grid, and the solutions compared. The two solutions agreed sufficiently well to imply convergence.

Ideally, the accuracy of a finite difference method is assessed by comparing the numerical solution with an analytical solution to the partial differential equations under consideration. The difficulty is that no analytical solutions exist to the equations under consideration. Therefore, it must be assumed that the applied equations developed in the preceding chapter adequately model the phenomena of interest and that experimental data can be used in lieu of analytical results. However, one must realize that the errors inherent in field observations may be as large or larger than those in the finite difference solutions. The next chapter is primarily devoted to comparing the model results with field observations to ascertain the accuracy of the model.

C. SUMMARY

This chapter is devoted to discussing the numerical method utilized in solving the partial differential equations and auxilary conditions that govern the phenomena of interest. After the applied equations are transformed to a nondimensional form, finite difference approximations are introduced in order that the equations can be solved on a digital computer. The particular difference utilized lead to an explicit calculation procedure that exhibits superior properties with respect to the smoothness of the solutions (i.e. oscillations are eliminated) and that takes little additional computer core over the minimum required for obtaining a solution. The method is subject to the CFL criterion and a stability limitation associated with diffusion; the latter is the most restrictive in the present problem. Next, the implementation of difference approximations for boundary conditions is considered in detail. The correct specification of boundary conditions requires much physical insight

and is critical to the success of the model. Presently simple, but adequate difference approximations to the boundary conditions are used, particularly at the free surface. The initial conditions necessary to implement the numerical method are not discussed in the present chapter but are elaborated on in the next chapter where their presentation is more natural. Finally, numerical viscosity, stability, convergence and accuracy of the numerical method are considered. A significant observation made from numerical experimentation is that an explicit numerical viscosity proportional to the square of the wave celerity, as opposed to the square of the maximum horizontal velocity component, is required for stability.

C'2

CHAPTER 4

RESULTS

Upon establishing that the hydrodynamics approach being exactly periodic within two tide cycles, this chapter presents the computer model results for three test cases. One of these cases is compared with prototype data for approximately the same boundary conditions. Then the controlling parameters for Mobile Bay as an estuarine system are identified and discussed. Finally, a purview of the results in the context of the objectives of this work is given.

A. APPROACH TO EXACT PERIODICITY

In reality, no two tide cycles for an estuary are identical primarily because the conditions at its boundaries are never constant or exactly periodic long enough for the system to achieve exact periodicity. These boundary conditions include river flows and their sediment loadings, tidal height, flow rate and salinity variations, and wind patterns and durations. Ideally, one would have sufficient prototype data to (1) input the histories of the boundary conditions to the computer model for as many tide cycles as are of interest, (2) accurately specify the initial conditions within the system and (3) verify the simulation results at later times. Practically, one seldom has the resources needed to collect for more than one or two diurnal tide cycles the detailed data required to specify the boundary condition histories and never to completely fix the conditions throughout the system at one instant in time. Still, if the boundary conditions are

nearly steady or periodic for several tide cycles, there is reason to believe that the flow patterns for the system repeat themselves after one or two cycles at the imposed conditions, assuming that the flow field within the estuary is not atypical initially. The hydrodynamics cannot be exactly periodic, however, because they are coupled with the salinity distribution. The latter does not respond as rapidly as the hydrodynamics to changes in the boundary conditions, particularly the river flows.

Support for the above conjectures on the response times of the flow and salinity patterns is derived from the computer model in the following way. Variables named UBAR, VBAR and SBAR are calculated for each time step by evaluating the change in the square of u, v and S, respectively, at each mesh point over the time step, summing these changes over all mesh points internal to the flow field, and then dividing by the product of the total number of points used and the size of the time step. If plots of UBAR, VBAR and SBAR versus time, referred to here as signatures, are identical for consecutive tide cycles, conditions within the system should vary with exact periodicity. The verb "should" is used because there is a possibility that u, v and/or S are undergoing redistribution from one tide cycle to the next in such a way that the corresponding signatures remain exactly periodic, but this situation is considered unlikely.

Signatures for U and S are given in Fig. 4.1 for ten tide cycles. The three combinations of boundary conditions examined and their durations are listed in Table 4.1. Note that sediment transport is not considered here. Also note that Case 4 is a repetition of Case 2; it was conducted to verify that the model results are reproducible when

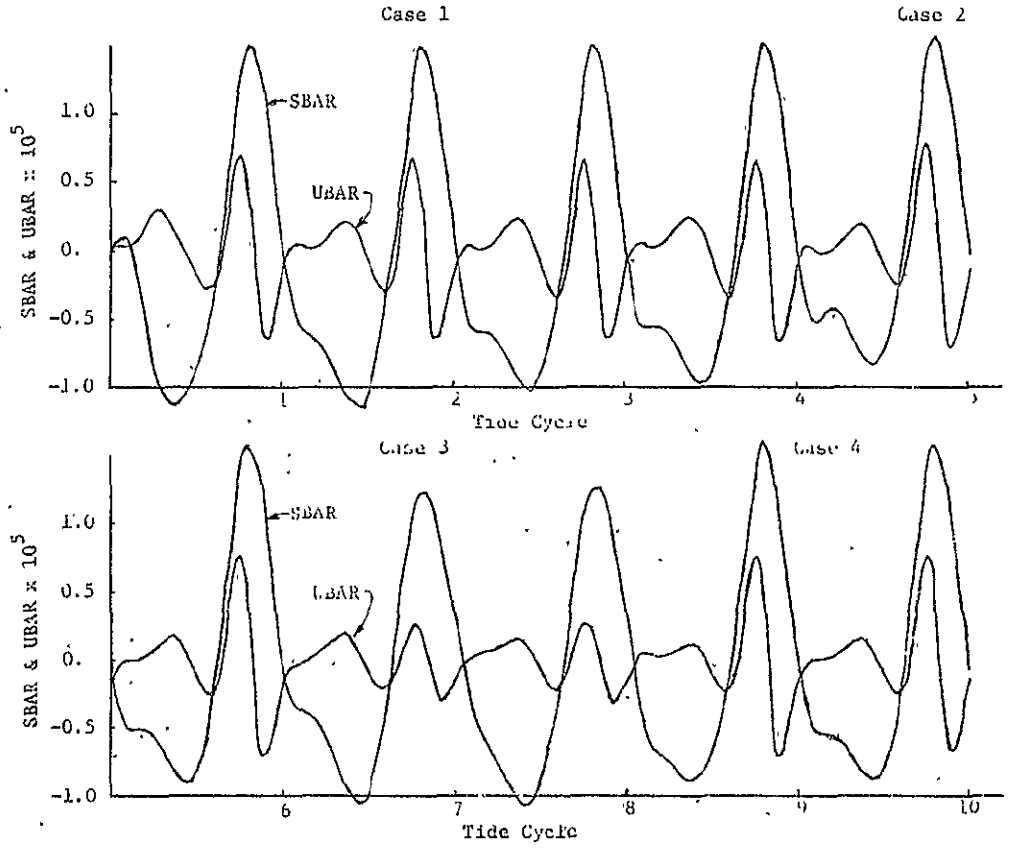


Figure 4.1 - Approach to Steady-State

TABLE 4.1

Test Cases

Case	Tide Cycles	Average River Flow $m^3 s^{-1}$	Wind Applied
1	1- 4	5780	No
2	5- 6	2980	No
3	7- 8	2980	Yes
4	9-10	2980	No

starting from slightly different initial conditions. With the exception of Case 1, the initial conditions for each case are the flow field properties from the preceding case. For Case 1 a grossly approximate salinity distribution that placed mostly fresh water at the north end of the bay and saltwater at the south end and a zero velocity field are the initial conditions. The boundary conditions were changed in a step-like manner between cases.

Fig. 4.1 reveals that for Case 1 the u signature almost repeated itself after the second cycle and did so exactly after the third. The S signature in Case 1 is qualitatively the same for cycles 3 and 4, but a slight increase in the minimum occurred in the latter. This prolonged change in the S signature is to be expected since the initial saltwater content of the bay was high for the average river discharge used in Case 1. Since the flow patterns repeated themselves beginning with the third cycle when starting from rough guesses at the initial conditions, they should be identical in the second and third cycles when more realistic initial conditions are supplied. Thus, the second tide cycle for a given set of boundary conditions and accurate initial conditions are accepted as being exactly periodic, at least with respect to the flow patterns. The salinity distributions may require several more cycles to reach this condition, but a two-cycle simulation provides an economic compromise between computer time and information about their final state.

B. COMPARISON OF COMPUTER MODEL AND PROTOTYPE DATA

Computer model results for Case 2 and prototype data taken from the 15-16 May 1972 field survey are compared for the stations and types of data indicated in Fig. 4.2. The types of data include velocity

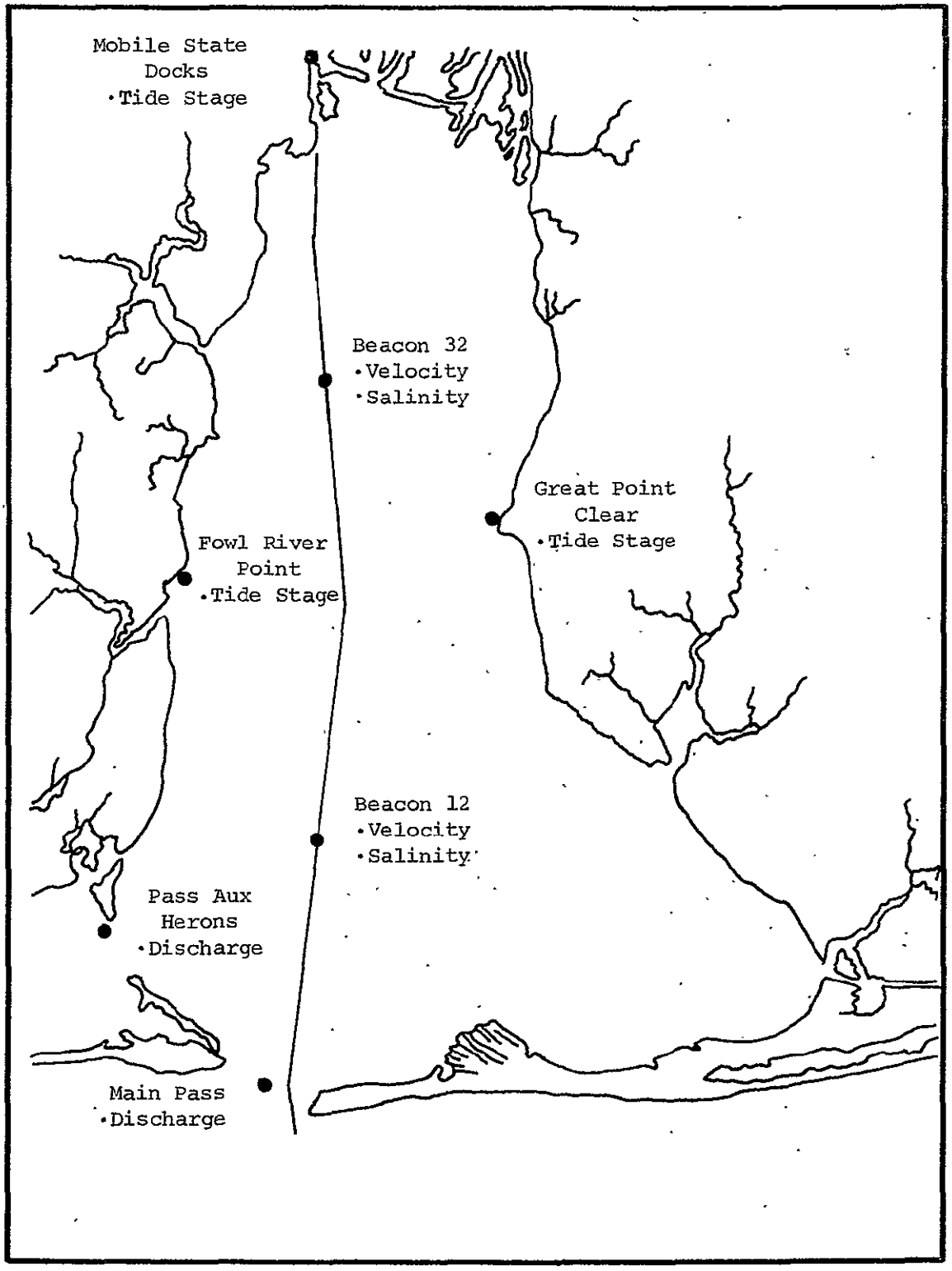


Figure 4.2 - Verification Points for Computer Model

components, salinities, tide curves and discharge rates. The comparisons are made primarily by plotting for each station the indicated data from both sources versus time. Unless it is noted otherwise, the prototype data are adapted from Lawing and others (1975). Of the three model cases, Case 2 most closely approximates the prototype data with respect to imposed boundary conditions. However, there are significant differences between the two. One is that the model combined river average discharge is 2980. $m^3 s^{-1}$ while the corresponding prototype discharge is 1798. $m^3 s^{-1}$. This discrepancy is a consequence of the fact that the river velocities are specified functions of time while the flow cross-sections and discharges are predicted by the model. Another is that an unsteady and unmeasured wind from the southwest over the southern portion of the bay influenced the prototype data, whereas there is no wind in Case 2. Finally, there are slight, but possibly significant differences between the prototype and model tidal inlet tide curves. These differences in the tide curves are described below.

North-south components of the free surface velocities and the salinities at Beacons 12 and 32 are compared in Figs. 4.3 through 4.6. The sign of the velocity component is positive if it points in the same direction as the positive x-axis; the sign is negative otherwise. The model and prototype velocity curves for Beacon 12 are much the same although a stronger flood flow in the model resulted in a maximum flood velocity of -0.7 m s^{-1} compared to -0.5 m $\cdot s^{-1}$ for the prototype. There are 2.5 to 3.0 hr differences in the times of the maximum ebb and flood velocities for the two curves. The free surface salinity comparison at Beacon 12 is not as positive as that for the velocity component. A strong variation in salinity from 30. PPT at high tide to 8. PPT just

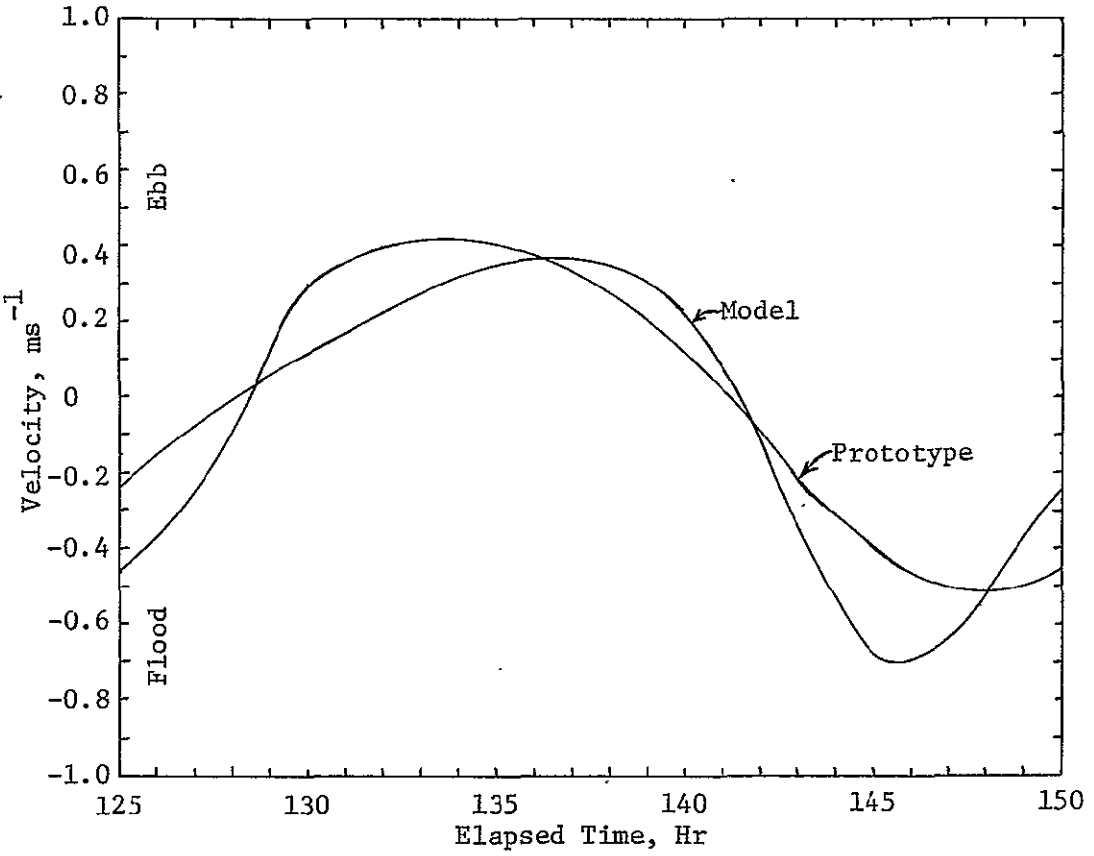


Figure 4.3 - Verification of Velocities at Beacon 12

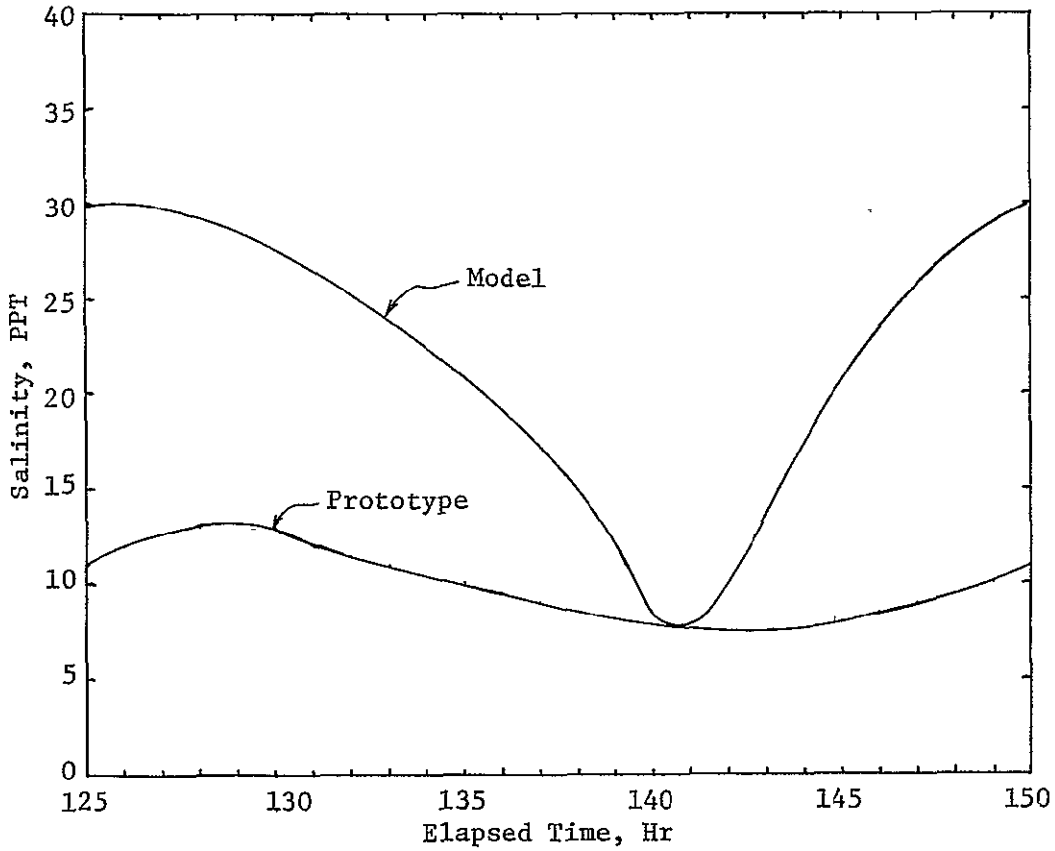


Figure 4.4 - Verification of Salinities at Beacon 12

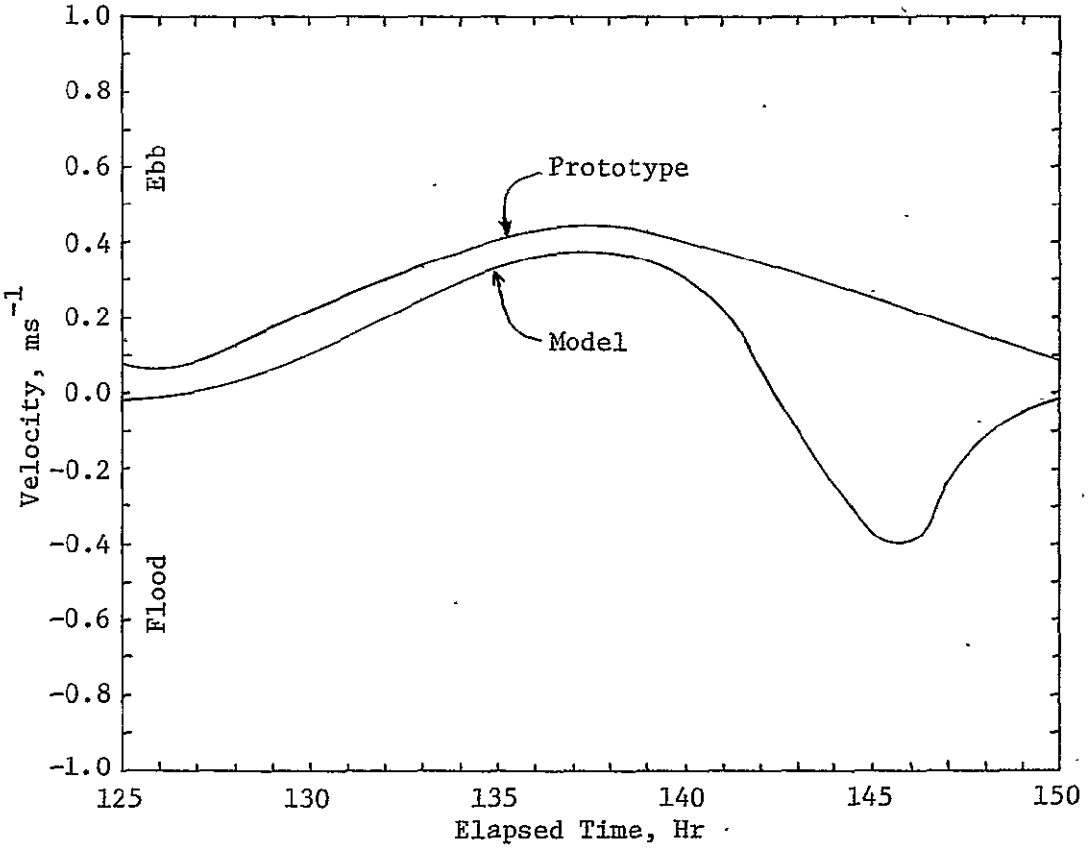


Figure 4.5 - Verification of Velocities at Beacon 32

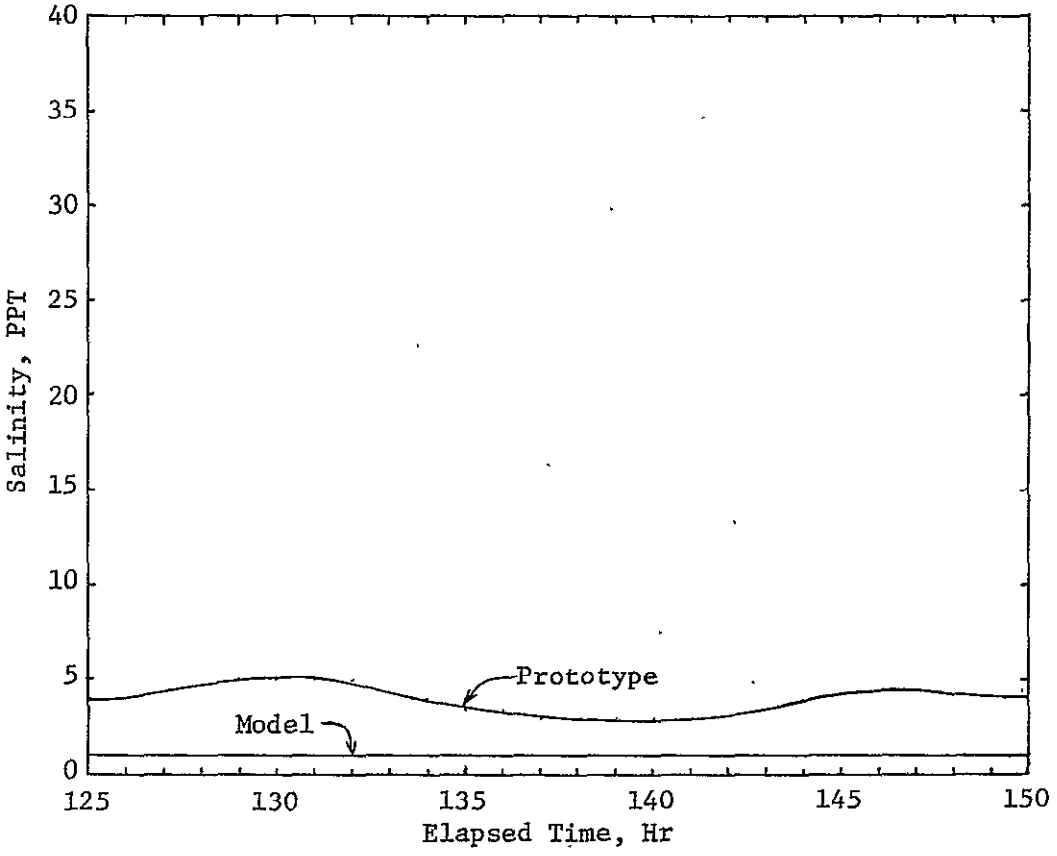


Figure 4.6 - Verification of Salinities at Beacon 32

after the model low tide at Main Pass occurred in the model, whereas the prototype variation is a much milder 13. to 7. PPT. The disparity between the model and prototype salinity curve is probably attributable to a strong but short flood tide in the model. For Beacon 32 the model predicted that the current would ebb over much of the tide cycle and flood with velocities up to -0.4 m s^{-1}. The salinity variation is small in both model and prototype with the latter salinity being approximately 4. PPT greater than the 1. PPT in the former. The model's strong flood tide caused the velocity reversal, but the failure of the prototype to reverse at the surface is unexpected considering the magnitude of the river flows and the tide conditions.

In addition to the free surface prototype data at Beacons 12 and 32 presented above, velocity magnitudes and salinities have been collected near the channel bottom (i.e., at 0.2 times the channel depth from the bottom). Directions have been assigned to the prototype velocity magnitudes by the Army Corps of Engineers on the basis of physical model results (McClelland, 1975). These data are compared in Tables 4.2 and 4.3. The velocity data at both stations agree with respect to direction and are of the same order of magnitude. From Table 4.2 one would conclude that near the bottom of the channel at Beacon 12 the flow is directed up the channel as the tide rises and passes through its high point and is directed down the channel as the tide falls and passes through its low point. Table 4.3 shows that at Beacon 32, the bottom flow is directed up the channel at all but low tide. The salinity data at both stations agree well over most of the tide cycle with most of the disagreement occuring at low tide in both cases.

TABLE 4.2

Beacon 12 Bottom Data

Elapsed Time hr	Tide Stage	Velocities		Salinities	
		Prototype $m\ s^{-1}$	Model $m\ s^{-1}$	Prototype PPT	Model PPT
131.25	falling	0.3	0.2	32.0	27.0
137.50	low	0.3	0.5	31.0	24.0
143.75	rising	-0.5	-0.9	28.0	27.0
150.00	high	-0.5	-0.2	31.0	30.0

TABLE 4.3

Beacon 32 Bottom Data

Elapsed Time hr	Tide Stage	Velocities		Salinities	
		Prototype $m\ s^{-1}$	Model $m\ s^{-1}$	Prototype PPT	Model PPT
131.25	falling	-0.2	-0.1	30.0	27.0
137.50	low	0.1	0.4	29.0	21.0
143.75	rising	-0.4	-0.4	29.0	27.0
150.00	high	-0.5	-0.5	26.0	27.0

The tide stage comparisons are presented in Figs. 4.7 through 4.11. The first two plots of this set contrast the tide stage functions supplied to the model for Main Pass and Pass Aux Herons with prototype curves. The model tide curves at the two tidal inlets agree with the prototype tide curves with respect to the time of high tide, but differ slightly as to the time of low tide. Low tide occurred approximately 0.75 hr earlier than the simple trigonometric function used in the model permits at Main Pass and 1.5 hr earlier at Pass Aux Herons. At both stations the model tide curves are about 10. cm higher than the prototype curves at high tide and 2. cm lower at low tide. The differences between the model and prototype curves are attributed to the following facts. The constants in the model tide stage functions were determined by least square fits to actual tide gage data that are not exactly periodic. The prototype curves, which were adapted from the Army Corps of Engineers report by Lawing and others (1975), were made exactly periodic by the Corps for a physical model study by adjusting the actual tide gage data. The two separate analyses of the original data resulted in the differences noted above.

Comparisons of the tide stage predictions for Fowl River Point, Point Clear and State Docks are presented in Figs. 4.9, 4.10 and 4.11, respectively. At all three stations, the model tracked the prototype data closely through falling tide and roughly predicted the correct times for low tide. However, it allowed a too rapid increase during the rising tide and overshot to produce high tides that are 25. to 30. cm too high and 2.5 to 3.0 hr premature. In Table 4.4 the differences in the tide elevation from State Docks to Main Pass at four points in the tide cycle are given. During falling and rising mean tide the

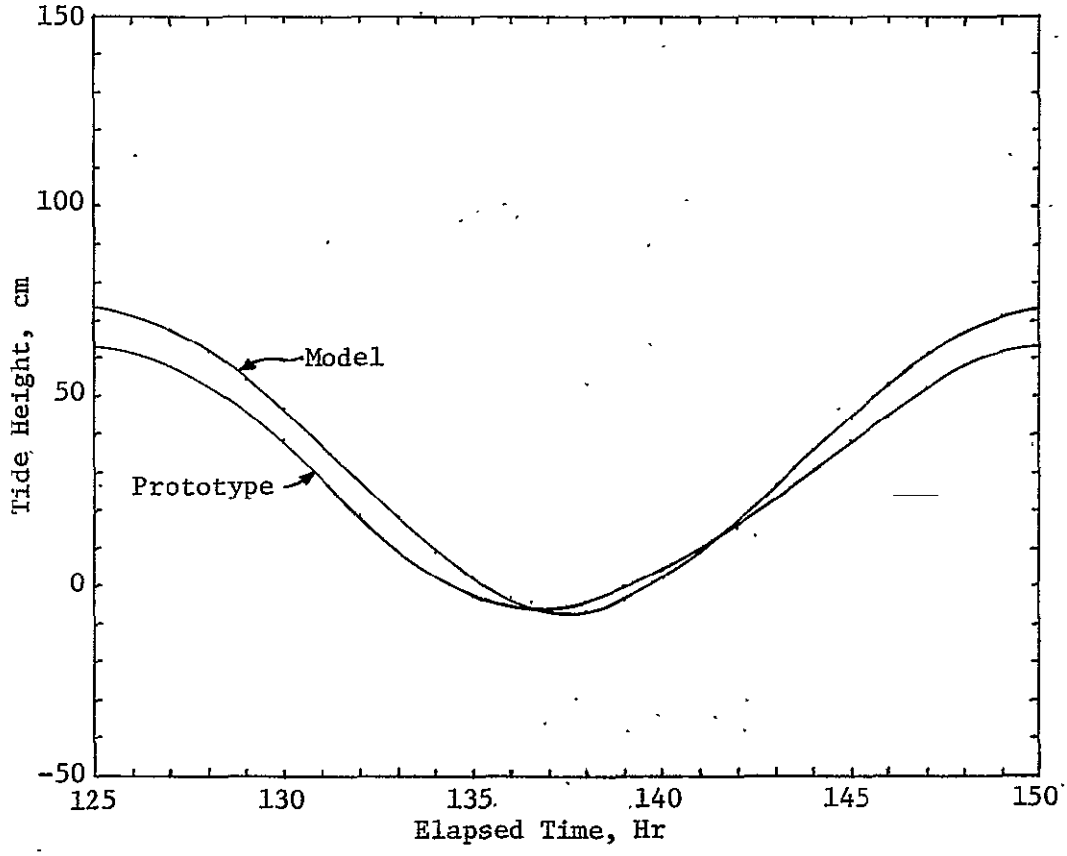


Figure 4.7 - Comparison of Idealized, Forced Model Tide at Main Pass with Prototype Tide

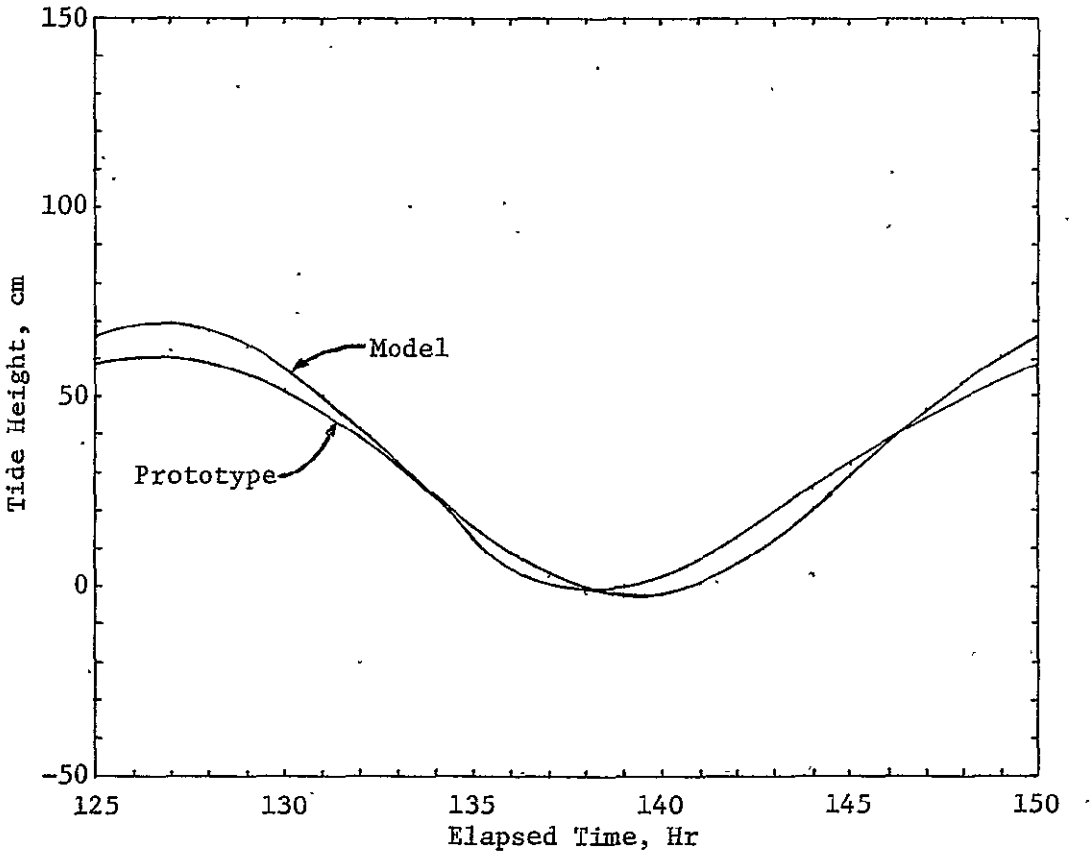


Figure 4.8 - Comparison of Idealized, Forced Model Tide at Pass Aux Herons with Prototype Tide

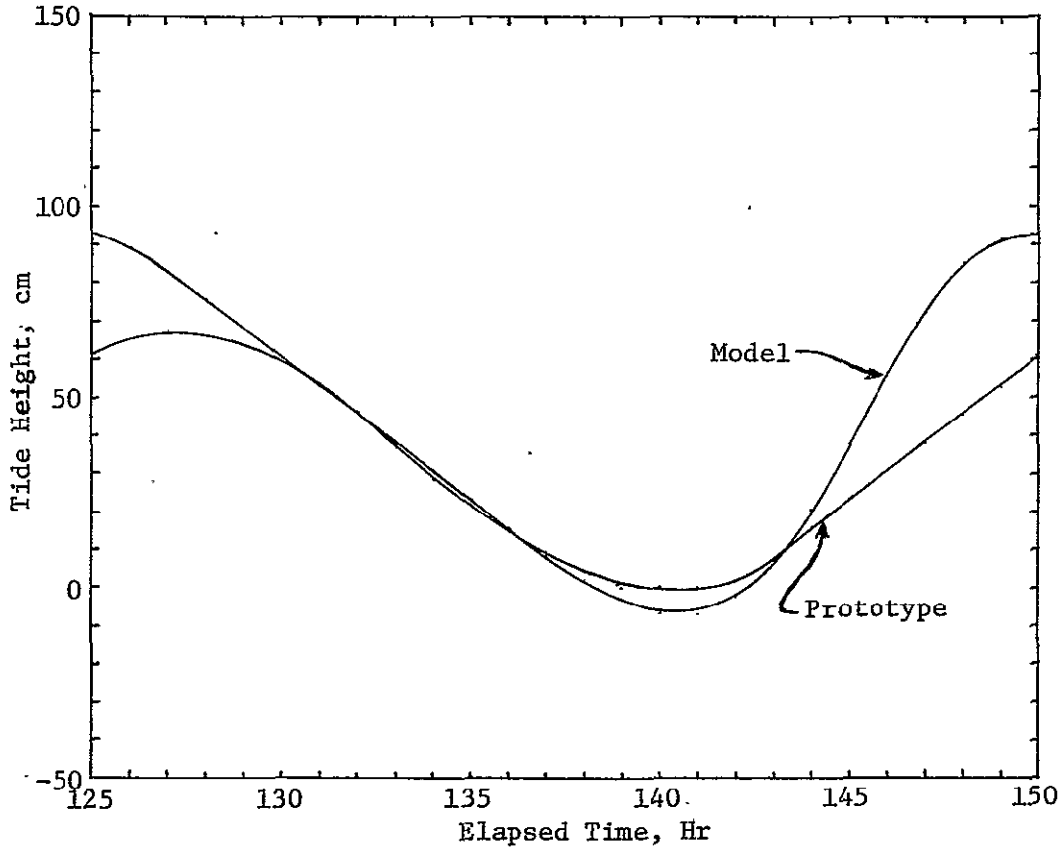


Figure 4.9 - Verification of Tides at Fowl River

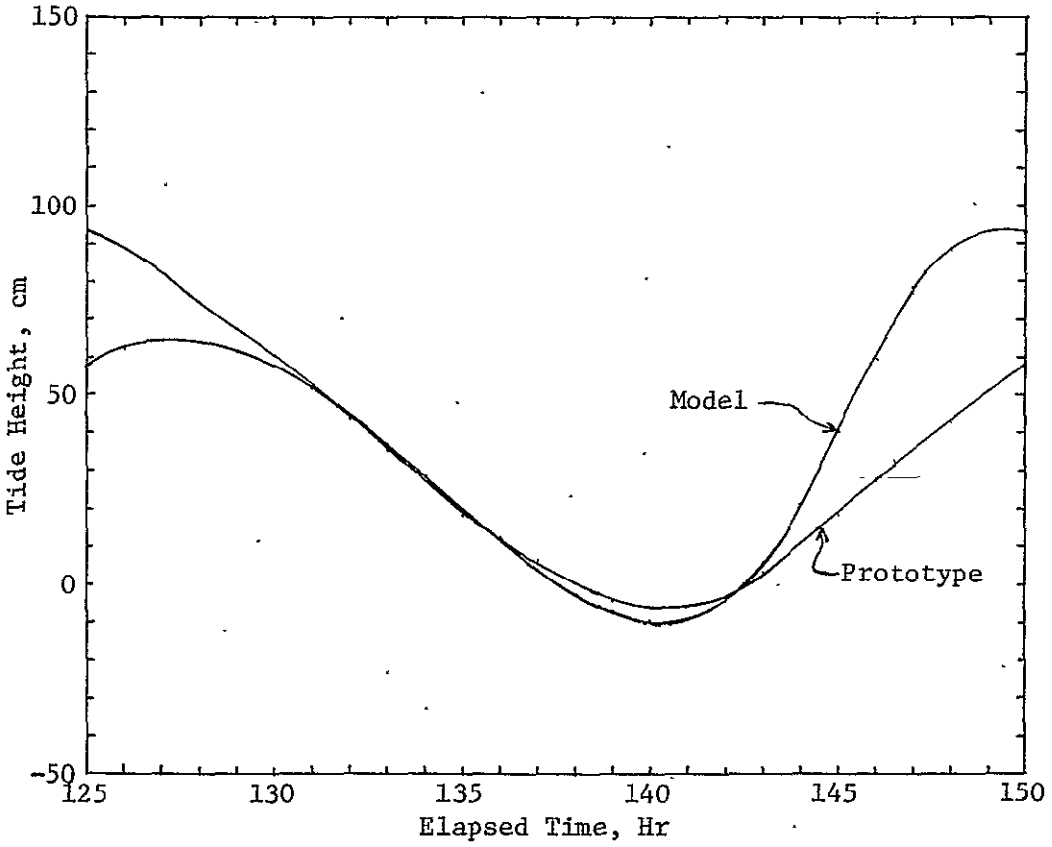


Figure 4.10 - Verification of Tides at Point Clear

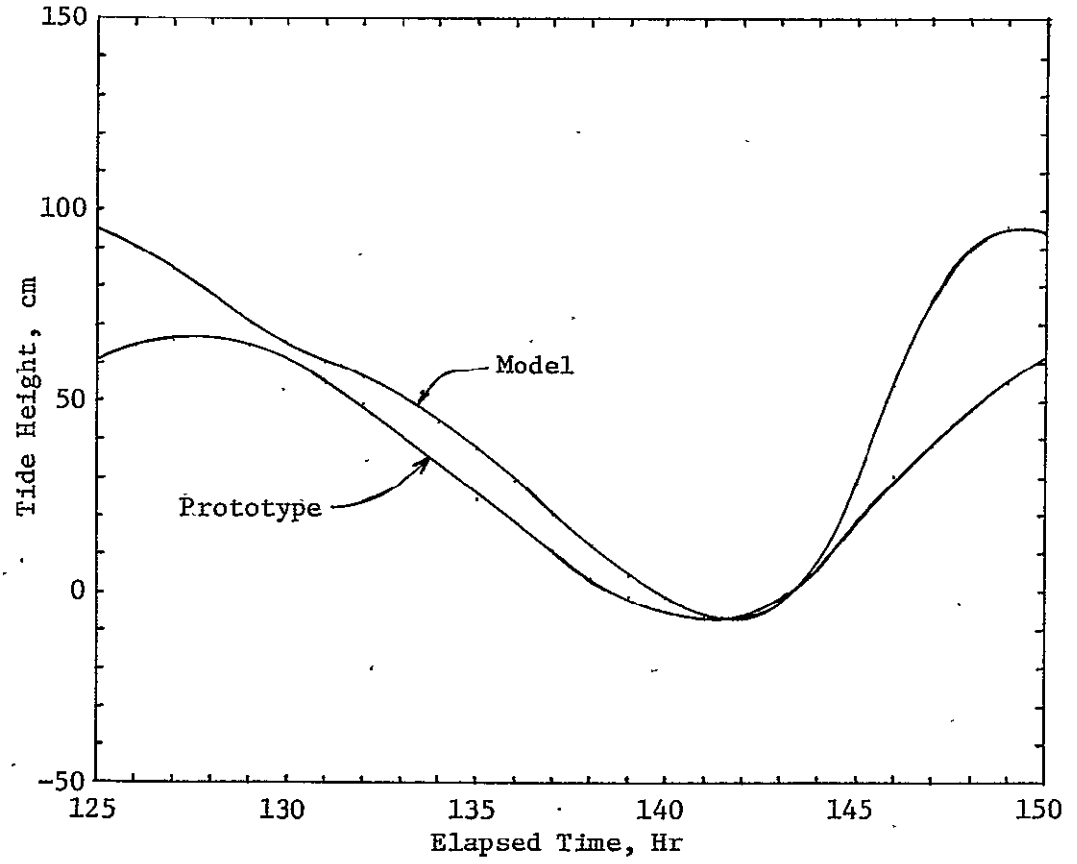


Figure 4.11 - Verification of Tides at State Docks

TABLE 4.4

Longitudinal Surface Differentials

Elapsed Time hr	Tide Stage	Differential	
		Prototype cm	Model cm
131.25	falling	-27.5	-27.0
137.50	low	-10.0	-24.0
143.75	rising	26.0	24.0
150.00	high	2.5	-21.0

agreement between model and prototype is good but at high and low tide it is poor. These amplitude and timing errors in the tide curves are the most telling deficiencies of the model as it presently stands since of the prototype data available, the tide curves are the data least subject to error. On the other hand, there is the possibility these discrepancies can be reduced by supplying the model with more accurate tide curves for Main Pass and Pass Aux Herons or by adjusting parameters such as the eddy viscosity. Further computations will be required to test these conjectures. But even with the tide stage errors, the goal of computing acceptable three-dimensional velocity and salinity distributions was achieved.

The instantaneous volumetric flow rates through Main Pass and Pass Aux Herons are compared in Figs. 4.12 and 4.13. The prototype curves are taken from Hill and April (1974). For Main Pass, the model predicted that ebbing commences at high tide; the prototype data indicate that ebbing does not start until 3.5 hr after high tide. Both the model and prototype data show that ebbing continues after low tide, for 3.25 hr. in the former and 4.5 hr in the latter based on the times of the low tides in the corresponding data. The peak model flooding rate of 44,000 m^3 s^{-1} is nominally 70% greater than the 26,000 m^3 s^{-1} for the prototype; the peak model and prototype ebbing rates are more nearly the same at 30,000 and 27,000 m^3 s^{-1}, respectively. For Pass Aux Herons, the model flow rate is a fixed at 20% of the rate through Main Pass with the same sign regarding ebb and flood and with no time lag to compensate for differences in times of flow reversals. The value of 20% was based on a consideration of estimates regarding the distribution of flows through the tidal inlets (McPhearson, 1970 and

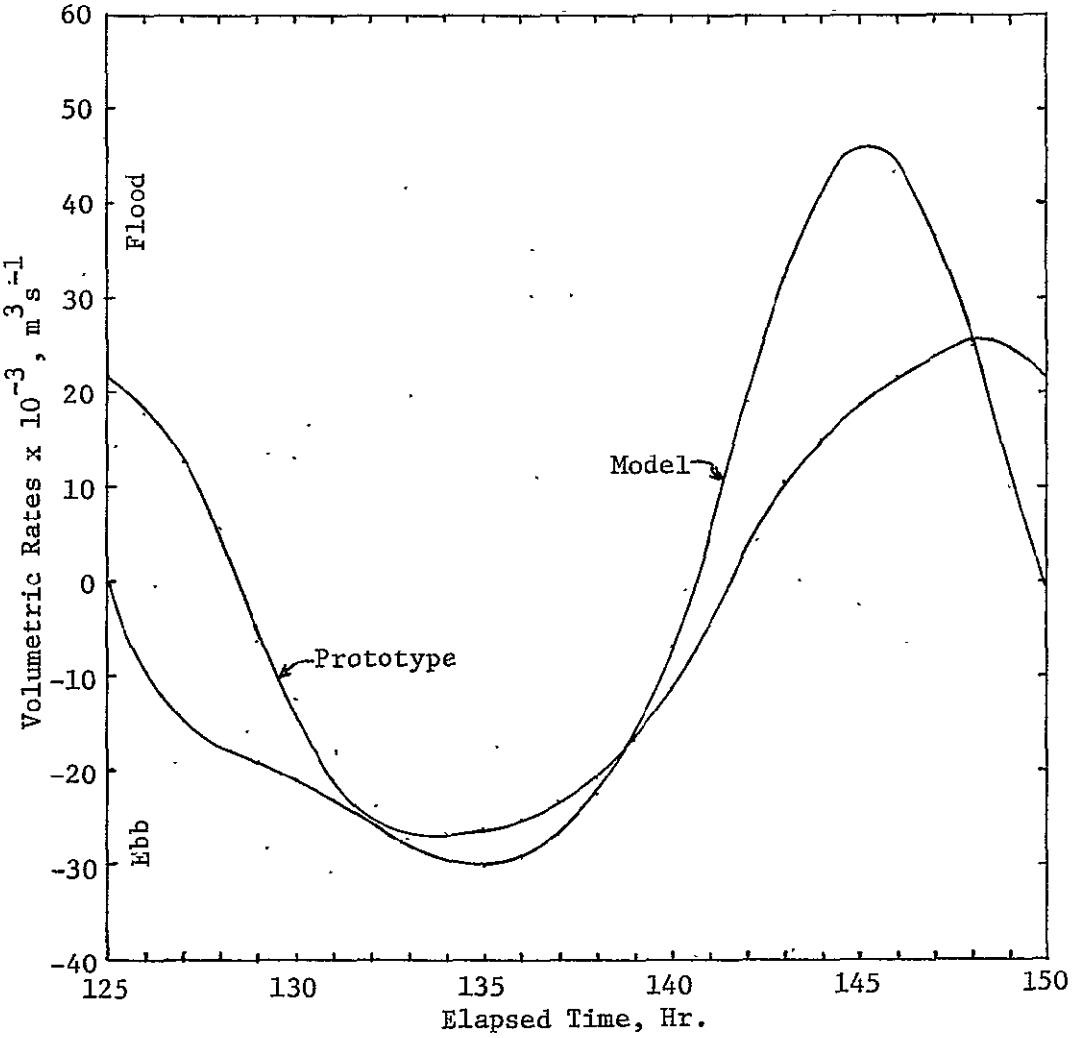


Figure 4.12 - Verification of Volumetric Rates at Main Pass

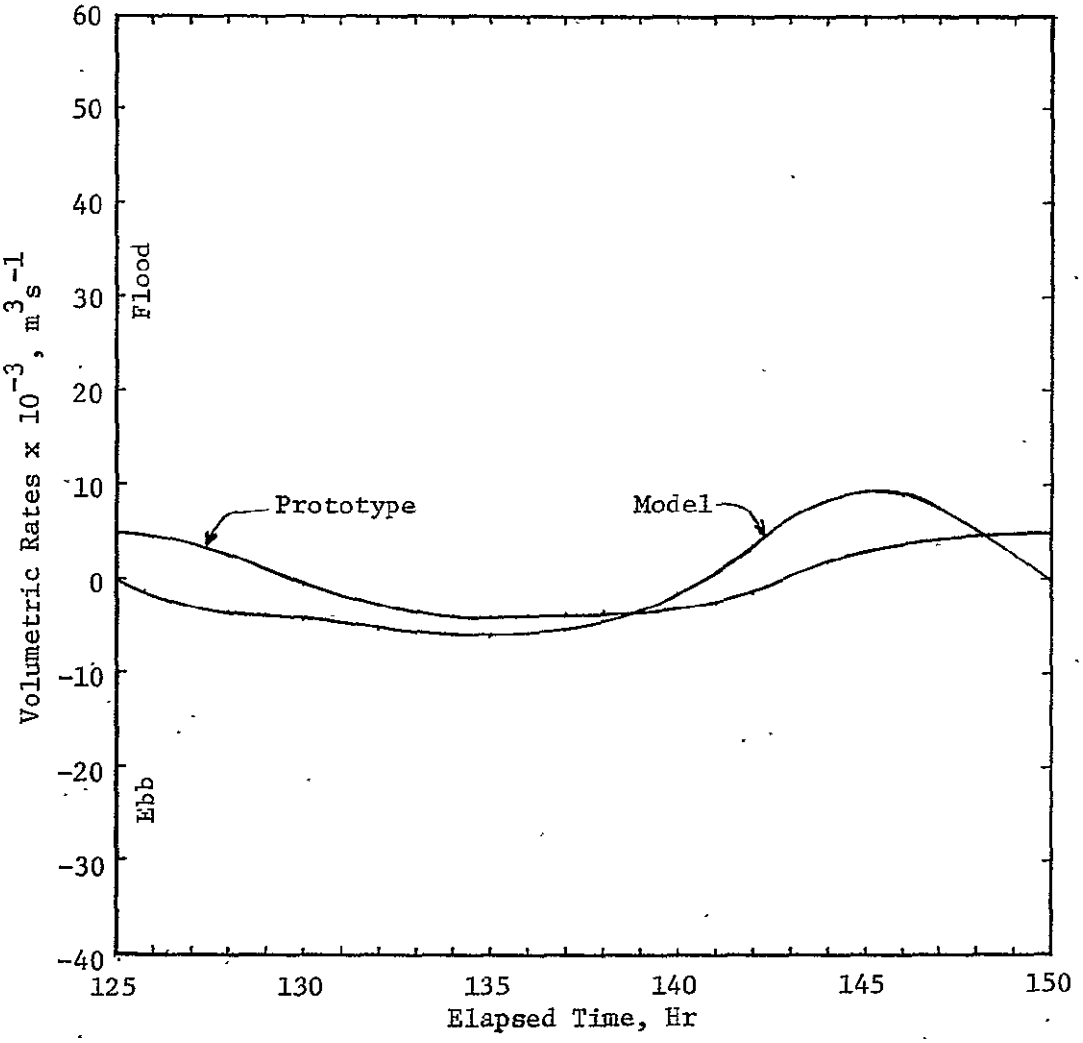


Figure 4.13 - Verification of Volumetric Rates at Pass Aux Herons

Doyle, 1975). This relationship was utilized only after many futile attempts were made at independently modeling the Pass Aux Herons flow rates. Hill and April (1974) encountered the same problem in their development of a vertically integrated, 2-dimensional model of Mobile Bay. The primary difficulty stems from the fact that the flow direction is strongly dependent on the difference in the surface elevation in the Mississippi Sound and just inside Mobile Bay. Any error in the prediction of the free surface elevation inside the bay can result in unrealistic flows through the Pass Aux Herons. Because the relationship between tidal flows is fixed in the model, the model ebb flow through Pass Aux Herons started at the same instant it did at the Main Pass (i.e., high tide at Main Pass), some 1.5 hr. before the prototype high tide at Pass Aux Herons. In the prototype, ebb flow started 2.75 hr. after high tide at Pass Aux Herons resulting in a total timing error of 4.25 hr. The fixed relationship also resulted in the model ebb flow continuing 1.25 hr. past the model low tide while the prototype data indicate ebbing continued for 4.75 hr. after the prototype low tide at Pass Aux Herons. The prototype data indicate that a one hour lag exists between the reversal of the tidal flow direction at Main Pass and Pass Aux Herons. Incorporation of this time lag in the computer model would reduce the above differences in the times of the flow reverals.

The results of integrating the instantaneous volumetric flow rate curves between flow reversals are given in Table 4.15. Because the model flood flow rates are larger and the flood durations are shorter than the corresponding prototype quantities, the average flood flow rate for the model is 66% greater than the prototype average at Main

TABLE 4.5

Tidal Discharges

Tidal Inlet	Net Ebb Flow 10^{-8}, m^3	Ebb Duration hr	Average Ebb Flow 10^{-3}, $m^3\ s^{-1}$	Net Flood Flow 10^{-8}, m^3	Flood Duration hr	Average Flood Flow 10^{-3}, $m^3\ s^{-1}$
			Model			
Main Pass	-11.5	15.6	-20.5	9.4	9.4	27.8
Pass Aux Herons	- 2.3	15.6	- 4.1	1.9	9.4	5.6
			Prototype			
Main Pass	- 9.0	13.2	-19.0	7.1	11.8	16.7
Pass Aux Herons	- 1.3	13.1	- 2.8	1.3	11.9	3.0

Pass and 87% greater for Pass Aux Herons. The average ebb flow rates for the model and prototype are somewhat closer in magnitude.

From the previous two paragraphs, the reader is certain to have the impression that the results of the tidal flow comparisons are negative. This impression should be mollified by the realization that the specification of model boundary conditions at tidal inlets is an exceedingly difficult task that requires much physical insight into the problem if it is to be done correctly. See Leendertse and Liu (1975) for an informative discussion of specifying tidal boundary conditions. Two difficulties center on fixing the free surface along a flow boundary as a function of position and time and on specifying velocity gradients in coordinate directions parallel and perpendicular to the open flow boundary. Seldom is this kind of data obtained in field surveys. A third difficulty concerns the salinity of the return flow to the estuary, particularly just after slack water. The composition of the returning water is strongly dependent on circulation patterns outside the bay. Beyond these considerations, there exists the possibility that some agent such as a local wind has influenced the prototype tidal data and the agent is not properly acccounted for in the model.

C. DISCUSSION OF COMPUTER MODEL TEST CASES

The results of a computer model of a multidimensional problem may be displayed graphically in two ways: (1) over the spatial extent at selected instants in time or (2) versus time at selected points in space. When the problem is three dimensional in space, the first type of display is generally limited to given cross-sections of the volume. The first approach is used extensively in this work. Because of their large number, these plots are located in Appendix A; they are discussed

below. The second approach is also used, and these plots are interspersed with their description in this section.

Appendix A contains plots for the four cases described in Table 4.5. The first three cases are presented with block borders that represent the limits of the computational grid used; the fourth case is presented with borders that represent the physical boundaries of Mobile Bay. Cases 1 through 3 contain the following type plots: (1) velocity vectors and normalized salinities (i.e., normalized density anomalies that have been multiplied by a factor of ten to facilitate plotting) in horizontal cross-sections at 0., 1.25 and 2.5 m below mean sea level (MSL), (2) free surface profiles, (3) velocity vectors and salinities in vertical cross-sections in planes perpendicular to the y-axis along the ship channel and 1.7 km to the west of the channel. These plots are given for the following four points in the tide stage at Main Pass: (1) falling mean tide, (2) low tide, (3) rising mean tide, and (4) high tide. Case 4 contains horizontal displays of velocity vectors and salinities at mean sea level and the free surface profiles at approximately 2.-hr intervals. The following table should help the reader visualize how Appendix A is arranged, and thus facilitate locating the figures as they are discussed. For Cases 1 through 3, the first five plots under velocity vectors and salinities are for falling mean tide, the next five for low tide, and so on. Under surface profiles for the first three cases the first plot is for falling mean tide, the next for low tide, and so on. As stated above, Case 4 contains plots at approximately 2.-hr intervals and only one horizontal plot of each kind for each interval. Since Case 4 is essentially a duplicate of Case 2, the discussion devoted to it will be minimal.

TABLE 4.6

Organization of Figures in Appendix A

Case	Velocity Vectors	Salinity Profiles	Surface Profiles
1	A.1 -A.20	A.21 -A.40	A.41 -A.44
2	A.45 -A.64	A.65 -A.84	A.85 -A.88
3	A.89 -A.108	A.109-A.128	A.129-A.132
4	A.133-A.144	A.145-A.156	A.157-A.168

On all of the plots in Appendix A, there is a figure labeled "Gulf Tide Stage" that indicates the current tide condition at the Main Pass. The arrowhead pointing toward the scale gives the tide elevation and the other arrowhead indicates whether the tide is rising or falling. On each figure there is an appropriate scale or reference to show the magnitude of the quantity plotted. Also note that the vertical scale in the north-south cross-section plots has been expanded by a factor of 2400 to make phenomena in those sections visible.

From the velocity vectors for falling mean tide, one can see that the rivers discharge into the bay at a few centimeters per second and that the tidal flows are out of the bay into the Gulf with an average velocity of 0.8 m s^{-1} in Cases 1 and 2 and 0.6 m s^{-1} in Case 3. There is a slight flow into Bon Secour Bay from the north and a 0.3 m s^{-1} current out of this area to the west. In the ship channel the flow is directed to the south at 0.1-0.3 m s^{-1} near the free surface and to the north in the same velocity range near the bottom except at the State Docks and in the southern most 10. km where the flows are directed to the south with velocities up to 0.7 m s^{-1}. The general movement in the bay is to the south with the pattern much the same for the two river discharges and with or without wind. Note that the wind pattern in Case 3 is indicated by the boldfaced vectors in Figure A.89. The wind speed varies from 6.1 m s^{-1} in the south to 4.7 m s^{-1} in the north.

At low tide the velocity vectors indicate that the rivers discharge into the bay at 0.3-0.5 m s^{-1} and that the tidal inlet flows ebb at 0.9 m s^{-1} in the first two cases and at 0.5 m s^{-1} in the third case. The current pattern in Bon Secour Bay is much the same as it is at falling mean tide except in Case 3 where the wind causes the main flow

to bypass the Bon Secour area and has set the shallow waters there in motion to the north against a slightly adverse surface gradient. From top to bottom along the entire length of the ship channel, the flow is ebbing with a maximum velocity of 0.8 m s^{-1} in Cases 1 and 3; the current is somewhat weaker for the lower river discharge of Case 2. As at falling mean tide, the general movement in the bay is to the south with the exception that the wind has resulted in a westerly movement in the southern third of the bay in Case 3.

As the tide rises past its mean position, the river discharges are in the 0.1-0.4 m s^{-1} range and are diminishing. There is a strong flood flow at the tidal inlets with the average of 1.0 m s^{-1} in Case 2 exceeding the 0.8 m s^{-1} in Case 1 with its higher river discharge rate. The average flood velocity in Case 3, 0.8 m s^{-1}, is weaker than in Case 2 because the wind results in storage of tidal waters which in turn reduces the favorable gradients in the free surface. In all the cases, there is a strong current into the Bon Secour area from Main Pass. In the ship channel, the flow floods in the south at 1.0-1.2 m s^{-1} and ebbs in the north. The ebb flow is stronger for Case 1 than Case 2 due to the difference in the river discharge rates and strongest in Case 3 because of the favorable surface gradient caused by the storage effect. The general movement in the bay is to the north except at the river mouths and along the southern edge of the bay where the movement is to the east. Currents along the western edge of the bay are enhanced by the wind.

At high tide the rivers are essentially stagnant in all cases. The tidal flow at Main Pass is weak and mixed with regard to direction except in Case 3 where the wind results in an average flood velocity of

0.2 m s^{-1}. The large velocity vector just outside the Pass Aux Herons is directed toward this pass in the first three cases because the imposed tide stage boundary condition in the pass is approximately 10. cm lower than the free surface at the location of the vector. The actual flow through Pass Aux Herons is 20% of that through Main Pass with the same direction with regard to ebbing and flooding. Therefore, the direction and magnitude of these currents are suspect in the first three cases. In Case 4, the free surface gradient in the Pass Aux Herons was set equal to zero (i.e., the surface height in the pass was equated to that at the first grid column inside the bay) and a more reasonable velocity pattern resulted. At high tide an unexpected vortex pattern rotating in a clockwise direction and having velocities up to 0.4 m s^{-1} appears in the Bon Secour area. At present, there are insufficient prototype data to verify the existence of such a flow pattern. The flow is predominately to the north in the ship channel with a maximum velocity of 0.7 m s^{-1} near the center. The channel current is beginning to reverse and ebb at Main Pass. The strength of the current at the north end of the channel is reduced in Case 1 because of the positive river discharge and in Case 3 because of adverse pressure gradients resulting from the storage effect. Except for the circulation pattern in the Bon Secour Bay, slack water exists in most of the bay with a slight southerly current of 0.1 to 0.2 m s^{-1} along the western edge of the bay in all but Case 3.

At this point it should be noted that the velocity distributions in the bay proper are relatively uniform with depth as can be seen in Figs. A.9, A.14, A.19 and others. A small reduction in the velocity occurs near the bottom due to viscous effects. However, in the ship-

ping channel intricate velocity distributions result that could not have been forecast by a vertically averaged model. One of the milestones of the present work is the successful interfacing of the three-dimensional model for the bay proper and the vertical, two-dimensional channel model.

Now the normalized density anomaly or salinity distributions for the cases will be discussed. In these figures, the integer numbers should be interpreted as follows:

Integer Numbers	Normalized Density Anomaly	Salinity Range PPT
0	0.00-0.05	0.0- 1.5
1	0.05-0.15	1.5- 4.5
2	0.15-0.25	4.5- 7.5
⋮	⋮	⋮
9	0.85-0.95	25.5-28.5
10	0.95-1.00	28.5-30.0

Since salinity is a concept familiar to most people working with estuarine systems, the discussion will be in terms of that variable.

On falling mean tide at Main Pass, the salinity in the northern third of the bay is less than 1.5 PPT. In Bon Secour Bay, the salinity ranges from 1.5 to 10.5 PPT in Cases 2 and 3. In Case 1, the higher salinity water along the southern edge of Bon Secour Bay is probably the remains of the unrealistically high initial salinity distribution that has not yet been transported out of the region. The salinity in the Main Pass is not uniform and depends strongly on the prevailing conditions. The salinity near the surface in the eastern quarter of Main Pass is apparently convected there from the Bon Secour area. The

maximum surface salinity in Case 1 is 19.5 PPT, while in Case 2 it is 22.5 PPT; the minimum is 4.5 PPT in both cases. The difference in the maxima results from the difference in river discharge rates and probably would be greater if Case 1 had reached exact periodicity with respect to the salinity distribution. The wind in Case 3 increases the maximum and minimum salinities in Main Pass to 28.5 and 16.5 PPT, respectively. An analogous increase in salinity exists at Pass Aux Herons in going from Cases 1 to 2 and from 2 to 3. If one uses 1.5 PPT to define an interface between low and high salinity water, this interface moves up the bay with decreasing river flow and with the application of wind from the south at equal river discharge rates. In particular, the wind pattern of Case 3 results in higher salinities along the western edge of the bay. Traveling south from the north end of the bay, one encounters this interface first in Case 3, second in Case 2 and third in Case 1. The ship channel is essentially full of 25.5 to 30.0 PPT salinity water below its interface with the bay proper. A smooth decrease towards fresh water occurs in the salinity profile as this interface is approached from below in the northern third of the channel. The saltwater intrusion in the channel is diminished from Case 2 by the higher river discharge of Case 1 and even more by the storage effect caused by the wind in Case 3.

At low tide the 1.5 PPT interface has moved down the bay while maintaining the same relative positions described in the previous paragraph for the first three cases. Now the northern half of the bay contains water with less than 1.5 PPT salinity. The salt content of the Bon Secour area as a whole has fallen slightly from that at falling

mean tide, but concentrations as high as 10.5 PPT are still present. The comment made above concerning the remains of the initial salinity distribution still applies to the extreme southeast corner of the bay in Case 1. The salinities in the tidal inlets are nominally 6.0 PPT lower than they were at falling mean tide in all cases, and the distribution across Main Pass is somewhat more uniform. The salt content of the northern third of the ship channel has decreased significantly over the quarter tide cycle with the change enhanced by the higher river discharge of Case 1 and the wind pattern of Case 3. The reduced salinity just below the bay-channel interface between 30. and 50. km is caused by a down flow of low salinity water from the bay proper during ebb tide.

When the tide stage at Main Pass reaches the rising mean position, the 1.5 PPT interface is at about the same position that it had at low time in Cases 1 and 2 and nominally 3. km further to the south in Case 3. The salinity distribution in the Bon Secour area did not change between low and rising mean tide. Since the currents in the tidal inlets are flooding strongly, the surface salinities there have increased and are approaching that of gulf water, 30.0 PPT. The fresh water content of the northern end of the ship channel has reached a maximum for the tide cycle with the saltwater wedge being 10. km further down the channel in Cases 1 and 3, respectively, than in Case 2. In all three cases the flooding flow at Main Pass is convecting gulf water into the channel.

At high tide, the 1.5 PPT salinity interface attains its northern most position in the three cases while maintaining the same relative positions described before. High salinity water is being convected

from the middle of the bay into the northwest corner of the Bon Secour area and low salinity water is being convected out of the area through the southwest corner. Except for the relatively constant salinity of the shallow waters to the east, the salinity in Bon Secour Bay increases by as much as 18. to 21. PPT over the last quarter tide cycle. Main Pass and the adjacent area contain gulf water as a consequence of the strong flooding. Pass Aux Herons contains water with a salinity equal to or slightly less than the 21.0 PPT of the Mississippi Sound. The flooding currents of the previous quarter tide cycle have convected saltwater up the ship channel. In Cases 1 and 2 pure gulf water has intruded half the length of the channel and in Case 3, almost a third the length. In Case 2, with its lower river discharge, the saltwater wedge has intruded to the channel's north end while the higher river discharge in Case 1 has retarded the intrusion significantly and the wind induced storage effects of Case 3 have retarded it even more.

For the particular set of river flows, tidal variation, and wind conditions investigated, the bay proper varies between being vertically homogeneous and slightly stratified. On the other hand, the ship channel is highly stratified below the channel-bay interface over much of the channel's length. Furthermore, the salinity distribution in the bay may require several tide cycles to reach exact periodicity after a substantial shift in the boundary conditions. In Case 1, the salinity in Bon Secour Bay continued to decrease four cycles after starting from inordinately high initial salinity in the area. Also, in Case 4 along the western edge of the bay above Fowl River Point, traces of higher salinity water forced there by the wind still remained two cycles after the wind has ceased.

The surface profile plots indicate that at falling mean tide there is approximately a 15. cm decline in the free surface elevation from north to south over the length of the bay excluding local effects at the river mouths and the tidal inlets. The differential is roughly half that given in Table 4.5, pa. 113, for the prototype between State Docks and Main Pass at the same point in the tide cycle. The surface gradient in the lateral direction is essentially zero except in the Main Pass. A combination of flow geometry and Coriolis effect generate a 5. to 15. cm rise from east to west across the width of this inlet. It should be emphasized that this lateral gradient is predicted by the model and is not an imposed boundary condition. The tide stage boundary condition, which is level in the lateral direction, is applied at an "imaginary" row of grid points one grid increment further out in the Gulf. The surface elevation at the north end of the bay is 3. cm higher in Case 1 than it is in Case 2 because of the difference in the river discharges; the surface profiles for these two cases are identical otherwise. In Case 3, the wind induced storage raises the surface by about 10. cm from that in Cases 1 and 2 over most of the bay except at the tidal inlets. At low tide, the surface of the entire bay has subsided 45. to 50. cm from its position at falling mean tide. The relative positions of the free surface are about the same between the three cases. Also, a very slight positive gradient from east to west exists as a result of the Coriolis effect. The wind pattern in Case 3 accentuates the lateral surface gradient. At rising mean tide, a 5. to 25. cm rise exists from north to south over the length of the bay. Cases 1 and 2 are nearly identical, while the free surface in Case 3 is about 20. cm higher at the north end of the bay and 5. cm higher at the

center than that in Case 2. No lateral gradients are discernible except at Main Pass. At Main Pass high tide, a 10. to 20. cm decline from north to south exists over the length of estuary with the free surface elevation being approximately the same in the three cases. A 2.5 cm rise occurs in the prototype at this point in the tide cycle. Laterally the surface is level in the northern half of the bay, but a 10. cm rise from west to east occurs in the Bon Secour area primarily because of the flow geometry.

The tide level curves for the designated stations in Fig. 4.2 are presented in Figs. 4.14, 4.15 and 4.16 for Cases 1, 2, and 3, respectively. An anomalous result that is immediately obvious in these plots is that the model high tide occurred at the State Docks as much as 0.75 hr before it did at the Main Pass. Figs. 4.3 and 4.7 show that the prototype high tide at State Docks lagged that at Main Pass by 2.5 hr. The relationship between the times of low tide at Main Pass and State Docks in the model are more nearly correct. Cases 1 and 2 are essentially identical; the primary difference is that the higher river flow in Case 1 has elevated the water levels at Point Clear, Fowl River Point and State Docks as much as 10. cm over those in Case 2. The times of high and low tide at the different stations are the same in the two cases. It should be noted that the model tide curves for the State Docks are influenced by the variation in the discharge rate for the Mobile River. This influence is evident in Figs. 4.14 and 4.15 where the increase in the river discharge during ebb tide retards the fall of the water level at State Docks. The tide curves in Case 3 indicate that the wind pattern reduced the fall of the water level in the middle and upper regions of the bay. In particular, low tide at

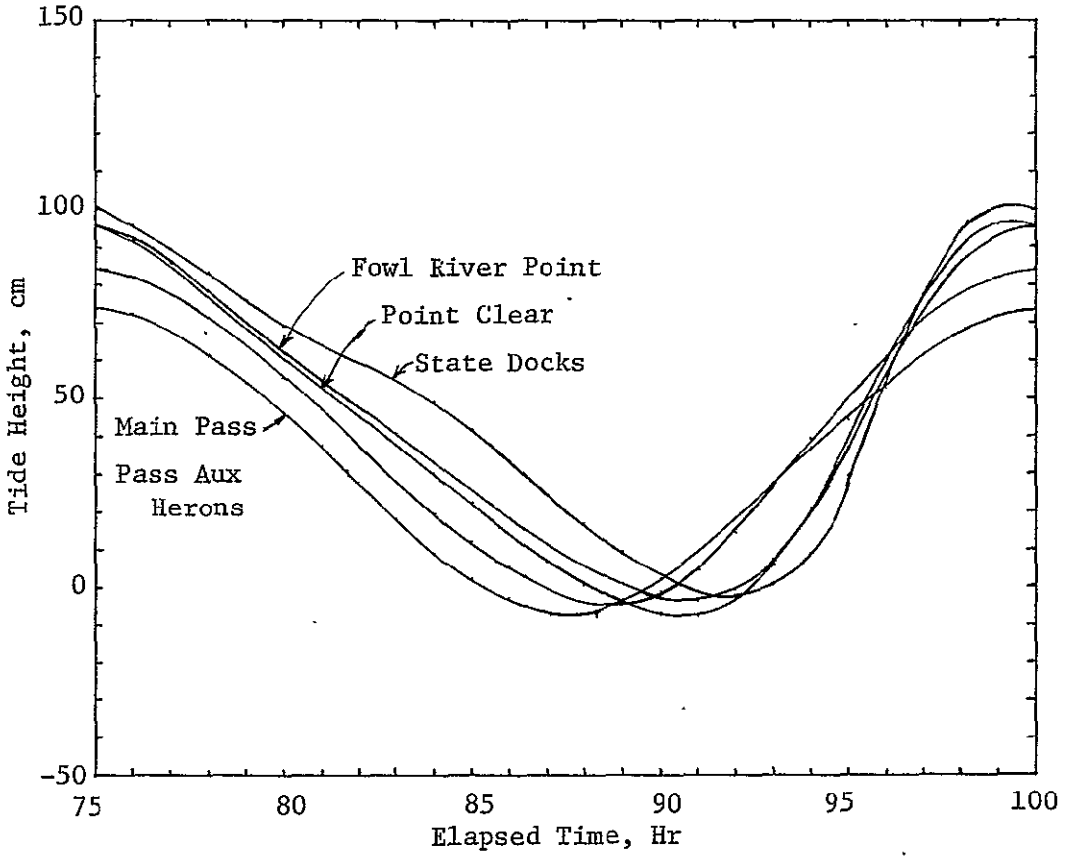


Figure 4.14 - Tides for Case 1

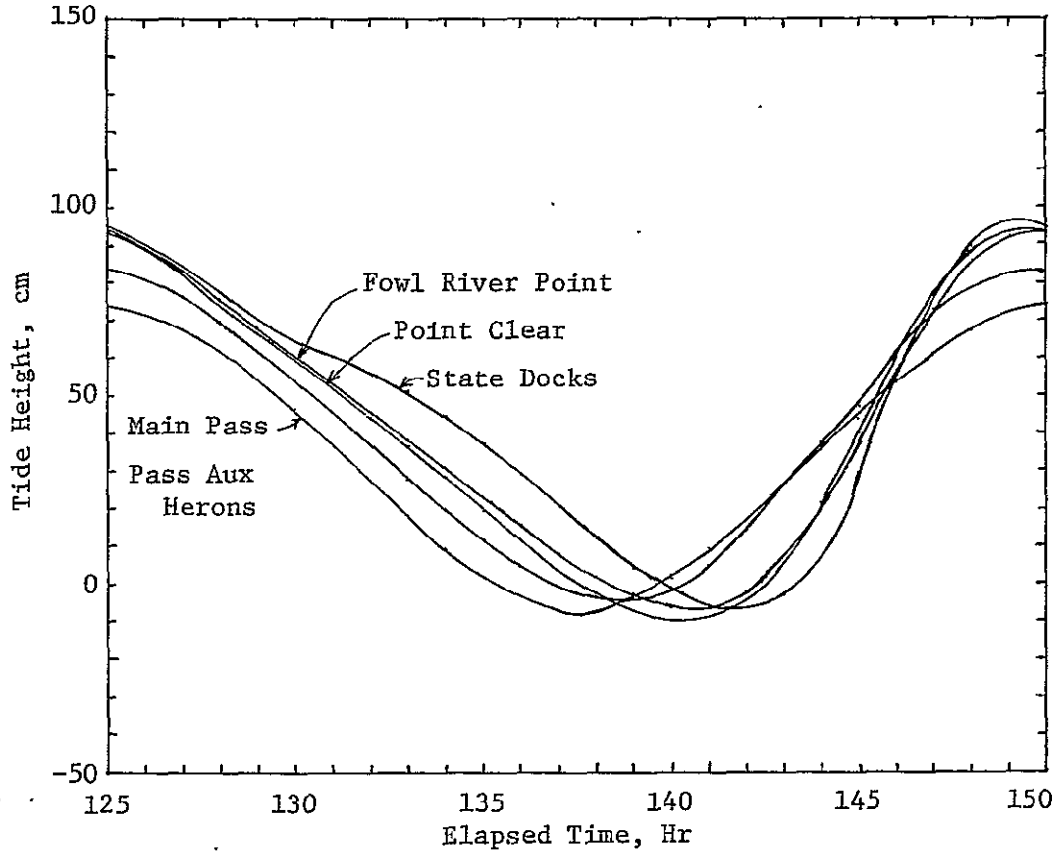


Figure 4.15 - Tides for Case 2

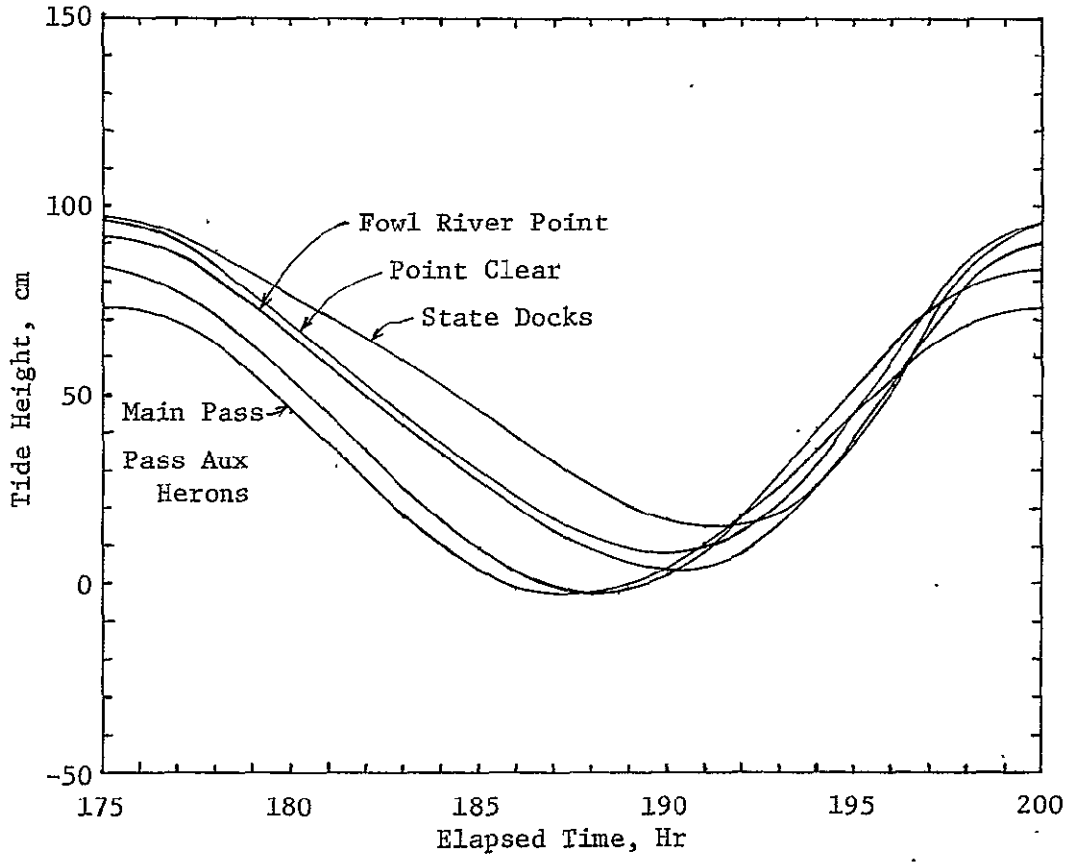


Figure 4.16 - Tides for Case 3

State Docks in Case 3 is 14. cm higher than it is at the corresponding station in Case 2. The times of low tide are approximately the same in Cases 2 and 3, but the wind in the latter case, somewhat surprisingly, delayed the time of high tide in the middle and upper regions of the bay by as much as an hour.

Figs. 4.17, 4.18 and 4.19 display the volumetric flow rates for the Main Pass, Pass Aux Herons and the combined rivers as functions of time. In addition, the rate of accumulation within the bay and an error calculated by summing the input rates and subtracting the accumulation rate are shown. The river velocities are specified functions of time in the model as described in Chapter 3. The average river velocity is 0.50 m s^{-1} in Case 1 and 0.27 m s^{-1} in Cases 2 through 4; the maximum variation is 0.52 m s^{-1} in all cases. However, the combined river rate depends on the flow cross-section of the river mouths which in turn depend on the free surface locations, variables calculated by the model. The tidal inlet flow rates and accumulation rate are calculated or predicted by the model. From Figs. 4.17 and 4.18, one can conclude that the 2.8 x 10^3 m^3 s^{-1} decrease in river flow rate from Case 1 to Case 2 is compensated for by a corresponding increase in the tidal flow rates so the accumulation rate remains approximately the same. A 1.5 hr increase in the duration of flood tide and a 9. x 10^3 m^3 s^{-1} decrease in the maximum flood flow rate from Case 2 to Case 3, indicated by Figs. 4.18 and 4.19, result because of wind induced storage in the bay. A point discernible in all three figures is that the error in the overall balance on the bay is the same order of magnitude as the Pass Aux Herons flow rate at any time in the tide cycle. The primary source of error is considered to be in the accumulation rate

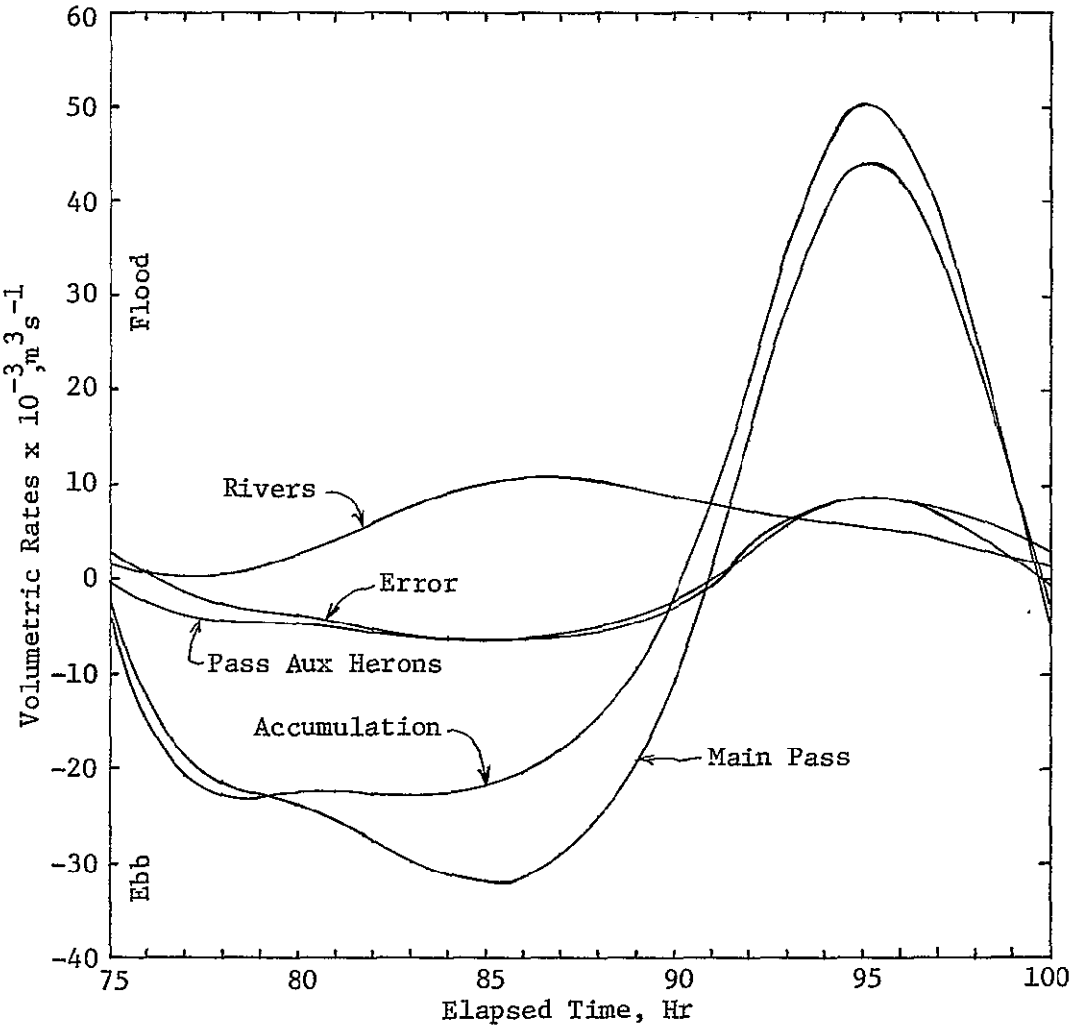


Figure 4.17 - Volumetric Rates for Case 1

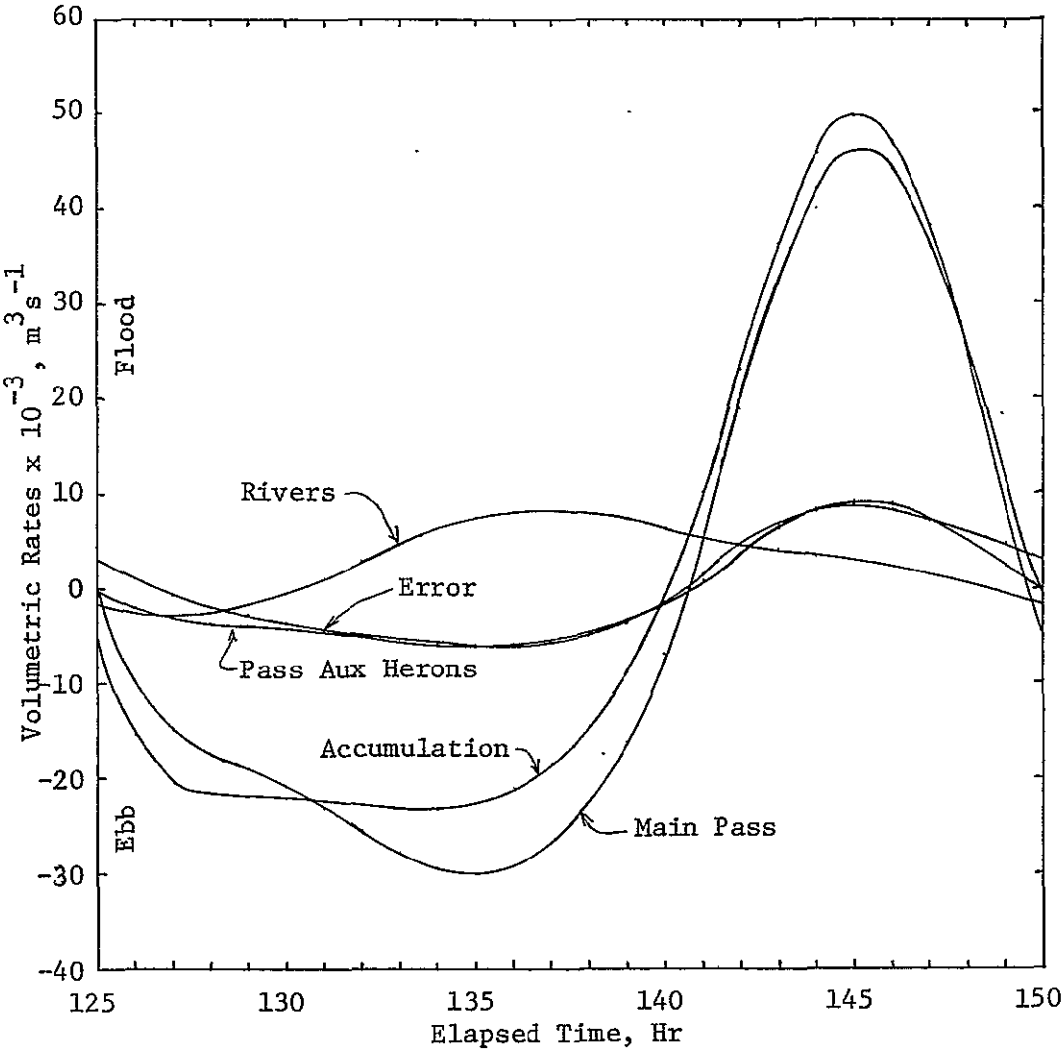


Figure 4.18- Volumetric Rates for Case 2

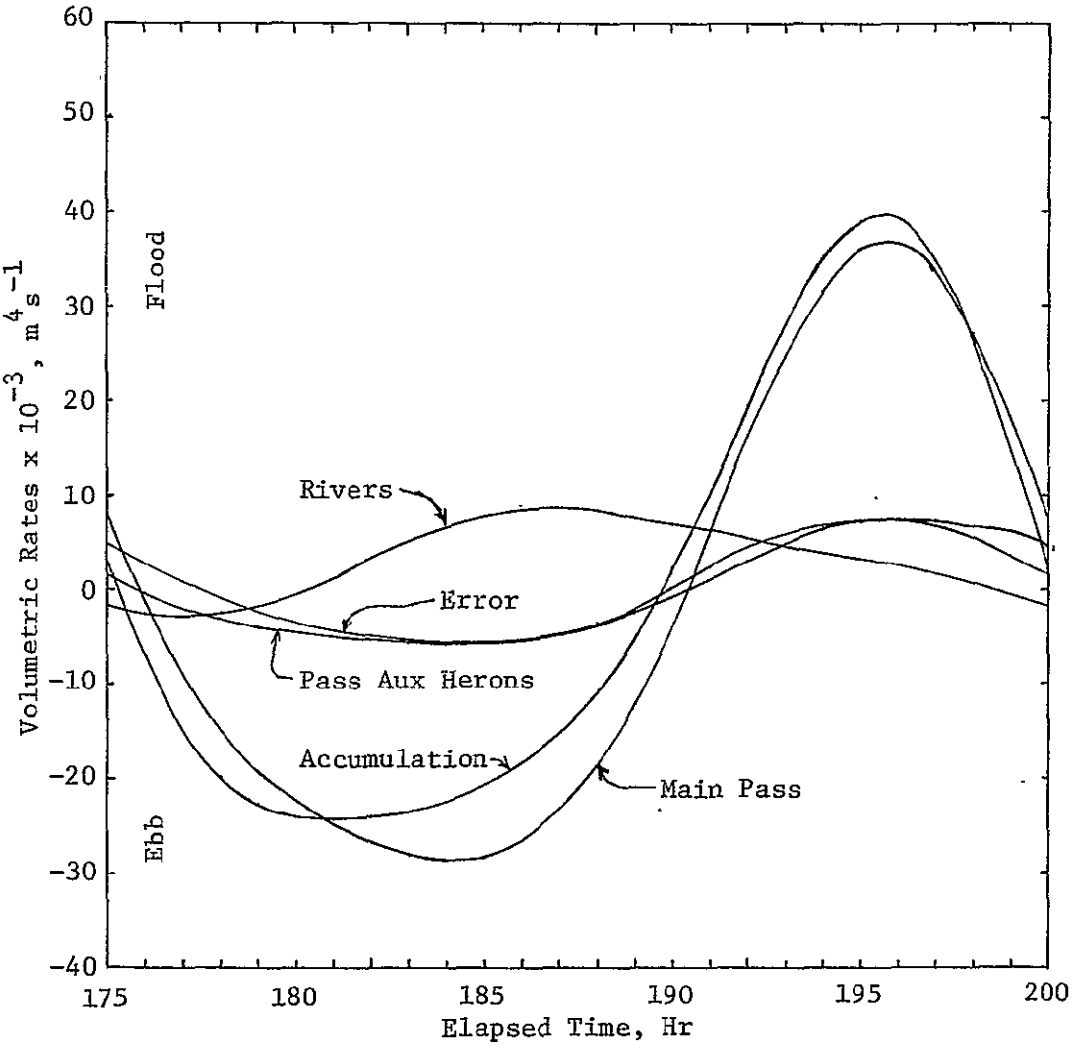


Figure 4.19 - Volumetric Rates for Case 3

term which depends directly on the rate of movement of the free surface. Any error in the calculated dynamics of the free surface, either from the finite difference scheme or the specification of boundary conditions, manifests itself as an error in the accumulation rate. In particular, if the surface rises and falls too slowly, then the error is proportional in phase and magnitude to the tidal flow rates which are significantly larger than the combined river flow rates over most of the tide cycle. The fact that the magnitude of the error is nearly the same as that of the Pass Aux Herons flow rate is thought to be coincidental.

D. IDENTIFICATION OF CONTROLLING PARAMETERS

The combined river flows have a significant influence on the composition and dynamics of Mobile Bay, particularly its upper region. The greater the freshwater flow, the lower the salt content of the bay for a given tide variation. The smaller the flow, the stronger the tendency for velocity reversals to occur in the river mouths and for salt water to intrude upstream. Conditions at the tidal inlets are complex and interactive. The flows through the Main Pass and the consequent free surface elevation and salinity distribution in the southern portion of the bay have a strong influence on what the flows and tide elevations in the Pass Aux Herons are like. The Pass Aux Herons flood current appears to deflect the Main Pass flow to the east during flood tide. Additionally, the flow through Pass Aux Herons depends on the conditions in the Mississippi Sound which are not necessarily the same as those in the Gulf of Mexico. The primary effects of local wind appears to be a perturbation in the velocity pattern throughout the

tide cycle and a redistribution of the salt water. It should be noted that a sustained wind over a long fetch of the Gulf of Mexico has a significant influence on the tide stage in the bay inlets. A wind from the north will tend to lower the average tide height in the inlets and the saltwater content of the bay, while a wind from the south will raise the average tide height and saltwater content. This wind-tide interaction has a much stronger influence on the estuary than a local wind alone.

As noted earlier, the accurate specification of the tidal inlet boundary conditions is critical to successful modeling of estuarine systems. This point is particularly applicable when the net flood and ebb tides are a significant fraction of the total volume of the system, say at low tide, as it is in Mobile Bay. Errors in these boundary conditions result in inaccurate simulations of the tidal flows and the attendant salinity distributions, if not instability in the calculation. Further study is required to determine the effect of the various approximations made in order to specify enough boundary conditions to solve the problem. For instance, the validity of setting $\partial u/\partial x$ and $\partial v/\partial x$ equal to zero or holding the salinity at the gulf water value at the "imaginary" grid points in the Gulf needs closer scrutiny. The former approximation implies that the flow is uniform as it passes through the boundary; the attainment of this condition depends strongly on the local flow geometry and the currents in the adjacent gulf area. The latter approximation is reasonable if the ebb flow is infinitely diluted by gulf water by the time the discharge reaches the last set of grid points. This dilution process probably does not happen in reality. Certainly, an accurate representation of the bathymetry of the bay is

needed. Because zero-order approximations are used to fit the geometry in order to facilitate formulation of boundary conditions and a narrow but deep portion of the Main Pass is ignored so as to minimize the size of the grid required, the topography of the Main Pass is poorly approximated at present, although the flow area is roughly correct. This discrepancy in geometry must have an influence on the tidal flows, probably in the direction of allowing the model tidal flows to be too large for a given tide stage variation.

Among the controlling parameters, one of the least understood is the eddy transport coefficient. Its impact on the tidal dynamics and salinity distribution in three-dimensional, time-dependent estuarine flow is a subject begging for further research. For instance, in the development of the present model an eddy viscosity of 15. $cm^2\ s^{-1}$ with a dampening factor based on the gradient Richardson number was used in the two-dimensional region of the ship channel. A recurring problem was the prolonged movement of an internal velocity wave up the channel toward the State Docks after the tide had begun to ebb at Main Pass. When the eddy viscosity was doubled to 30. $cm^2\ s^{-1}$, the velocity wave movement diminished and reversed during ebb flow as one would expect. To accurately model the unsteady flow and salinity patterns in an estuary, one must experiment extensively with the eddy transport coefficient formulations.

E. PURVIEW OF RESULTS

The primary purpose of this research is to develop a working model of three-dimensional, time-dependent flows in estuarine systems, and this goal has been achieved. The computational procedure is stable as

evidenced by a calculation involving 15,000 time steps of 60. seconds and ten tide cycles. Starting from different initial conditions and computing to the same final state demonstrate that the results are reproducible. Time steps approaching the limits set by the CFL criterion and the diffusion stability limit can be taken. The model conserves the total water volume relatively well. Furthermore, the model accounts for the interactions of the essentially two-dimensional ship channel with the three-dimensional bay, a non-trivial task.

This tool can be used to assess different formulations for eddy transport coefficients and different prescriptions of conditions at open boundaries such as the river mouths and tidal inlets. With a minimum of program modifications, the effects of changes in the bay geometry can be evaluated. Furthermore, since wind and tide conditions can be varied independently, the model can be used to investigate a variety of wind-tide conditions with ease.

All the relevant phenomena with their associated parameters are incorporated in the model. It is sufficiently general in scope that it can be used to simulate any estuary considered to be isothermal. It is particularly applicable to those having velocity reversals with depth and saltwater intrusion. What is now needed is a study to improve the simulative capacity of the model by selecting more accurate values and representations of the controlling parameters. This study will require a coordinated effort by the modeler and those groups making field surveys to identify and measure the parameters critical to accurate simulations and to verify the expected and unexpected results, both from the prototype and the model.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

The primary conclusion from this research is that the computer model is a viable means of studying three-dimensional, time-dependent estuarine flows. The nature of this viability is explained in the following comments: (1) Because of the three-dimensionality and time-dependency, use of the computer model with reasonable spatial resolution requires a significant amount of CPU time and core storage. For instance, in the present work approximately 220 K bytes of core and either 36.2 min/25-hr. tide cycle on the IBM 370, Model 158 or 54.8 min/25-hr. tide cycle on the IBM 360, Model 65 were required. Since the maximum stable time step is related to the smallest horizontal grid increment through the CFL criterion and the vertical grid increment through the diffusional stability limitation both CPU time and core storage requirements escalate with increased spatial resolution. With the present model an acceptable compromise between resolution and computer resources can generally be found. (2) Because there is no truly typical set of conditions for a given estuary, a potential user should expect to have to investigate and adjust certain items that were not fully investigated herein. For instance, the eddy viscosity and diffusivity probably vary with prevailing wind conditions; this variation was not considered in this research. Also,

at low river flows the boundary conditions on the species coutinuity equation would have to be modified to account for saltwater intrusion, a circumstance not handled by the model as it presently stands. (3) A potential user should have at his disposal computer plotting facilities. Graphical displays are essential to being able to interpret the large volumes of numbers generated when describing a general flow field.

Further conclusions are as follows: (1) The model velocity and salinity patterns compare favorably with those obtained from field measurements. (2) Sustained local winds have a signigicant influence on the velocity and salinity distributions as in Case 3. (3) High river discharges are capable of displacing salt water from the bay as in Case 1. The second and third conclusions agree with field observations also. However, quantitative data sufficient to check the second conclusion do not exist.

Recommendations for improving the computer model apply primarily to the numerical method. They are as follows: (1) The effect of the magnitude of the artificial or explicit numerical viscosity in the water level finite difference equation should be analyzed more fully. This parameter should be large enough to insure stability but not so large as to generate unwanted "diffusional effects". (2) At the expense of additional computations, the use of an explicit artificial viscosity in the water level finite difference equation can be eliminated by solving the equation with an alternating-direction-implicit

scheme. However, the programming becomes difficult when the geometry is complex. (3) The location of the water level is taken into account in the overall continuity equation (i.e. the water level equation) and the pressure terms of the momentum equations where it is essential. Improved solutions should result from modifying the finite difference formulations of the convective and diffusive terms of the momentum and species continuity equations when applied at grid points adjacent the free surface so that they also take into account the water level location. (4) "Semi-implicit" finite difference formulations of the second-order derivatives (i.e. the diffusion terms) in the momentum and species continuity equations that eliminate the diffusional stability limitation have been proposed (Roache, 1975). Their use should be considered. (5) The numerical method utilized here is termed a one-step procedure because only one estimate of each dependent variable is made for a new time level. A one-step procedure was used because it is computationally fast and the available CPU time was limited. If computer resources are not a limiting factor and accuracy of the solution is important, two-step procedures, where corrections to the first estimates at a new time level are made, should be investigated.

The recommendation for improving the mathematical model per se is that a better representation of turbulence, for instance a turbulent kinetic energy or Reynold's stress closure model, should be used instead of constant eddy transport coefficients modified by gradient Richardson number dependent dampening coefficients. It should also be noted that the particular set of boundary conditions

employed at the tidal inlets in the present work are not unique. For instance, if velocity profile data are available, these can be used as boundary conditions and the tide height variation with time predicted.

Recommended extensions and applications of the computer model include the following: (1) The model can be used to assess the impact of islands formed as a consequence of dredging operations on circulation and salinity patterns, a topic of vital interest to the oyster industry in Mobile Bay. This application is obvious and would require no modifications to the computer code. It was not pursued in the present research due to a lack of computer resources. (2) The computer code with a sediment transport model incorporated in it would be of value in scheduling maintenance-type dredging operations in Mobile Bay. (3) As mentioned in Chapter 1, the computer model should be useful in investigating the complex surface phenomena that cause the main ship channel to be strongly contrasted with the shallow waters on either side in visible spectrum photographs taken from satellites.

An item that may be classified as both a conclusion and a recommendation is that future computer modeling efforts and field surveys should be coordinated. In this way attempts to measure parameters critical to accurate simulations will be made and expected and unexpected results from the prototype and model can be verified.

REFERENCES

Austin, G. B. Jr. (1954), "On the Circulation and Tidal Flushing of Mobile Bay, Alabama, Part I," Ref. 54-20T, Dept. of Oceanography, Texas A&M.

Batchelor, G. K. (1967), An Introduction to Fluid Dynamics, Cambridge University Press, London.

Bird, R. B., W. E. Stewart and E. N. Lightfoot (1960), Transport Phenomena, J. Wiley and Sons, New York.

Bowden, K. F. (1967), "Circulation and Diffusion," Estuaries, ed. George H. Lauff, American Assoc. for the Advancement of Science, Washington, D. C.

Bowden, K. F. and P. Hamilton (1975), "Some Experiments with a Numerical Model of Circulation and Mixing in a Tidal Estuary," Estuarine and Marine Science, Vol. 3, No. 3, pp. 281-301.

Cameron, W. M. and D. W. Pritchard (1963), "Estuaries," The Sea, ed. M. N. Hill, Vol. 2, John Wiley and Sons, New York.

Codell, R. B. (1973), "Digital Computer Simulation of Thermal Effluent Dispersion in Rivers, Lakes, and Estuaries," AD-771 940, U. S. Army Missile Research, Development and Engineering Laboratory, Redstone Arsenal, Alabama.

Defant, A. (1961), Physical Oceanography, 2 Vols., Pergamon Press, New York.

Doyle, H. (1975), U. S. Army Engineer District, Mobile, Alabama, private communication.

Dyer, K. R. (1973), Estuaries: A Physical Introduction, John Wiley and Sons, London.

Farmer, R. C. (1976), private communication.

Farmer, R. C., W. R. Waldrop, F. H. Pitts and K. R. Shah (1975), "Development of a Three-Dimensional Time-Dependent Flow Field Model," NASA-CR-120762, Marshall Space Flight Center, Huntsville, Alabama.

Fisher, H. B. (1976), "Mixing and Dispersion in Estuaries," Annual Review of Fluid Mechanics, Vol. 8, pp. 107-133.

Hamilton, P. (1975), "A Numerical Model of the Vertical Circulation of Tidal Estuaries and Its Application to Rotterdam Waterway," Geophysical J. of the Royal Astronomical Soc., Vol. 40, pp. 1-21.

Harsha, P. T. (1971), "Free Turbulent Mixing: A Critical Evaluation of Theory and Experiment," AEDC-TR-71-36, Arnold Engineering Development Center, Arnold Air Force Station, Tennessee.

Hill, D. O. and G. C. April (1974), "A Hydrodynamic and Salinity Model of Mobile Bay," BER Report No. 169-112, Univ. of Alabama.

Hinze, O. J. (1959), Turbulence: An Introduction to Its Mechanism and Theory, McGraw-Hill, New York.

Horne, R. A. (1969), Marine Chemistry, Wiley-Interscience, New York.

Hsu, S. A. (1972), "Wind Stress on a Coastal Water Surface," Proc. 13th Coastal Eng. Conf., pp. 2531-2541.

Kinsman, B. (1965), Wind Waves: Their Generation and Propagation on the Ocean Surface, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

Lawing, R. J., R. A. Boland and W. H. Bobb (1975), "Mobile Bay Model Study," Tech. Rep. H-75-13, U. S. Army Waterways Experiment Station, Vicksburg, Mississippi.

Leendertse, J. J., R. C. Alexander and S.-K. Liu (1973), "A Three-Dimensional Model for Estuaries and Coastal Seas: Volume I, Principles of Computation," Rep. No. R-1417-OWRR, Rand Corp., Santa Monica, California.

Leendertse, J. J., and S.-K. Liu (1975), "A Three-Dimensional Model for Estuaries and Coastal Seas: Volume II, Aspects of Computation," Rep. No. R-1764-OWRT, Rand Corp., Santa Monica, California.

Lumley, J. L. and B. Khajeh-Nouri (1974), "Computational Modeling of Turbulent Transport," Advances in Geophysics, Vol. 18A, pp. 169-192.

Kamenkovich, V. M. (1975), "Basic Concepts in Modeling the Ocean Circulation," Numerical Models of Ocean Circulation, National Academy of Sciences, Washington, D. C.

McClelland, H. (1975), U. S. Army Engineer District, Mobile, Alabama, private communication.

McPhearson, R. M. Jr. (1970), "The Hydrography of Mobile Bay and Mississippi Sound, Alabama," J. Marine Science Alabama, Vol. 1, No. 2, pp 1-83.

Nichols, B. D. and C. W. Hirt (1973), "Calculating Three-Dimensional Free Surface Flows in the Vicinity of Submerged and Exposed Structures," J. of Computational Physics, Vol. 12, pp. 234-246.

Oden, J. T., O. C. Zienkiewicz, R. H. Gallagher and C. Taylor (Editors) (1974), Finite Elements in Fluid Flow, UAH Press, Huntsville, Alabama.

Officer, C. B. (1976), Physical Oceanography of Estuaries (and Associated Coastal Waters), John Wiley and Sons, New York.

Reynolds, A. J. (1974), Turbulent Flows in Engineering, John Wiley and Sons, New York.

Reynolds, W. C. (1976), "Computation of Turbulent Flows," Annual Review of Fluid Mechanics, Vol. 8, pp. 183-208.

Richtmyer, R. D. and K. W. Morton (1967), Difference Methods for Initial-Value Problems, Interscience, New York.

Roache, P. J. (1975), "Computational Mechanics," Lecture Notes in Mathematics, Vol. 461, pp. 195-256.

Roache, P. J. (1976), Computational Fluid Dynamics, Hermosa Publishers, Albuquerque, New Mexico.

Roberts, B. R. and R. L. Street (1975), "Two-Dimensional, Hydrostatic Simulation of Thermally-Indluenced Hydrodynamic Flows," Tech. Rep. No. 194, Civil Engineering Dept., Stanford Univ.

Schumann, U. (1975), "Linear Stability of Finite Difference Equations for Three-Dimensional Flow Problems", J. of Computational Physics, Vol. 18, pp. 465-470.

Shah, K. R. and R. C. Farmer (1976), "A Quasi-Steady Mathematical Model of Mobile Bay," Chemical Engineering Dept., Louisiana State Univ.

Slattery, J. C. (1972), Momentum, Energy, and Mass Transfer in Continua, McGraw-Hill, New York.

Spraggs, L. D. and R. L. Street (1975), "Three-Dimensional Simulation of Thermally-Influenced Hydrodynamic Flows," Tech. Rep. No. 190, Civil Engineering Dept., Stanford Univ.

Stoker, J. J. (1957), Water Waves: The Mathematical Theory with Applications, Interscience Publishers, Inc., New York.

Street, R. L. (1976), "Simulation of Thermally-Influenced Hydrodynamic Flows: Final Report," Tech. Rep. No. 203, Civil Engineering Dept., Stanford Univ.

Tennekes, H. and J. L. Lumley (1972), A First Course in Turbulence, The MIT Press, Cambridge, Massachusetts.

Townsend, A. A. (1976), The Structure of Turbulent Shear Flow, 2nd Ed., Cambridge University Press, London.

Turner, J. S. (1973), Buoyancy Effects in Fluids, Cambridge University Press, London.

Waldrop, W. R. and R. C. Farmer (1973), "Three-Dimensional Flow and Sediment Transport at River Mouths," Tech. Rep. No. 150, Coastal Studies Institute, Louisiana State Univ.

Waldrop, W. R. and R. C. Farmer (1974), "Three-Dimensional Computation of Buoyant Plumes," J. Geophysical Research (Oceans and Atmospheres), Vol. 79, No. 9, pp. 1269-1276.

Waldrop, W. R., R. C. Farmer and P. A. Bryant (1974), "Saltwater Intrusion into a Flowing Stream," Tech. Rep. No. 161, Coastal Studies Institute, Louisiana State Univ.

Yih, C.-S. (1965), Dynamics of Nonhomogeneous Fluids, The Macmillan Company, New York.

Zienkiewicz, O. C. (1971), The Finite Element Method in Engineering Science, McGraw-Hill, London.

NOMENCLATURE

c	empirical constant or conversion factor
C_v	heat capacity at constant volume
D	diffusivity
f	Coriolis parameter
g	gravitational acceleration
h	instantaneous flow depth
H	length characteristic of flow depth
i,j,k	spatial indices
I	identity element
k	friction factor or constant
k_o	Von Karman's constant
K	thermal conductivity
ℓ	Prandtl mixing length
L	length scale
m	empirical constant
n	normal distance to a surface or time index
N	kinematic viscosity
p	pressure
P	empirical constant
q	empirical constant or turbulent kinetic energy per unit mass of fluid
q_r	thermal radiation flux
Q	dummy variable

Ri	gradient Richardson number
s	salinity
S	normalized density anomaly
t	time
T	temperature or period of a tide cycle
u,v,w	velocities or components of velocity vector
u	friction velocity
U	depth mean velocity
V	velocity scale
x,y,z	Cartesian coordinates
X,Y	stretched coordinates
z_o	size of a wall roughness element

Greek

α	constant in state equation or a viscosity or diffusion coefficient
β	density difference between gulf and fresh water or an empirical constant
γ	explicit artificial or numerical viscosity
ε	dissipation rate of turbulent kinetic energy per unit mass of fluid
θ	angle with respect to positive x axis in the horizontal plane
κ	thermal conductivity
ν	kinematic viscosity
ρ	density
τ	shear stress
τ_o	wall shear stress

ϕ	phase lag
Φ_v	viscous dissipation function
ω	mass fraction
Ω	maximum variation in ω with distance or angular velocity

Subscripts

A	chemical species A, in particular, total salt
avg	tide cycle average
B	chemical species B, in particular, water
(e)	effective value
f	fresh water value
g	gulf water value
max	maximum value
(t)	turbulence induced value
var1,var2	first and second variations over tide cycle, respectively
w	wind
o	initial value or reference quantity
1	boundary value

Operators

div	divergence
$\frac{D}{Dt}$	substantial derivative
EXP	exponential
grad	gradient
I_z^h	trapezoidal integral
$\underline{Q}\underline{Q}$	dyadic product
x	cross vector product
$\cdot$	inner vector product

$\frac{\partial}{\partial Q}$	partial derivative with respect to Q
Δ	incremental difference
Δ_x^c	first-order centered difference
Δ_{xx}^c	second-order centered difference
Δ_x^w	windward difference
Σ	summation
$\int$	integral
$\langle\ \rangle$	ensemble average

Special symbols

Q'	turbulent fluctuation in Q about ensemble mean or first derivative of Q with respect to distance
Q''	second derivative of Q with respect to distance
Q^*	nondimensional variable
$\overline{Q}$	ensemble mean
Q	scalar
$\underset{\sim}{Q}$	vector
$\underset{\approx}{Q}$	tensor

APPENDIX A

TEST CASES

This appendix contains the Varian plots of the computer model results for the four cases described in Table 4.1, pa. 92, and discussed in Chapter 4, beginning on pa. 89.

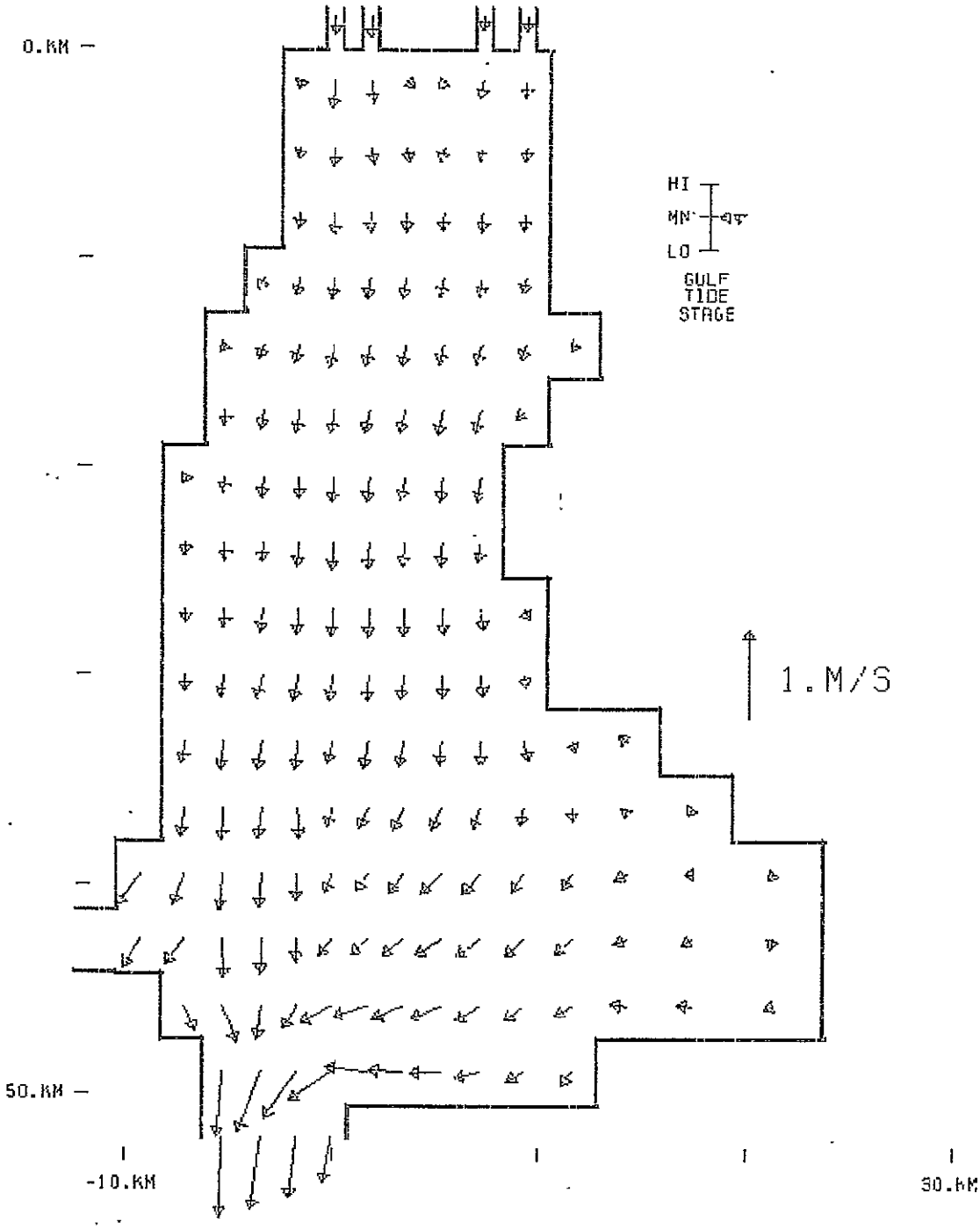


Figure A.1 -

Case 1

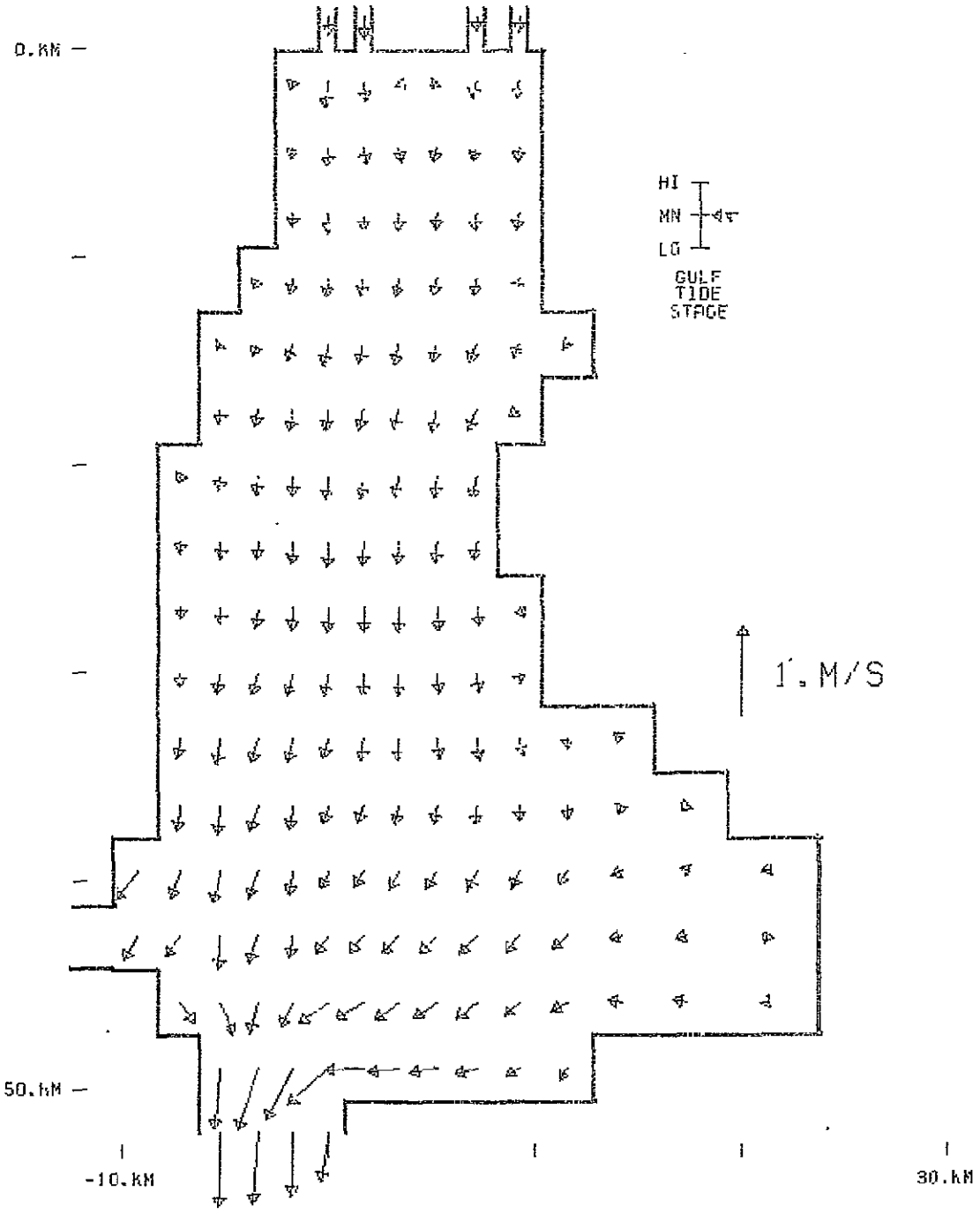


Figure A.2 -

Case 1

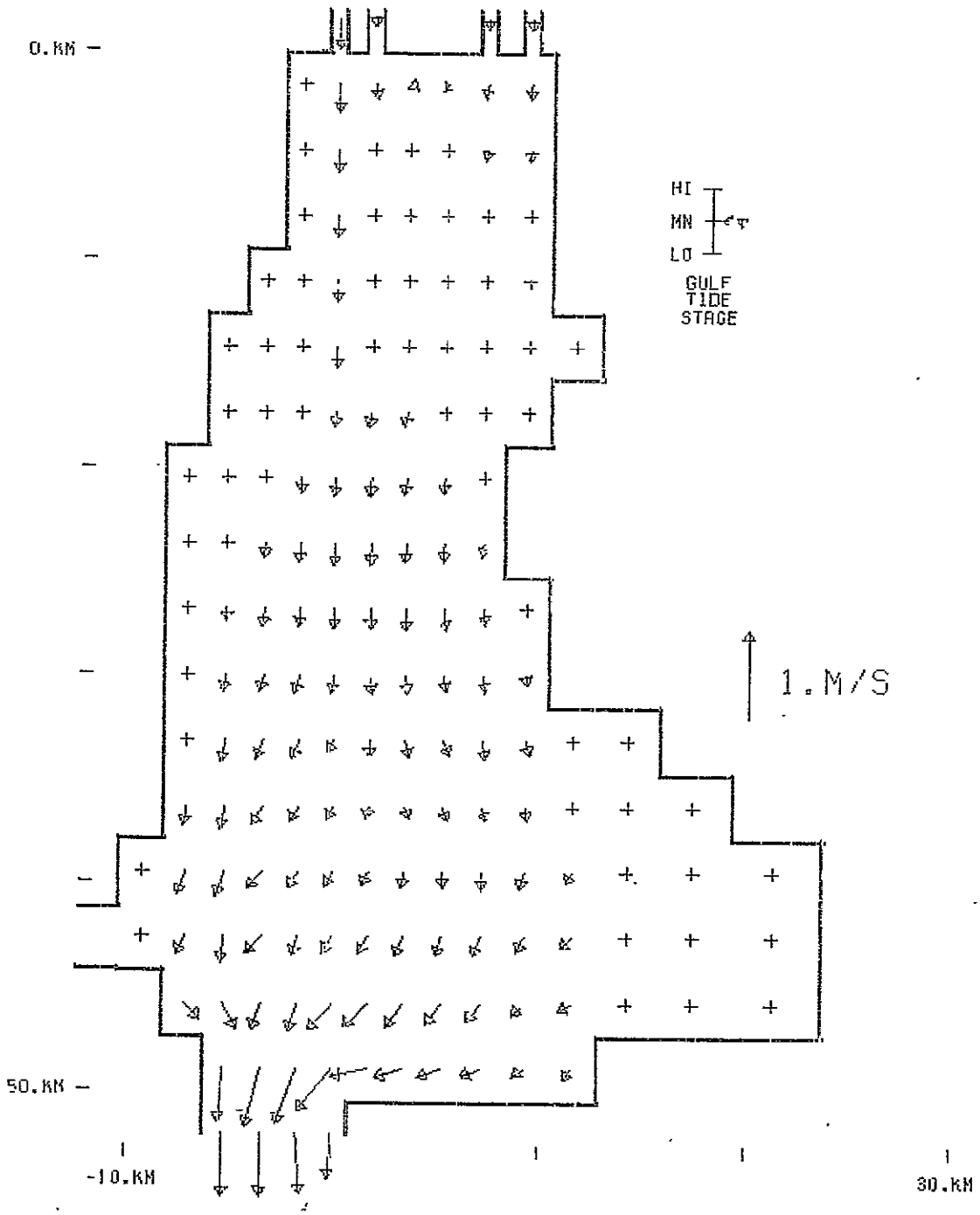


Figure A.3 -

Case 1

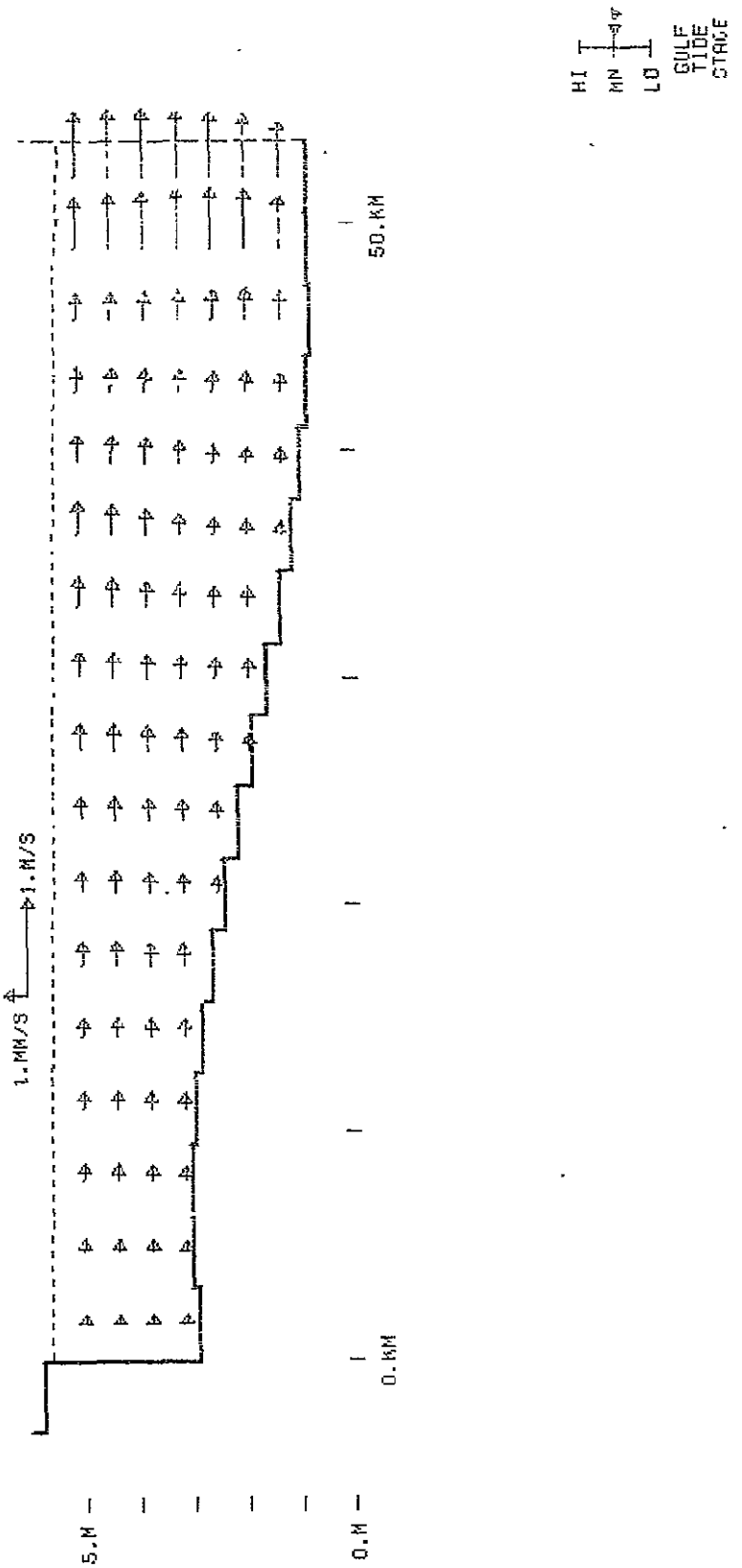


Figure A.4 - Case 1

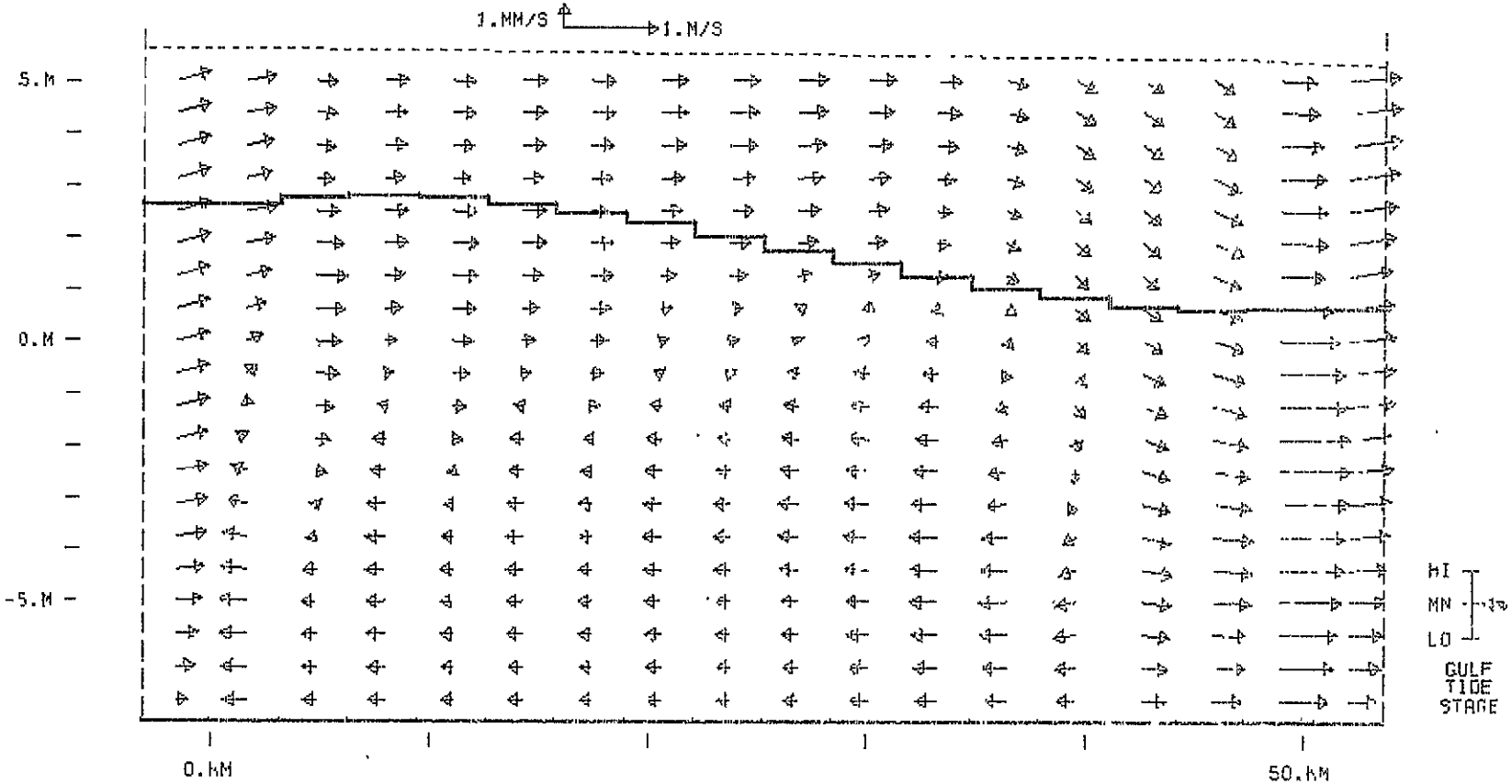


Figure A.5 - Case 1

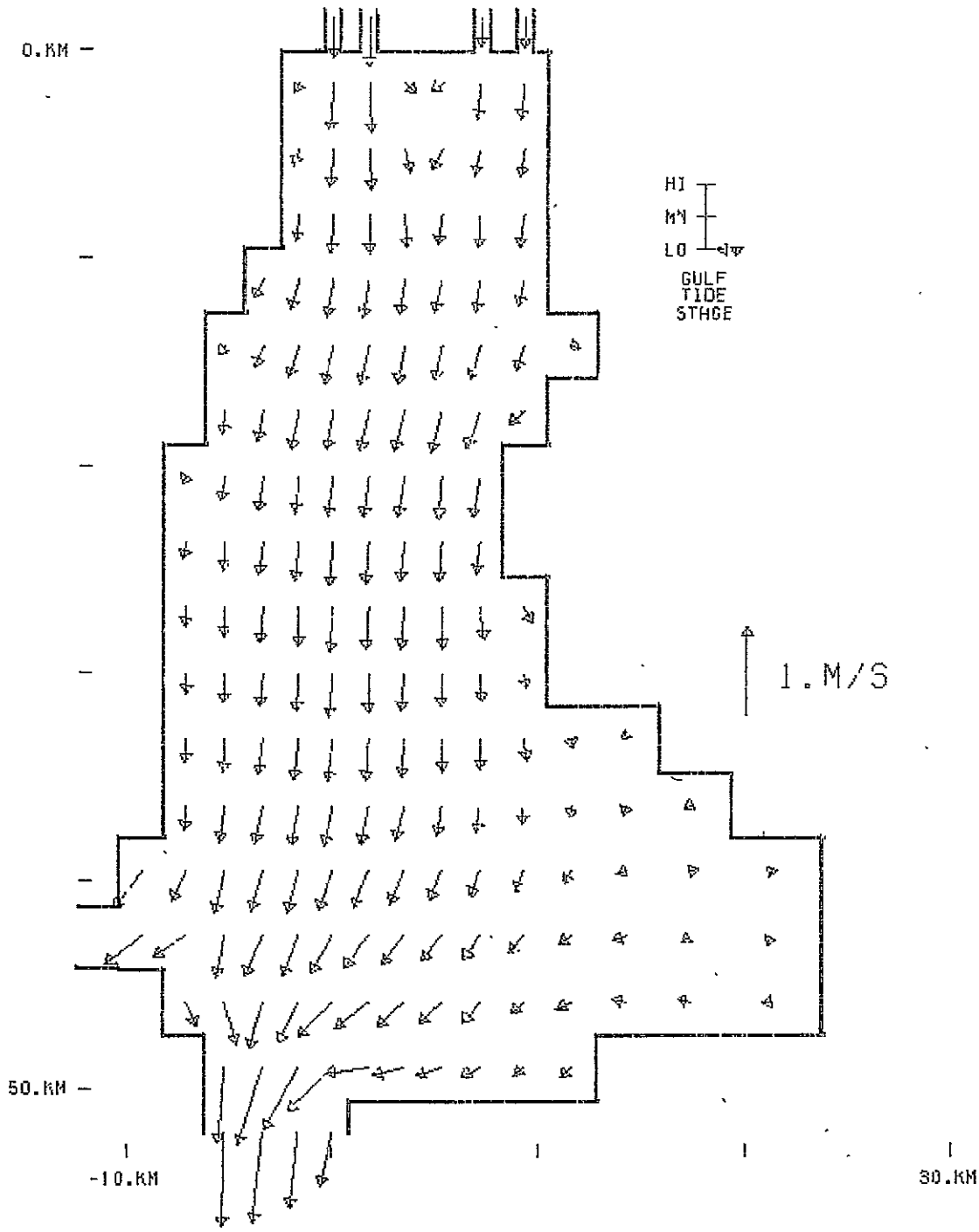


Figure A.6 -

Case 1

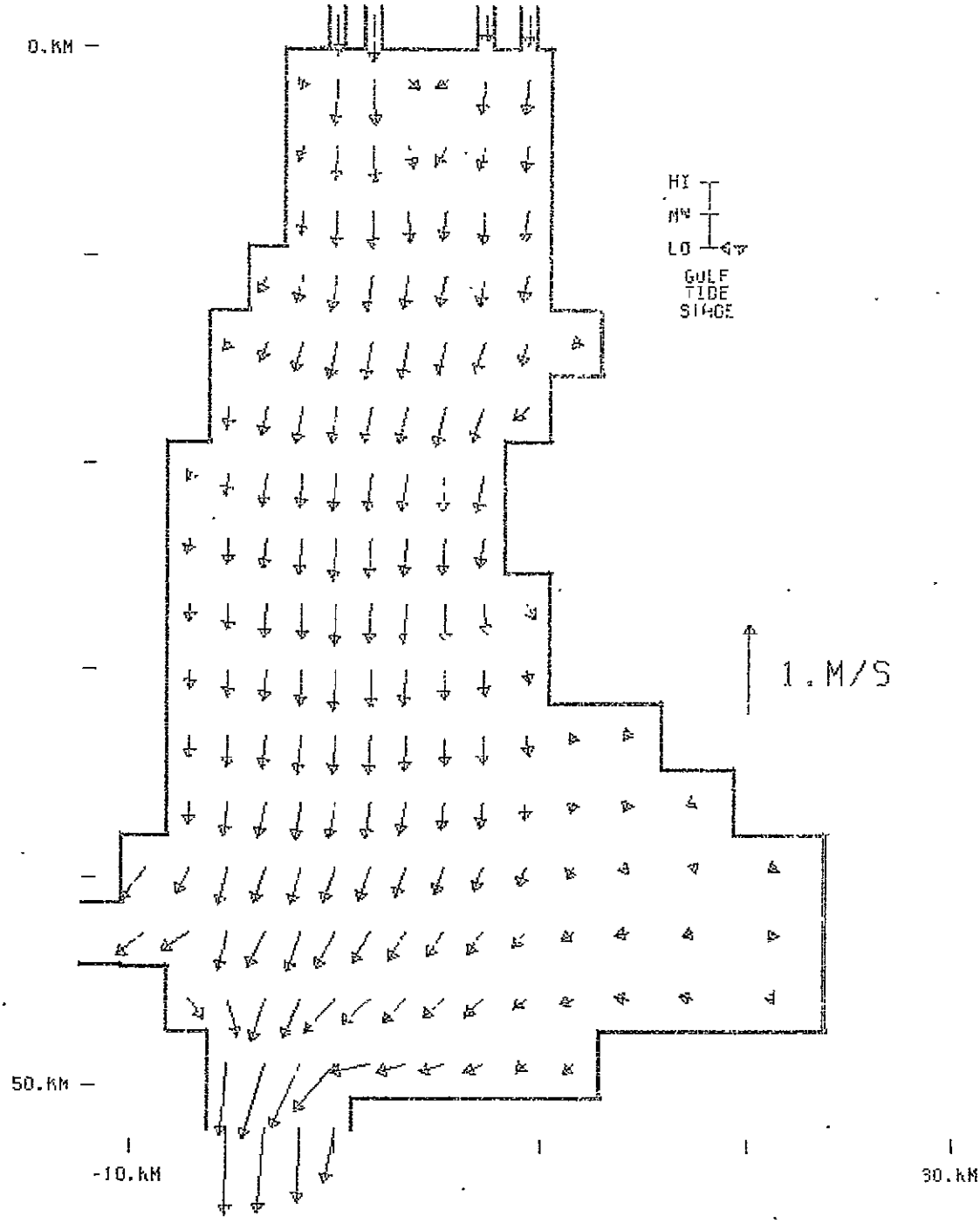


Figure A.7 -

Case 1

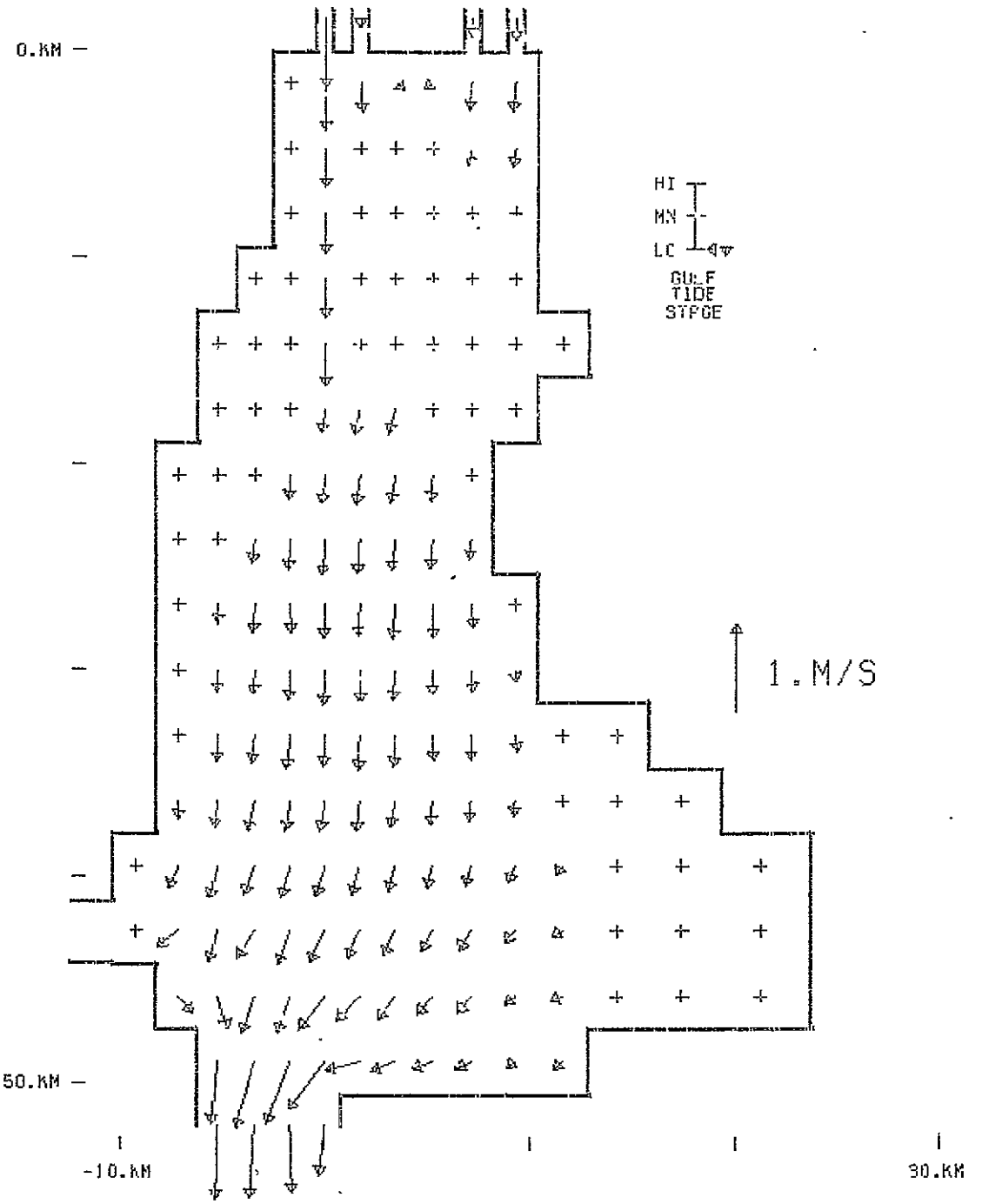


Figure A.8 -

Case 1

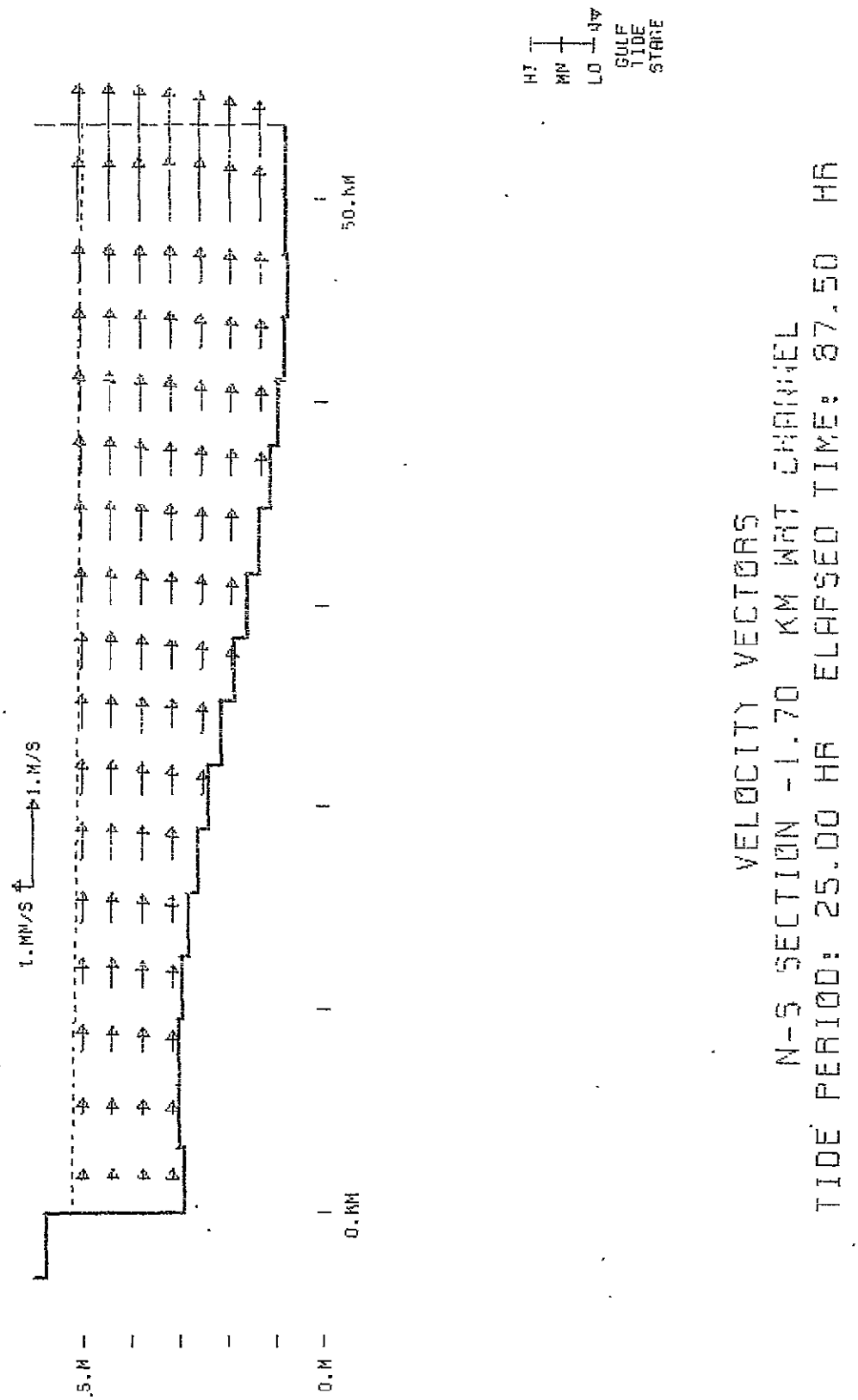


Figure A.9 - Case 1

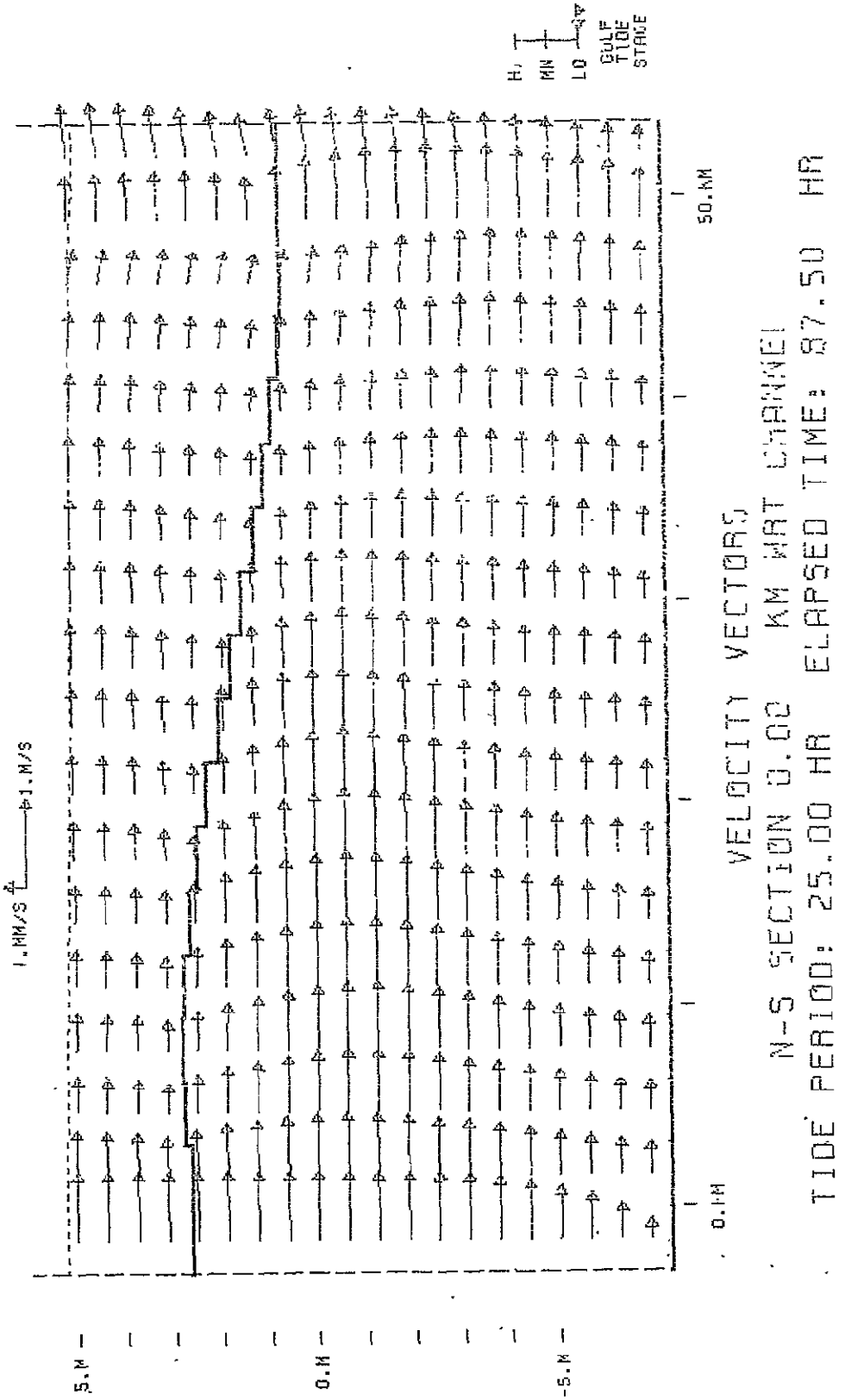


Figure A.10 - Case 1

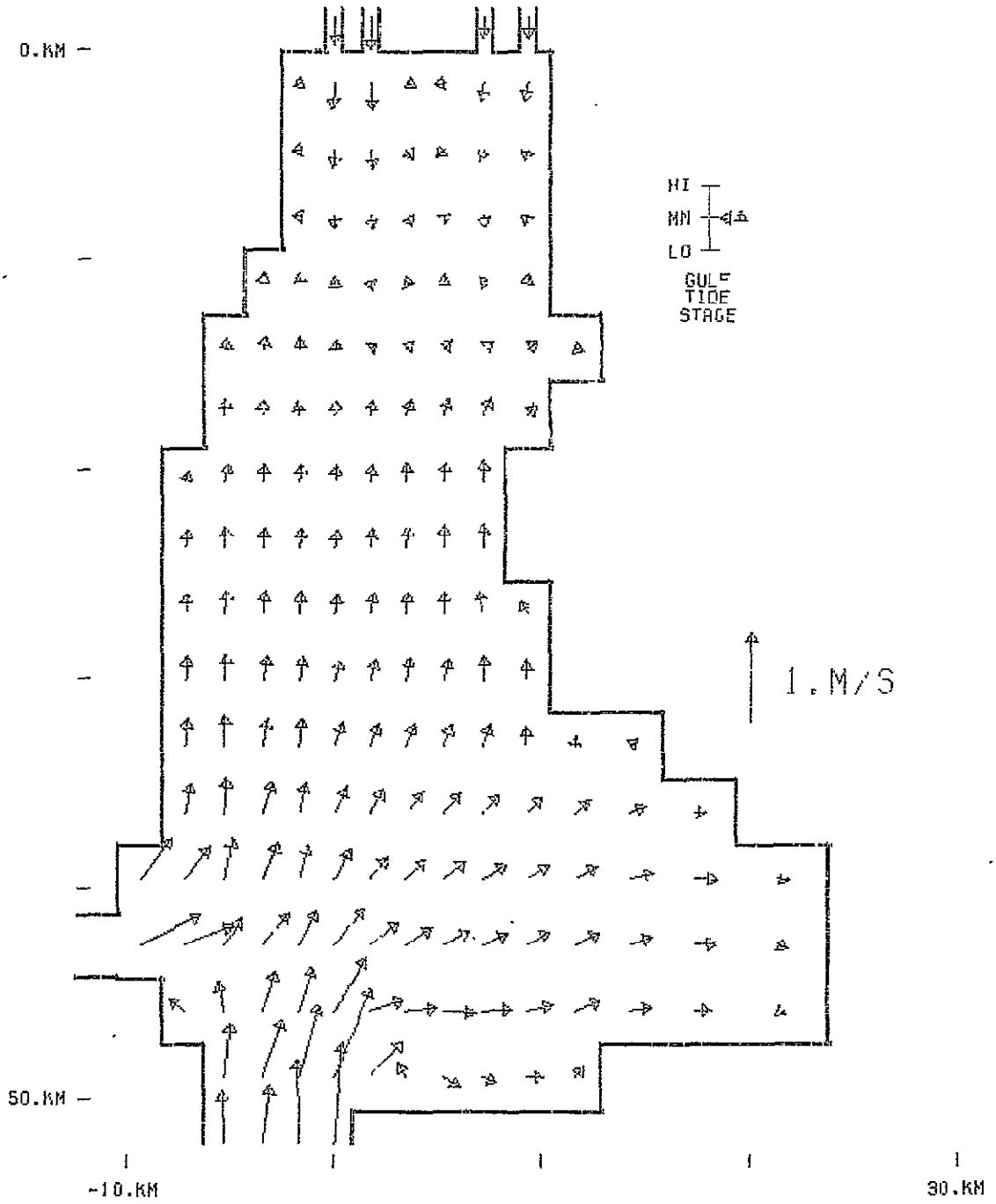


Figure A.11 - Case 1

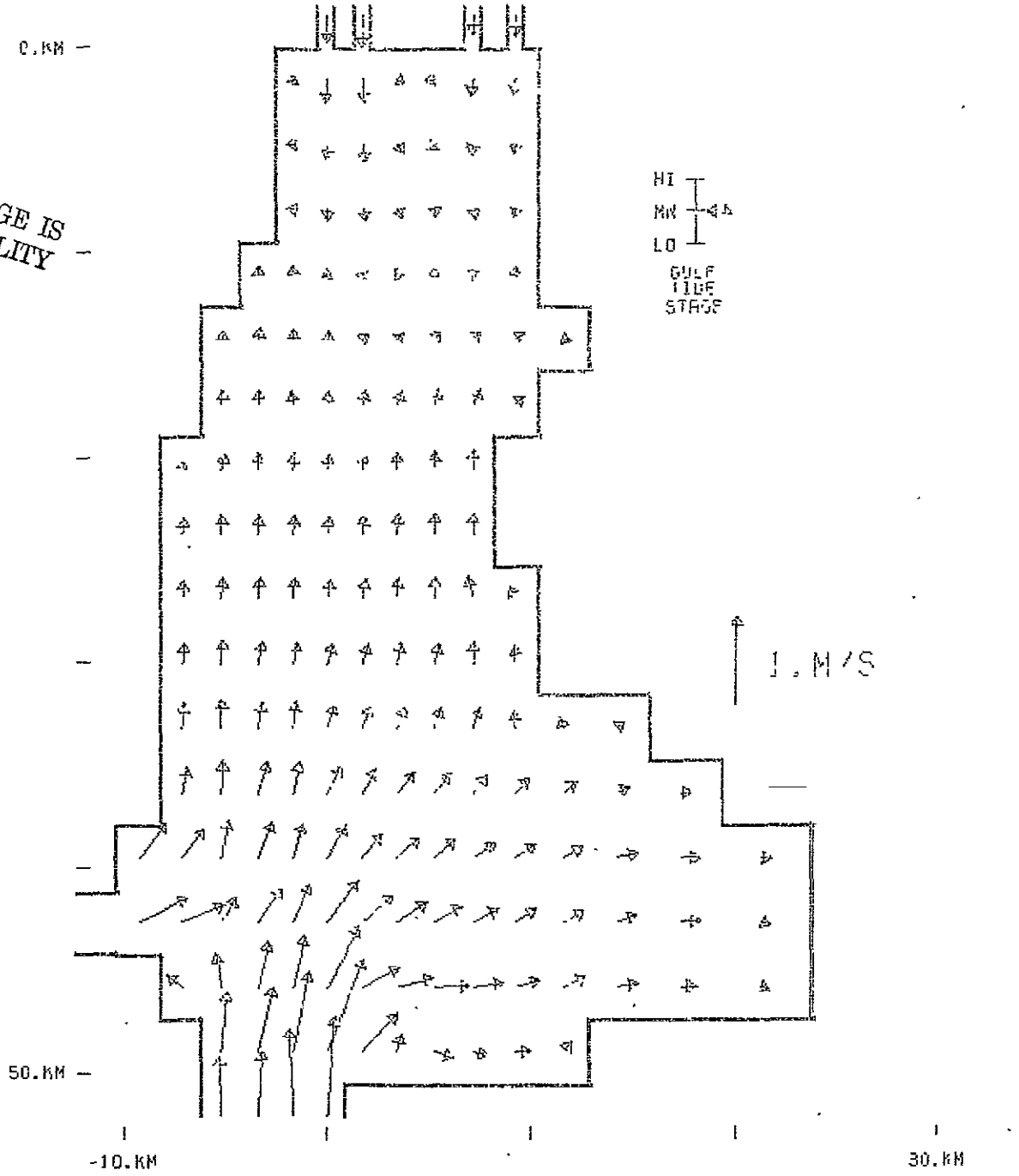


Figure A.12 -

Case 1

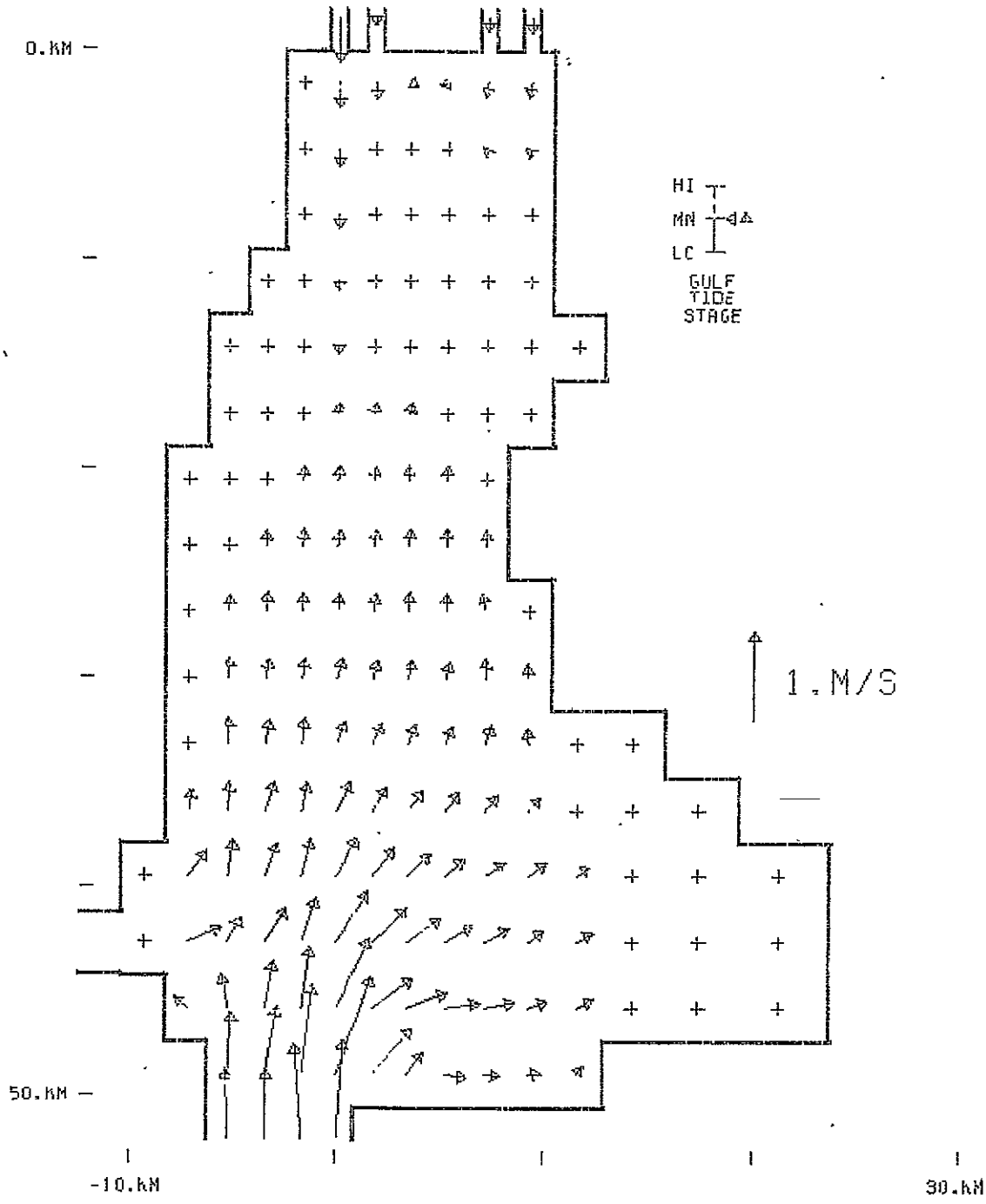


Figure A.13 -

Case 1

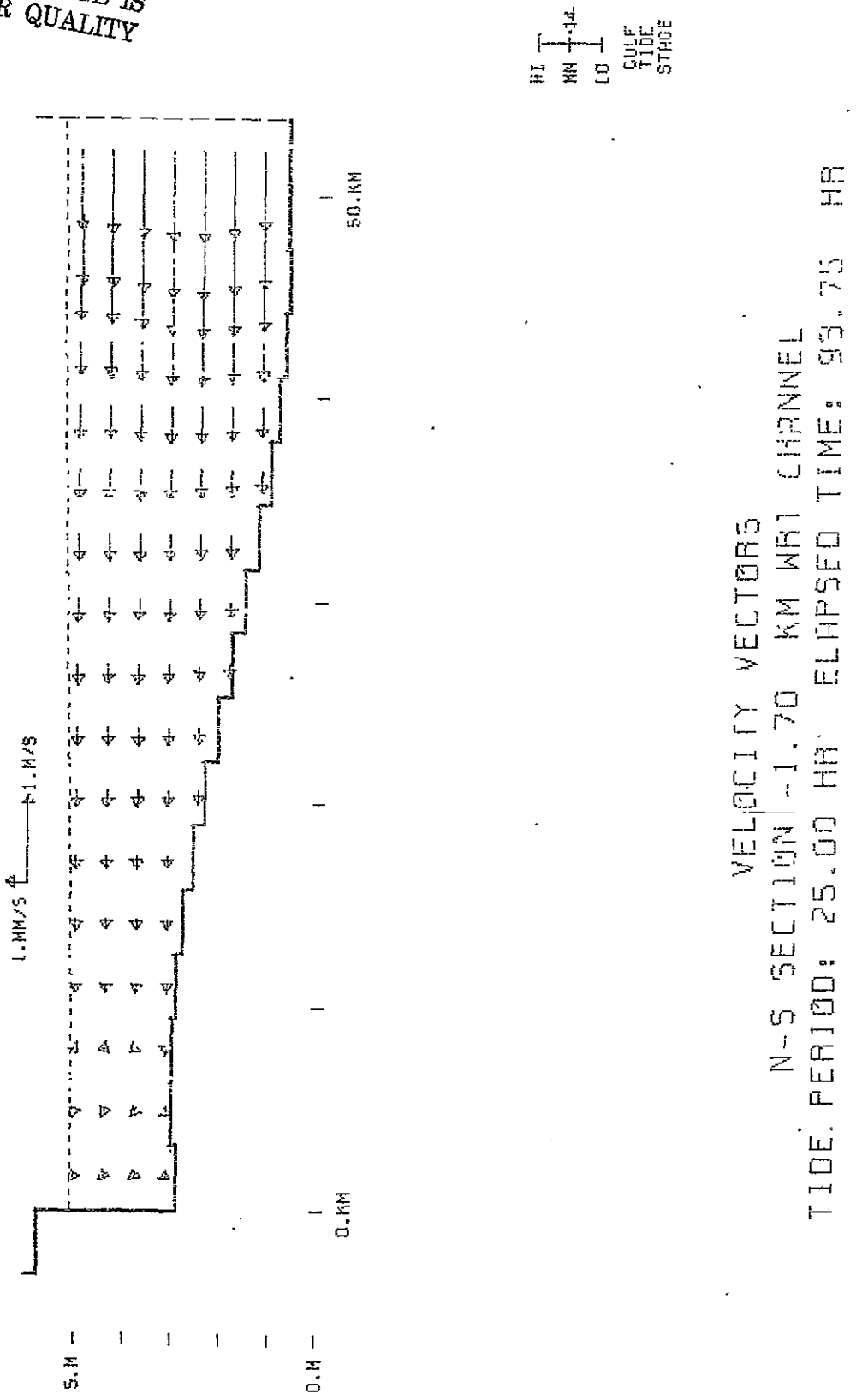


Figure A.14 - Case 1

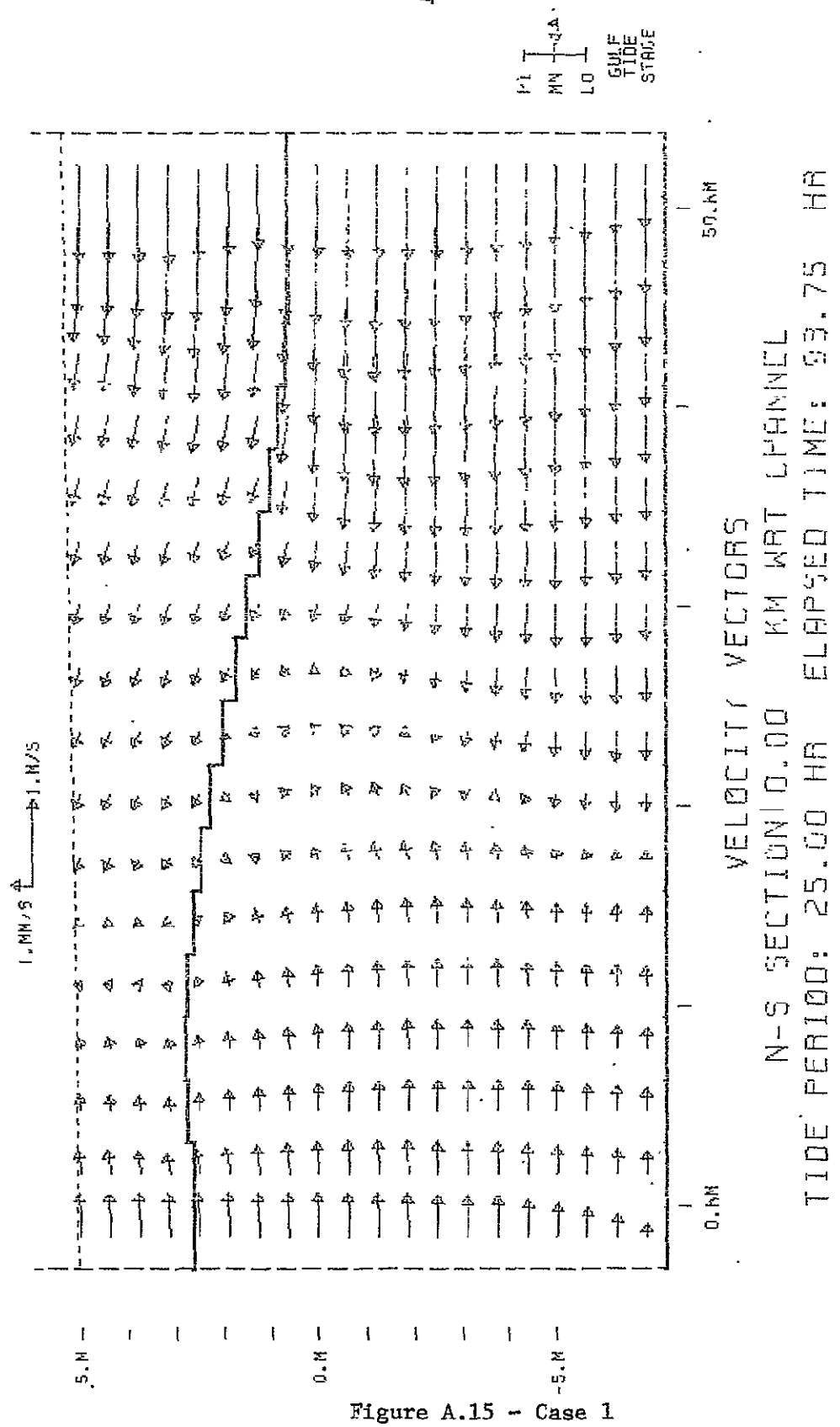


Figure A.15 - Case 1

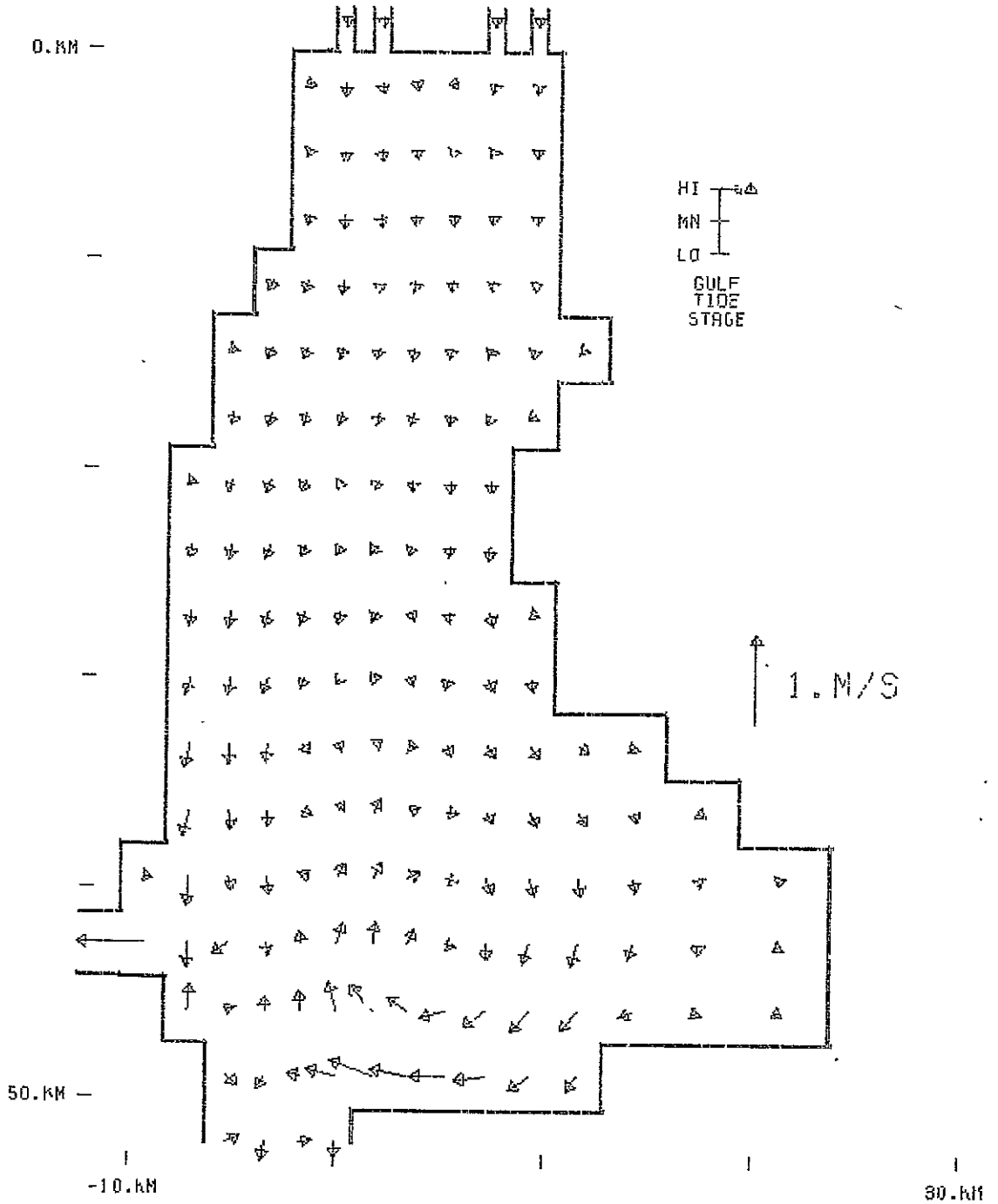


Figure A.16 -

Case 1

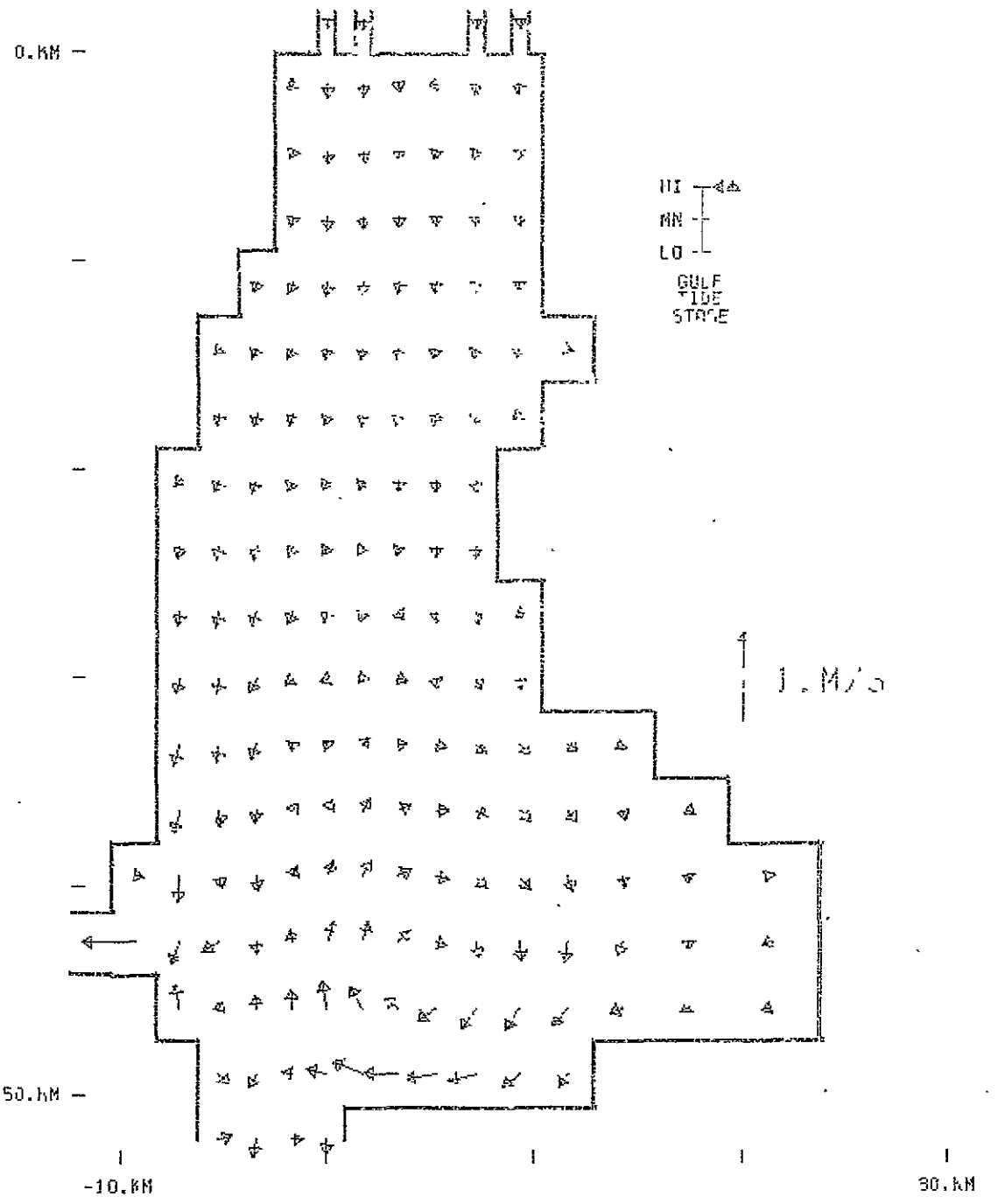


Figure A.17 -

Case 1

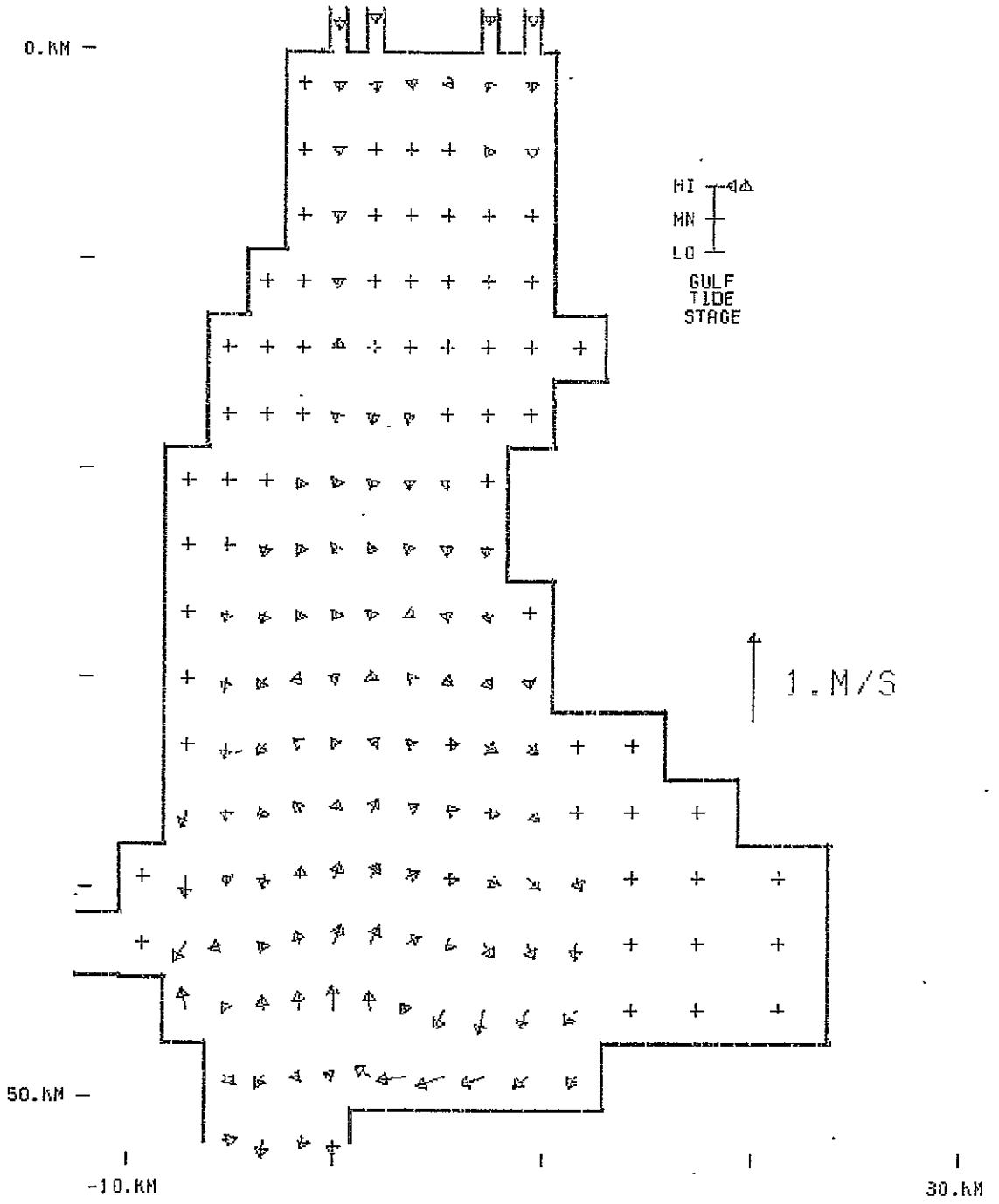


Figure A.18 -

Case 1

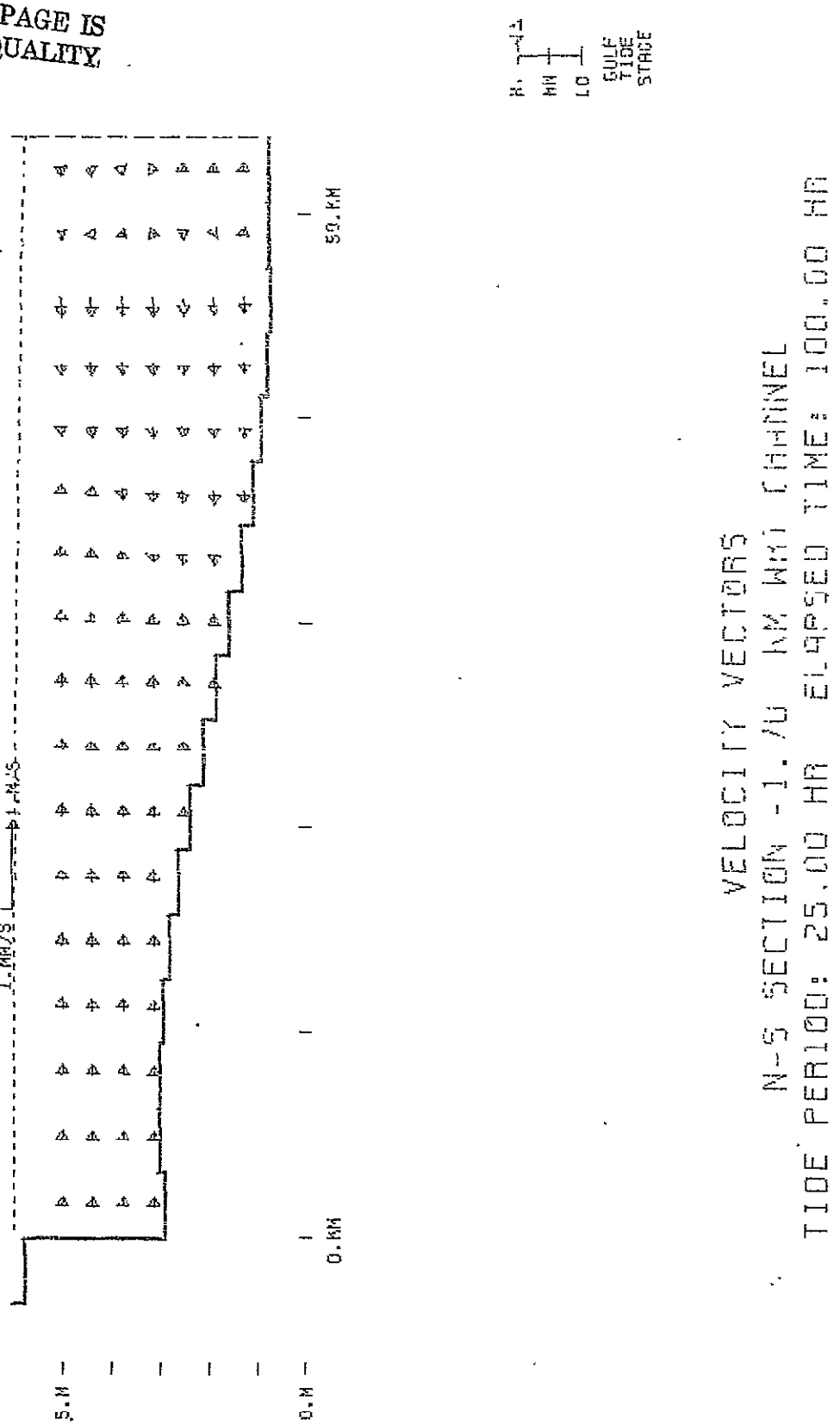


Figure A.19 - Case 1

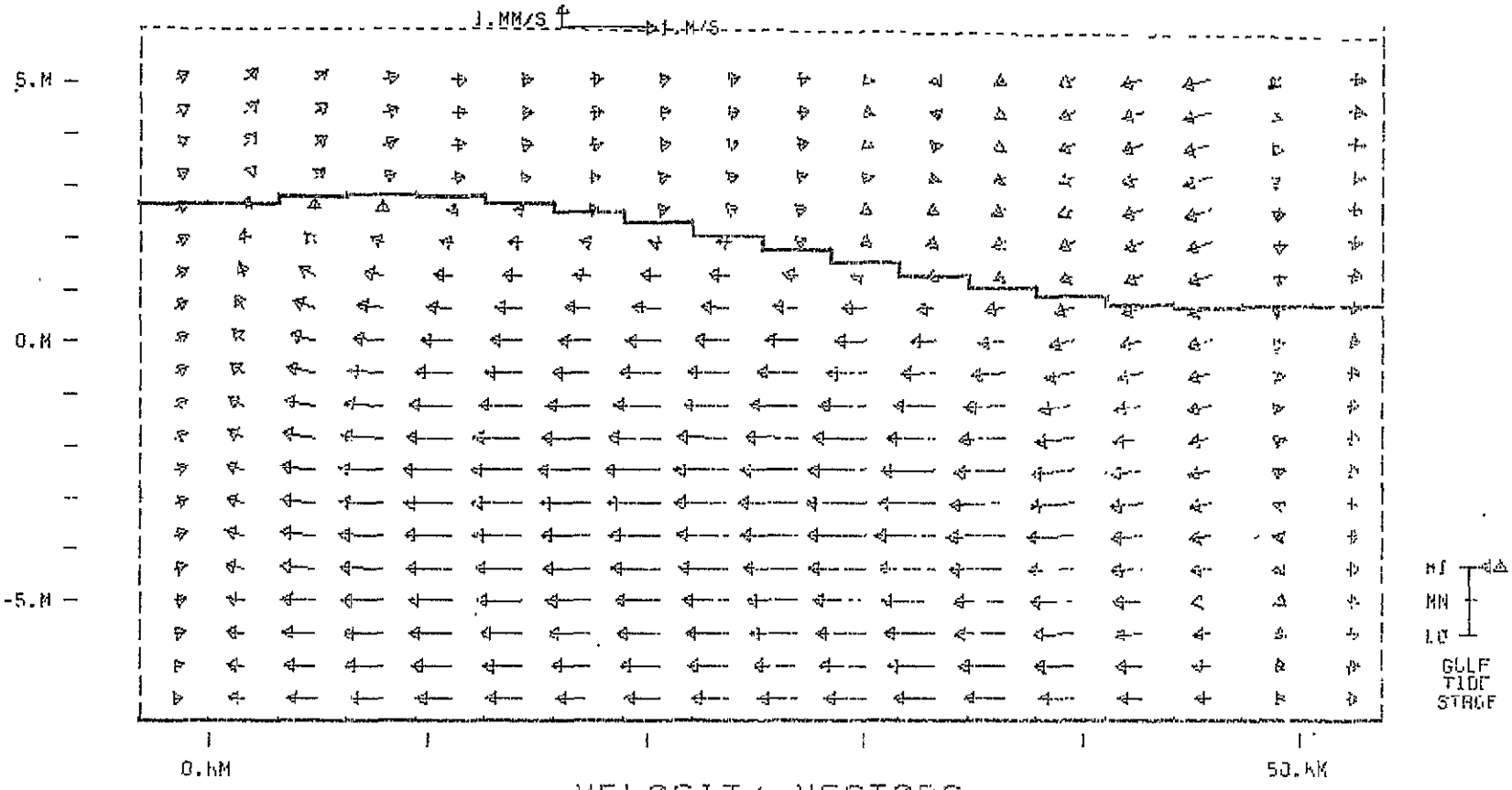


Figure A.20 - Case 1

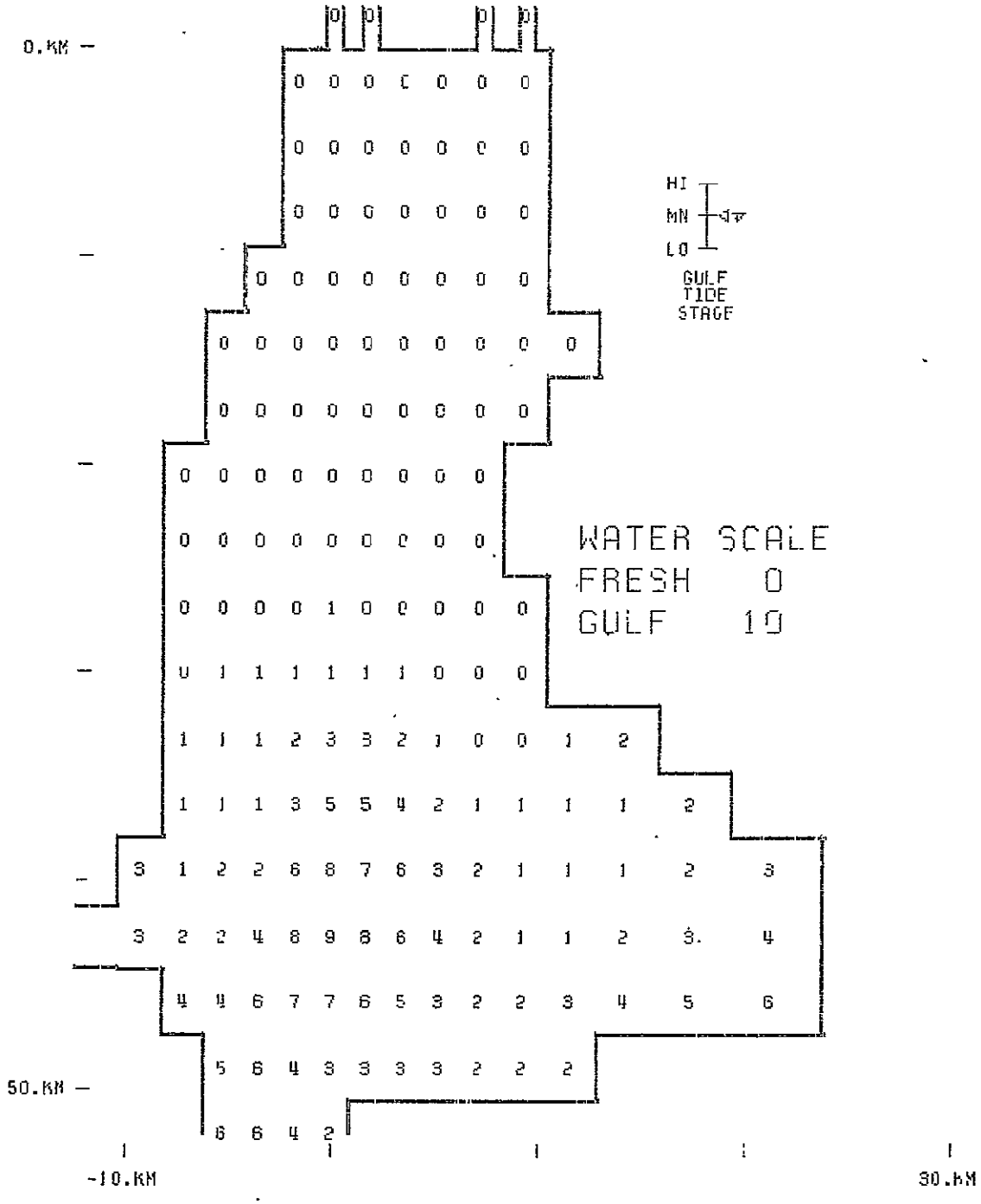


Figure A.21 -

Case 1

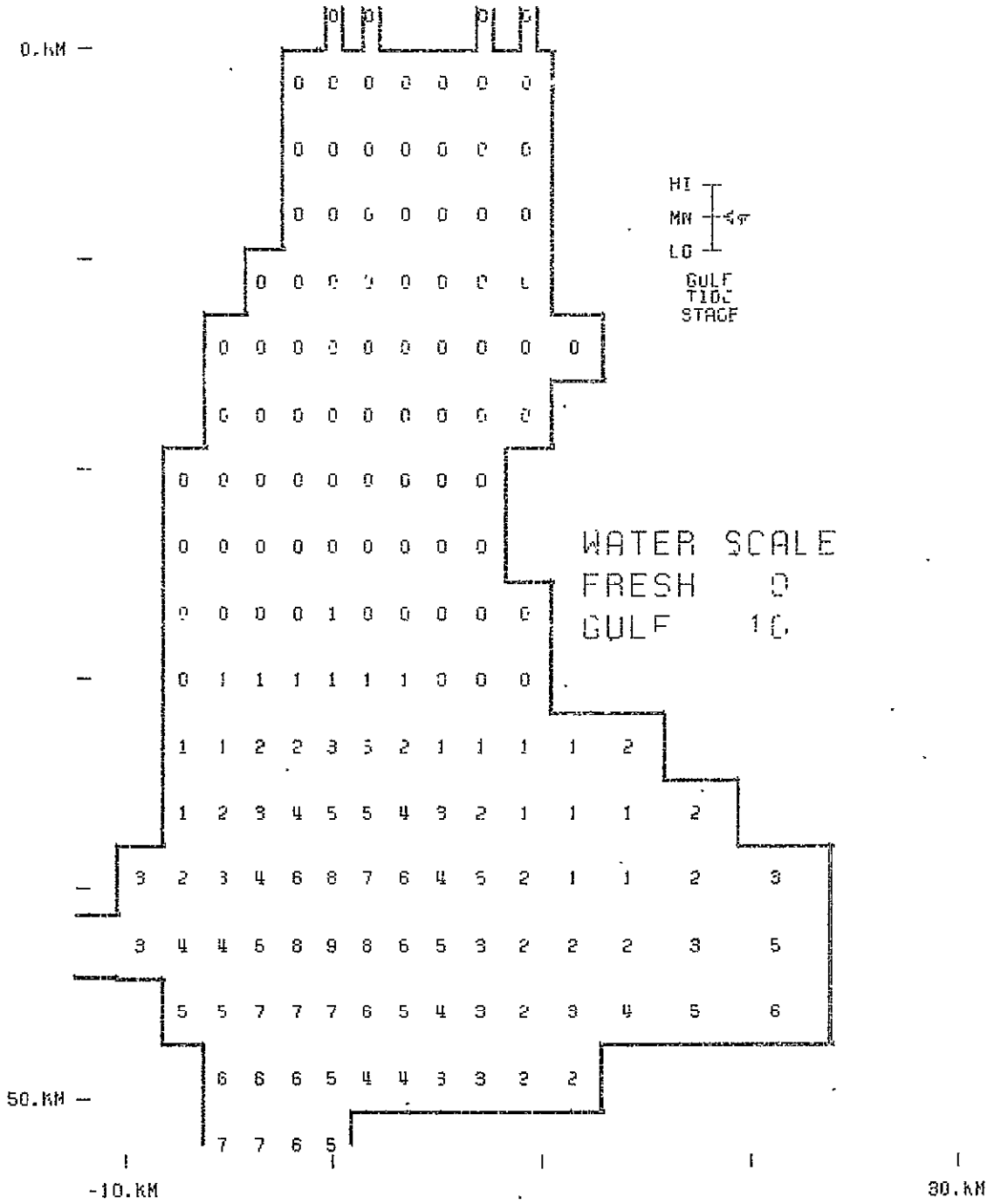


Figure A.22 -

Case 1

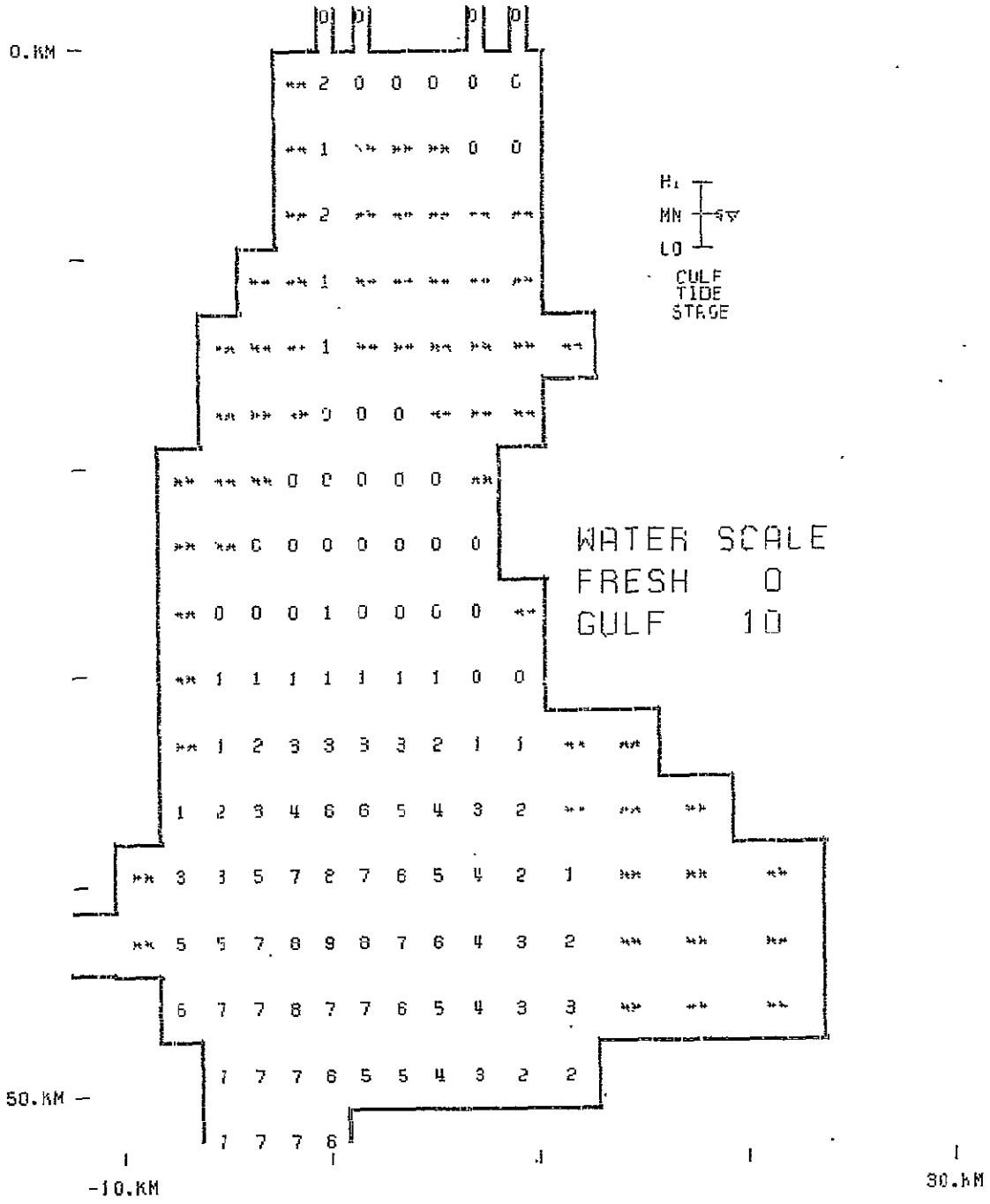


Figure A.23 -

Case 1

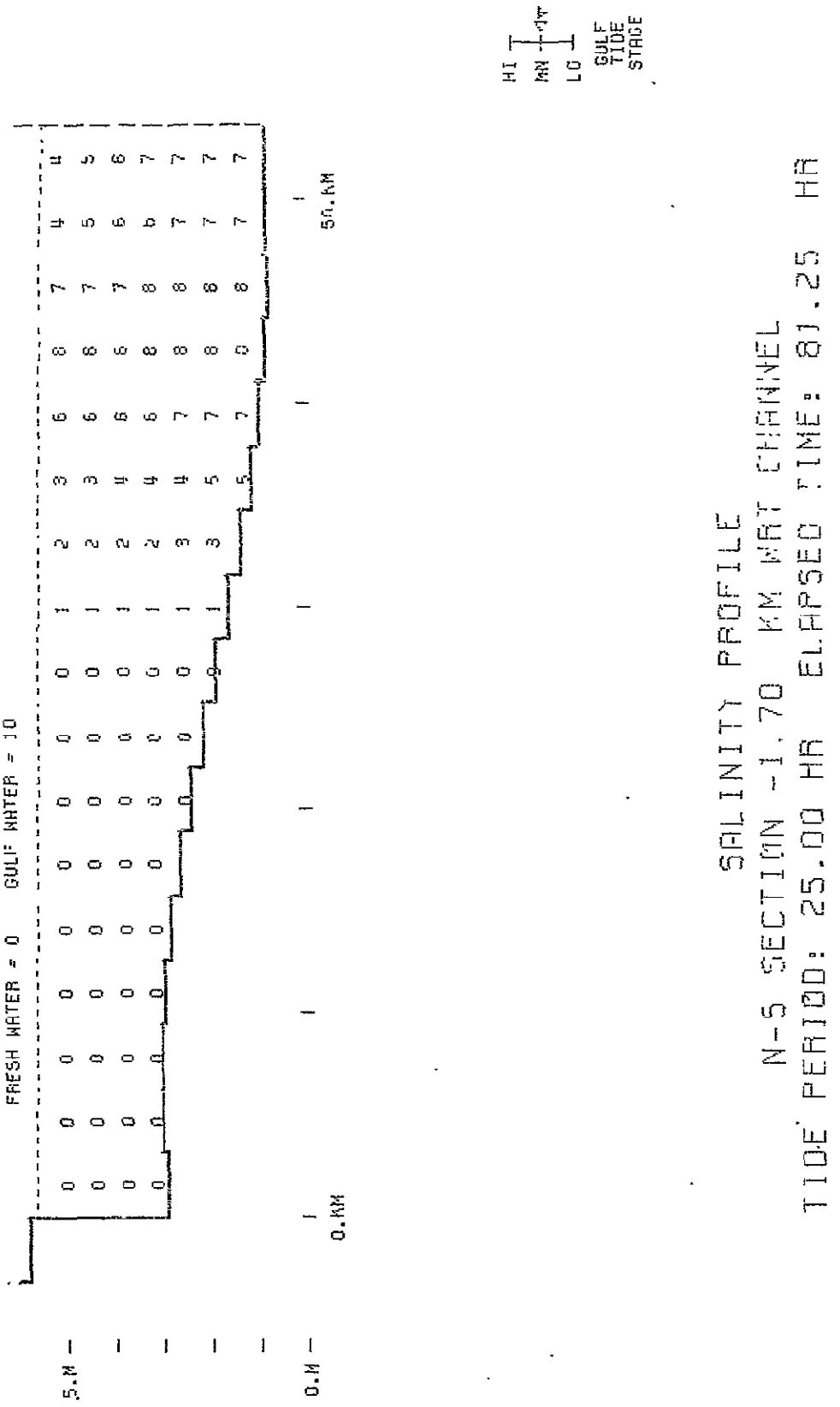


Figure A.24 - Case 1

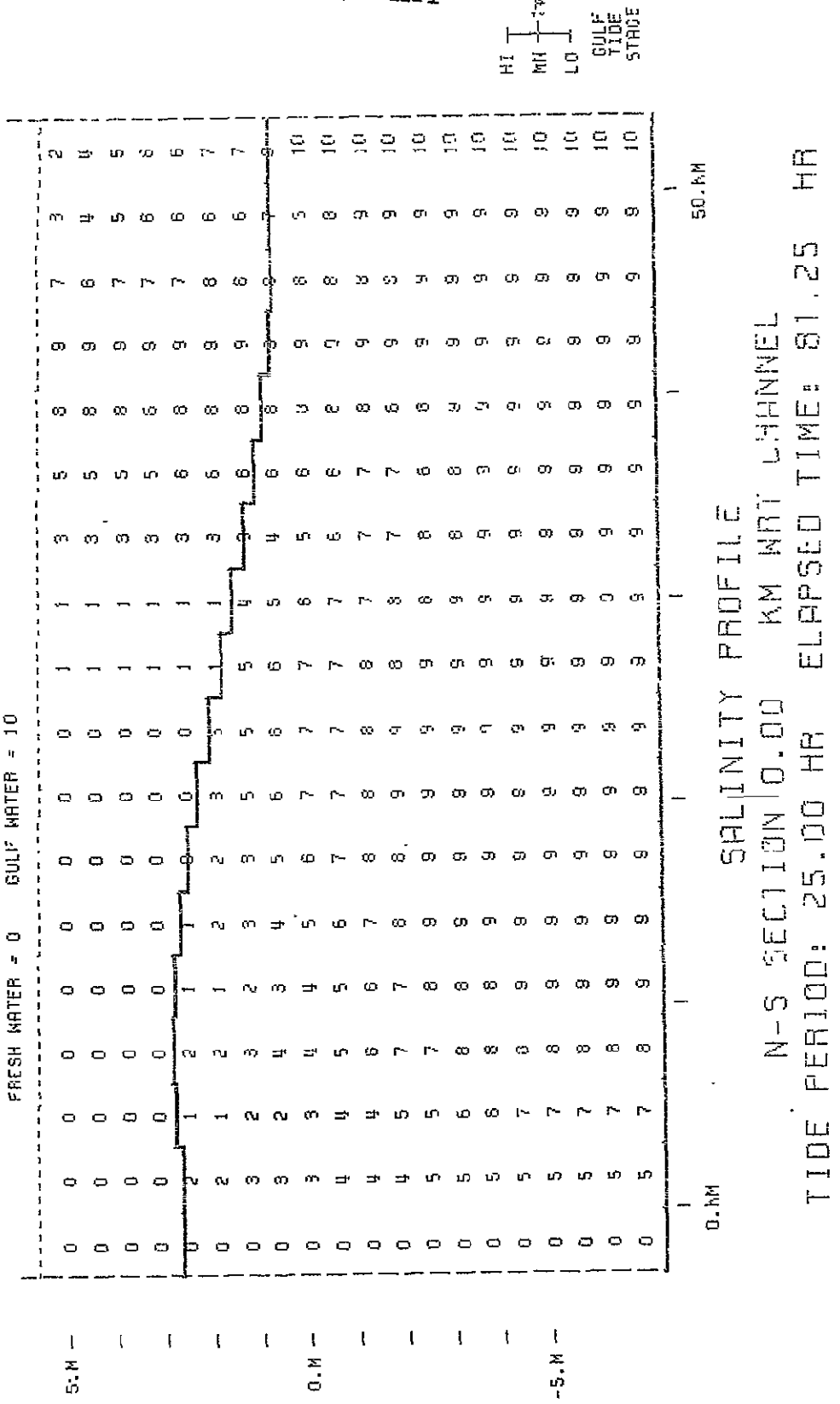


Figure A.25 - Case 1

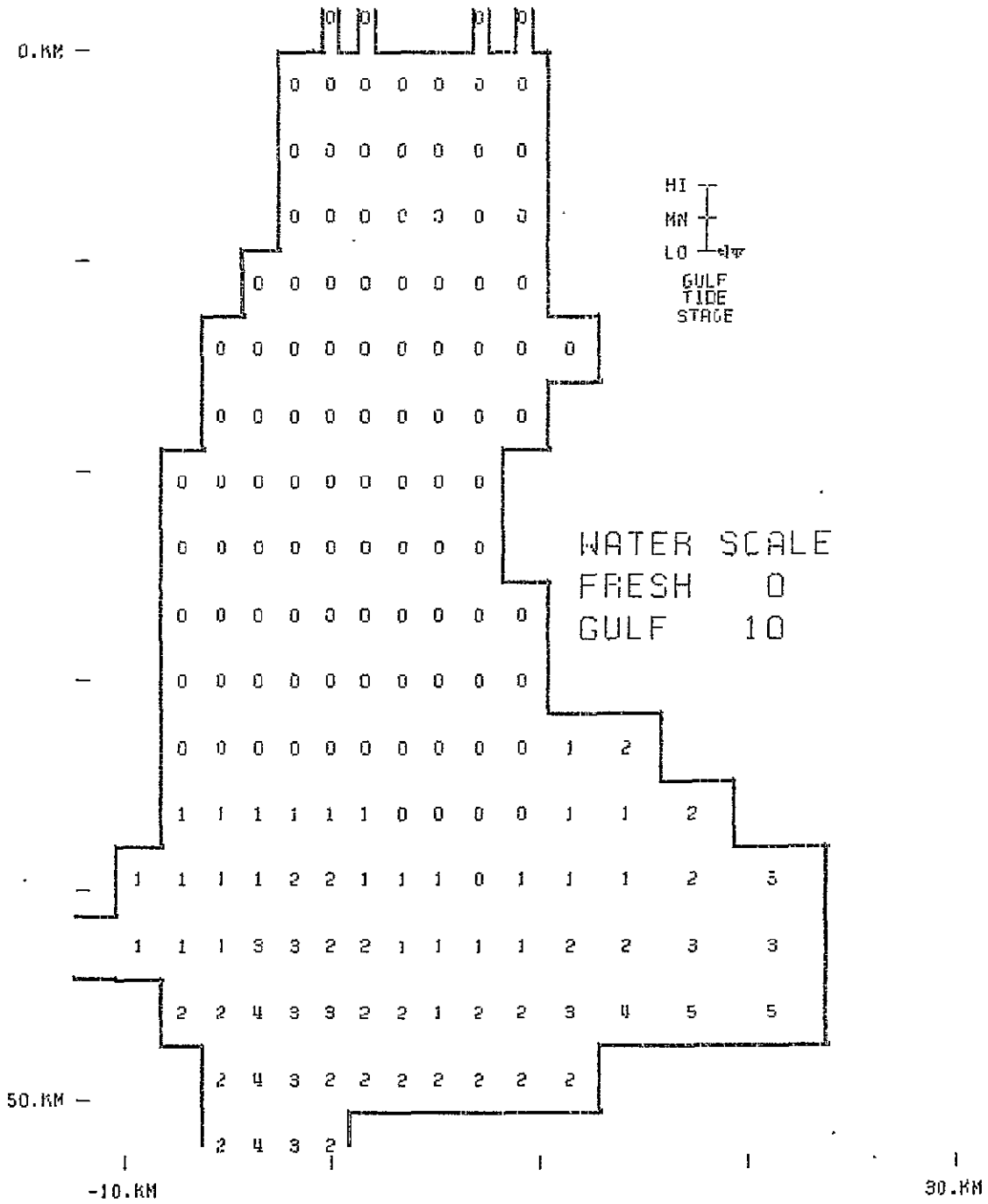


Figure A.26 -

Case 1

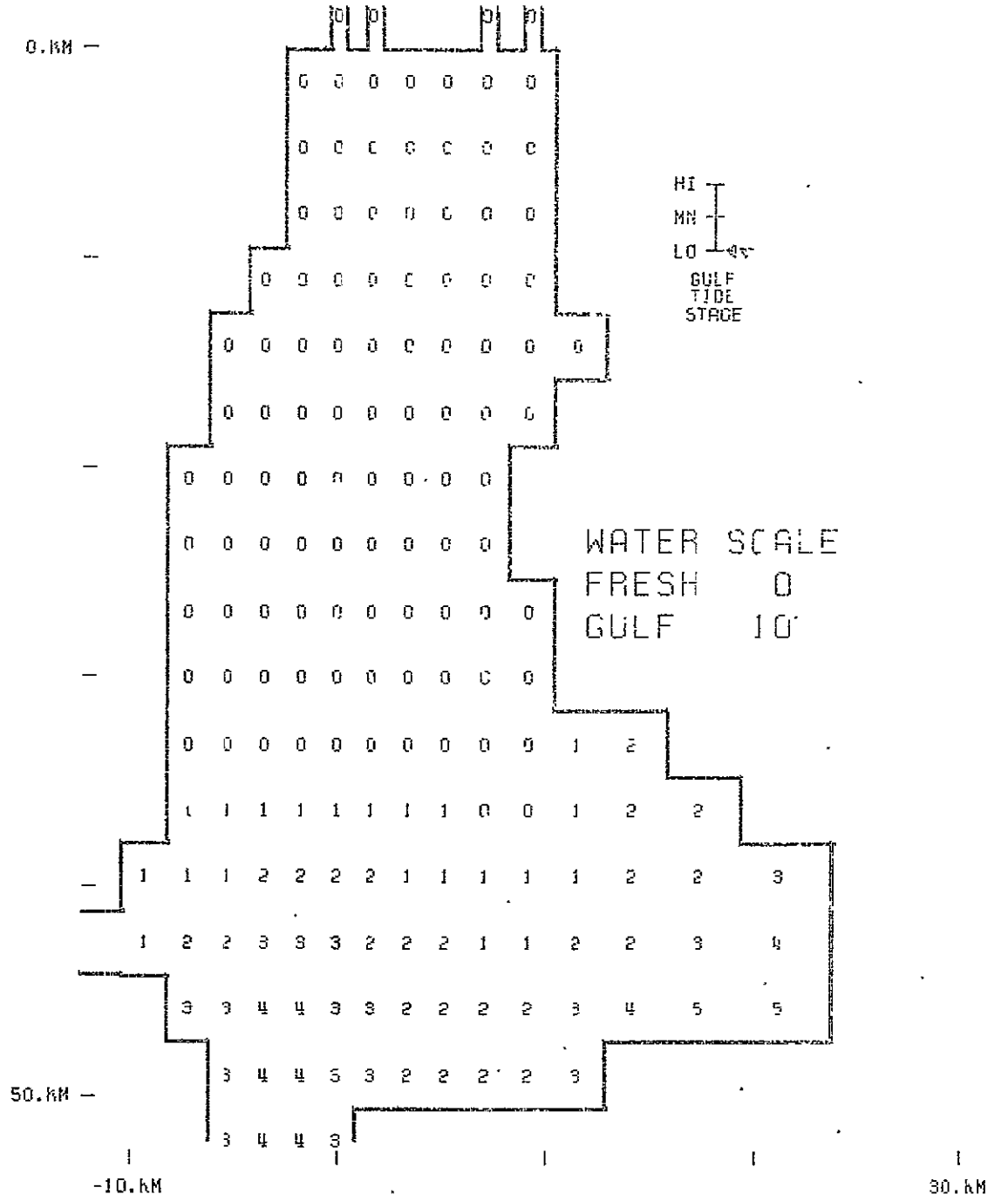


Figure A.27 -

Case 1

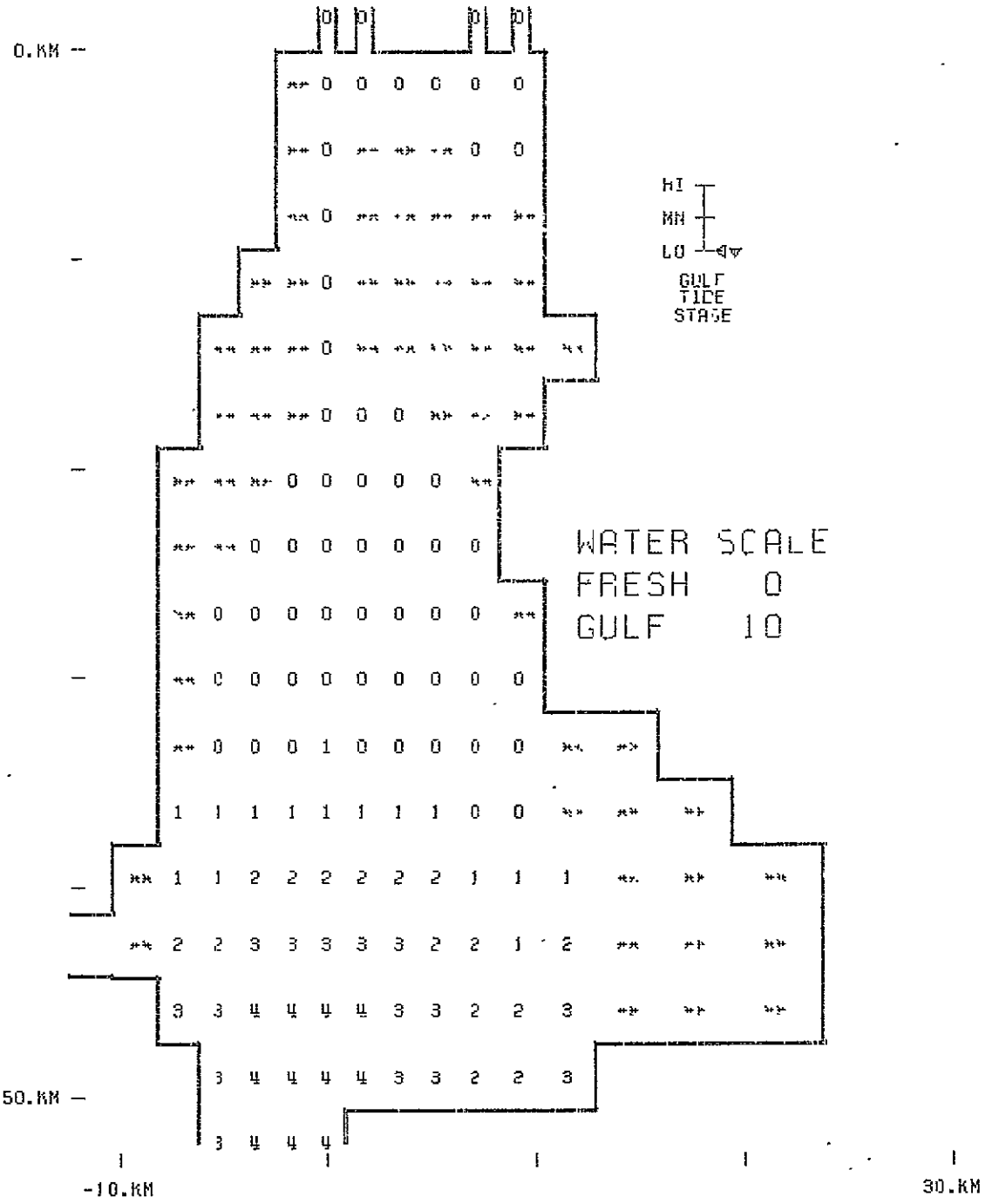


Figure A.28 - Case 1

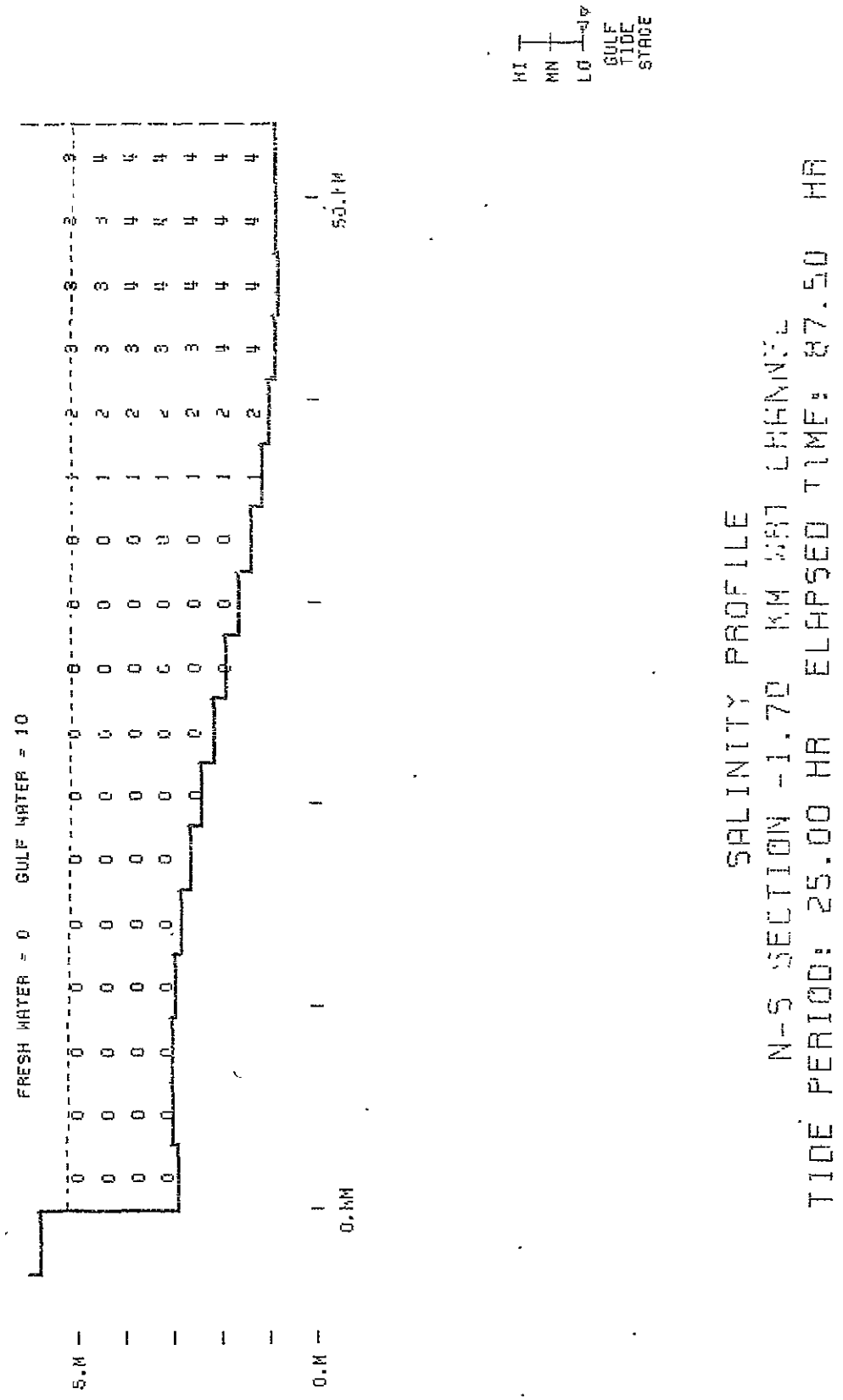


Figure A.29 - Case 1

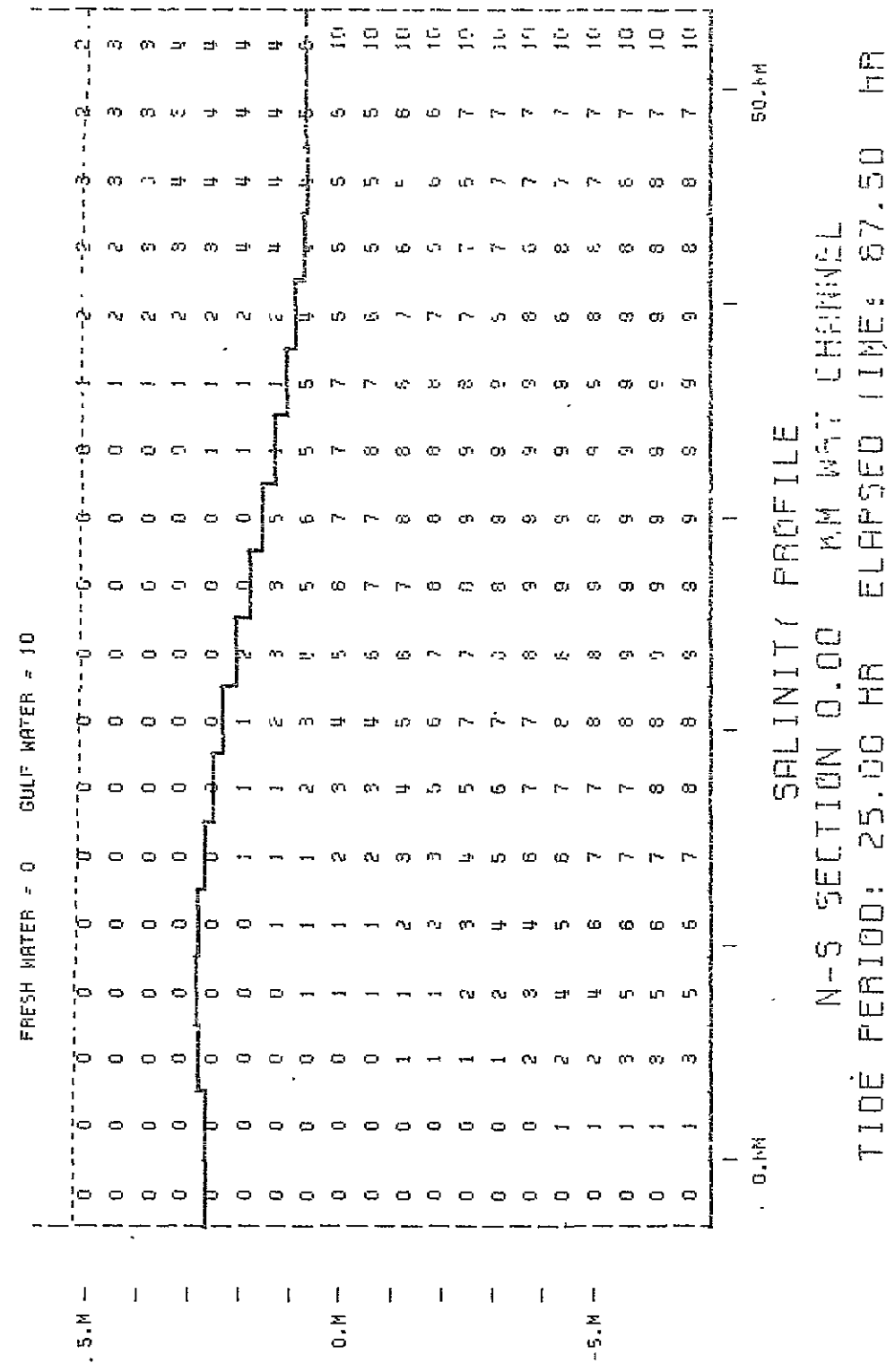


Figure A.30 - Case 1

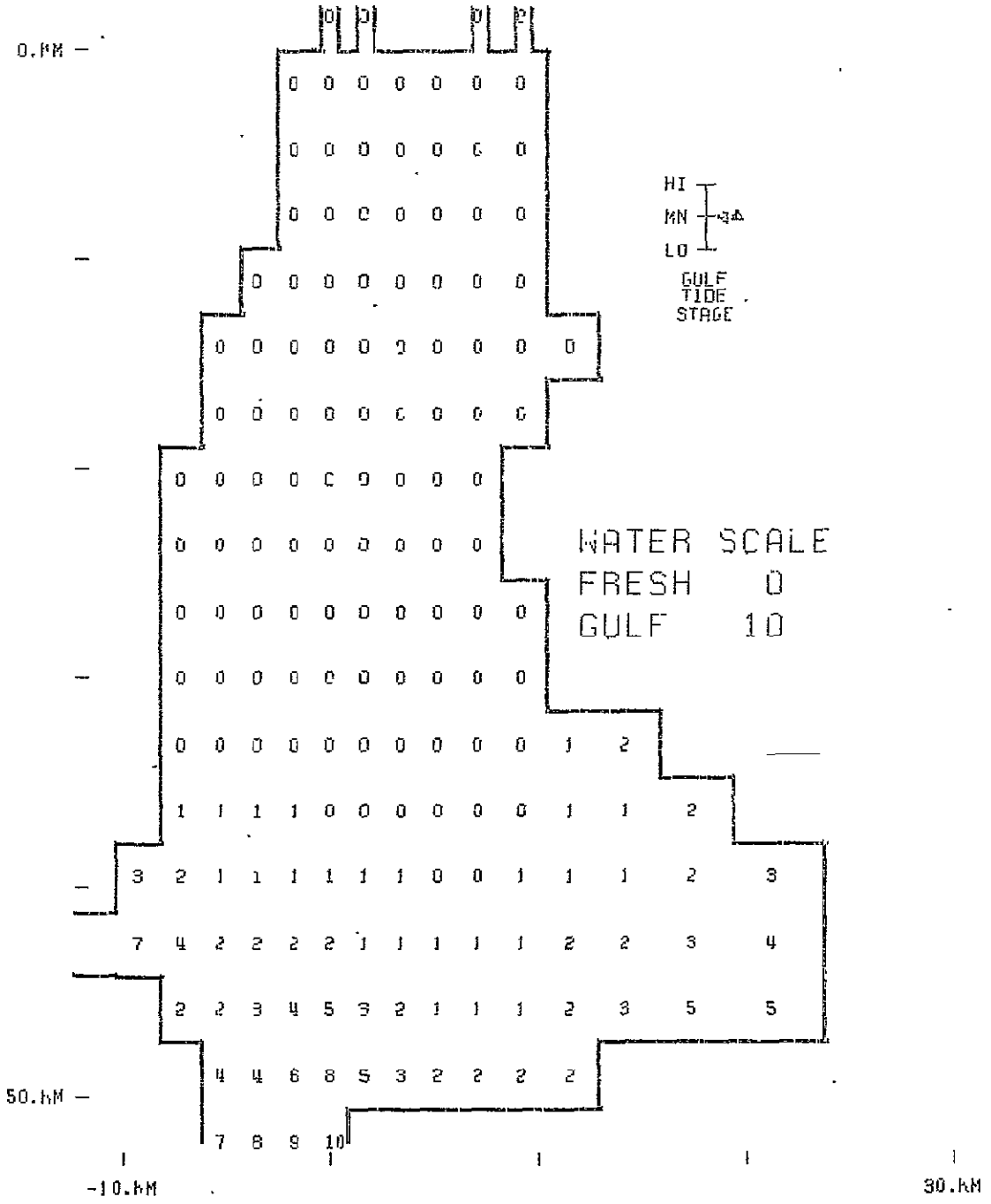


Figure A.31 -

Case 1

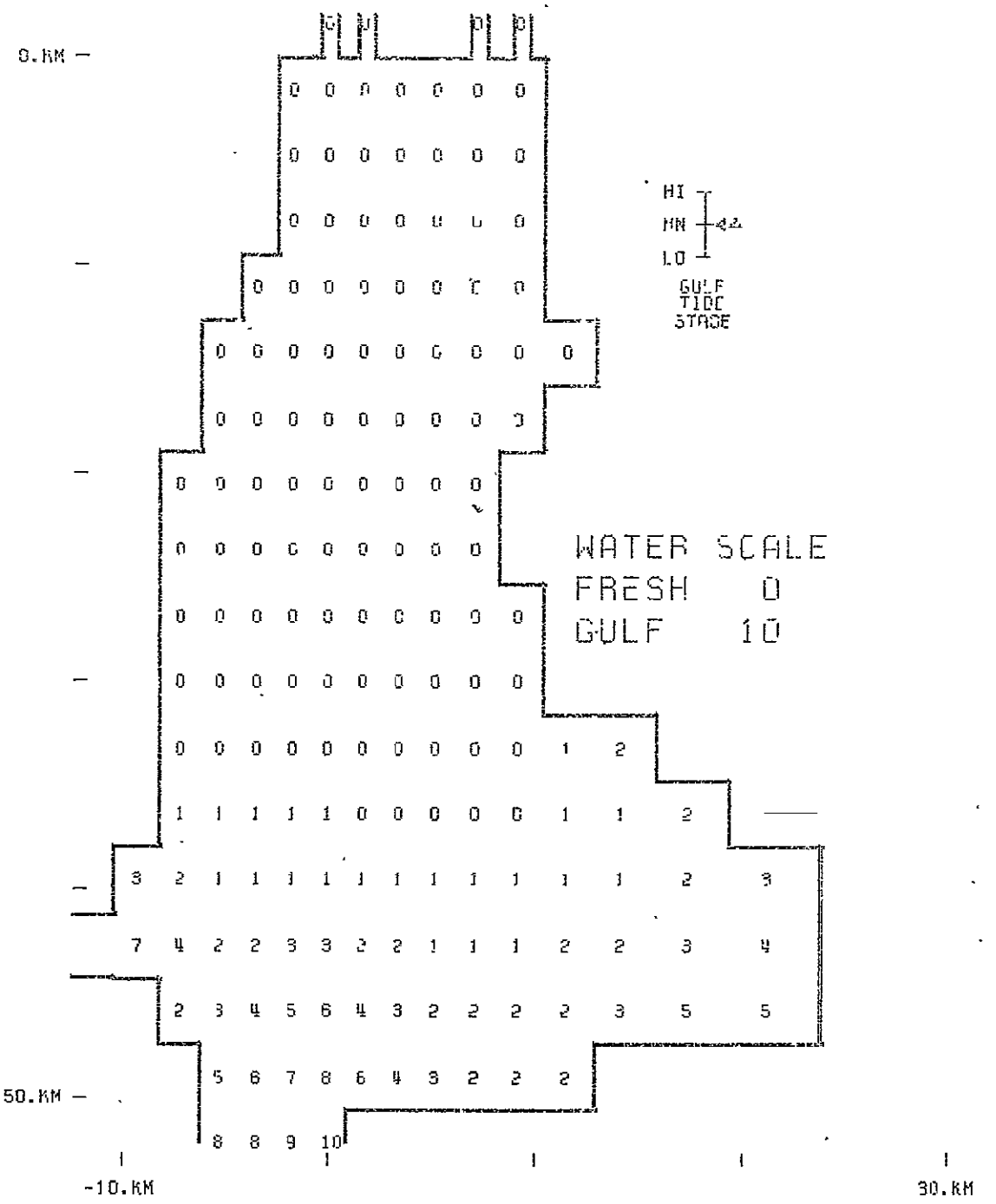


Figure A.32 - Case 1

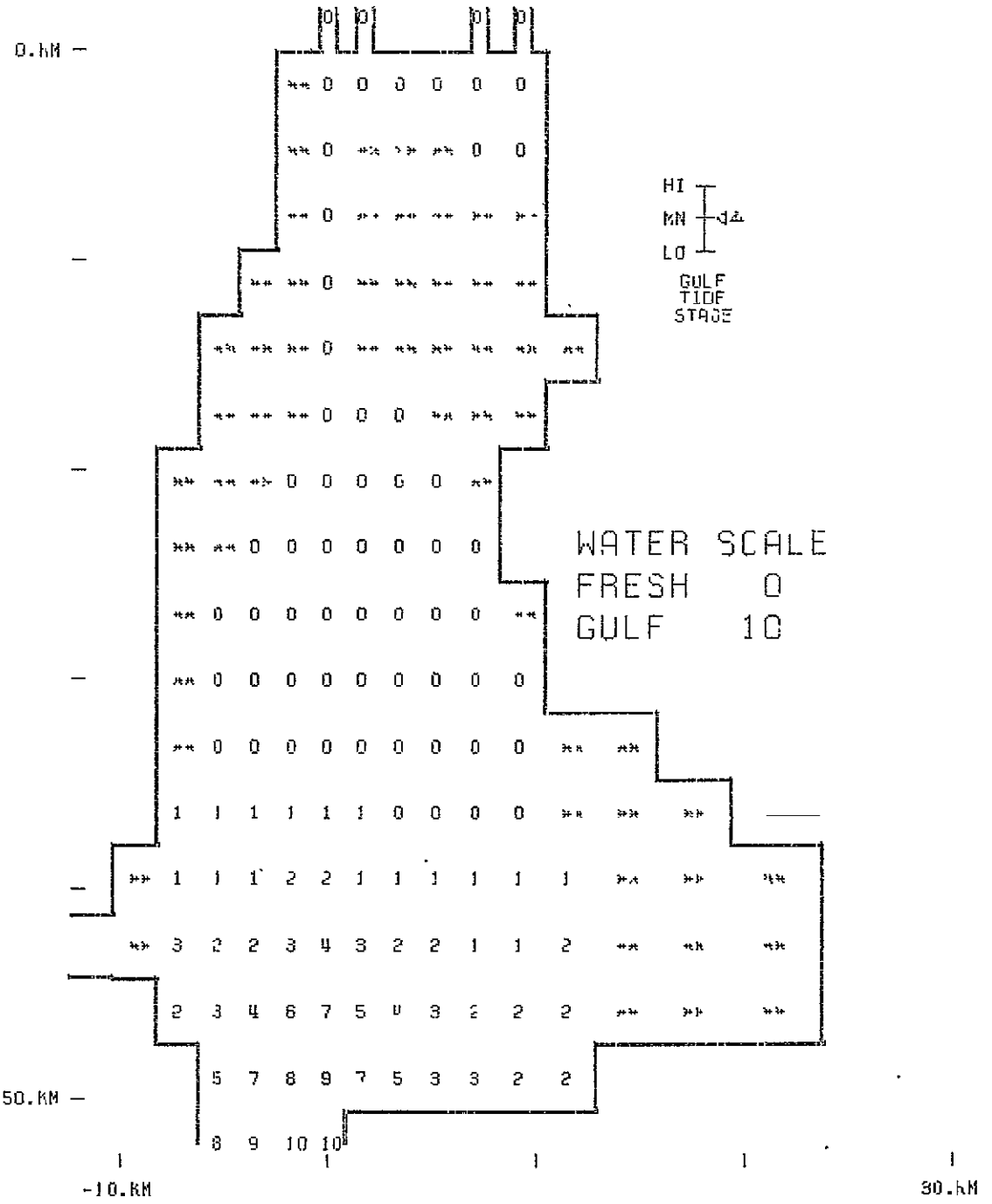


Figure A.33 -

Case 1

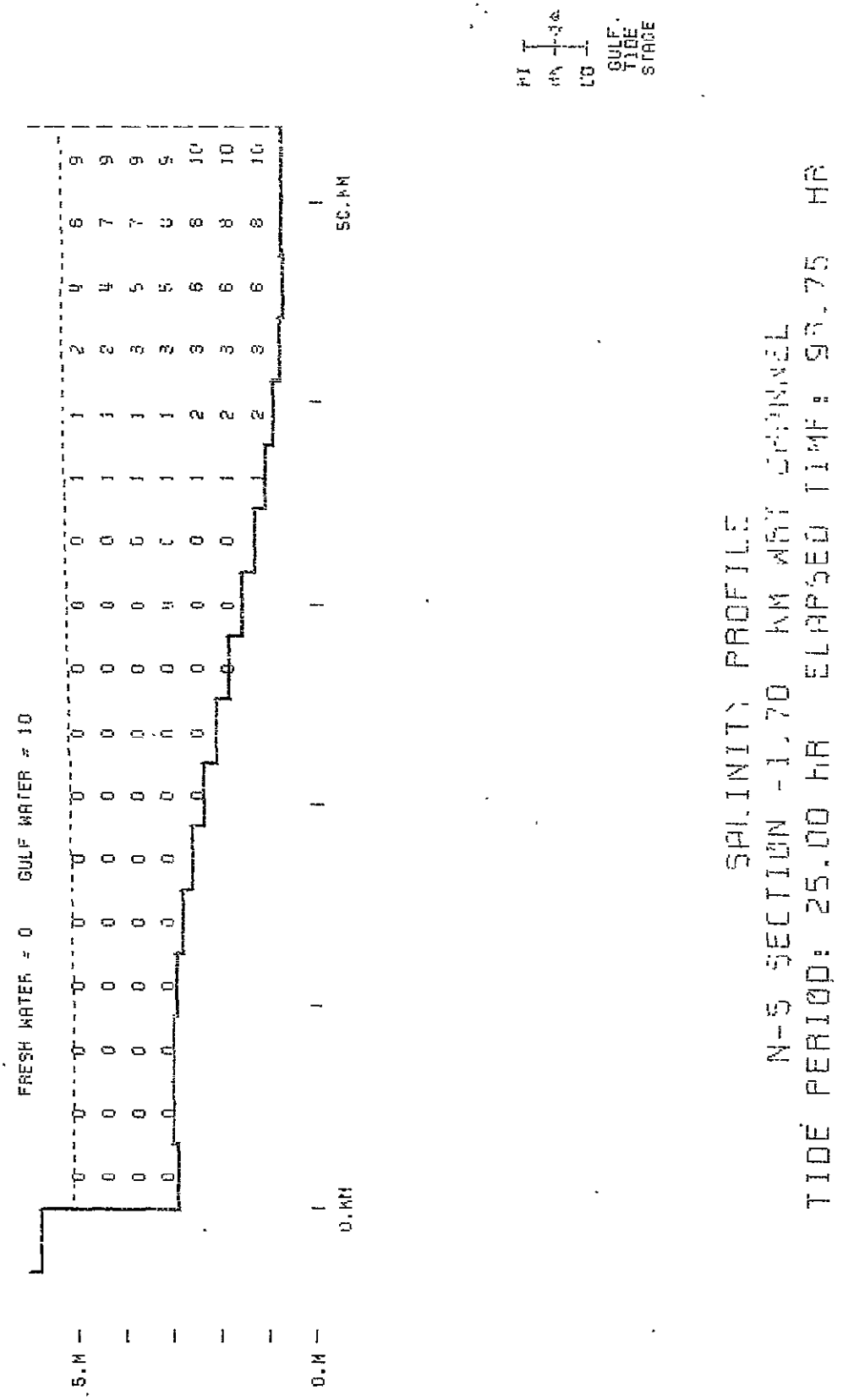


Figure A.34 - Case 1

ORIGINAL PAGE IS
OF POOR QUALITY

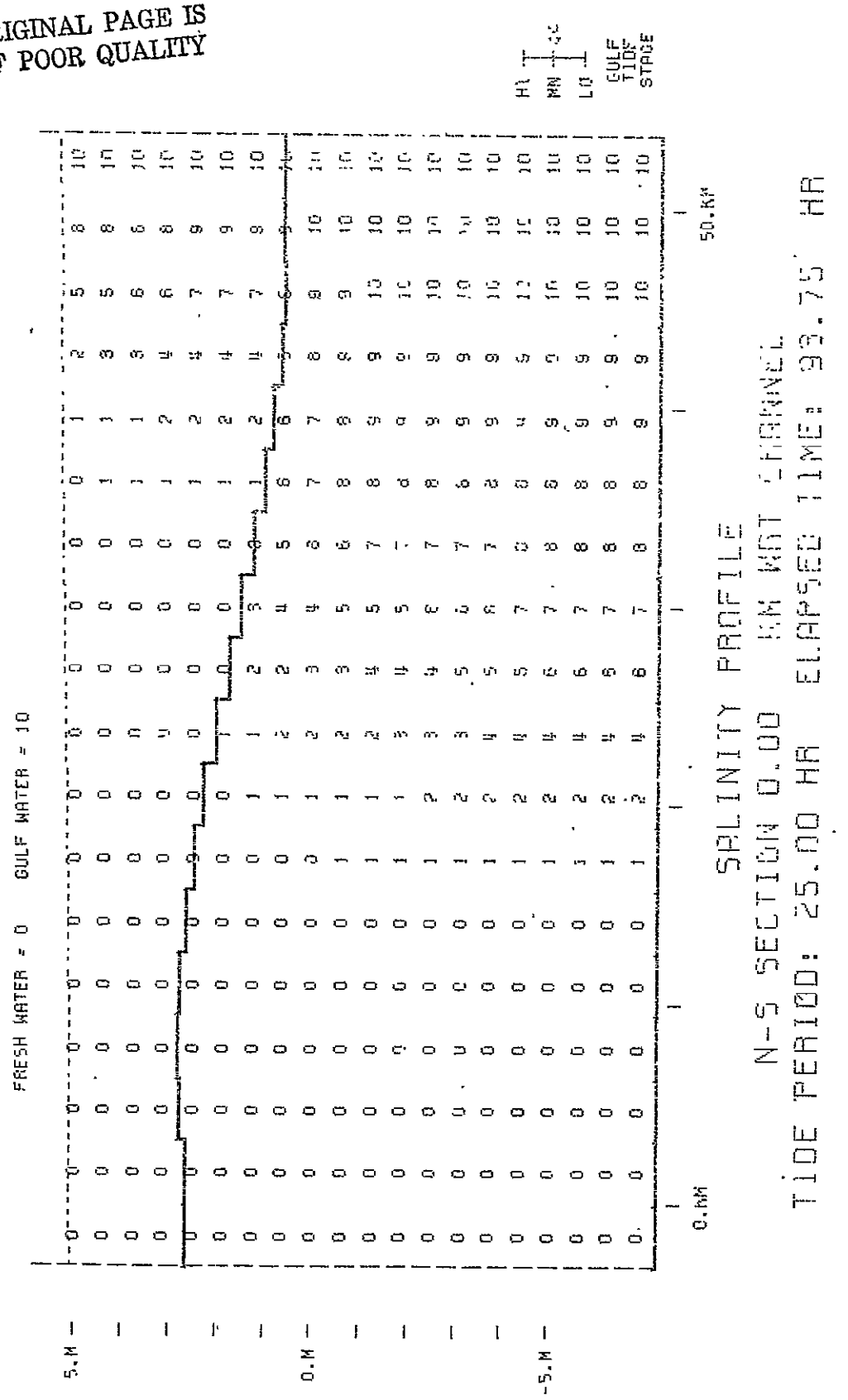


Figure A.35 - Case 1

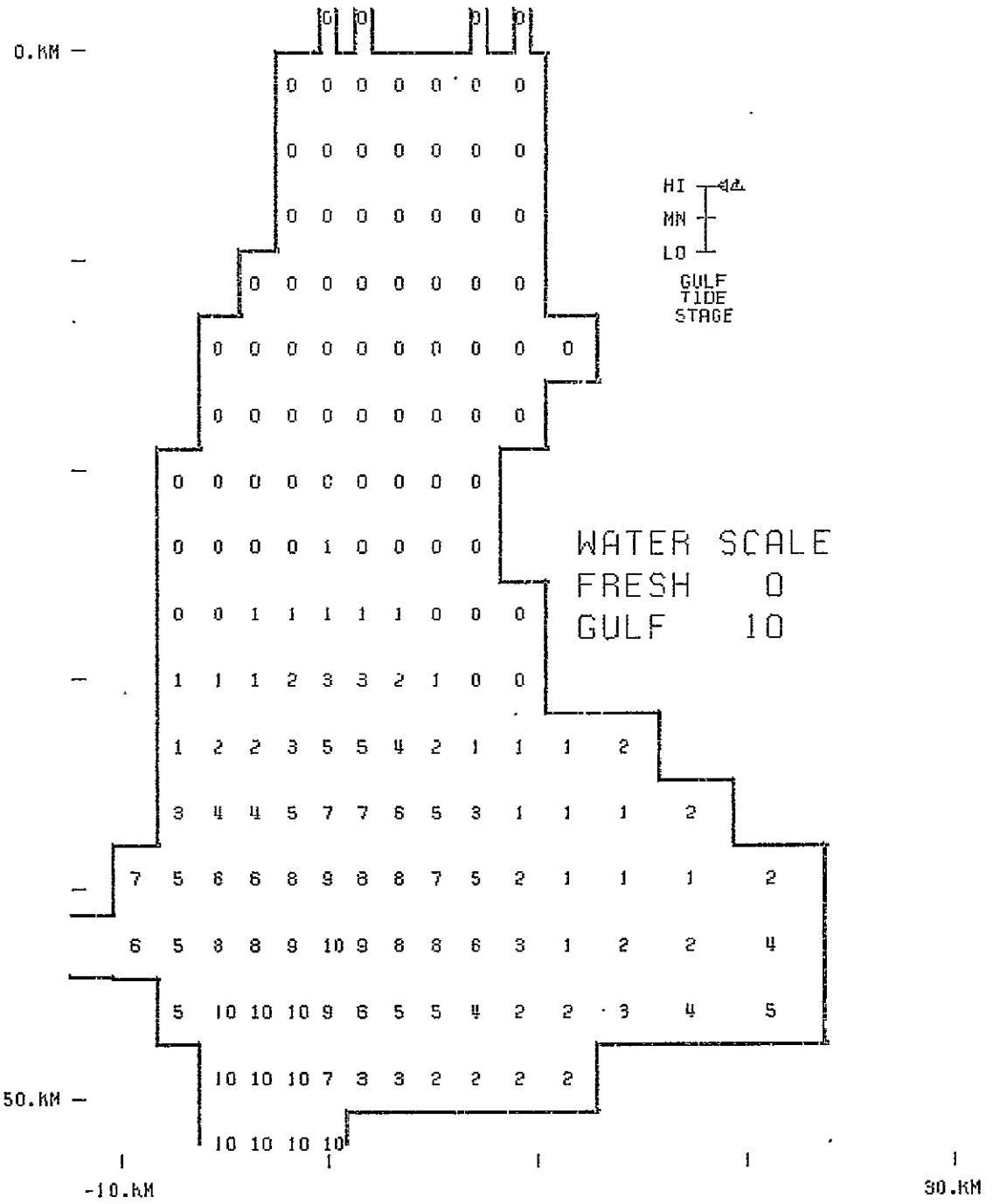


Figure A.36 -

Case 1

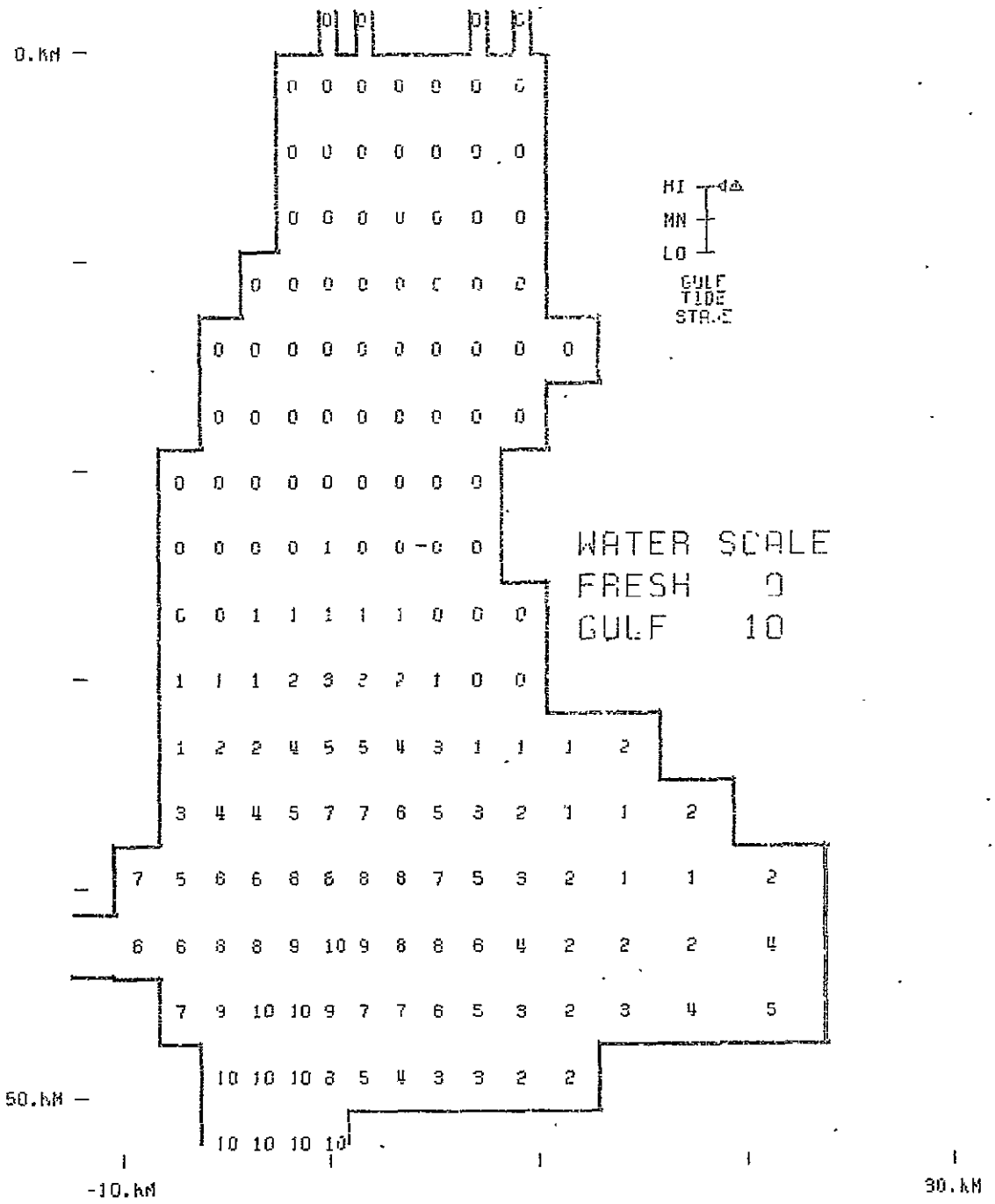


Figure A.37 - Case 1

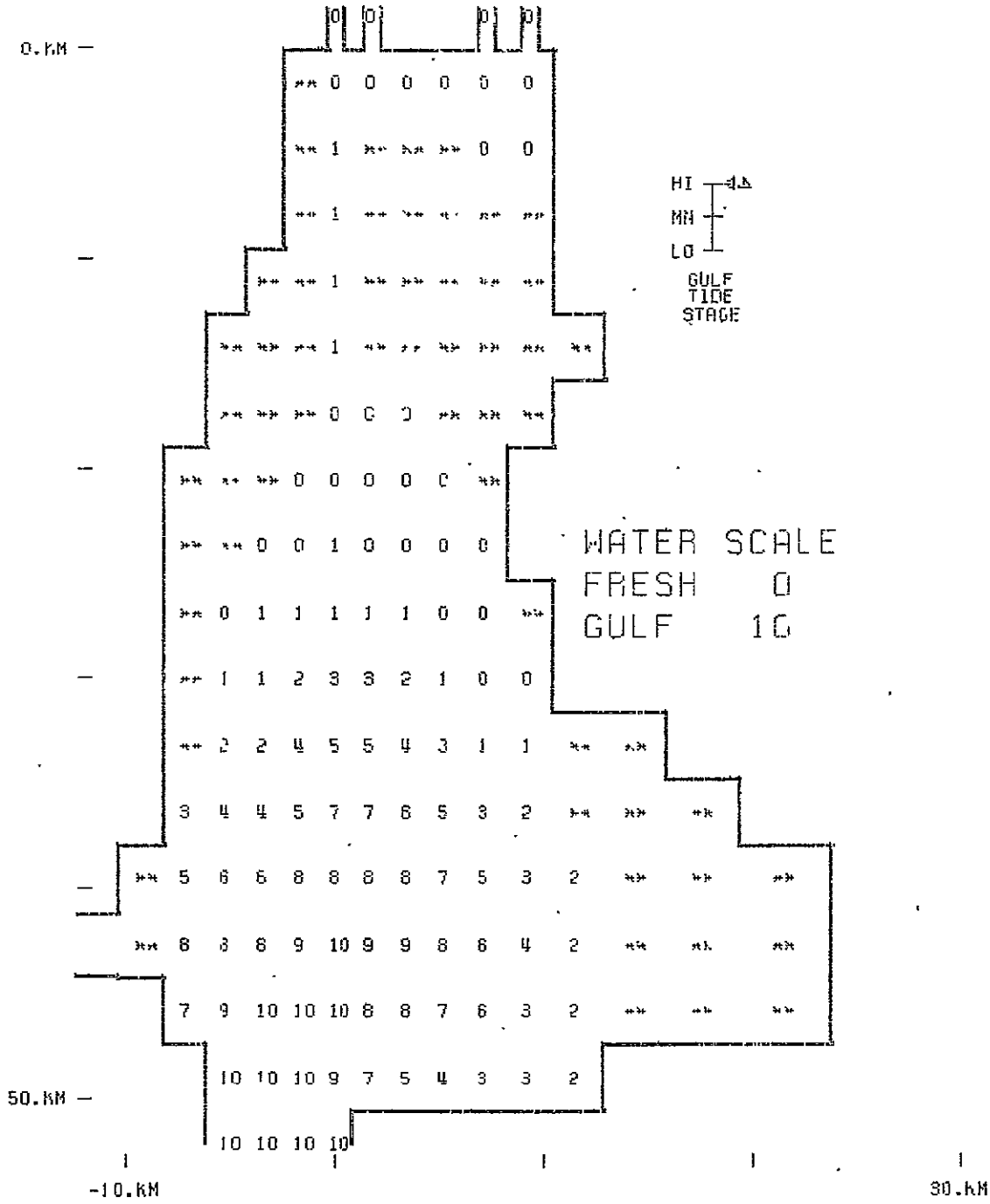


Figure A.38 - Case 1

ORIGINAL PAGE IS
OF POOR QUALITY

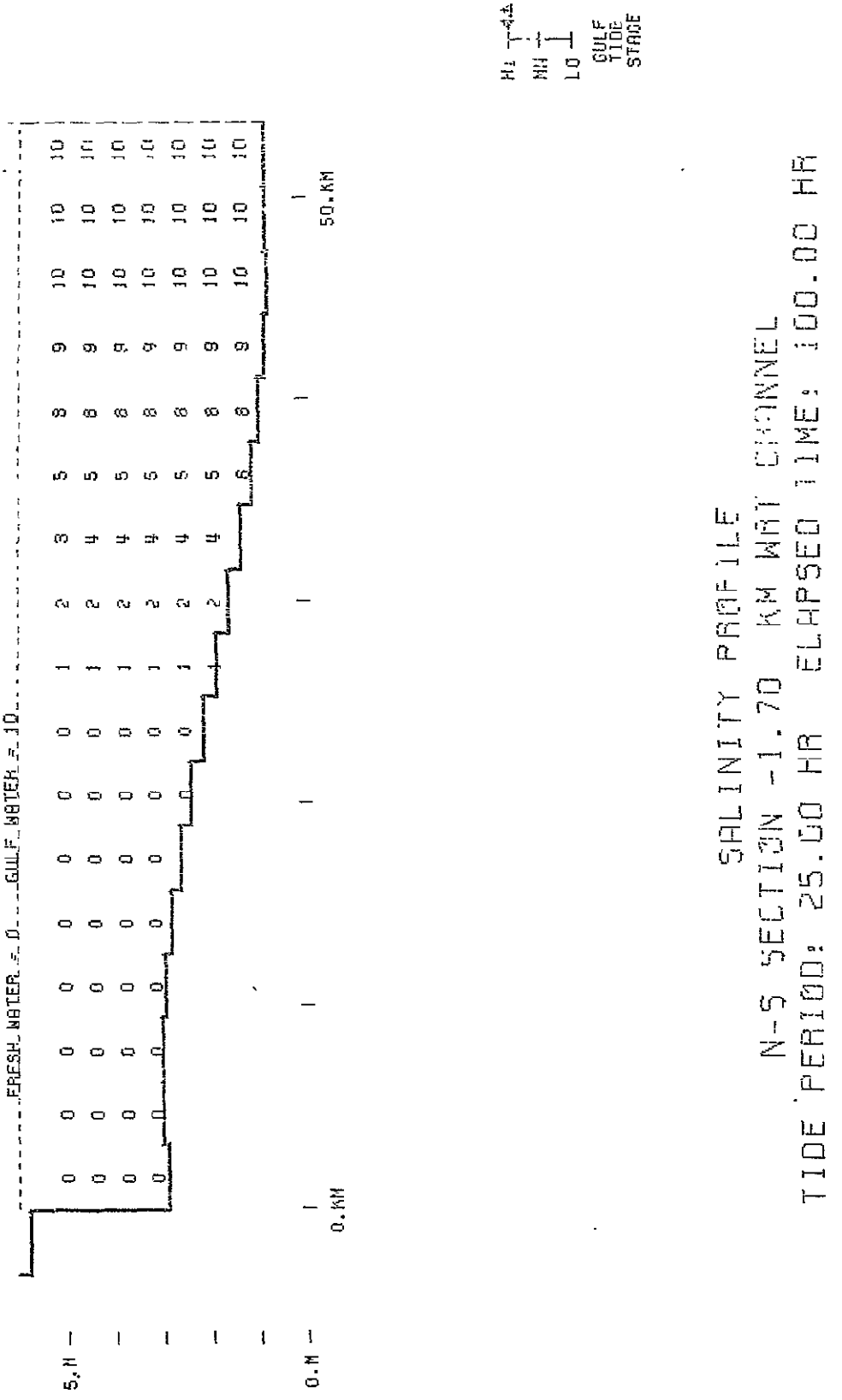


Figure A.39 - Case 1

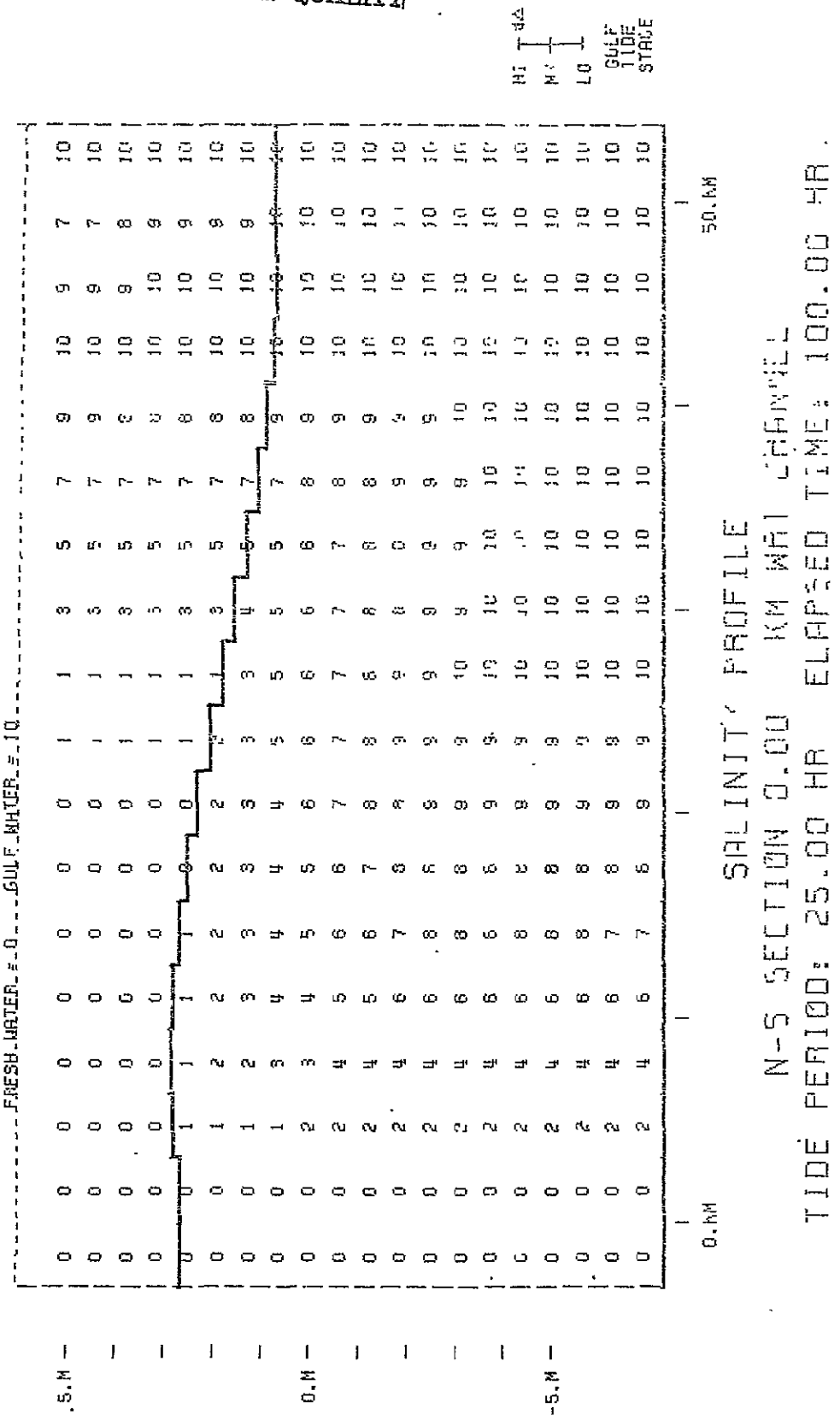


Figure A.40 - Case 1

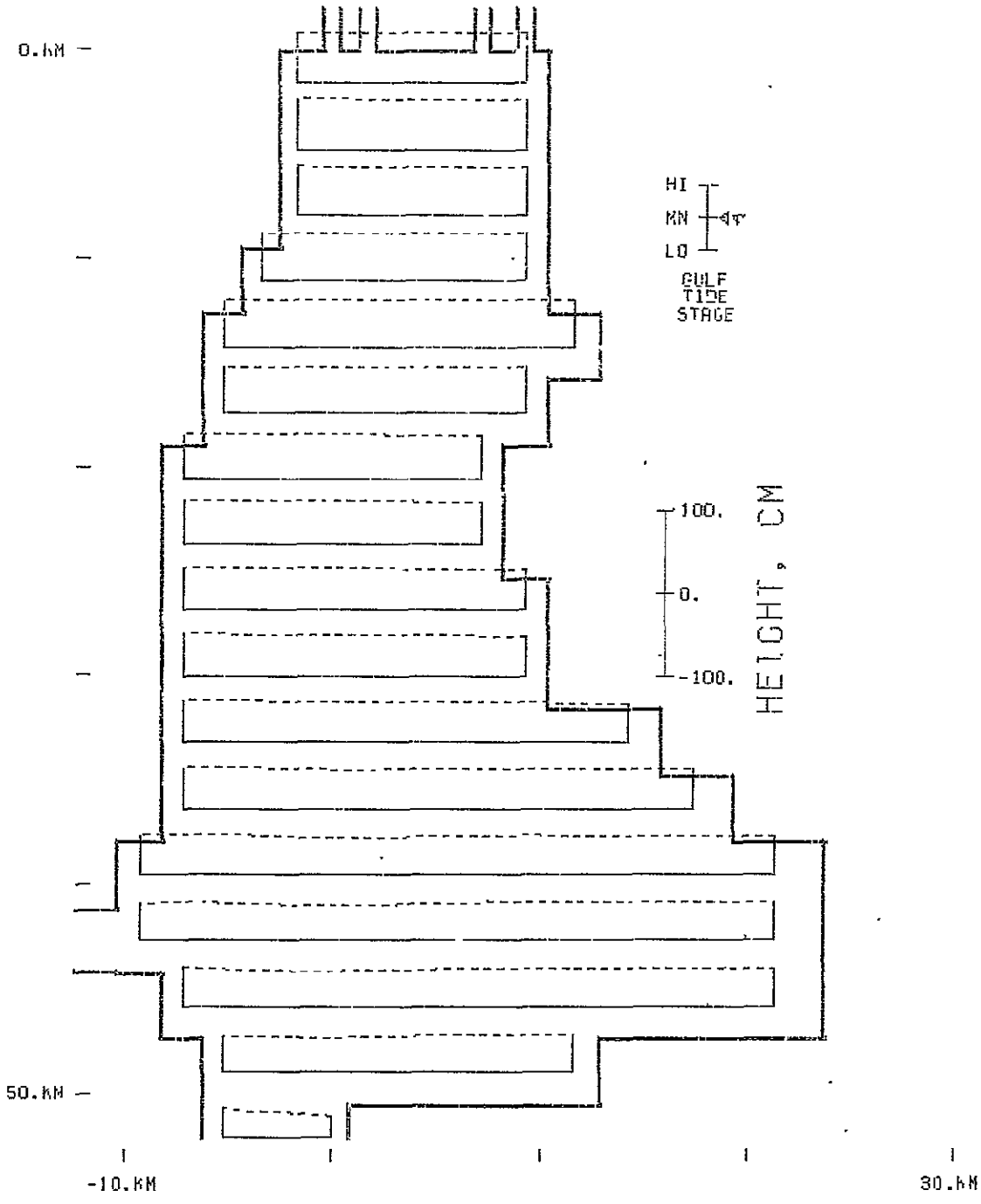


Figure A.41 -

Case 1

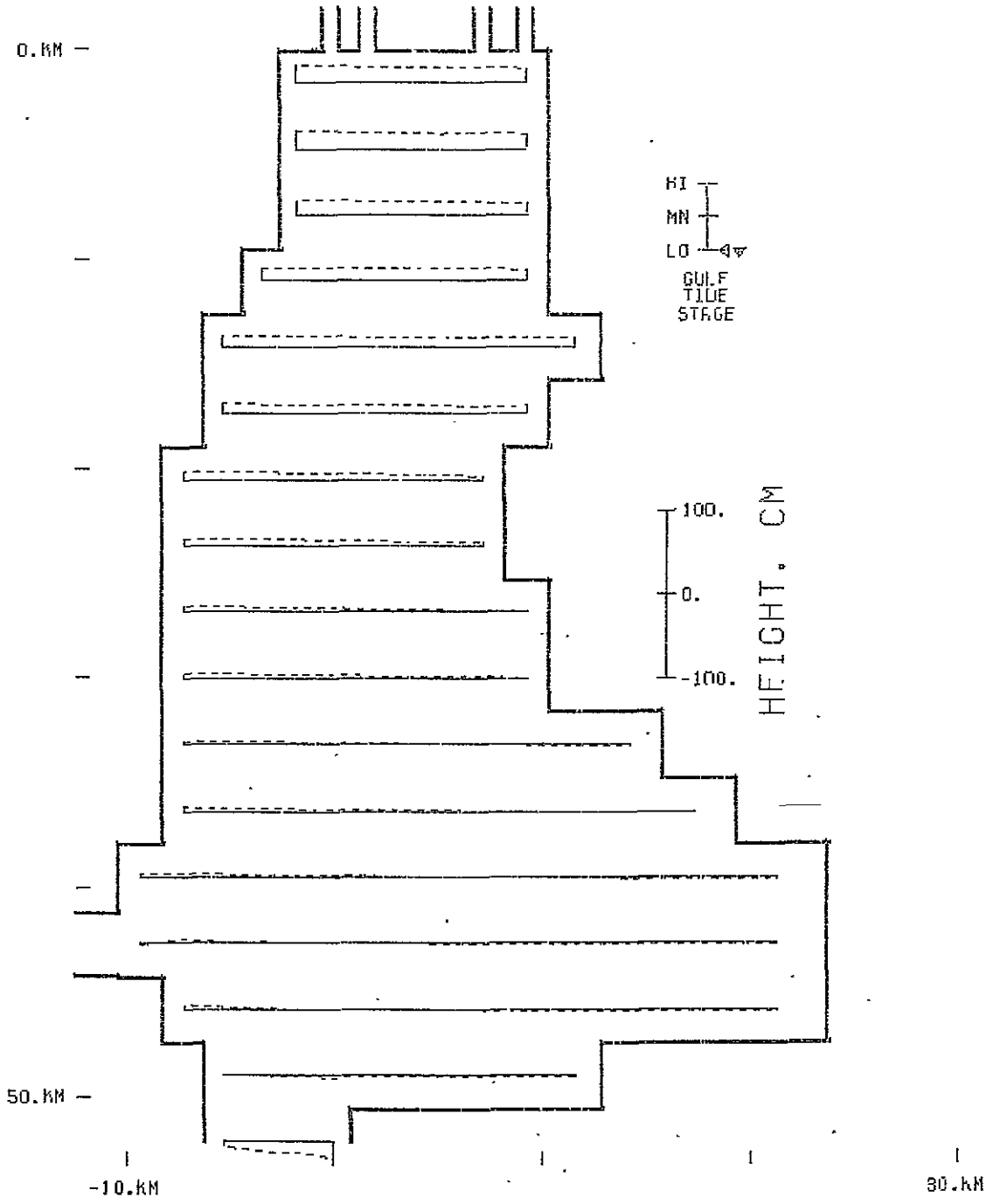


Figure A.42 -

Case 1

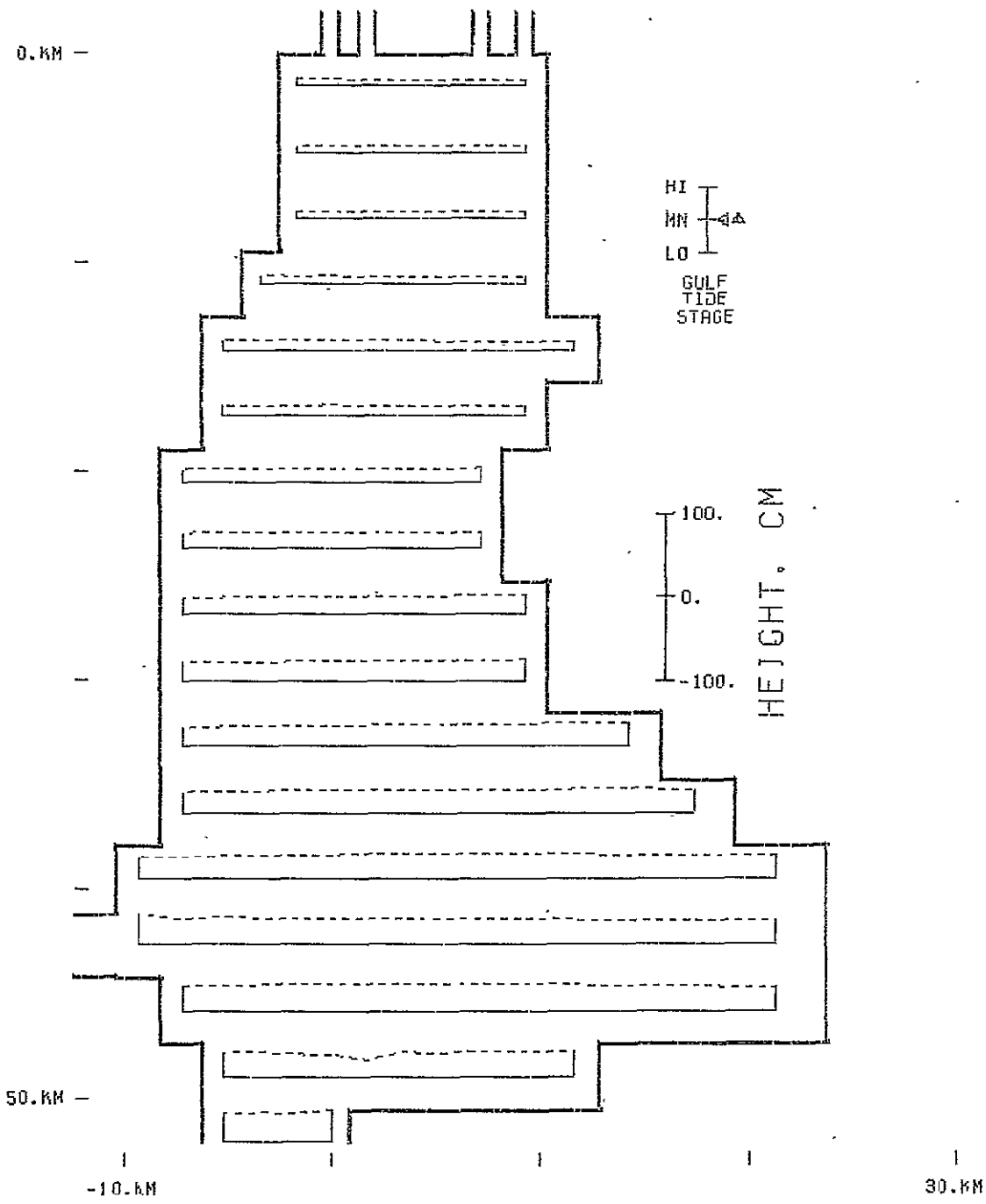


Figure A.43 -

Case 1

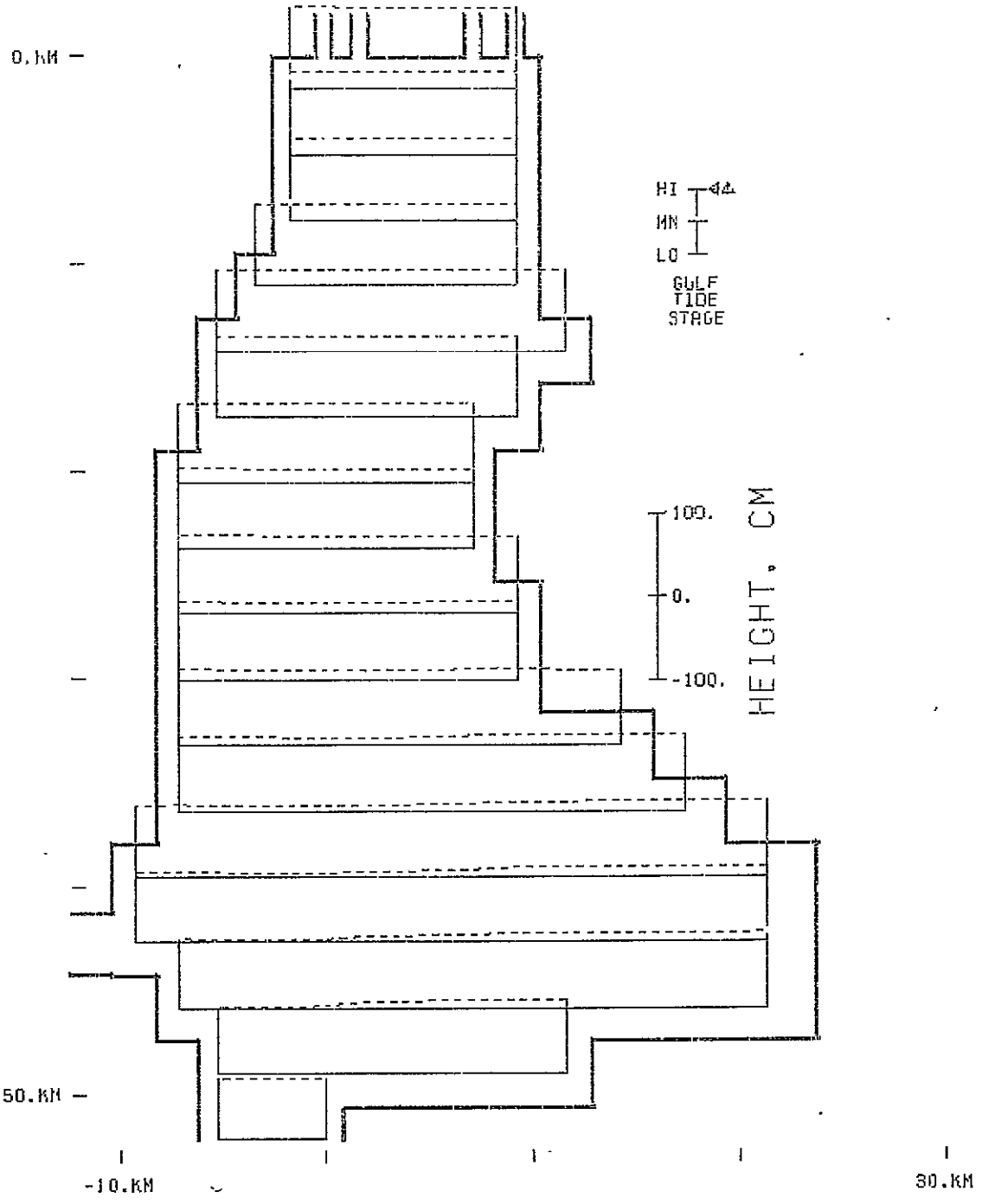


Figure A.44 -

Case 1

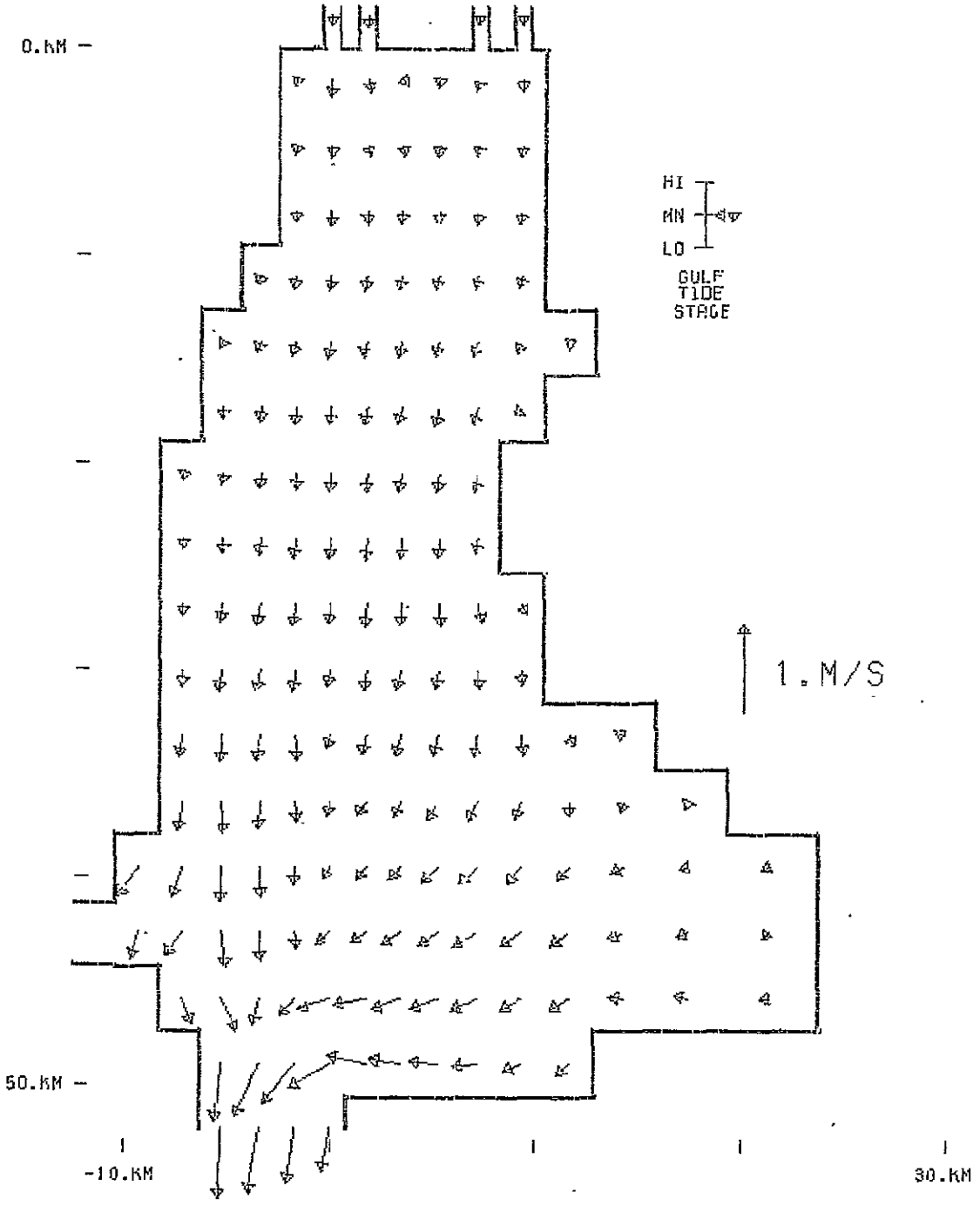


Figure A.45 -

Case 2

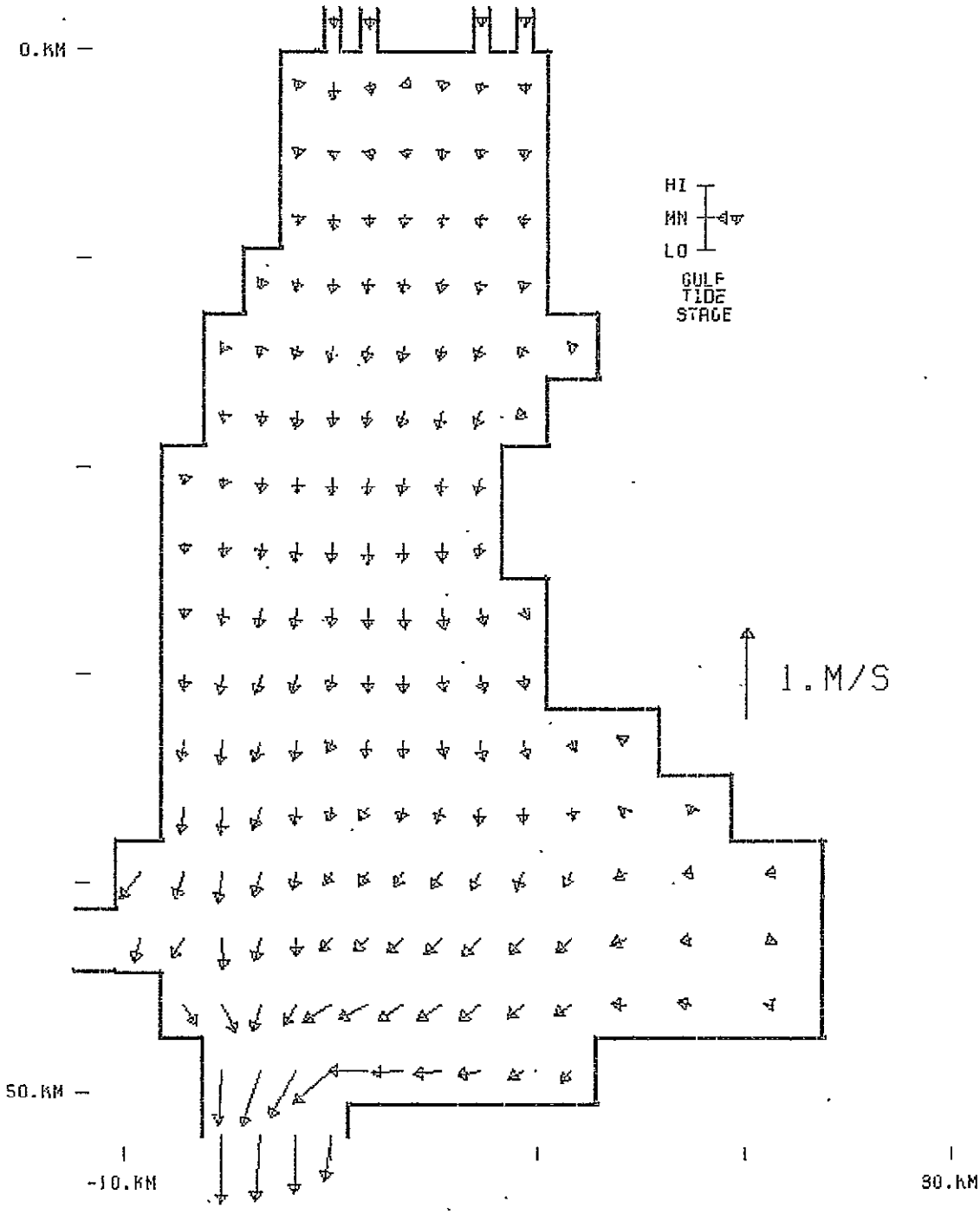


Figure A.46 -

Case 2

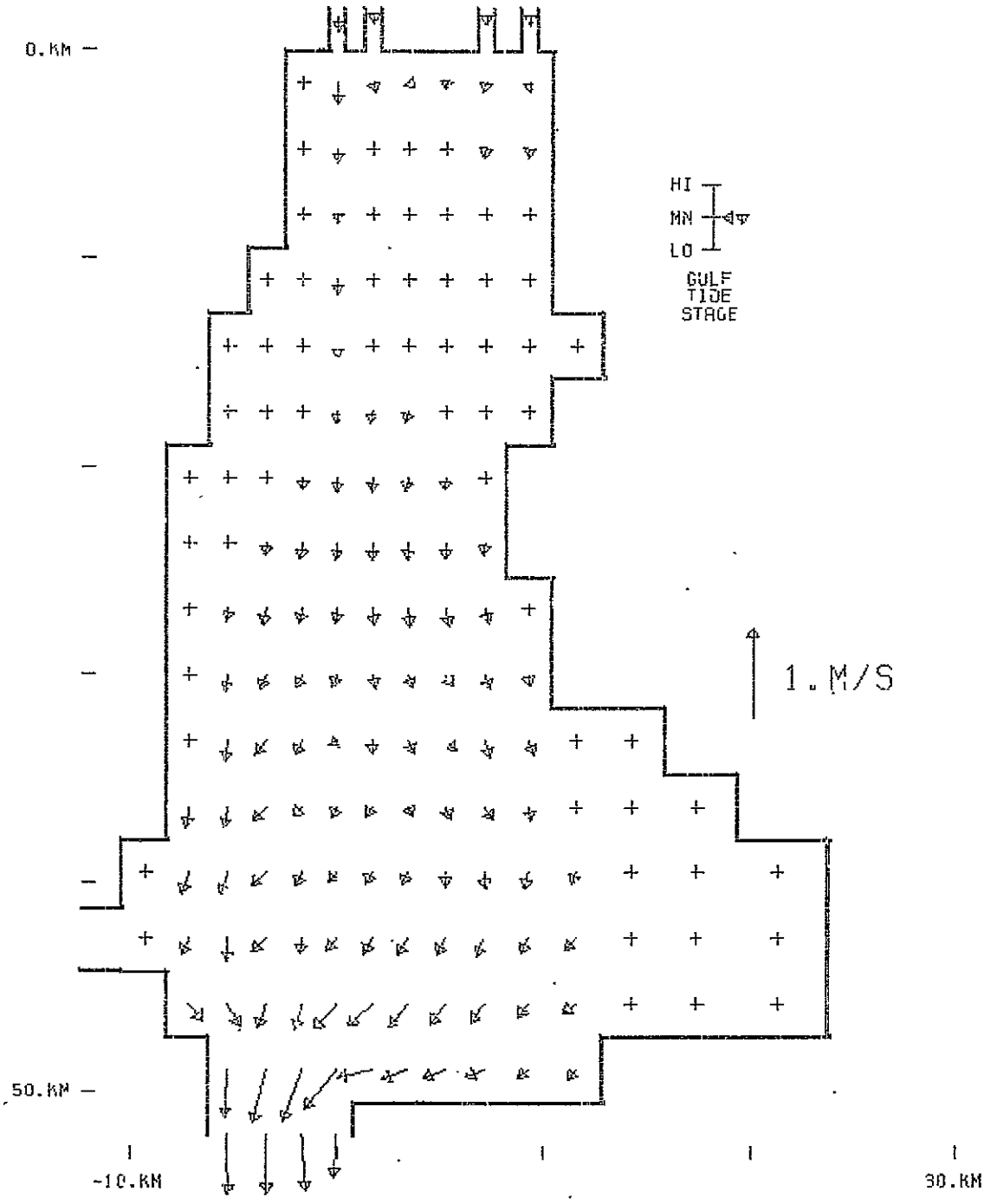


Figure A.47 -

Case 2

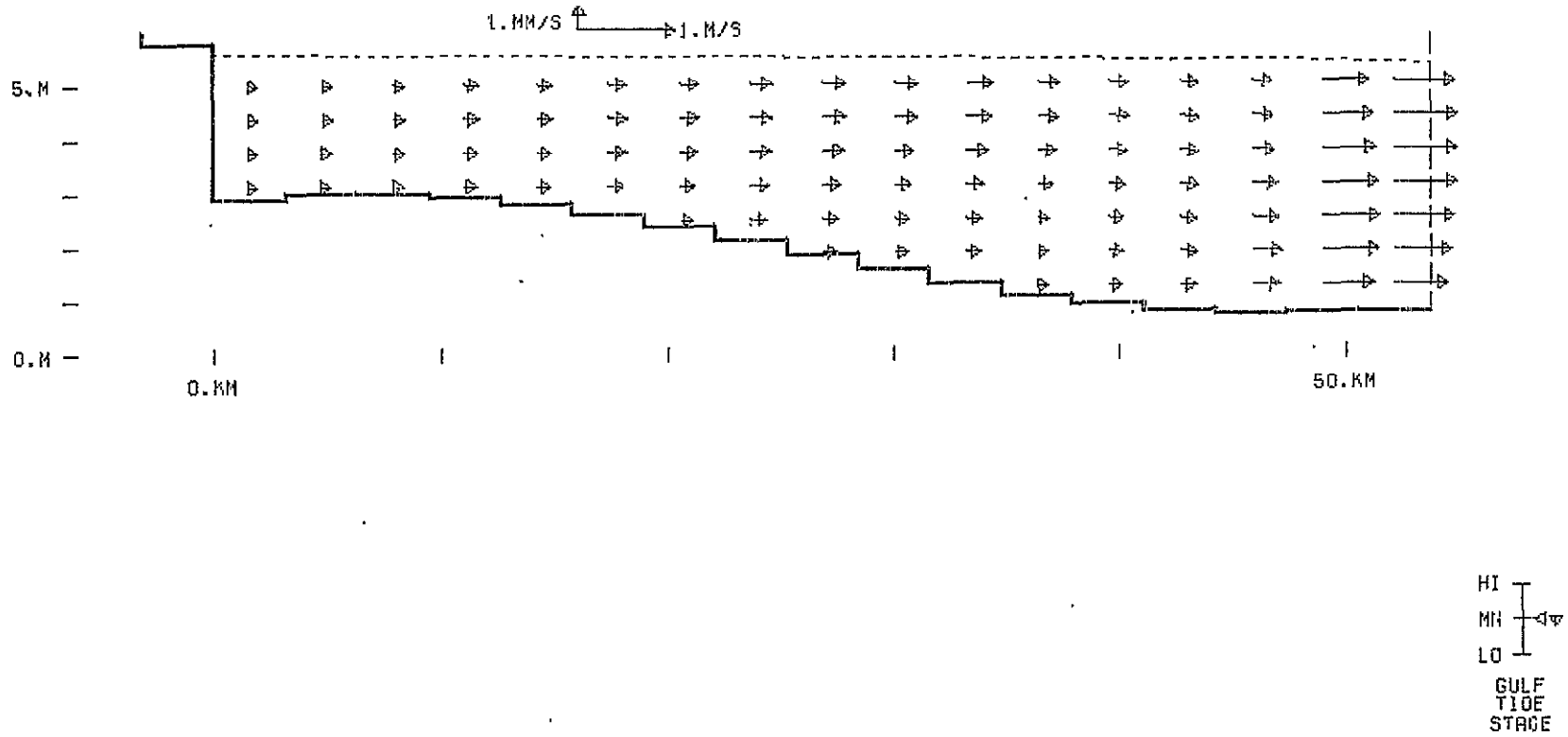


Figure A.48 - Case 2

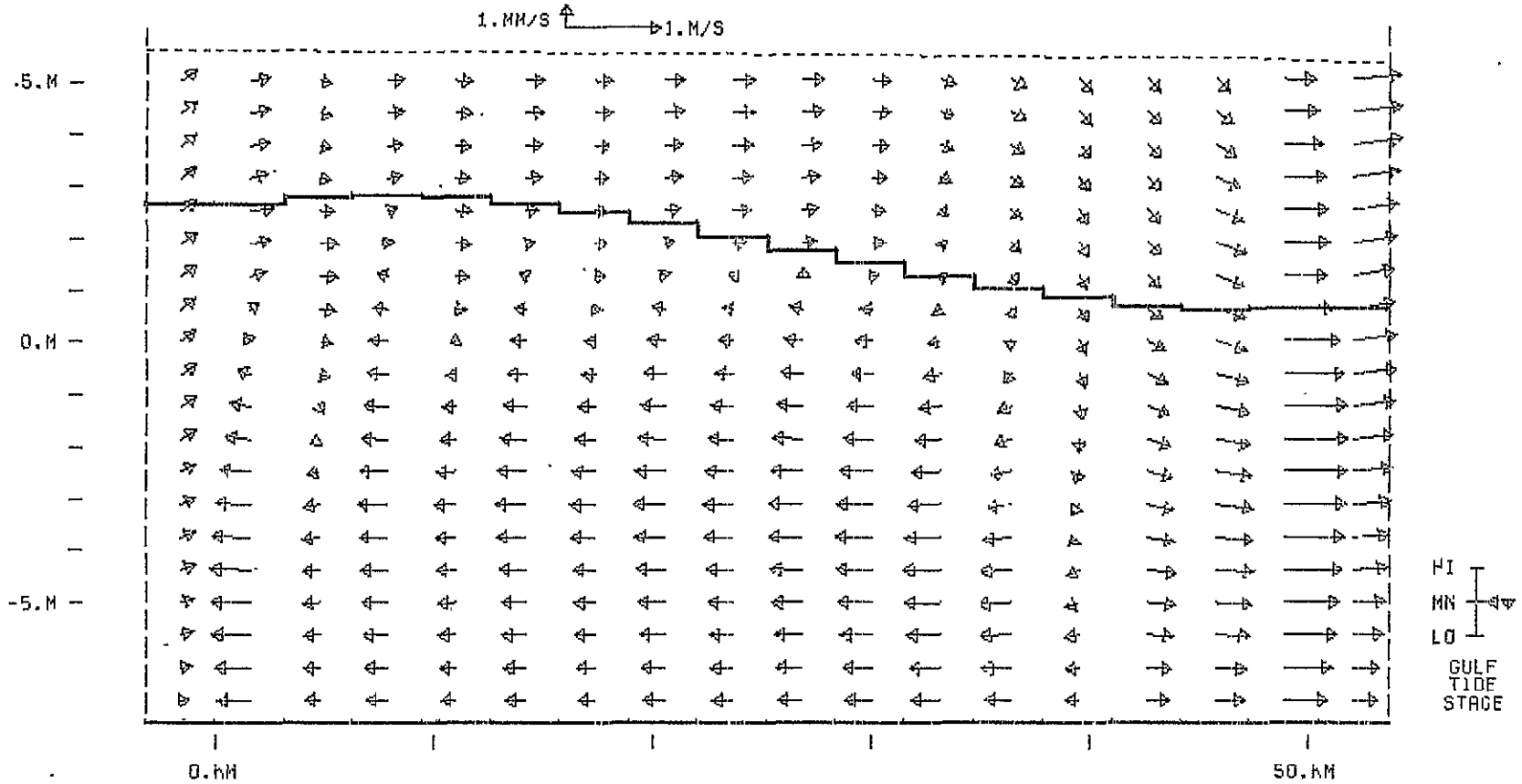


Figure A.49 - Case 2

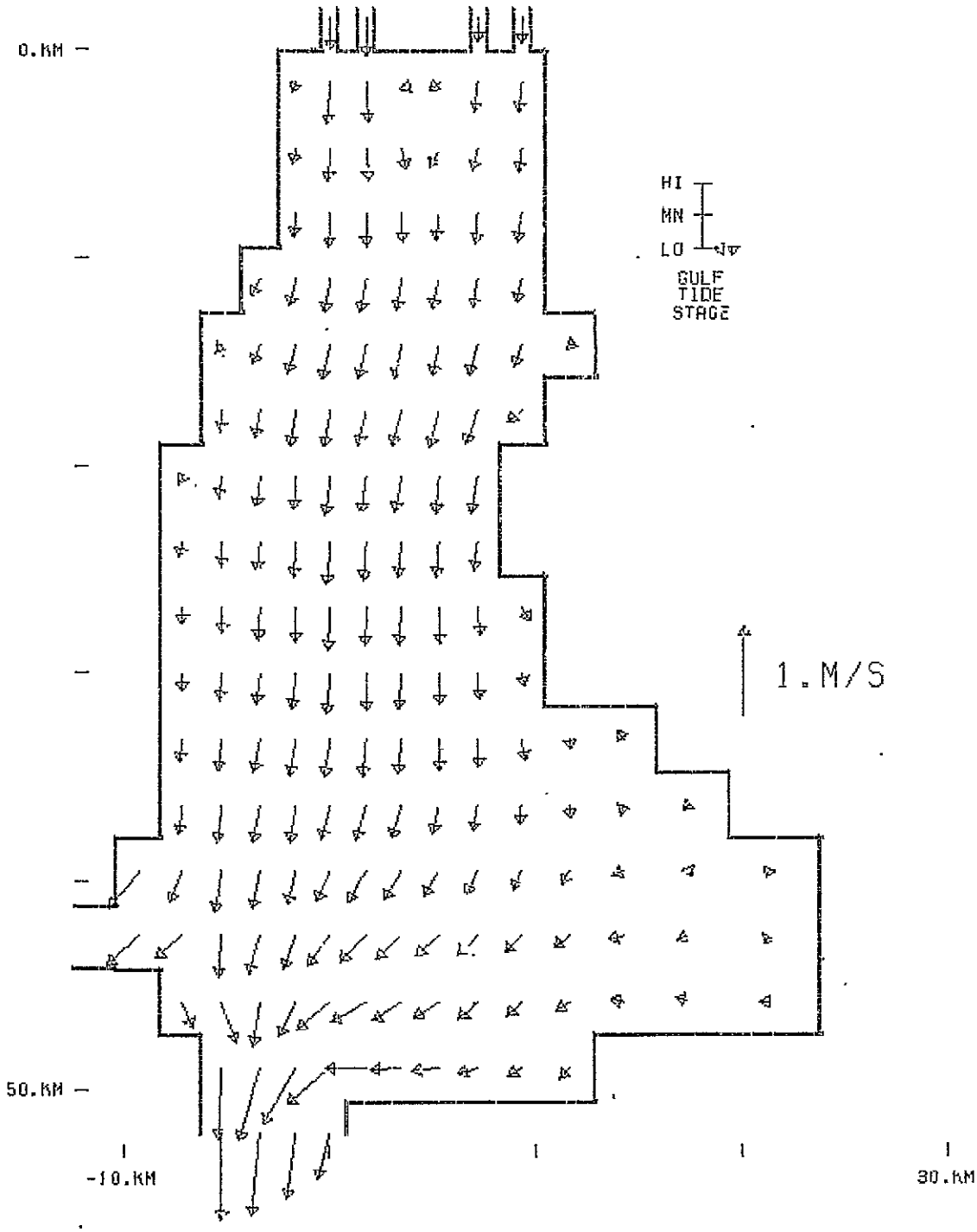


Figure A.50 -

Case 2

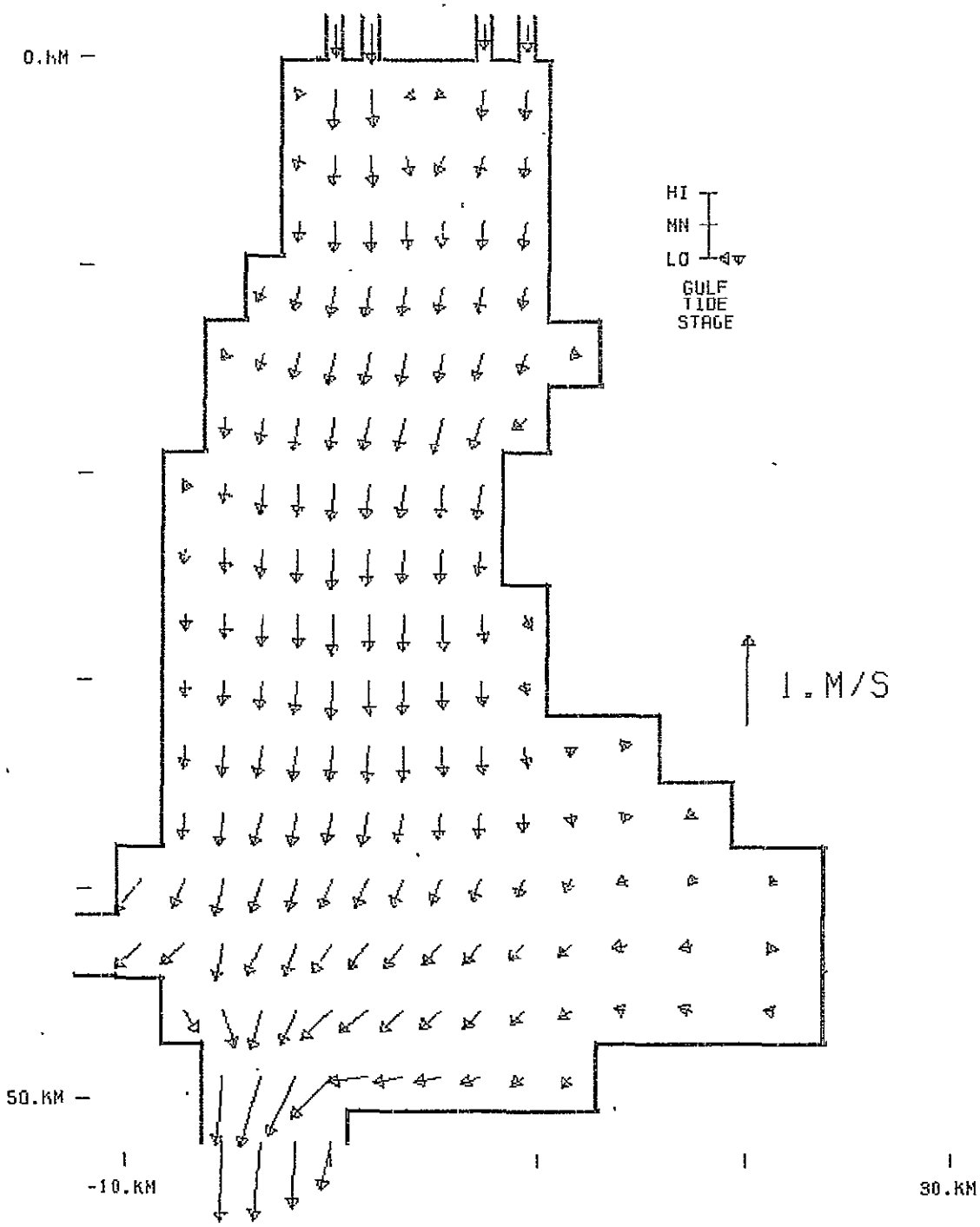


Figure A.51 -

Case 2

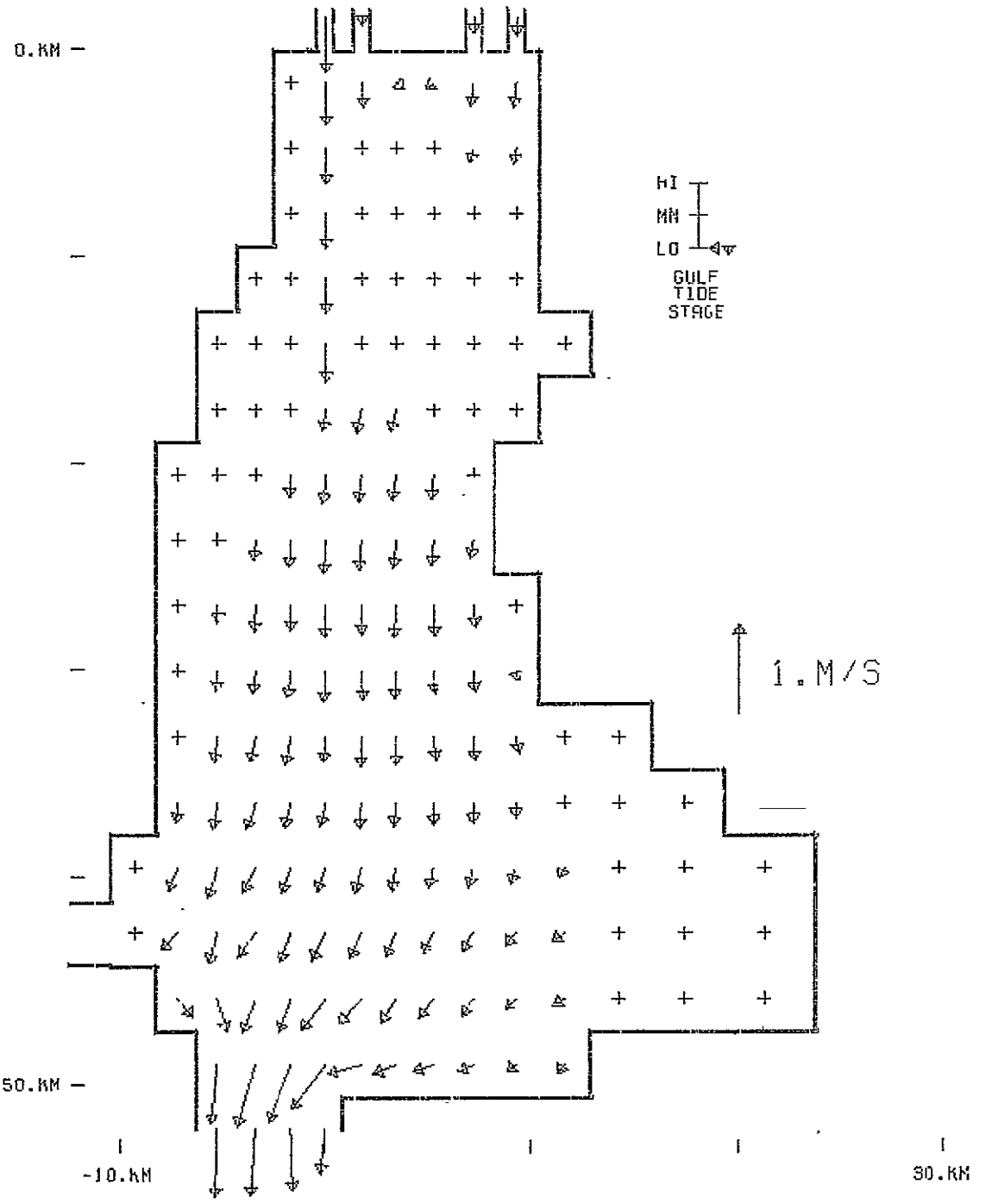
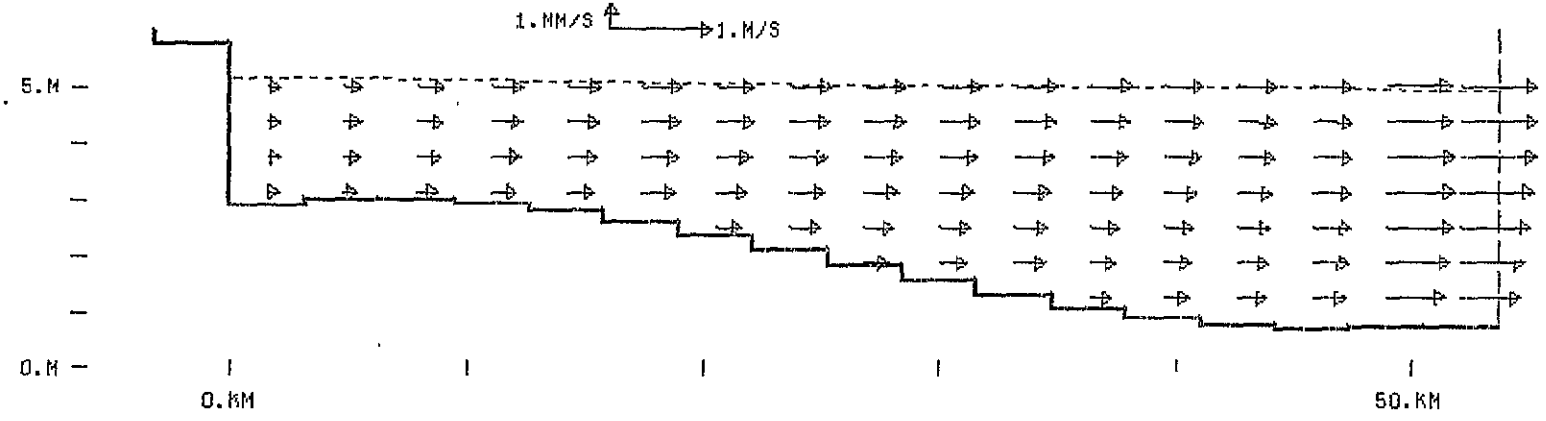
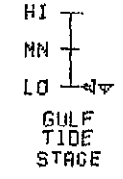


Figure A.52 -

Case 2





VELOCITY VECTORS
N-S SECTION -1.70 KM WRT CHANNEL
TIDE PERIOD: 25.00 HR ELAPSED TIME: 137.50 HR

Figure A.53 - Case 2

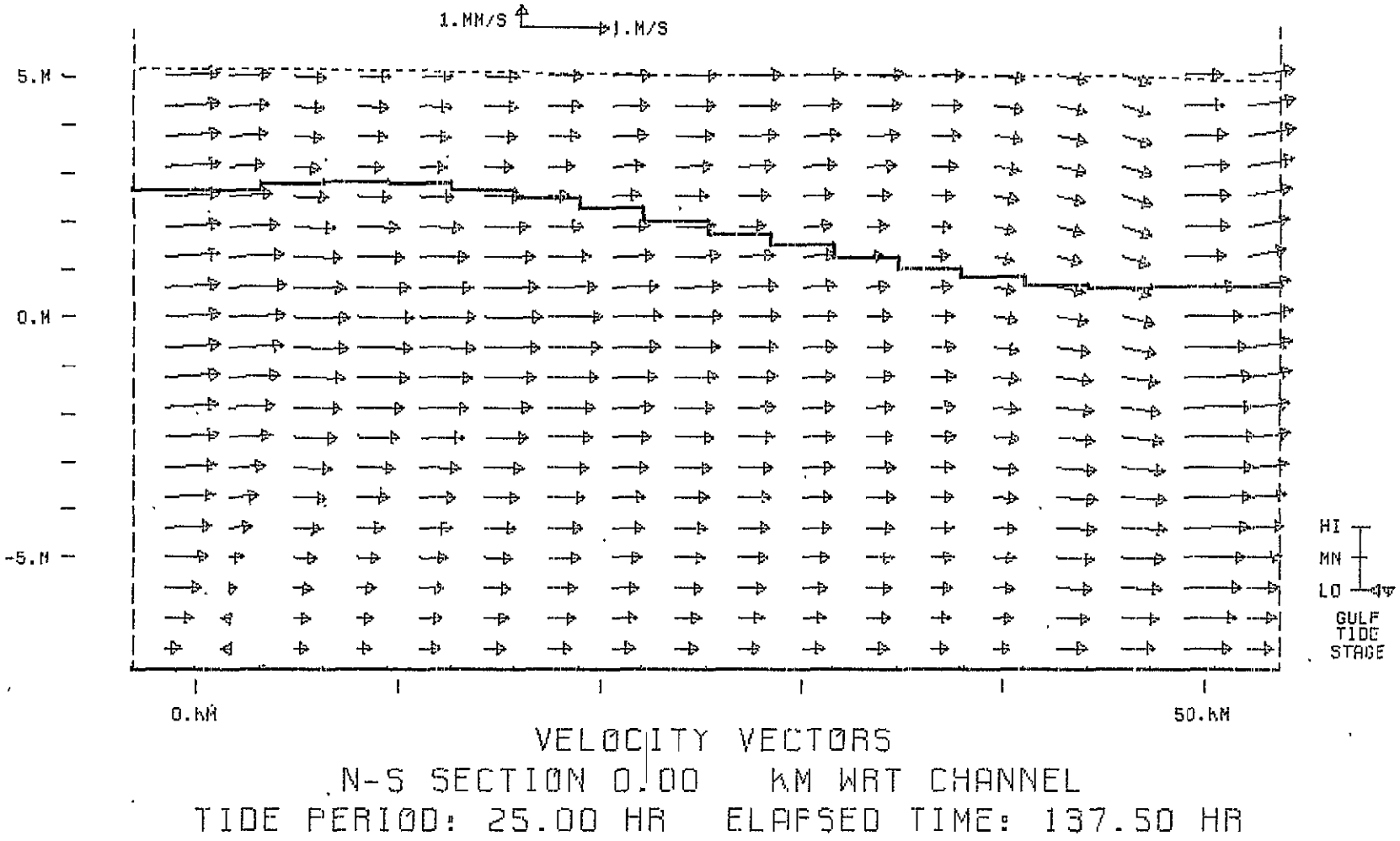


Figure A.54 - Case 2

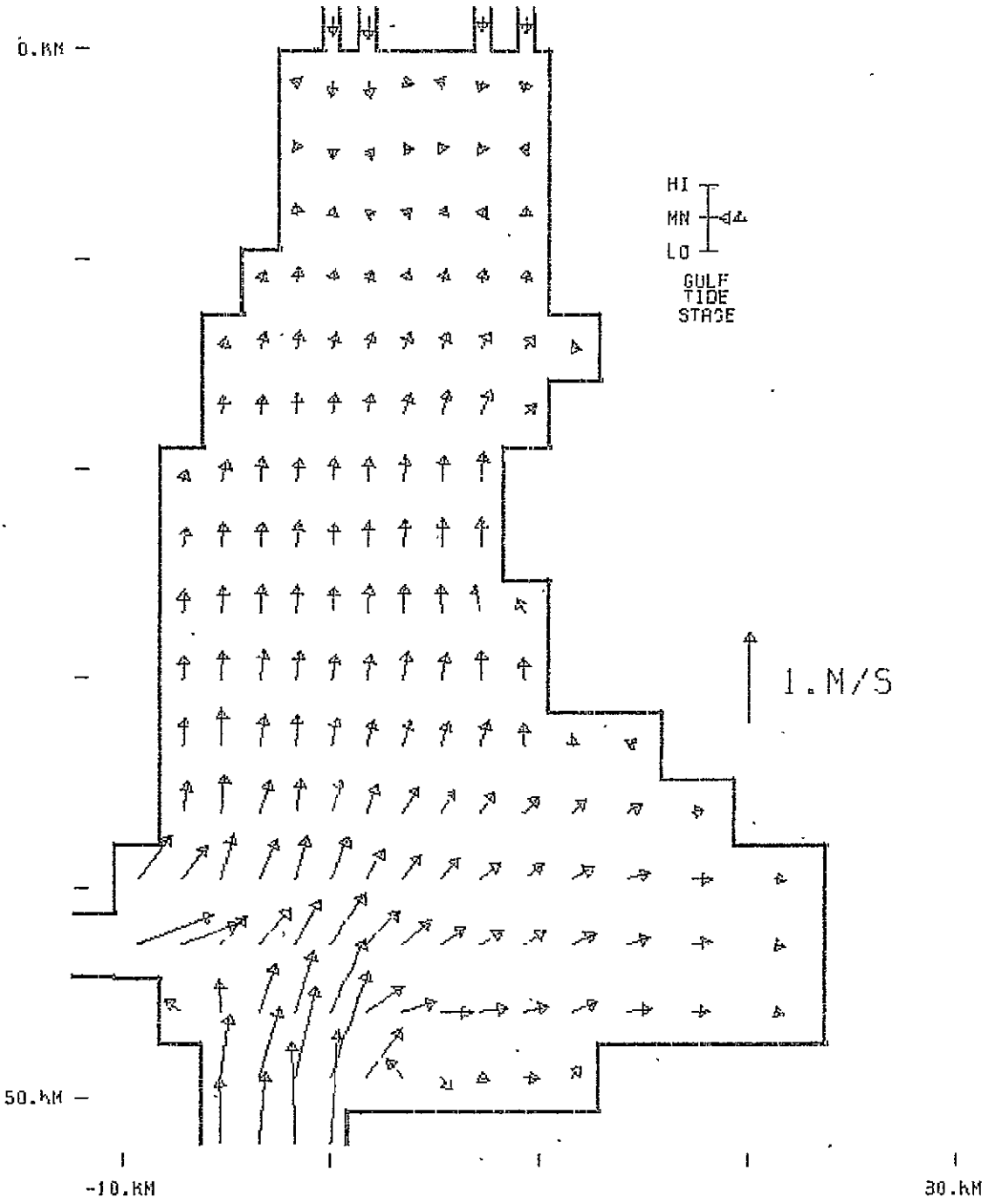


Figure A.55 -

Case 2

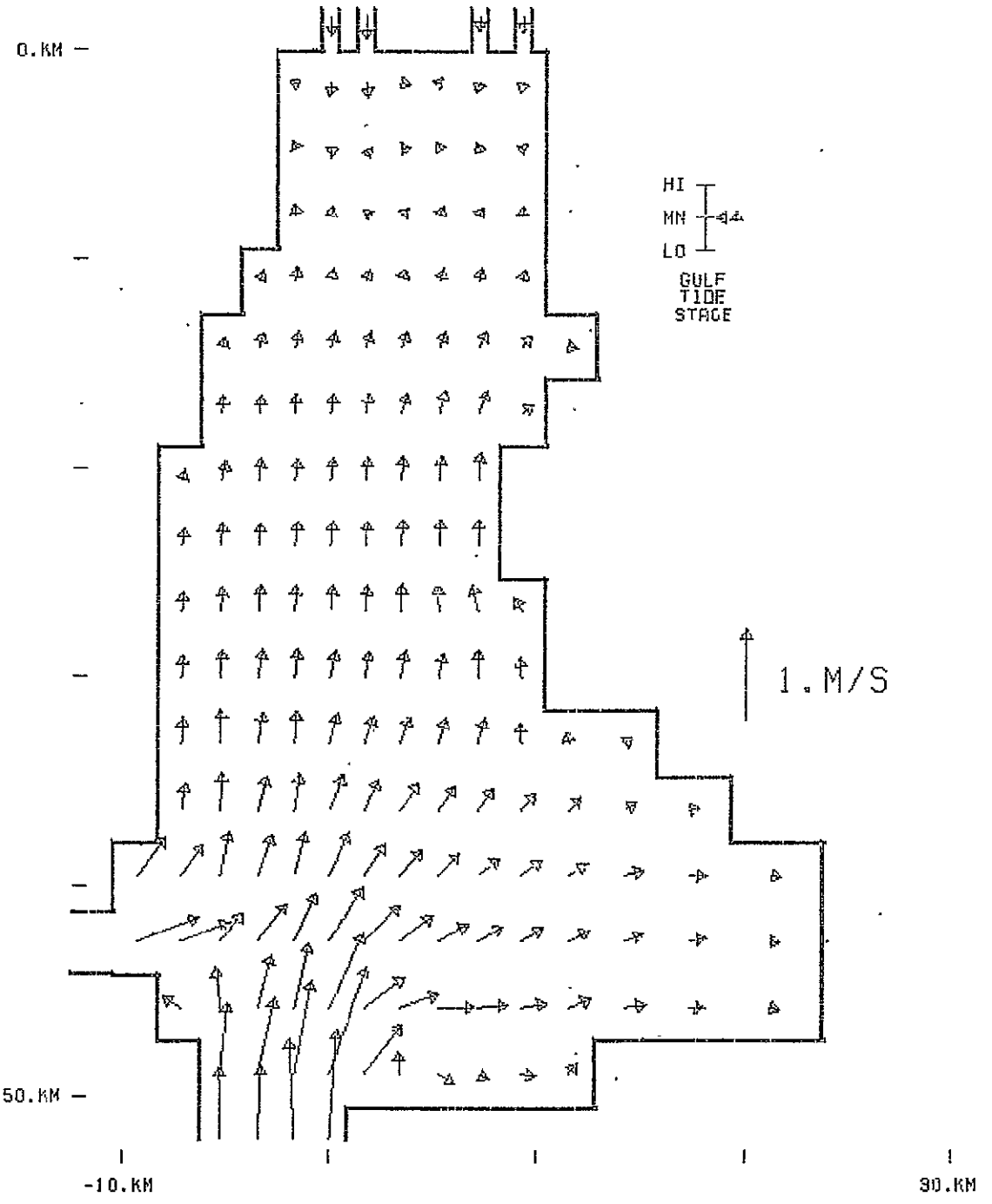


Figure A.56 -

Case 2

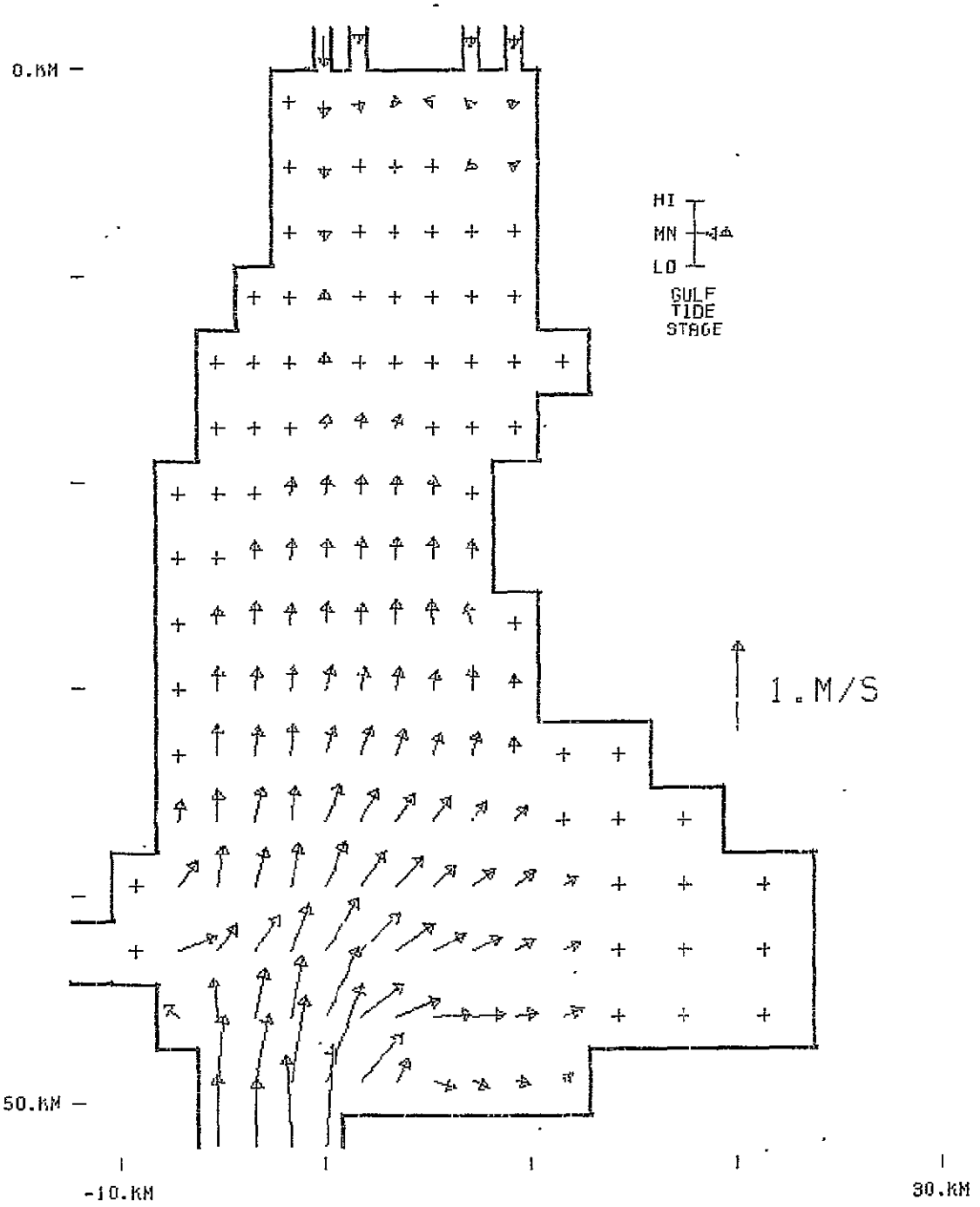
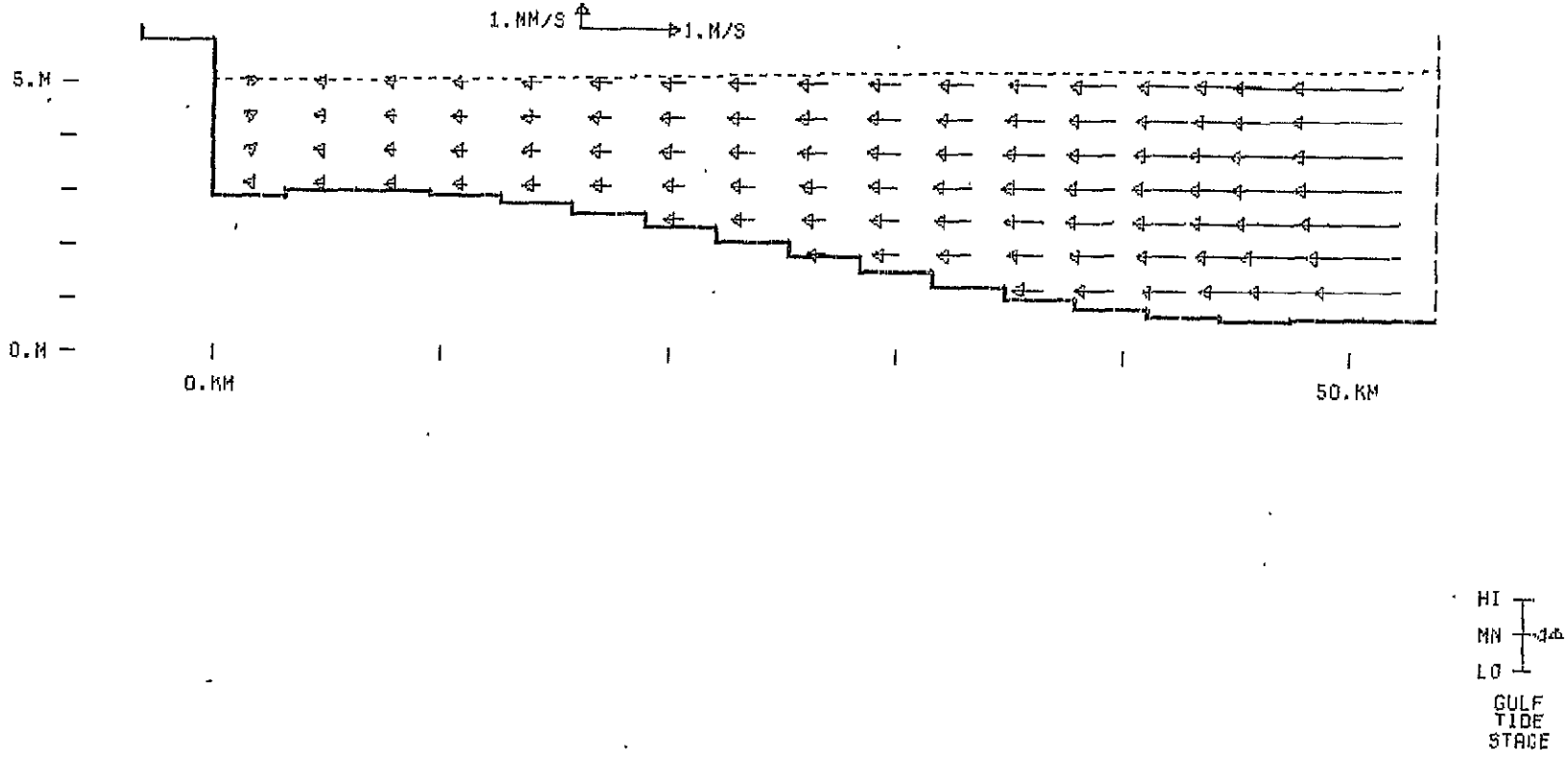


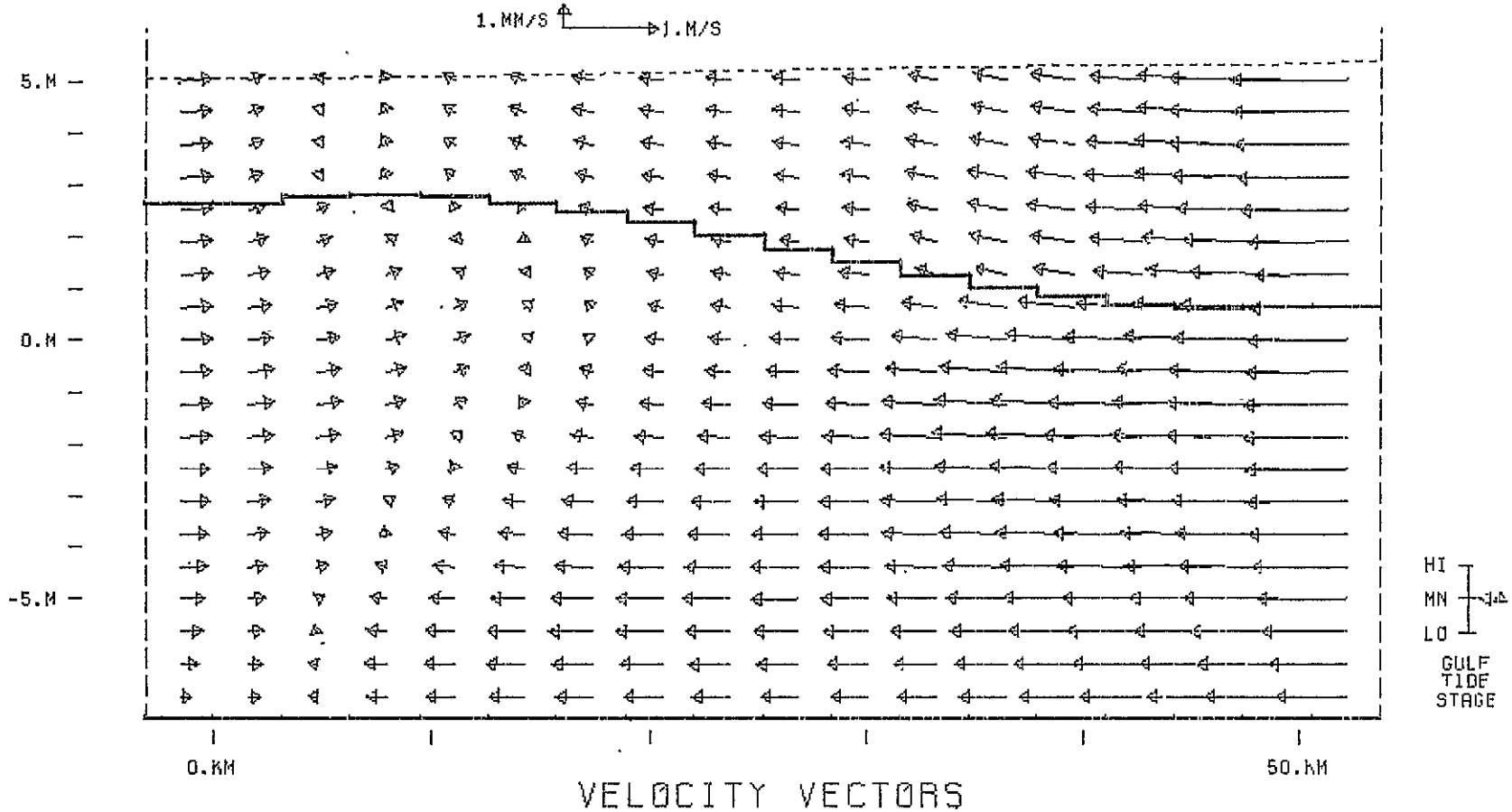
Figure A.57 –

Case 2



VELOCITY VECTORS
N-S SECTION -1.70 KM WRT CHANNEL
TIDE PERIOD: 25.00 HR ELAPSED TIME: 143.75 HR

Figure A.58 - Case 2



VELOCITY VECTORS
N-S SECTION 0.00 KM WRT CHANNEL
TIDE PERIOD: 25.00 HR ELAPSED TIME: 143.75 HR

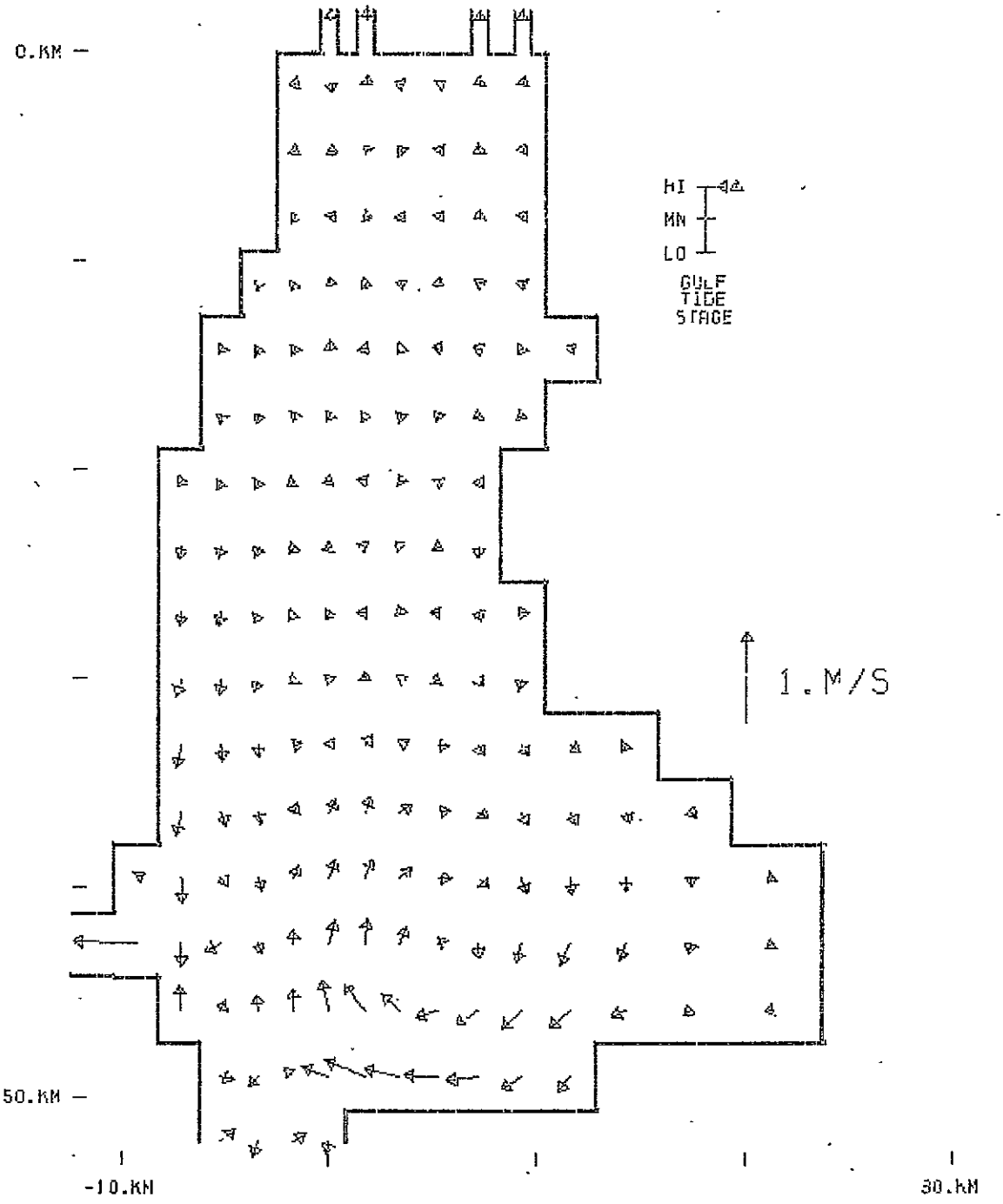


Figure A.60 -

Case 2

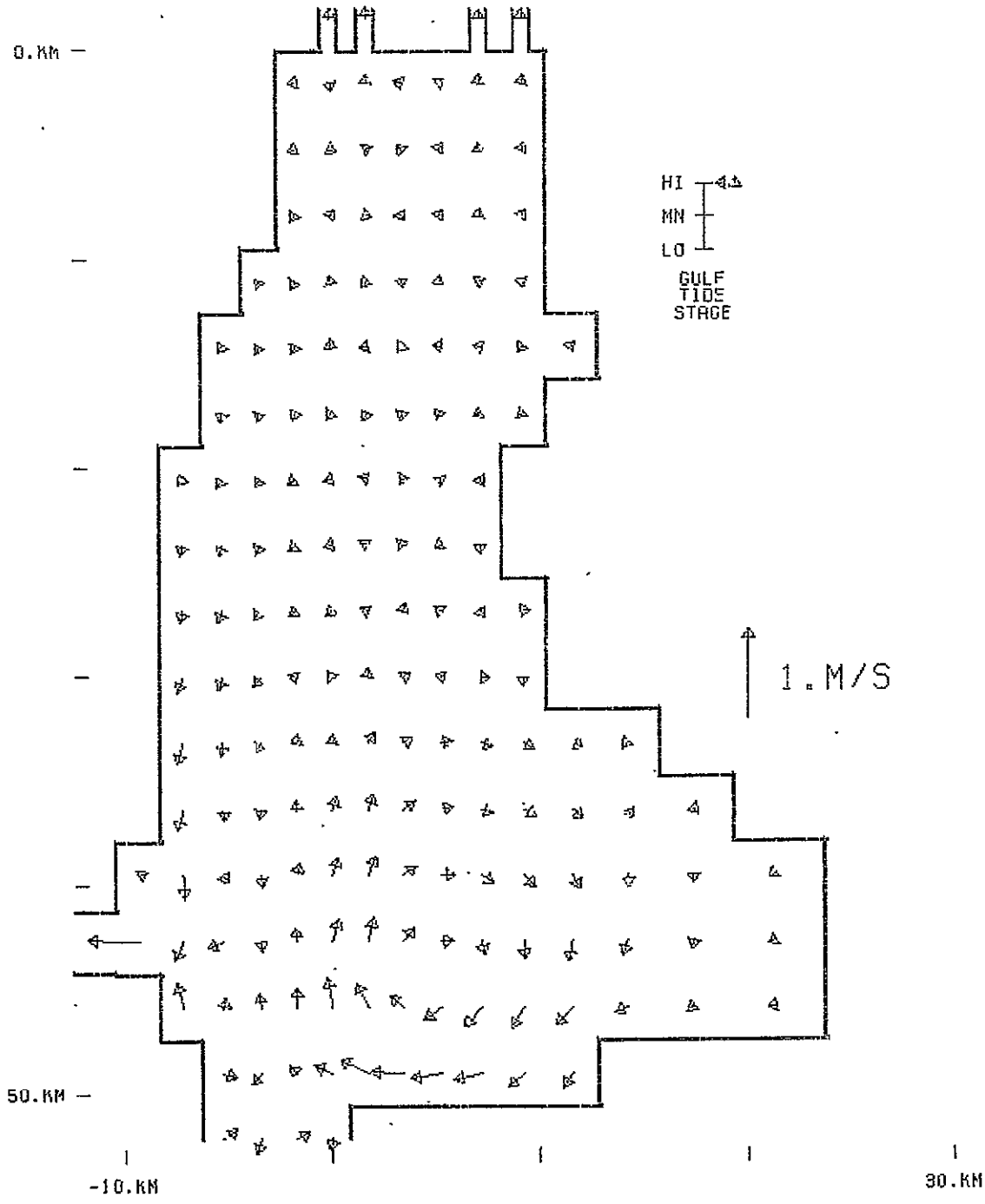


Figure A.61 -

Case 2

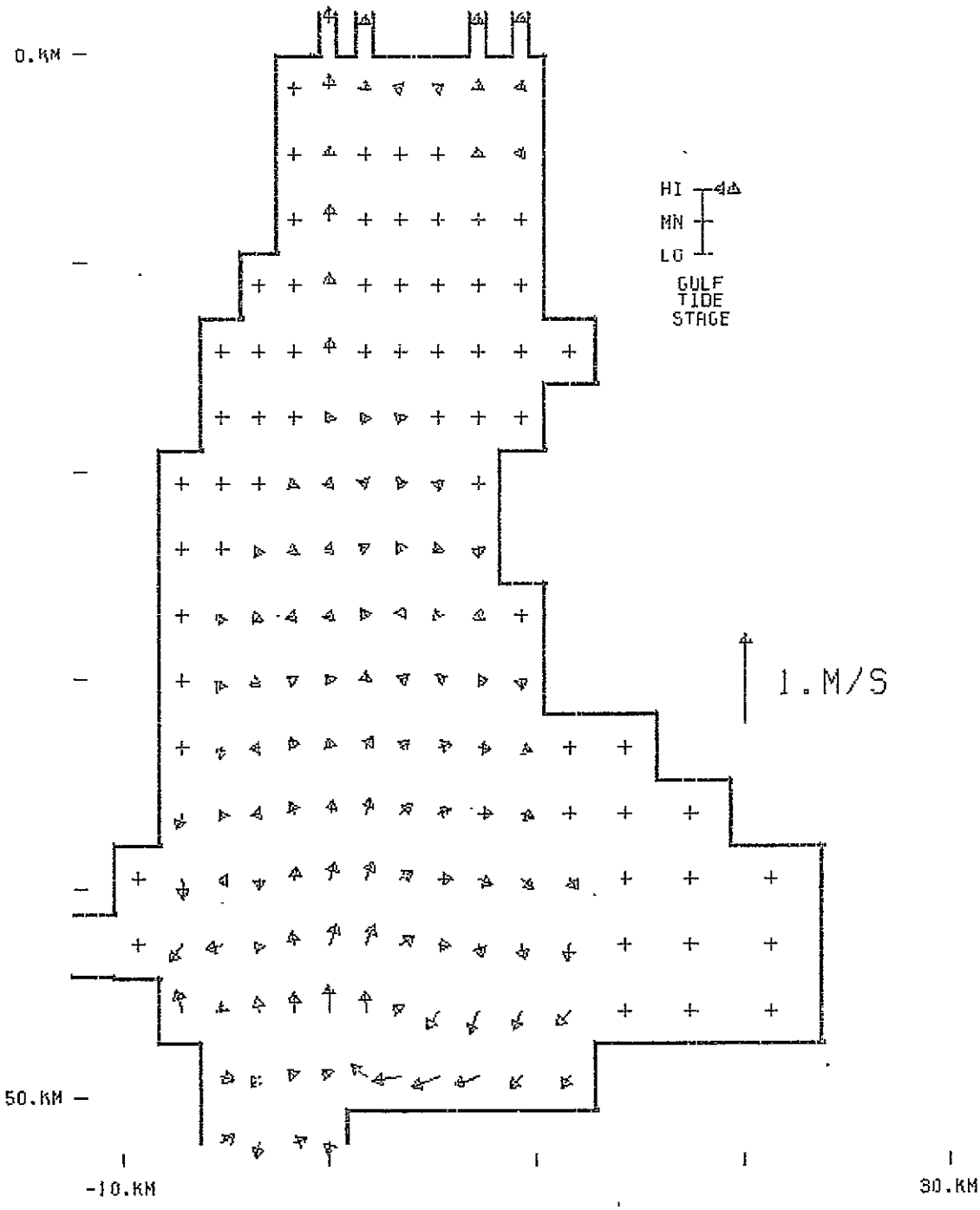
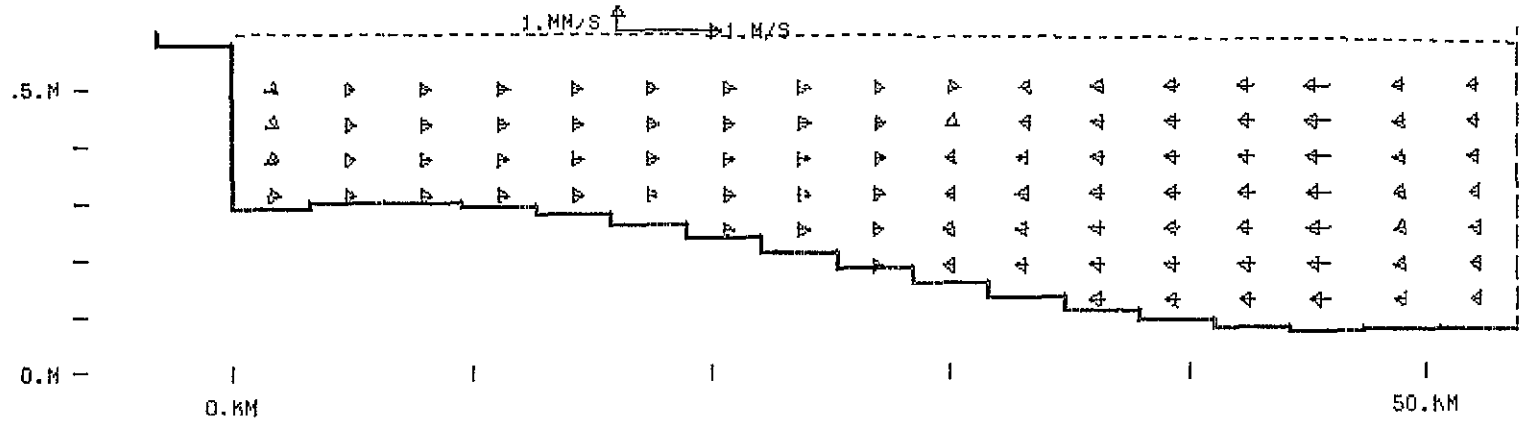


Figure A.62 -

Case 2



VELOCITY VECTORS
N-S SECTION -1.70 KM WRT CHANNEL
TIDE PERIOD: 25.00 HR ELAPSED TIME: 150.00 HR

Figure A.63 - Case 2

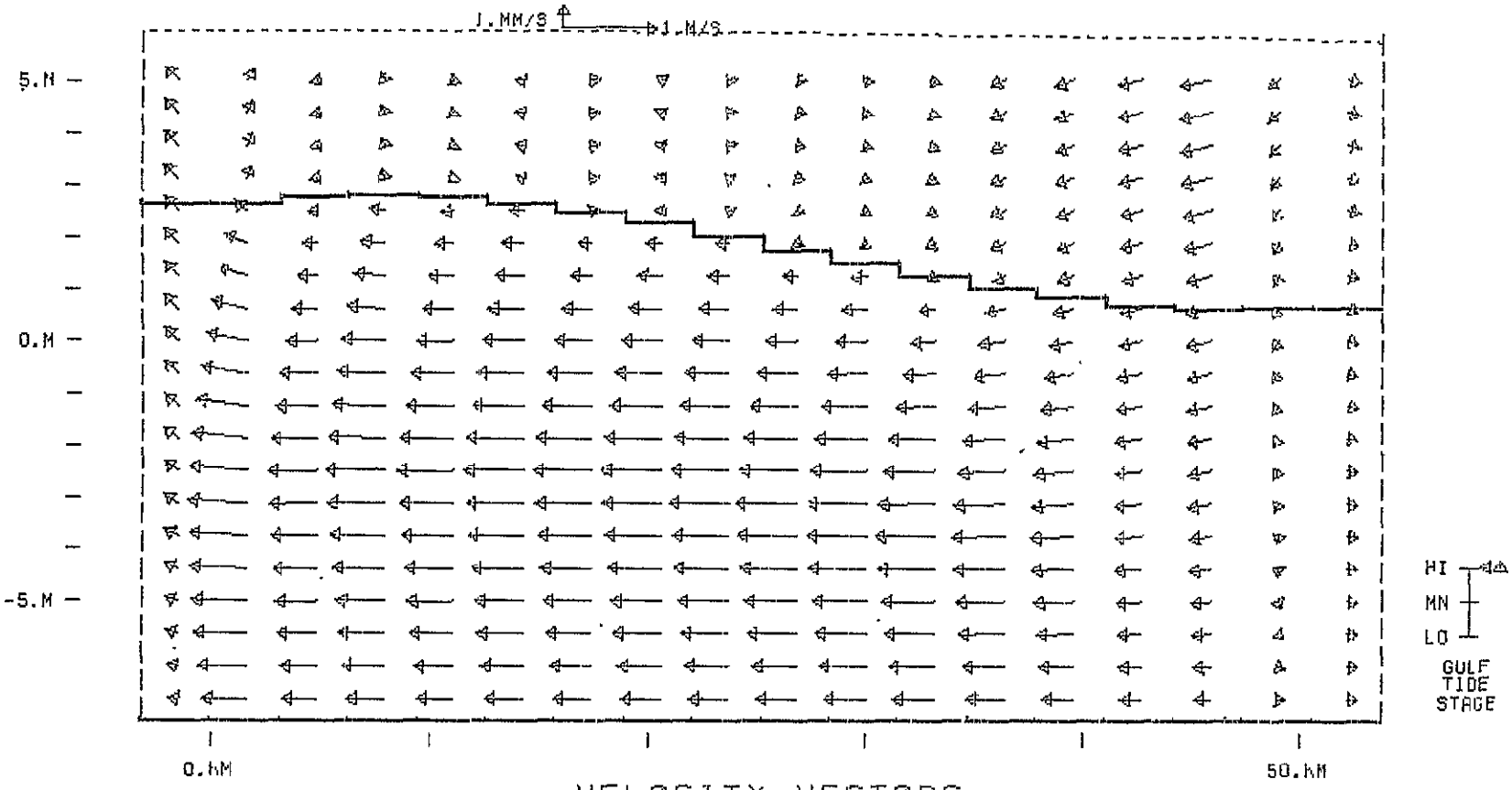


Figure A.64 - Case 2

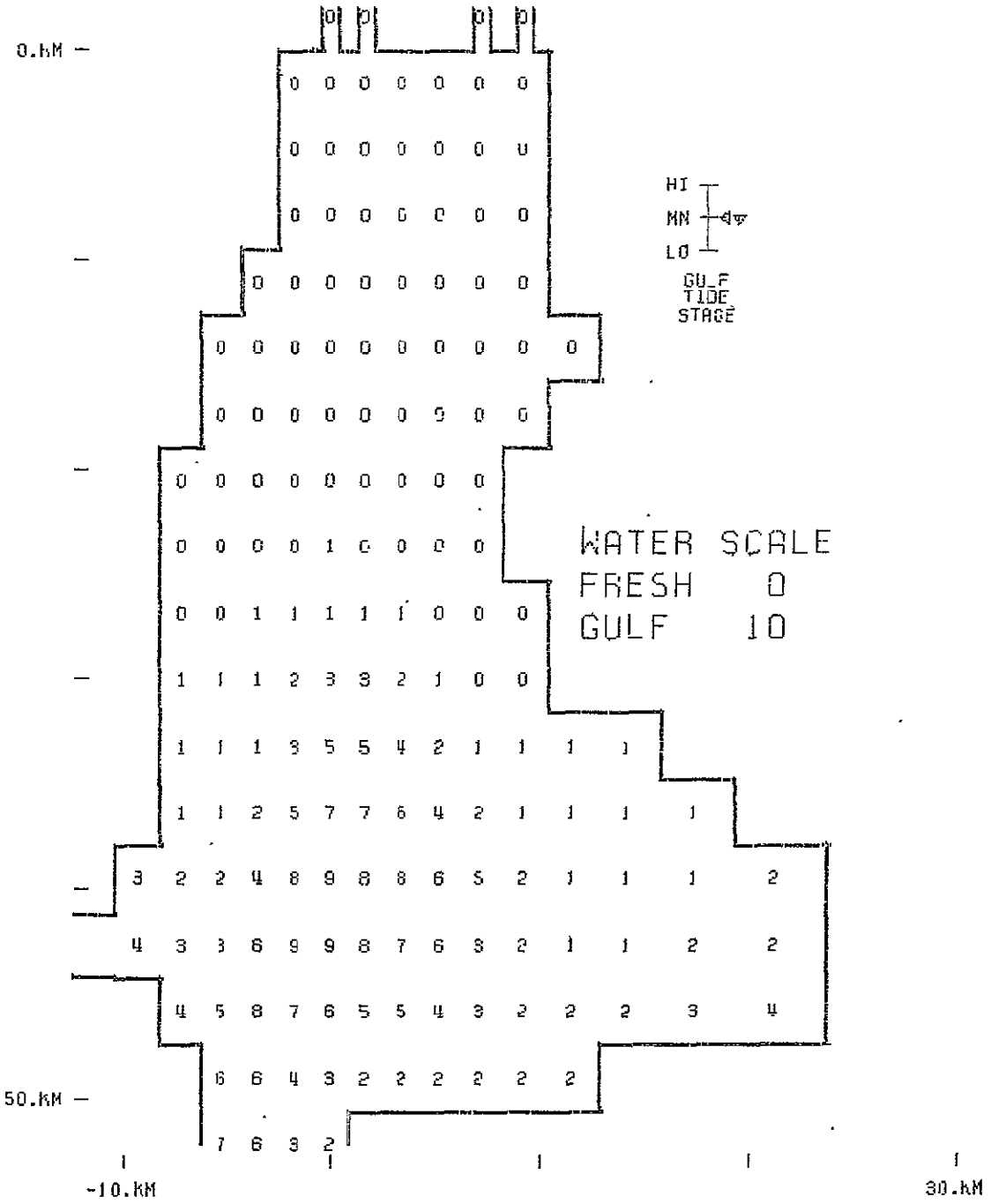


Figure A.65 -

Case 2

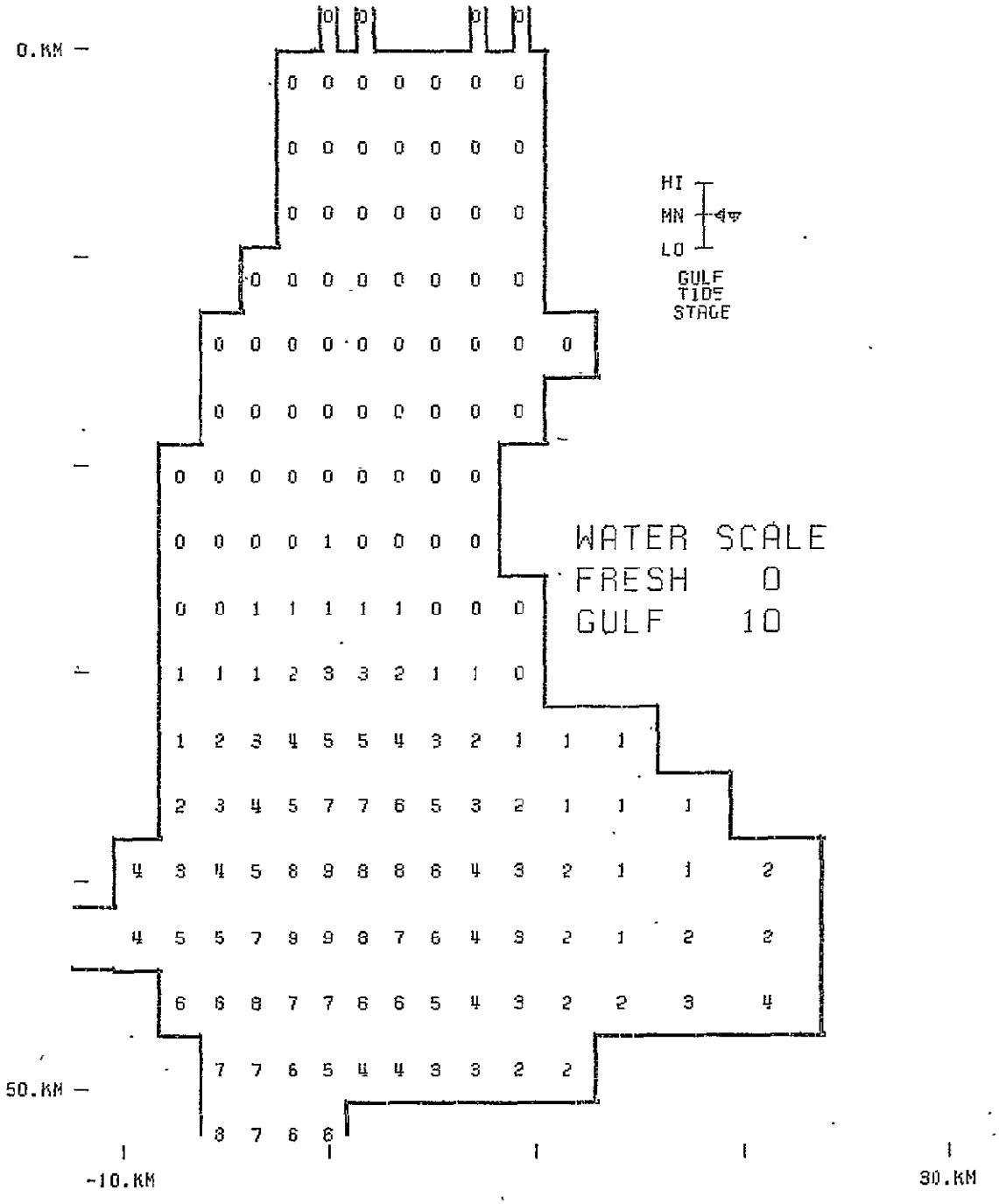


Figure A.66 -

Case 2

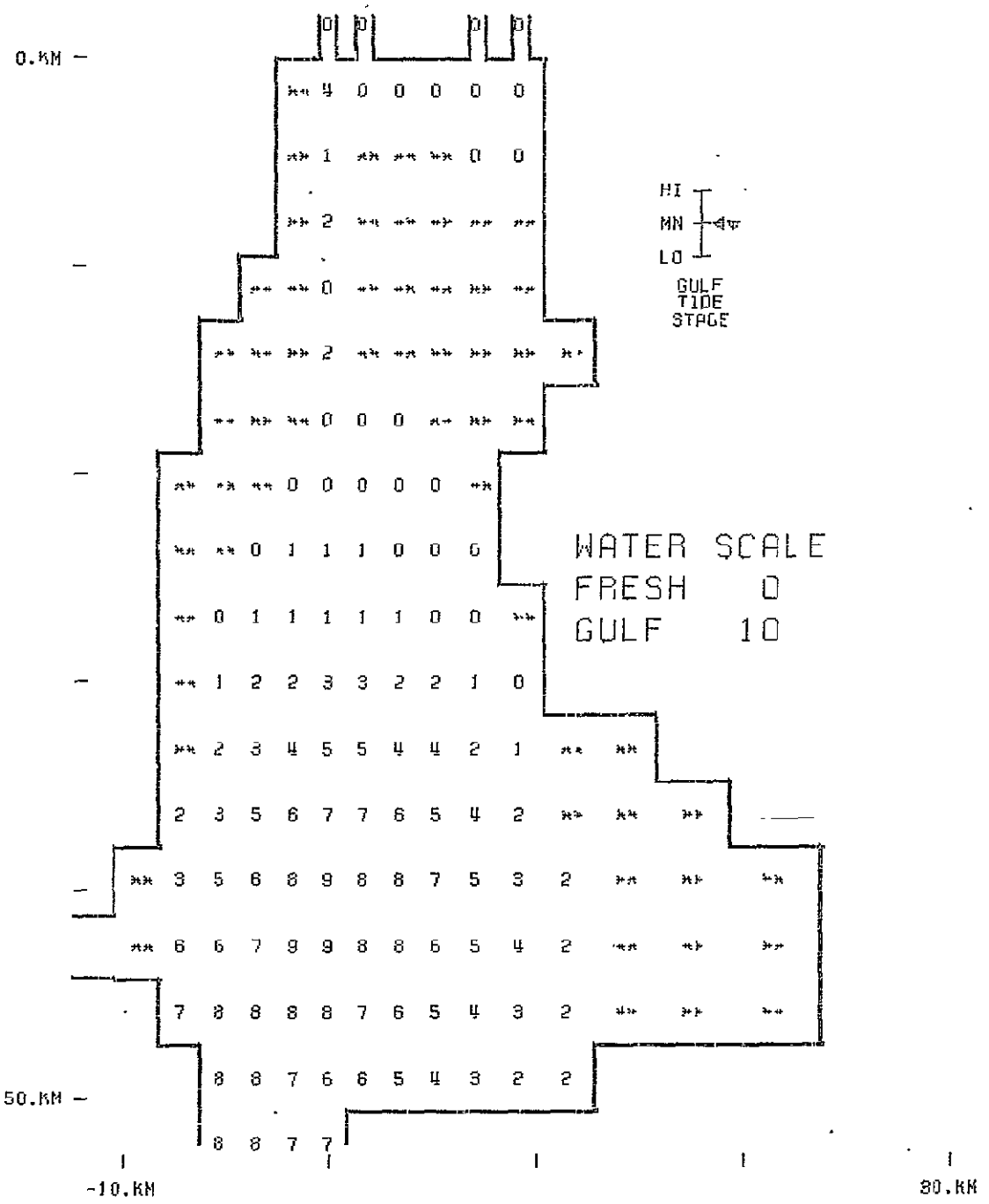
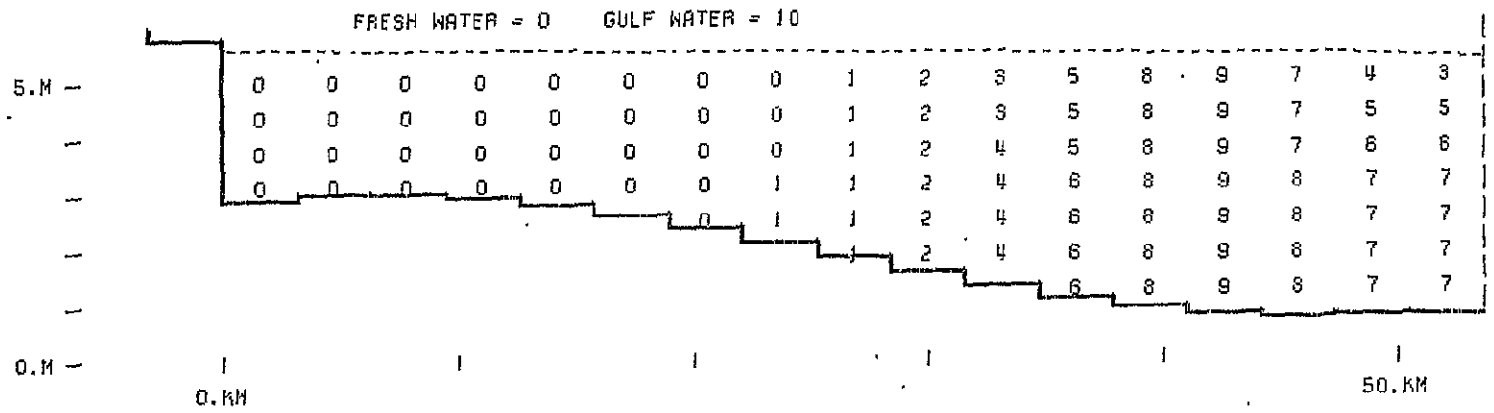


Figure A.67 - Case 2



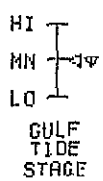


Figure A.68 - Case 2

SALINITY PROFILE
N-S SECTION -1.70 KM WRT CHANNEL
TIDE PERIOD: 25.00 HR ELAPSED TIME: 131.25 HR

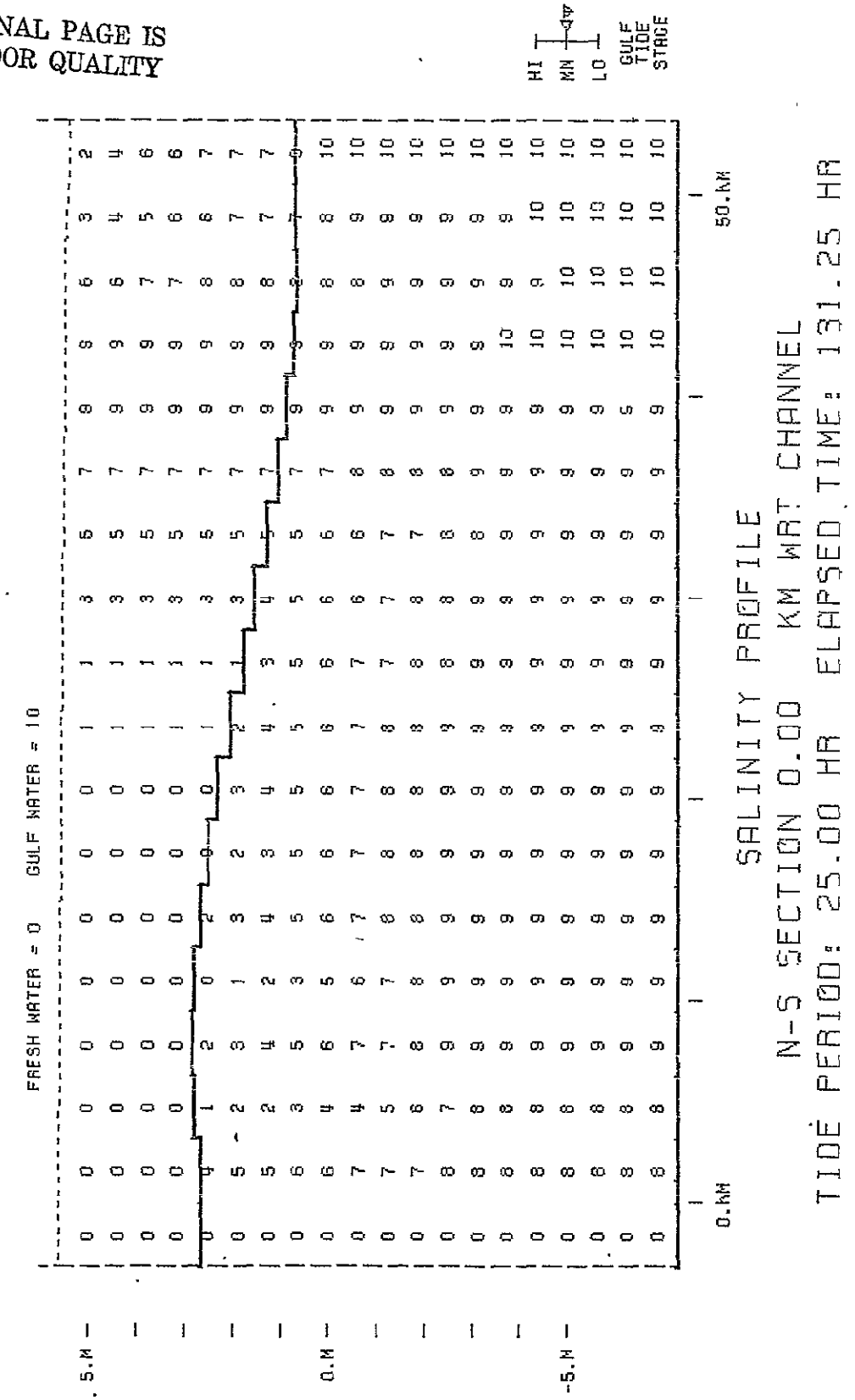


Figure A.69 - Case 2

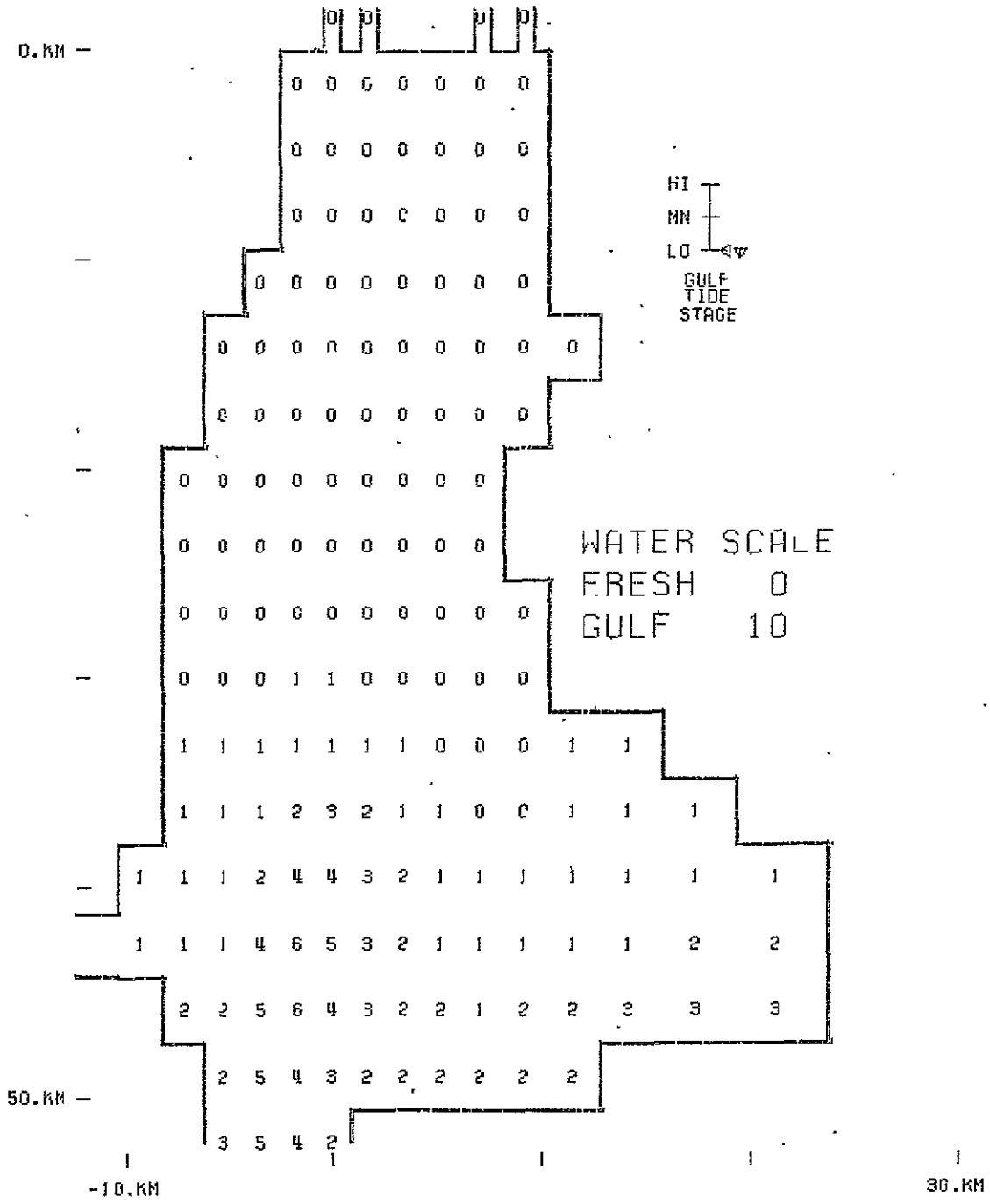


Figure A.70 - Case 2

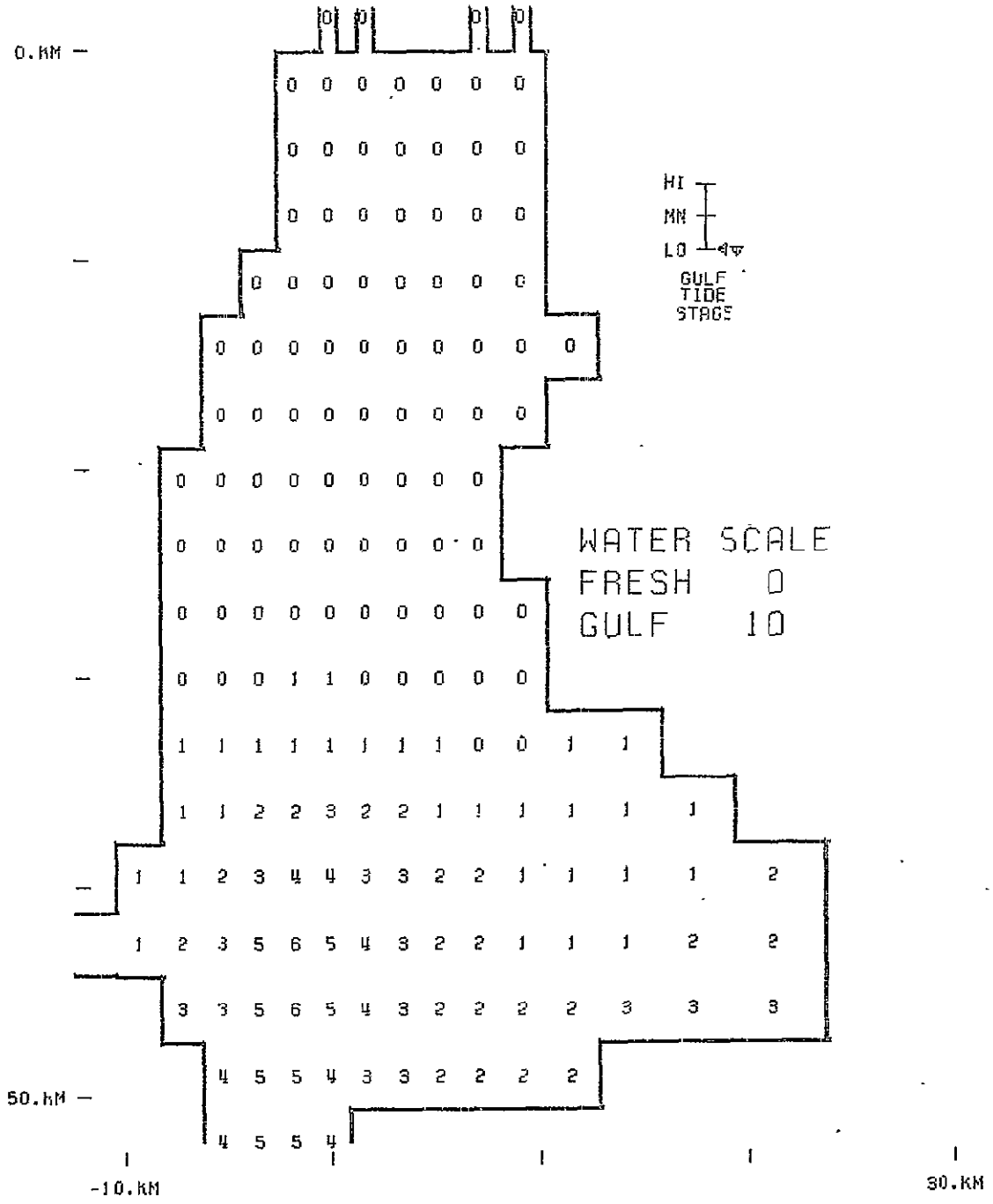


Figure A.71 -

Case 2

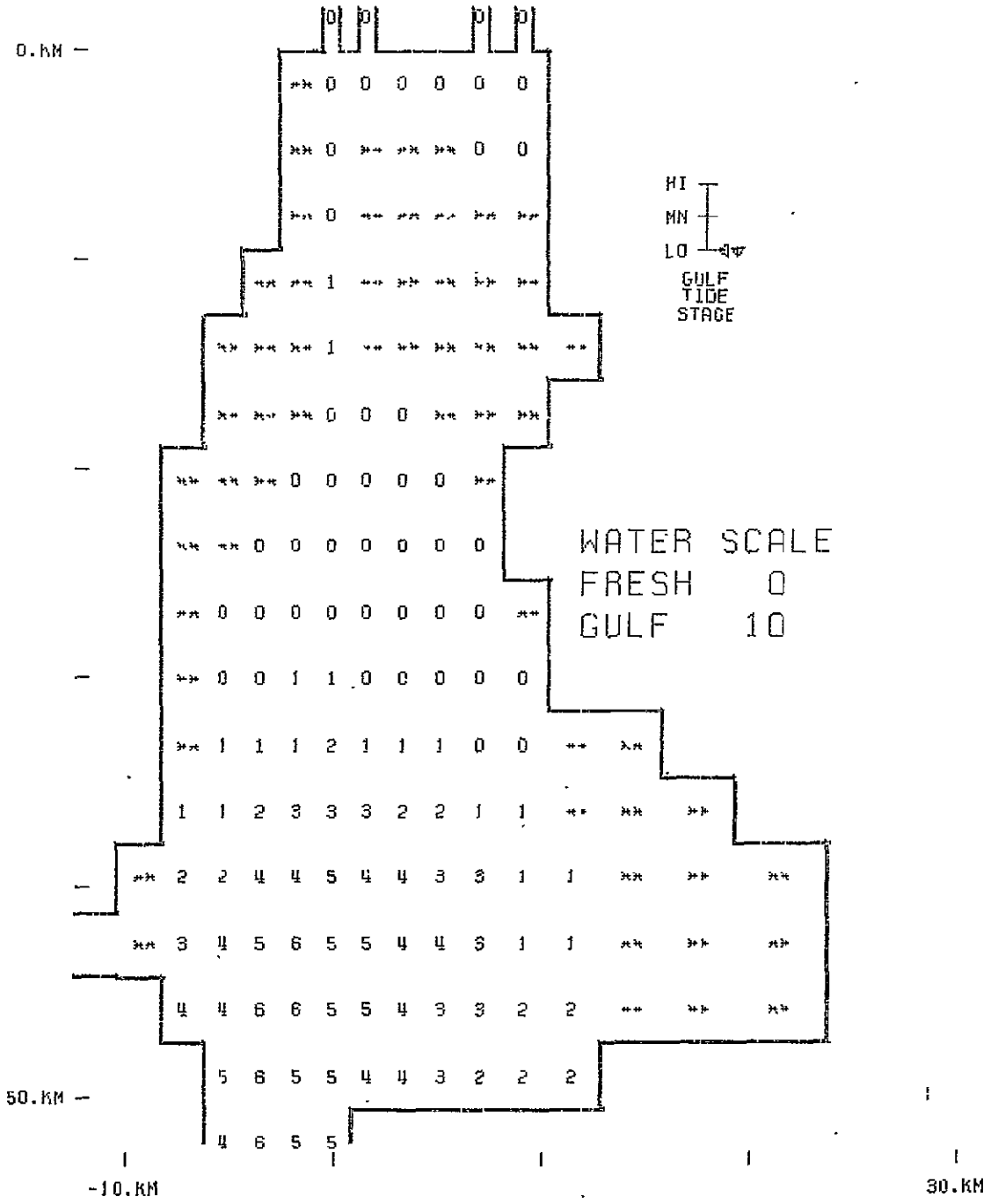
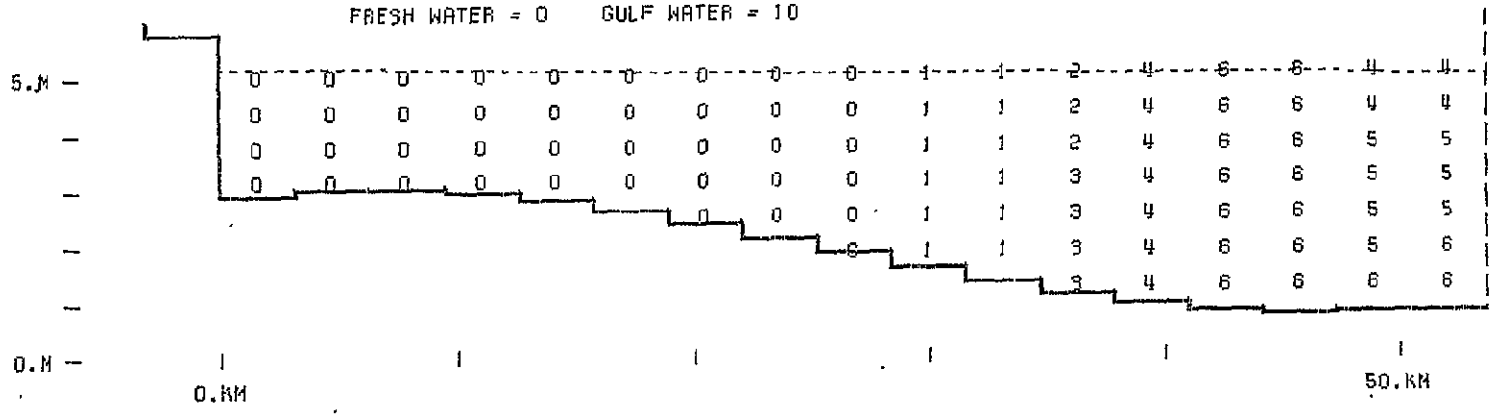


Figure A.72 -

Case 2



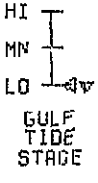


Figure A.73 - Case 2

SALINITY PROFILE
N-S SECTION -1.70 KM WRT CHANNEL
TIDE PERIOD: 25.00 HR ELAPSED TIME: 137.50 HR

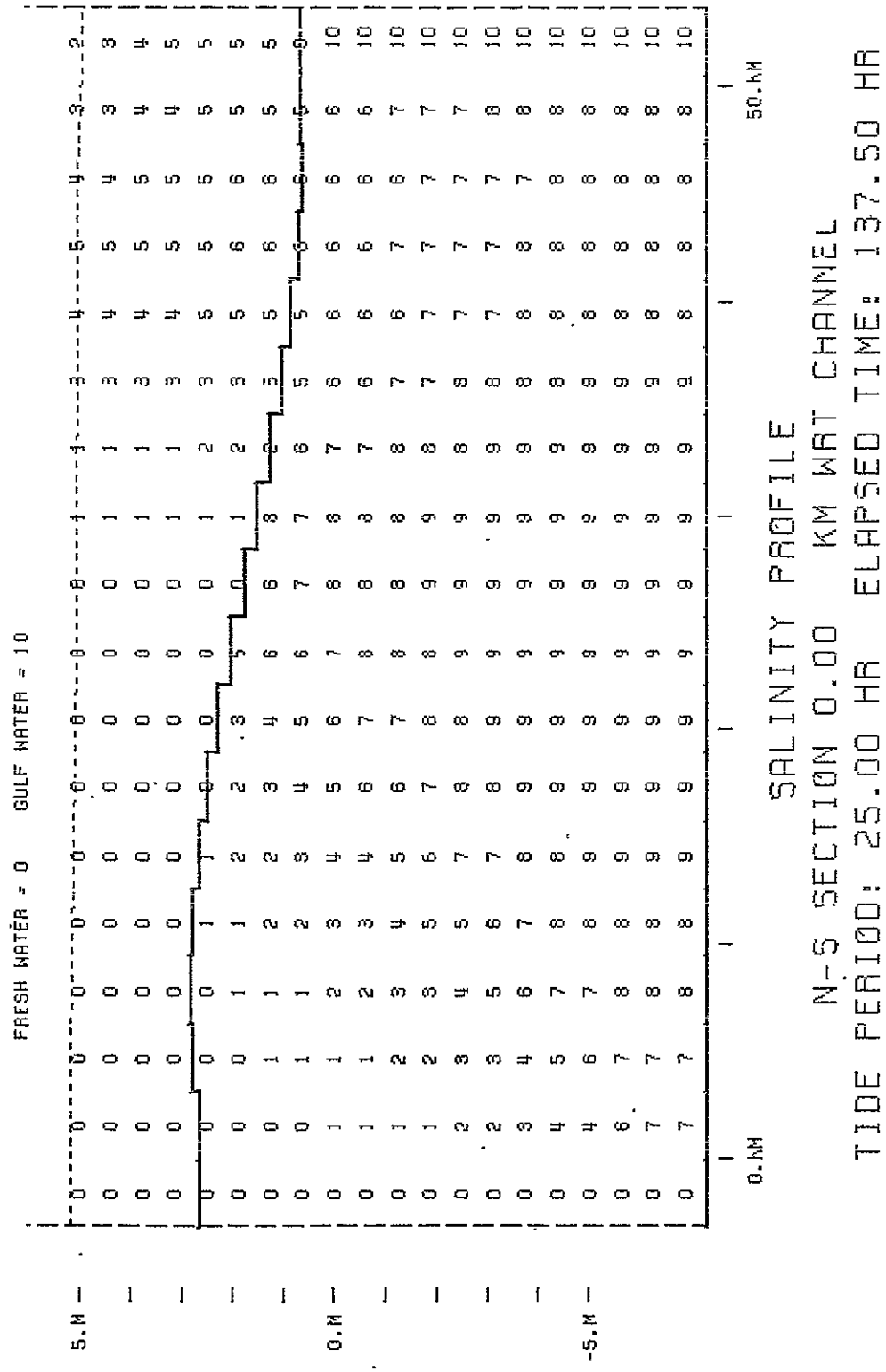


Figure A.74 - Case 2

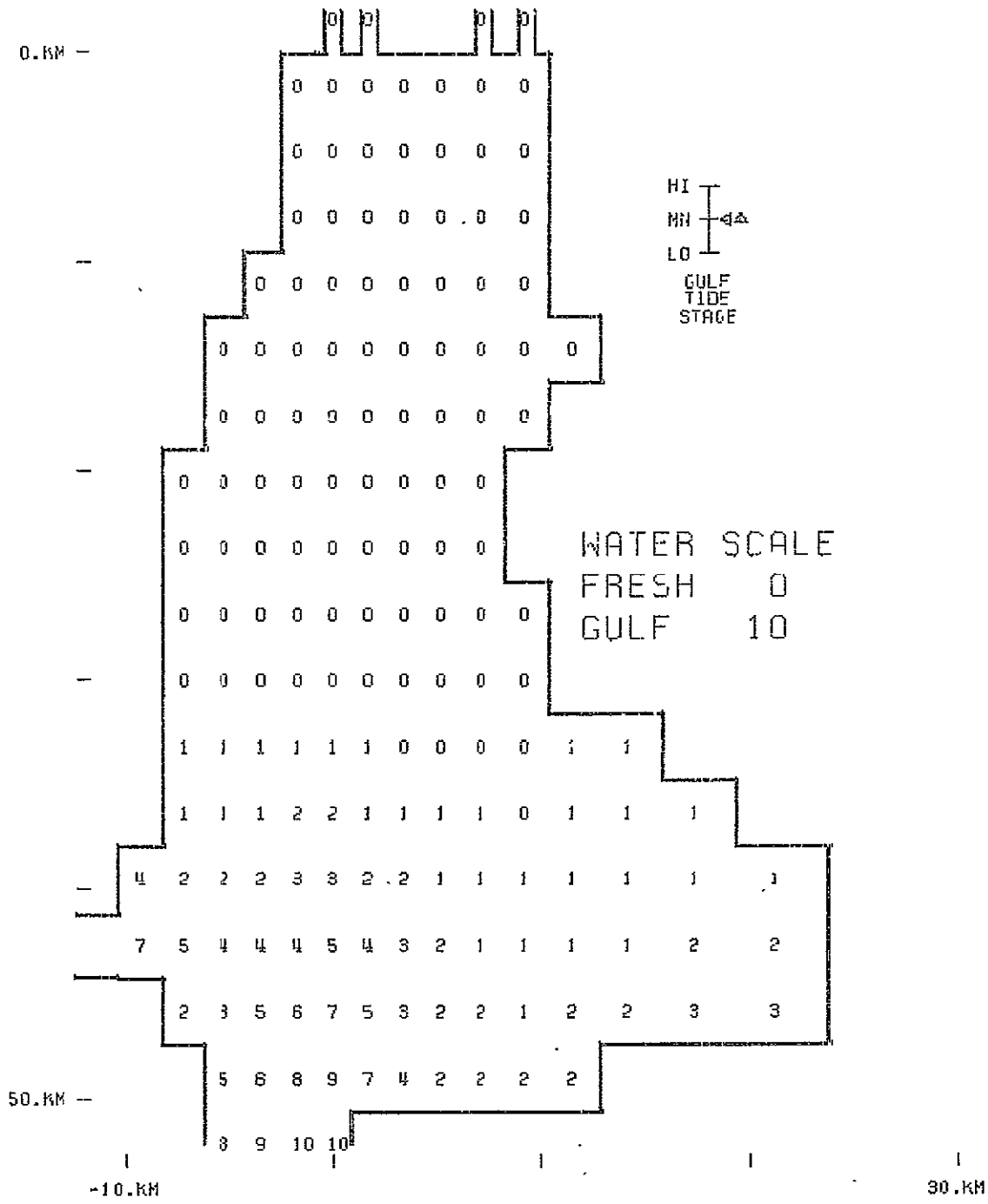


Figure A.75 -

Case 2

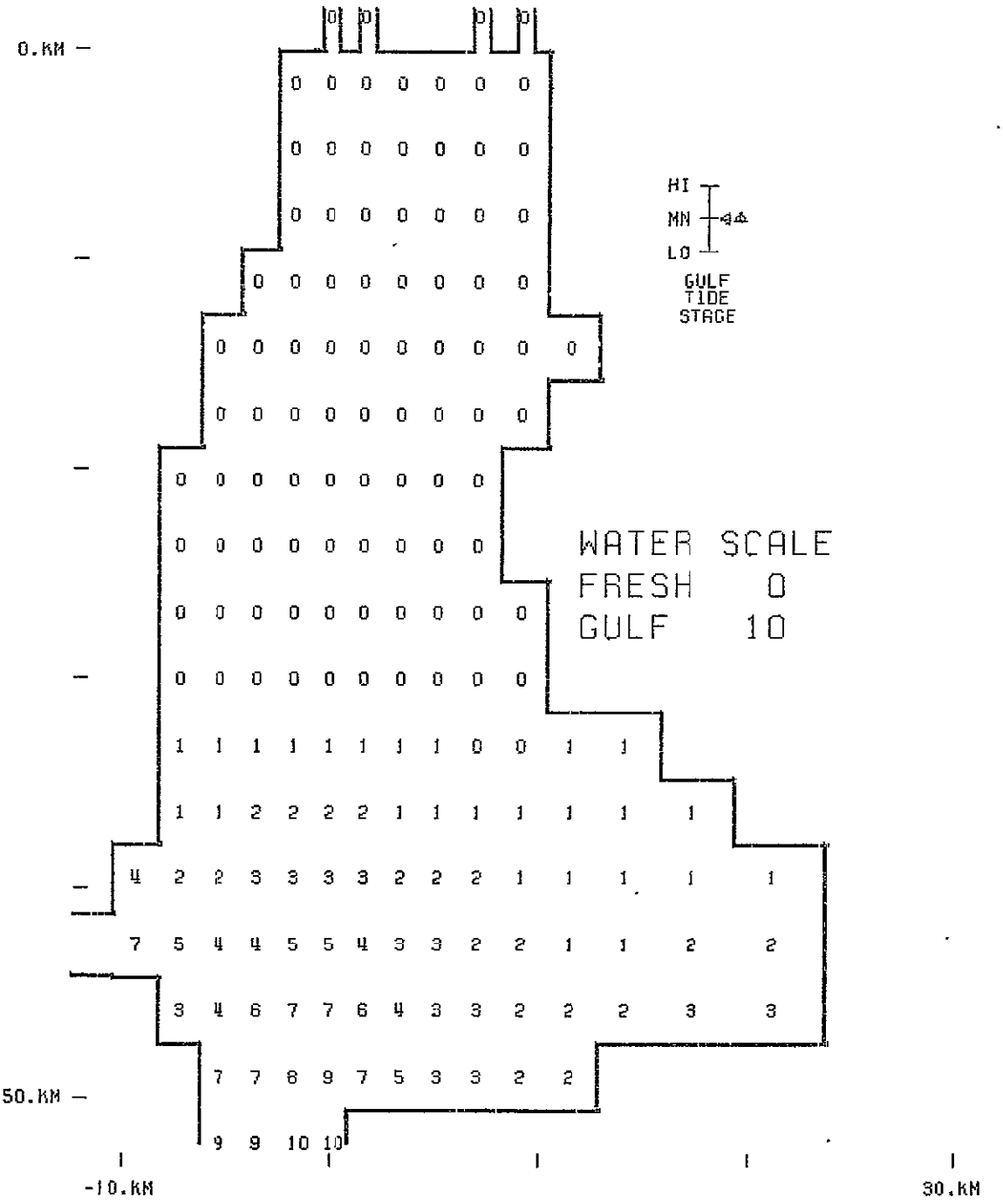


Figure A.76 -

Case 2

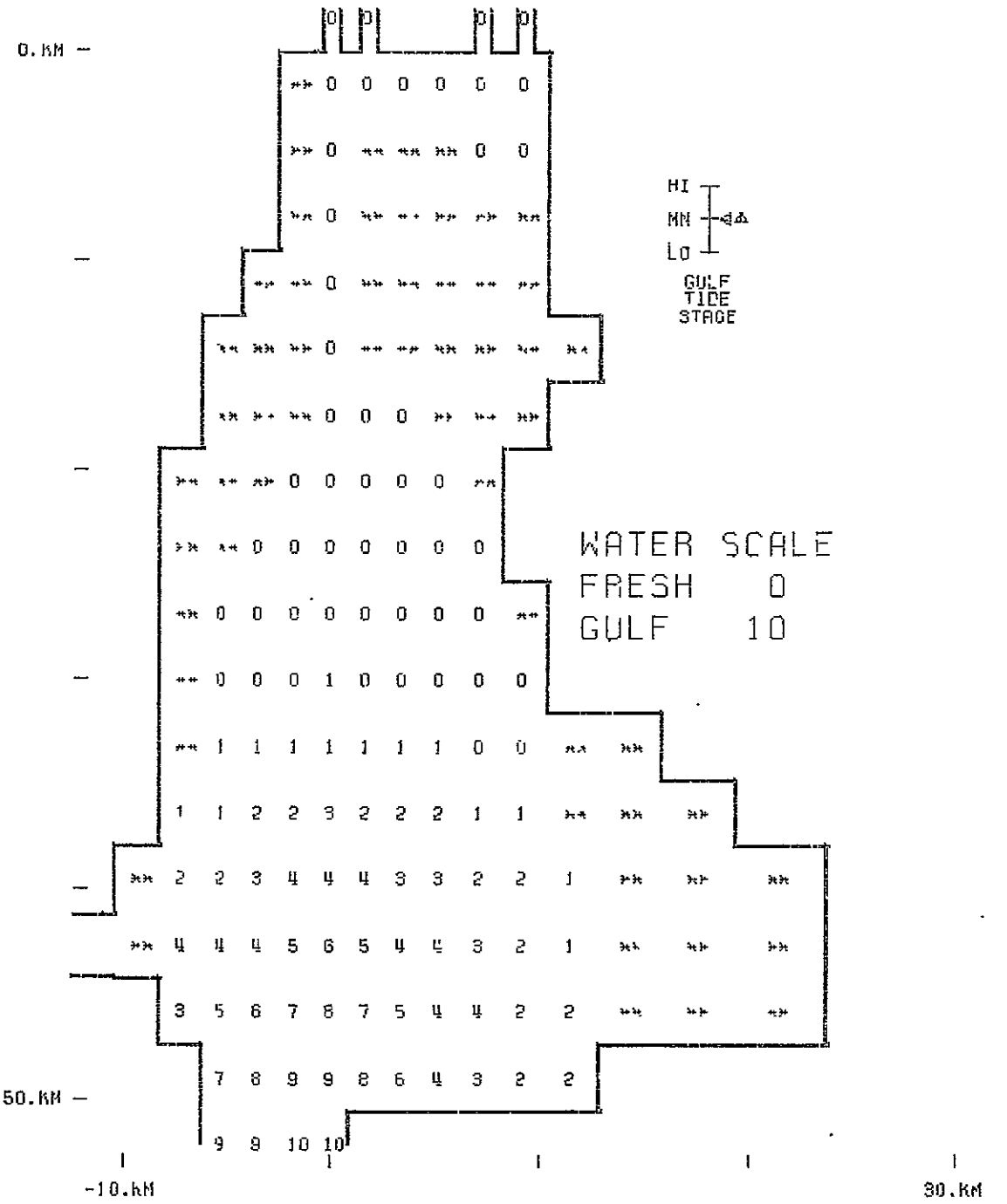


Figure A.77 -

Case 2

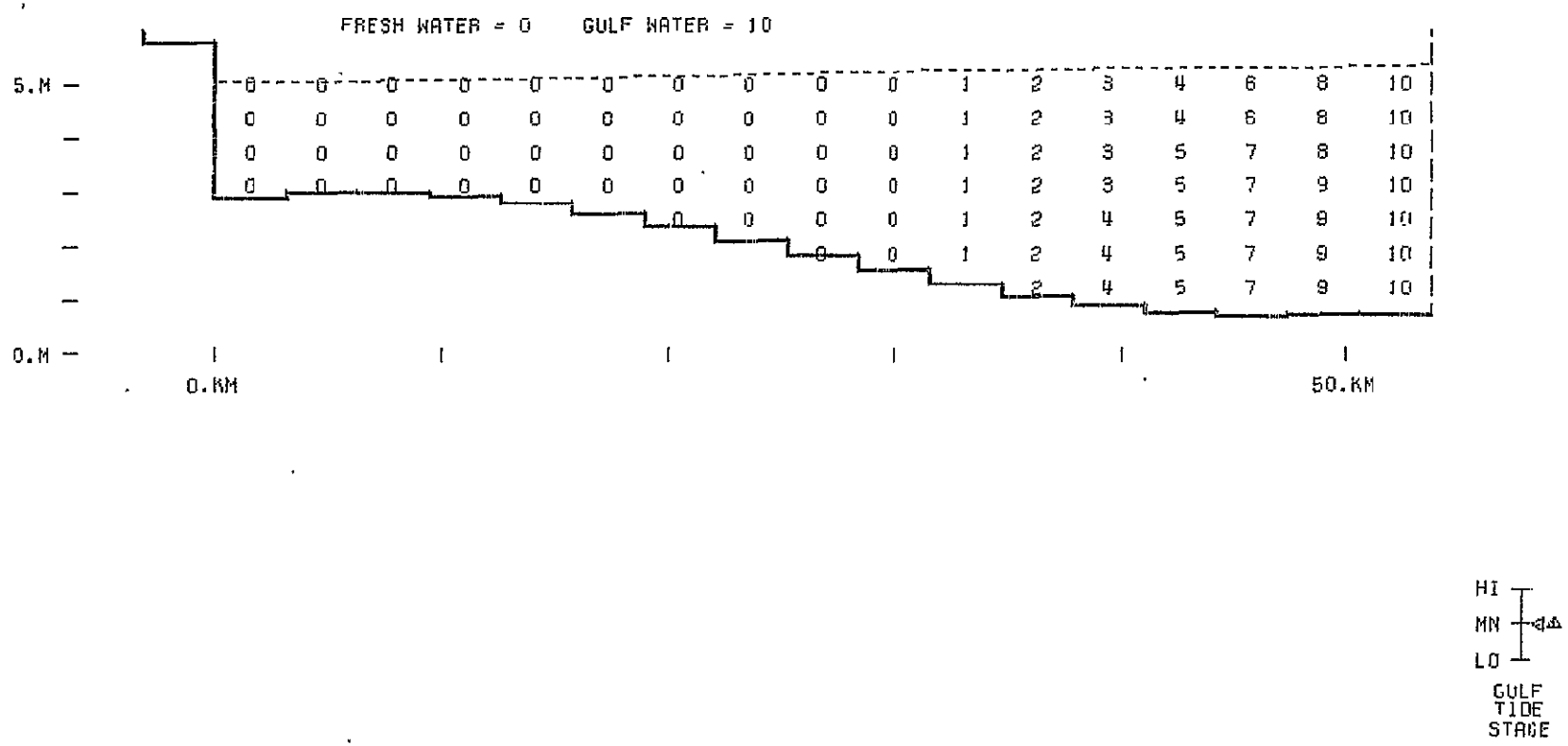


Figure A.78 - Case 2

SALINITY PROFILE
N-S SECTION -1.70 KM WRT CHANNEL
TIDE PERIOD: 25.00 HR ELAPSED TIME: 143.75 HR

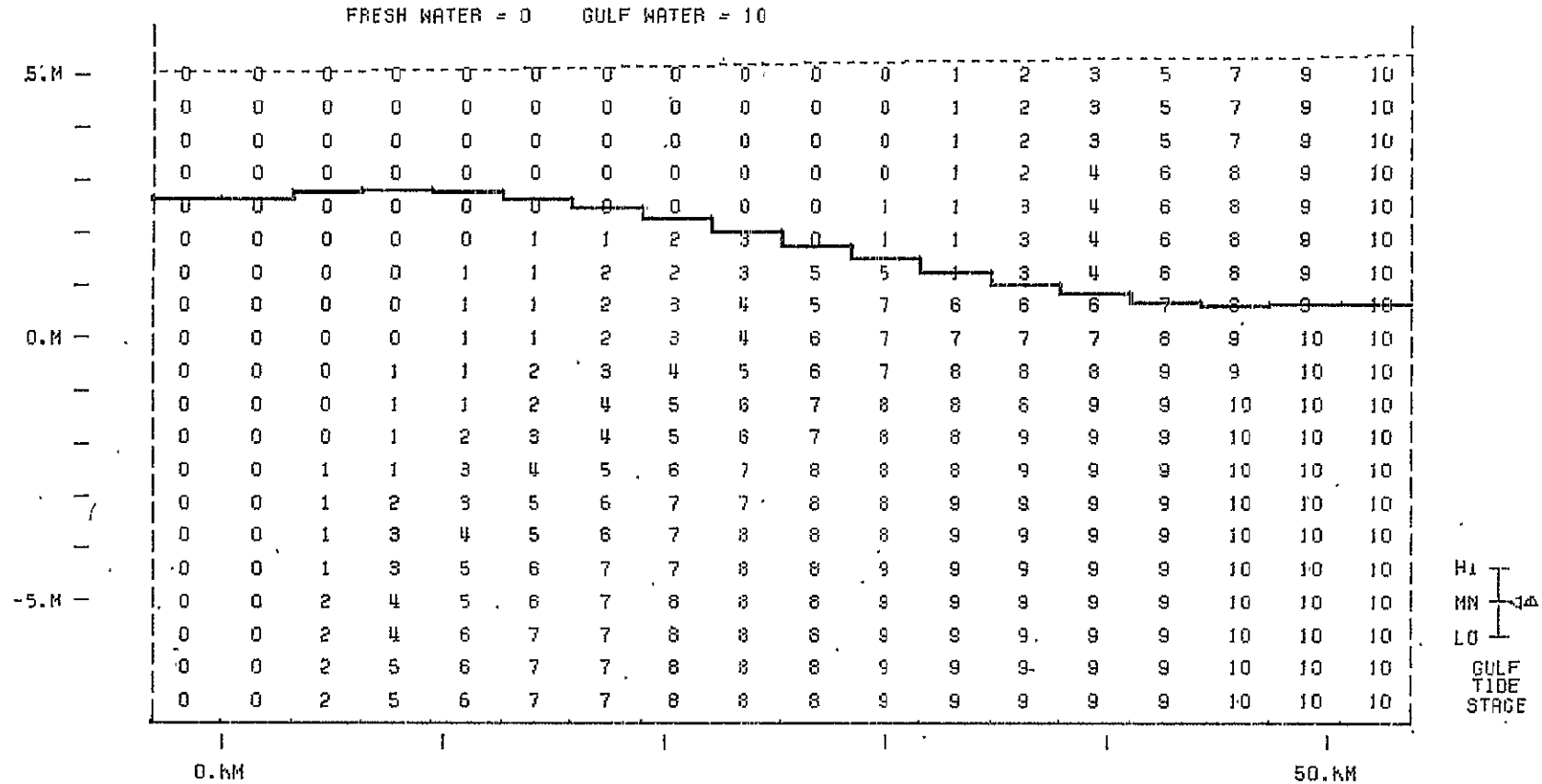


Figure A.79 - Case 2

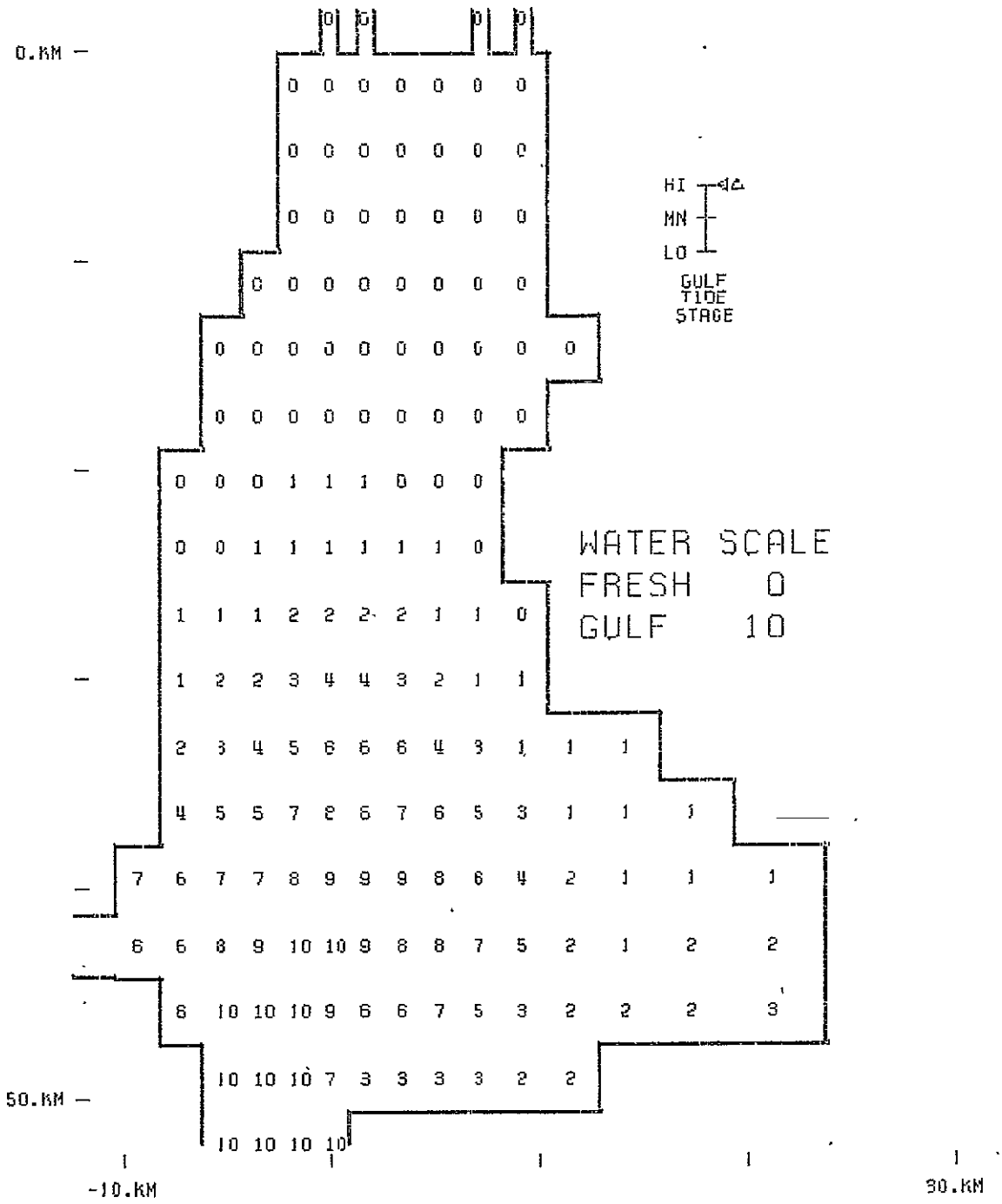


Figure A.80 -

Case 2

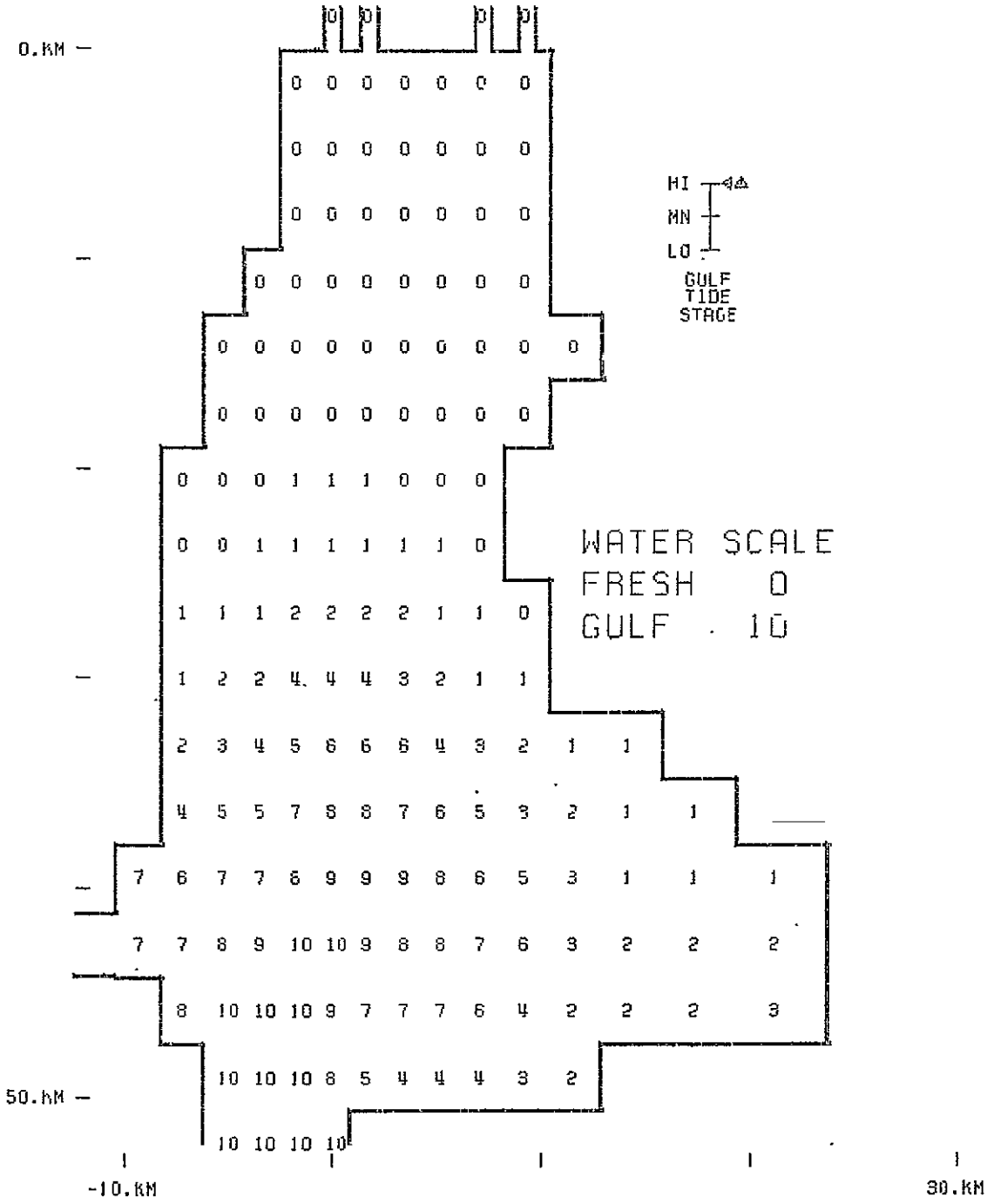


Figure A.81 - Case 2

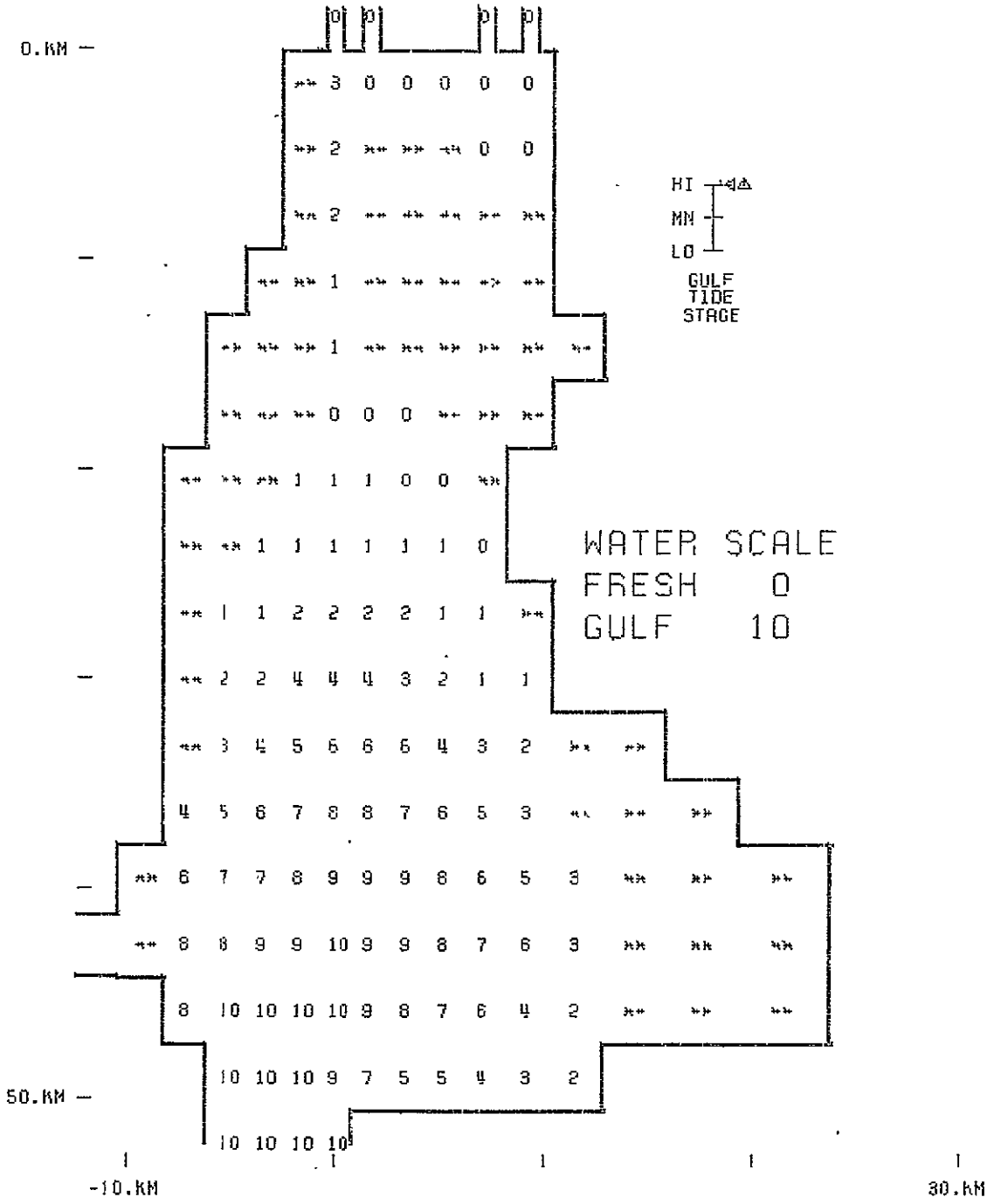
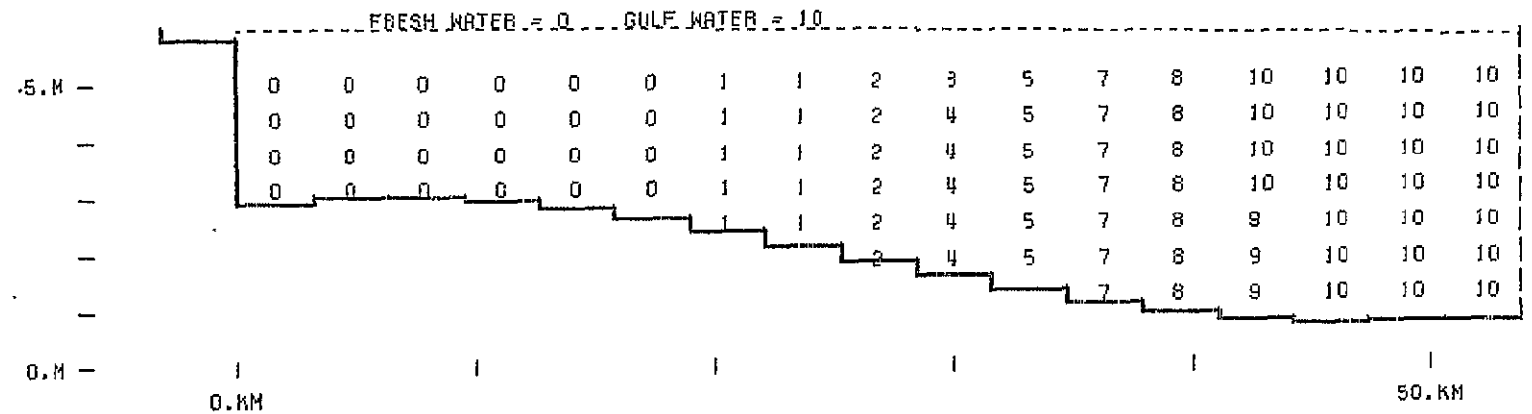
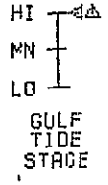


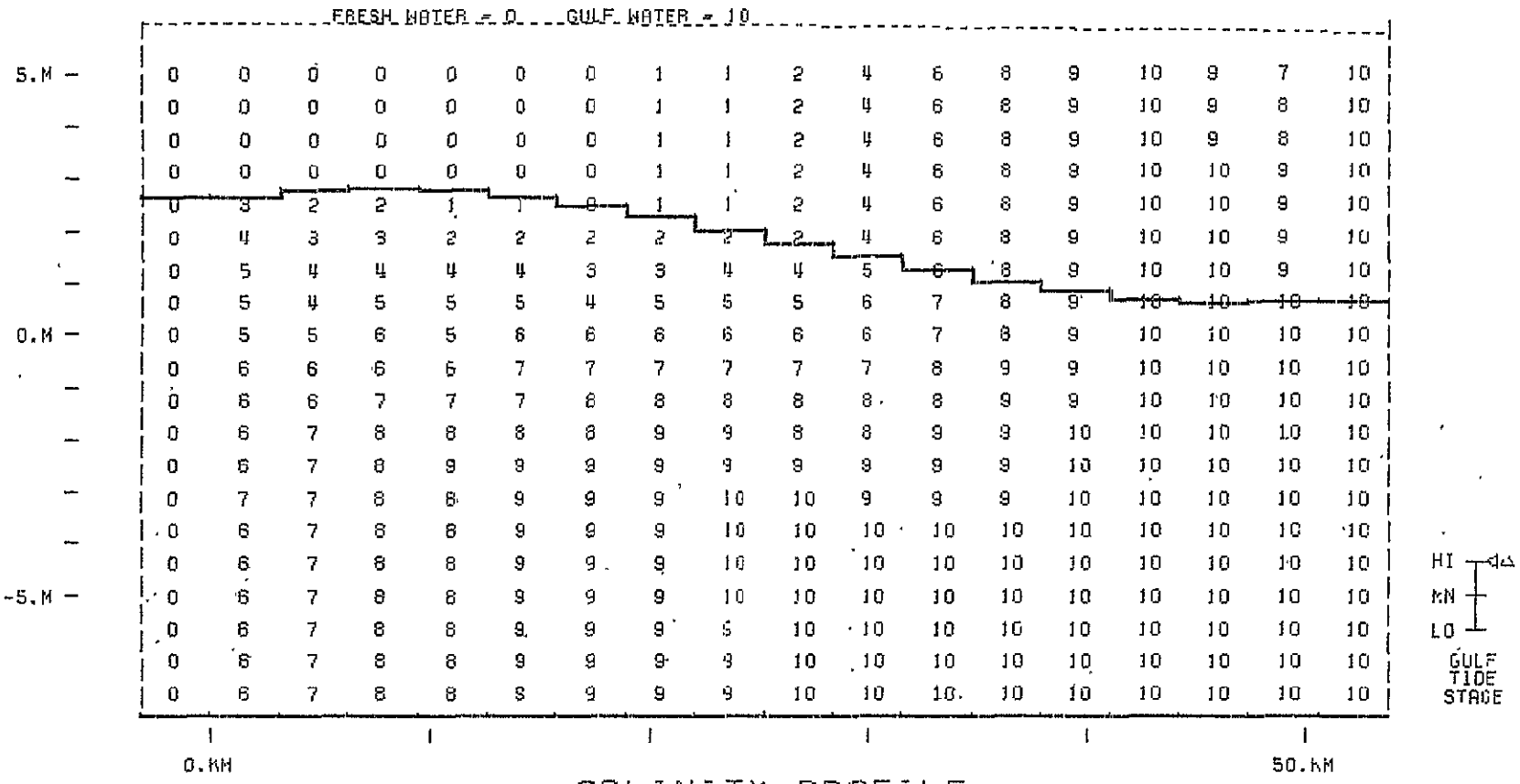
Figure A.82 - Case 2





SALINITY PROFILE
N-S SECTION -1.70 KM WRT CHANNEL
TIDE PERIOD: 25.00 HR ELAPSED TIME: 150.00 HR

Figure A.83 - Case 2



SALINITY PROFILE
N-S SECTION 0.00 KM WRT CHANNEL
TIDE PERIOD: 25.00 HR ELAPSED TIME: 150.00 HR

Figure A.84 - Case 2

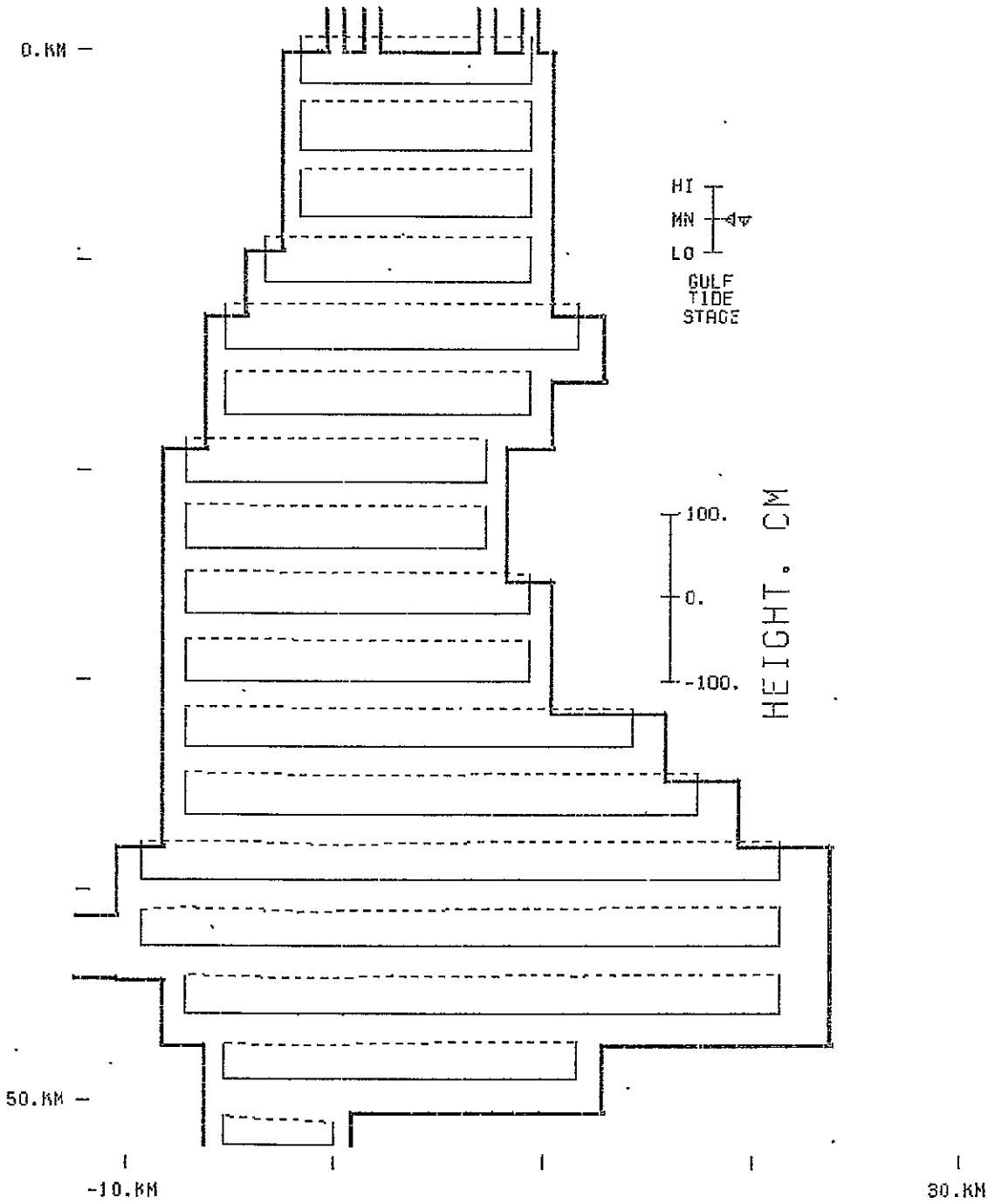


Figure A.85 - Case 2

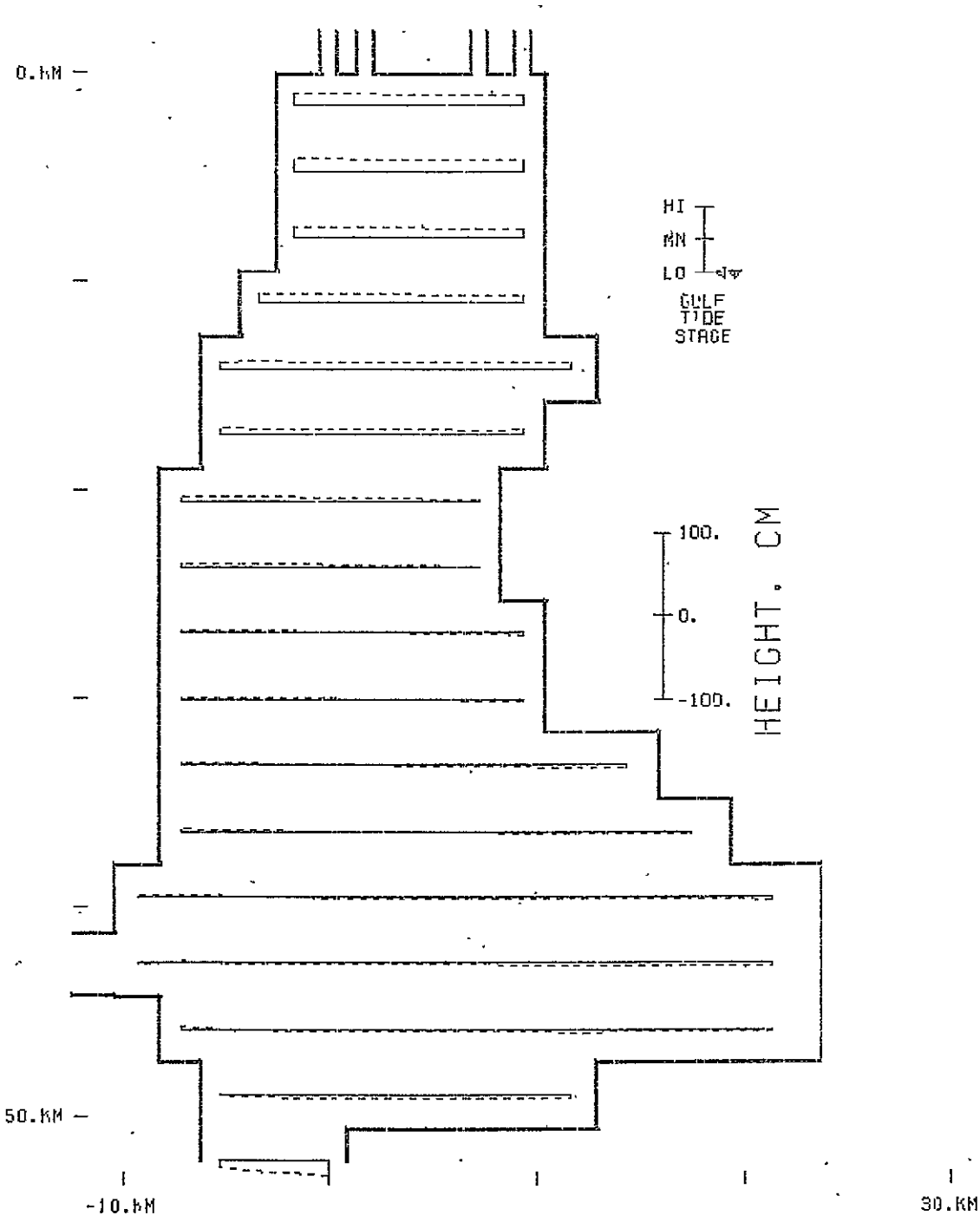


Figure A.86 -

Case 2

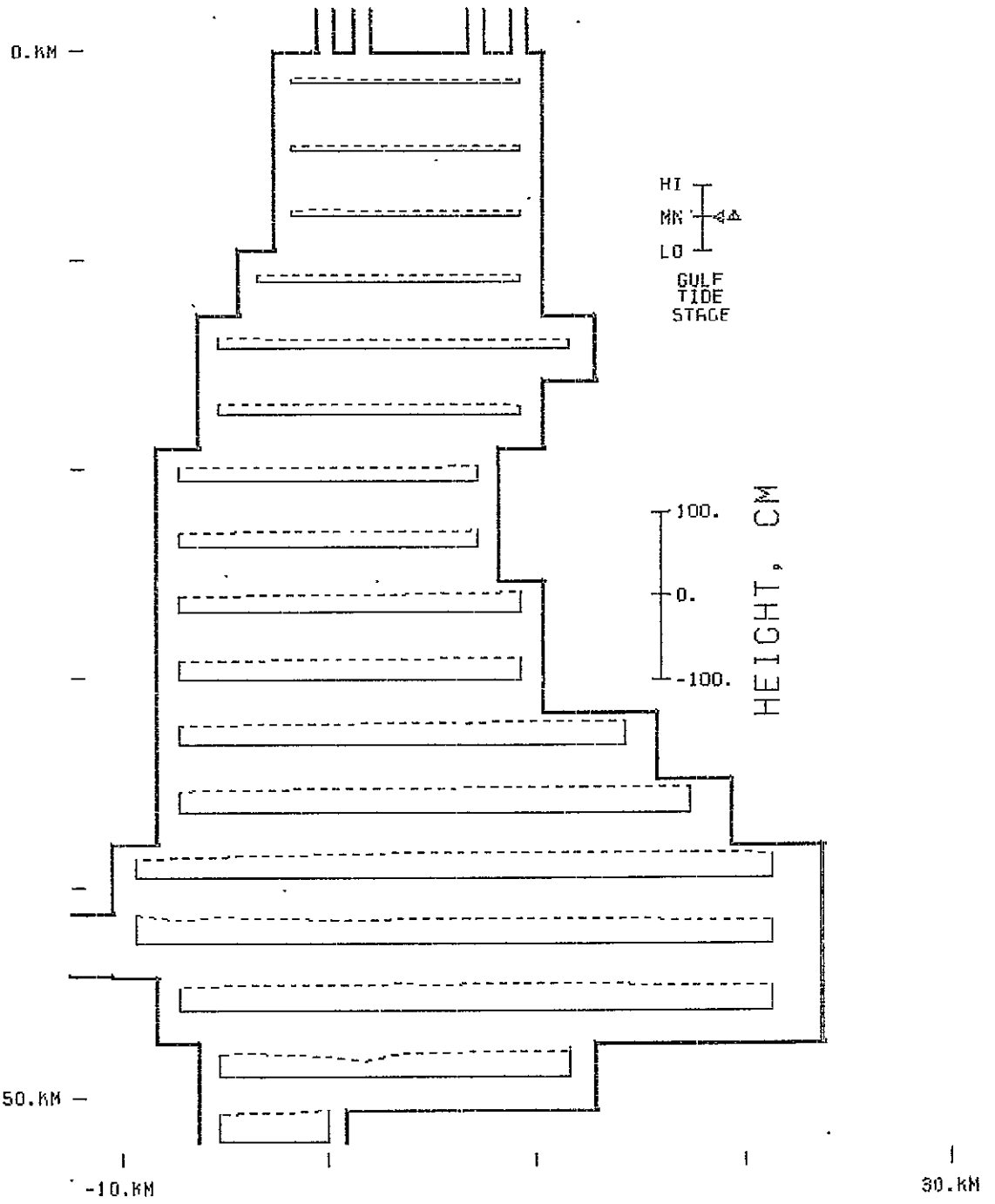


Figure A.87 -

Case 2

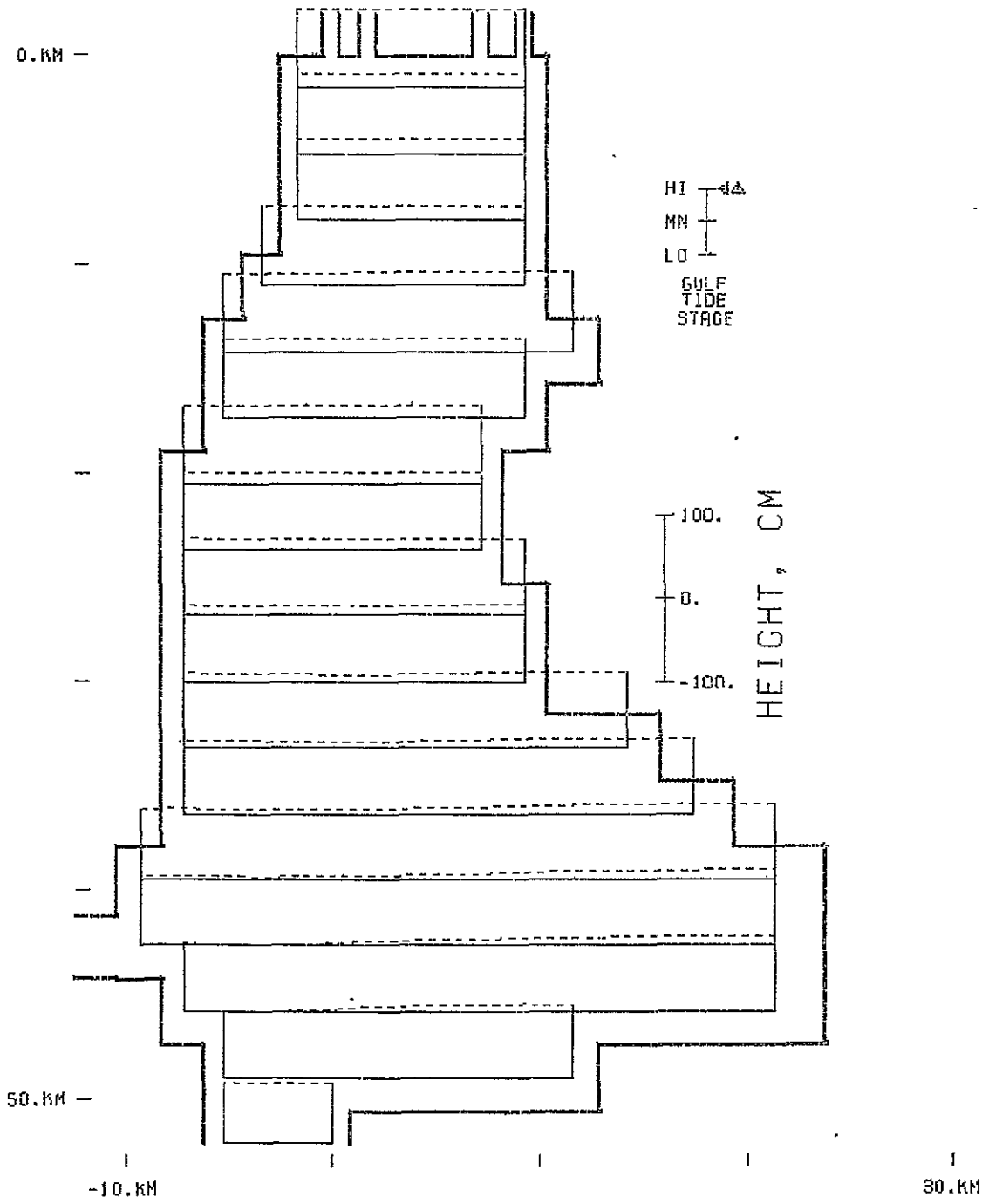


Figure A.88 -

Case 2

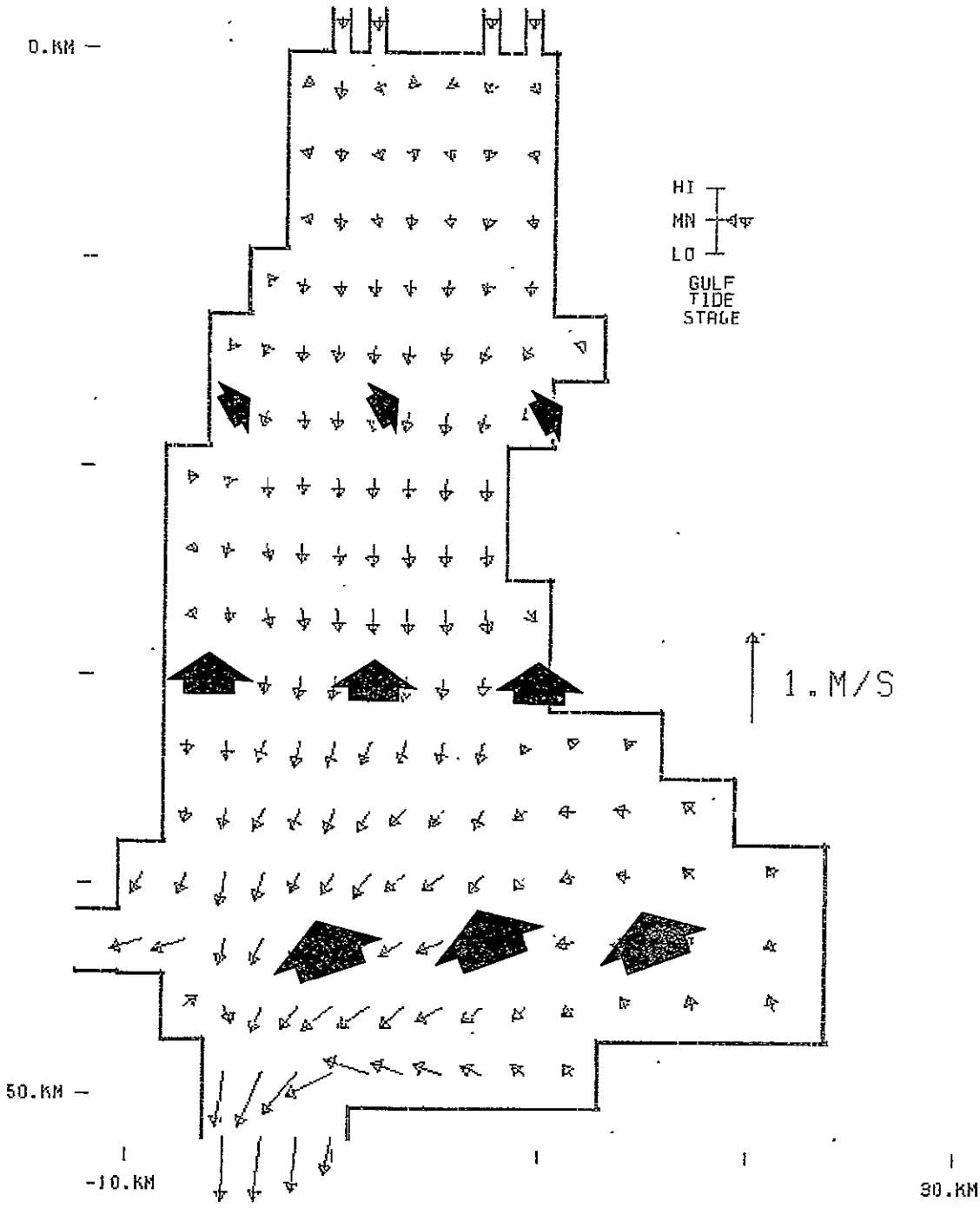


Figure A.89 -

Case 3

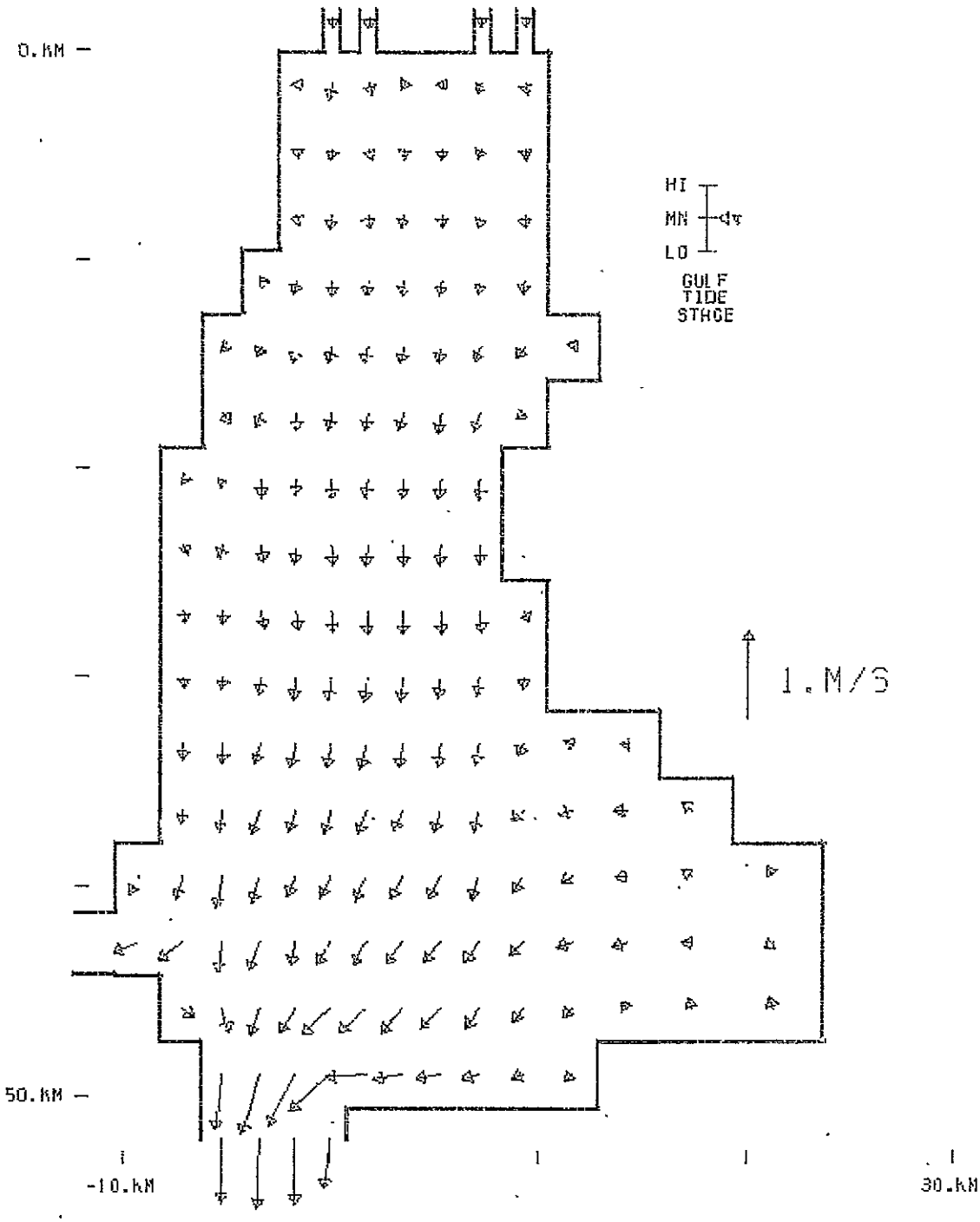


Figure A.90 -

Case 3

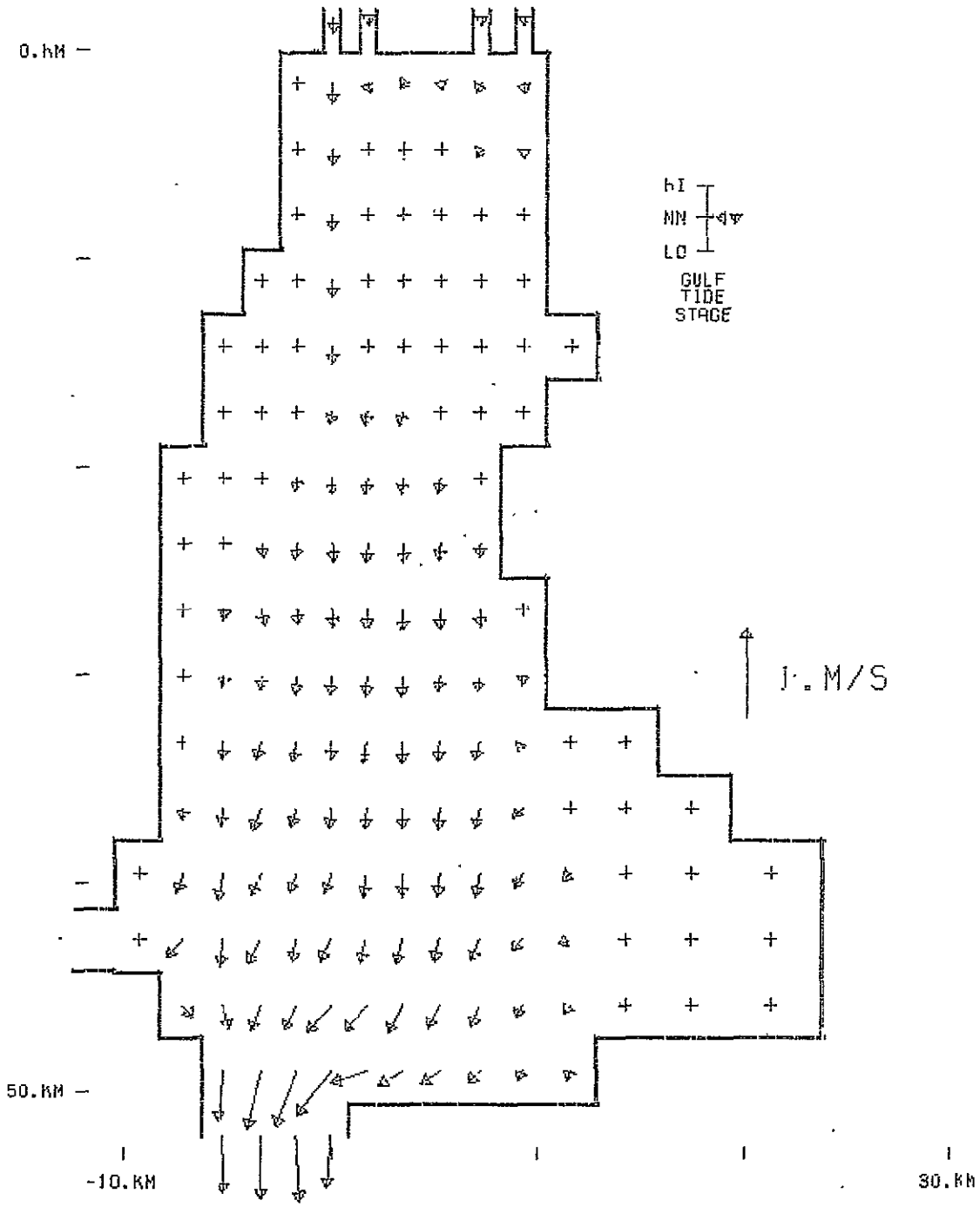
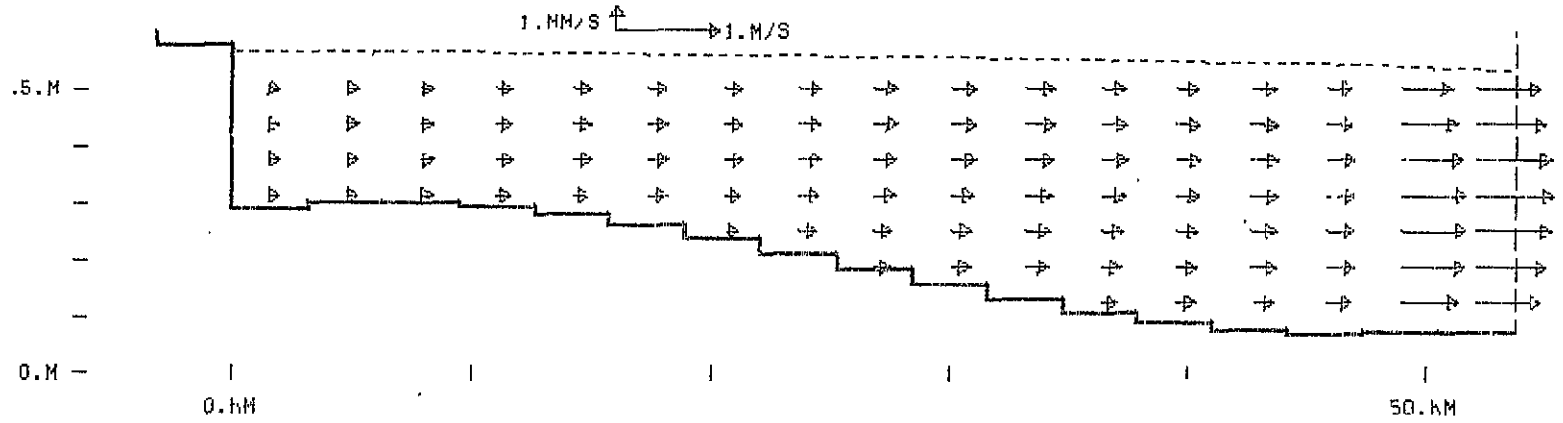
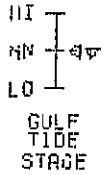


Figure A.91 -

Case 3





VELOCITY VECTORS
N-S SECTION -1.70 KM WRT CHANNEL
TIDE PERIOD: 25.00 HR ELAPSED TIME: 181.25 HR

Figure A.92 - Case 3

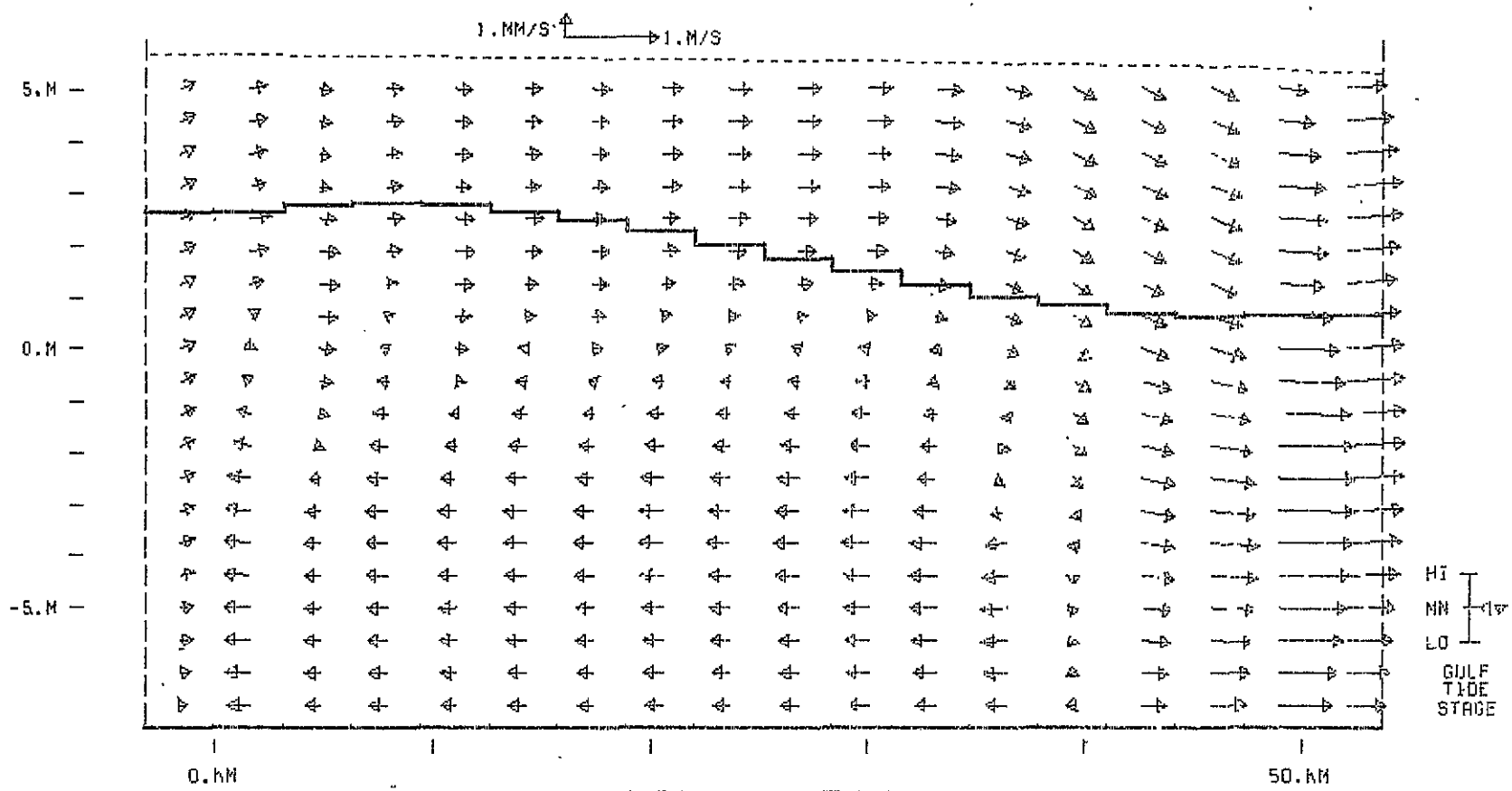


Figure A.93 - Case 3

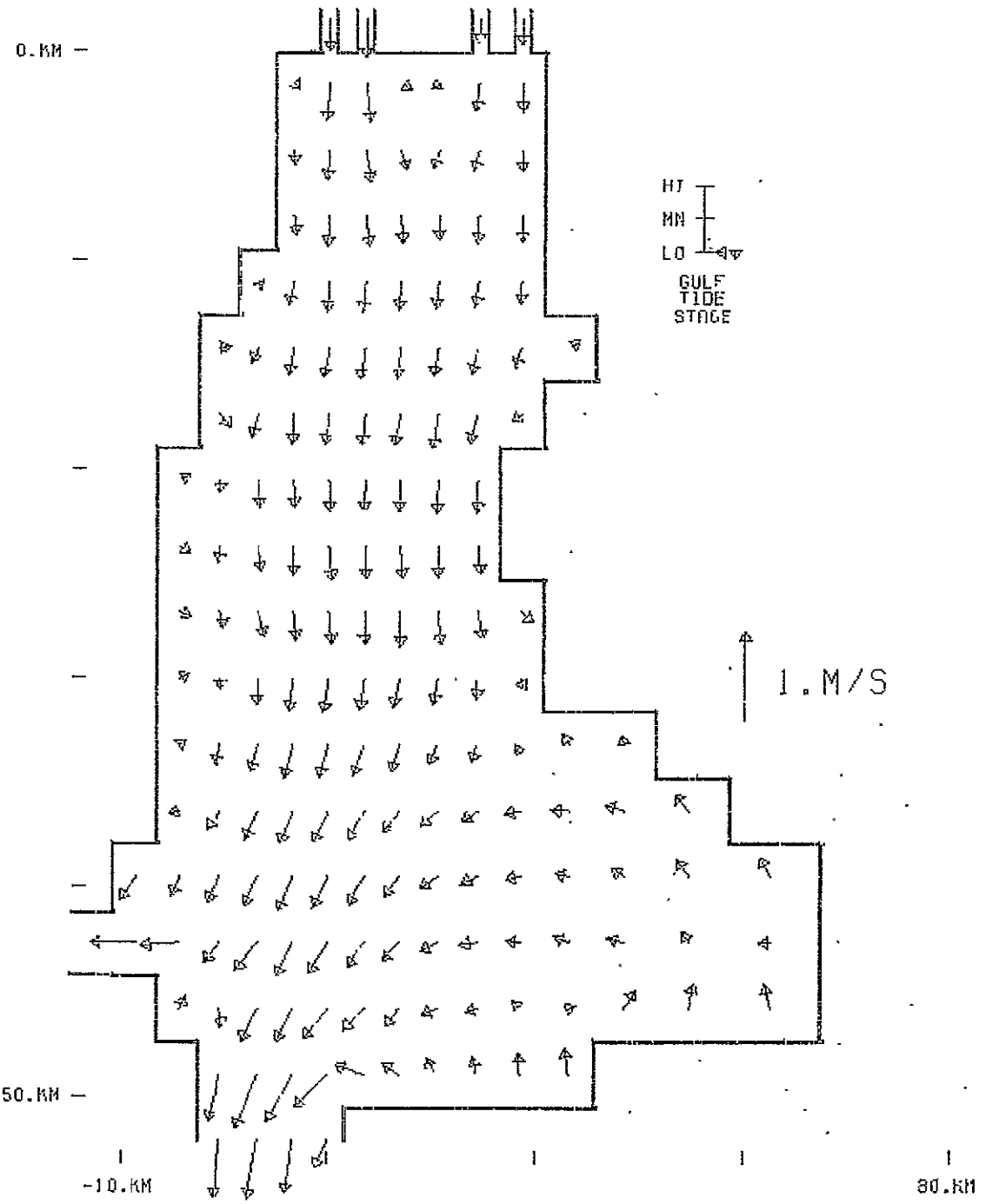


Figure A.94 -

Case 3

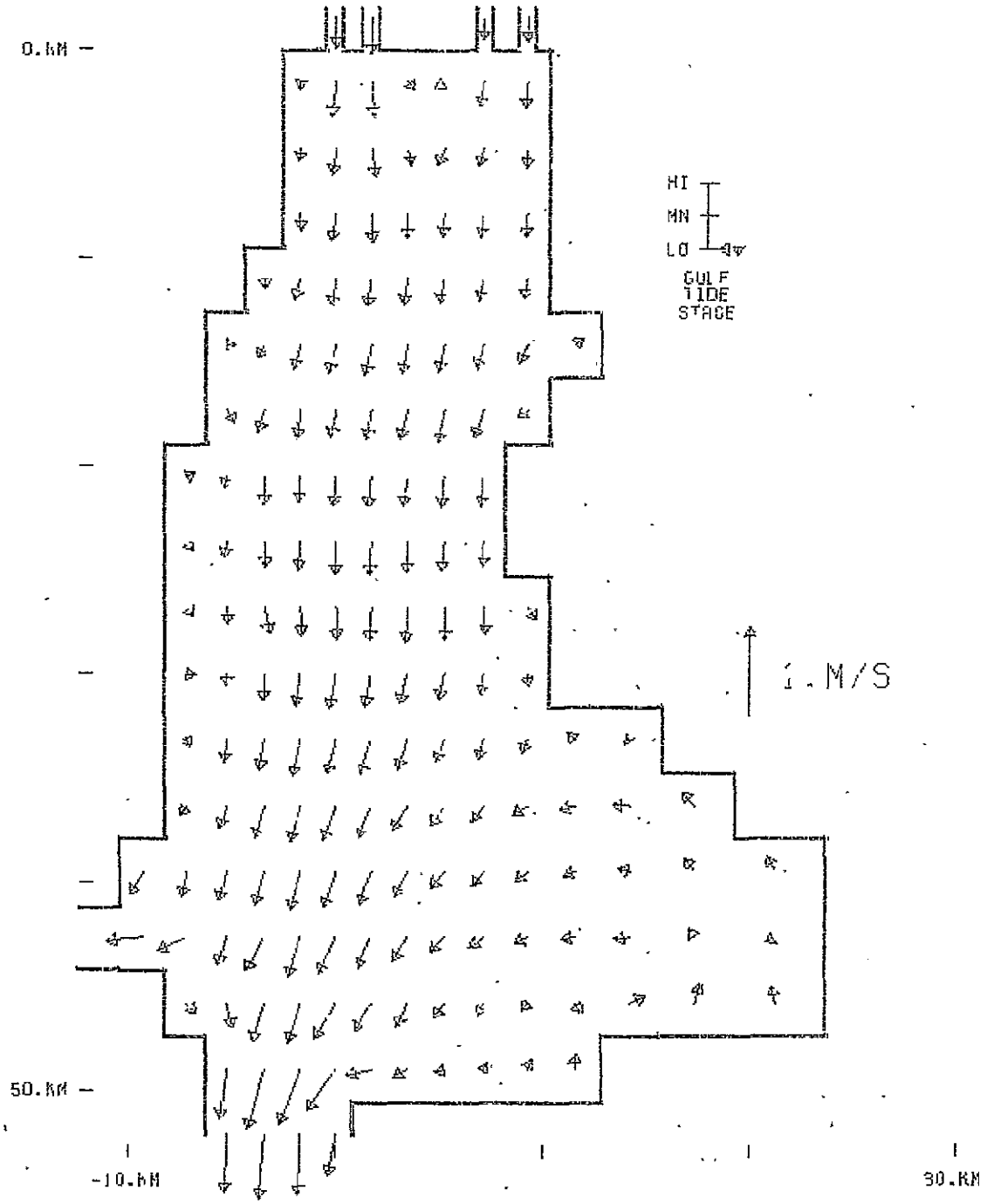


Figure A.95 -

Case 3

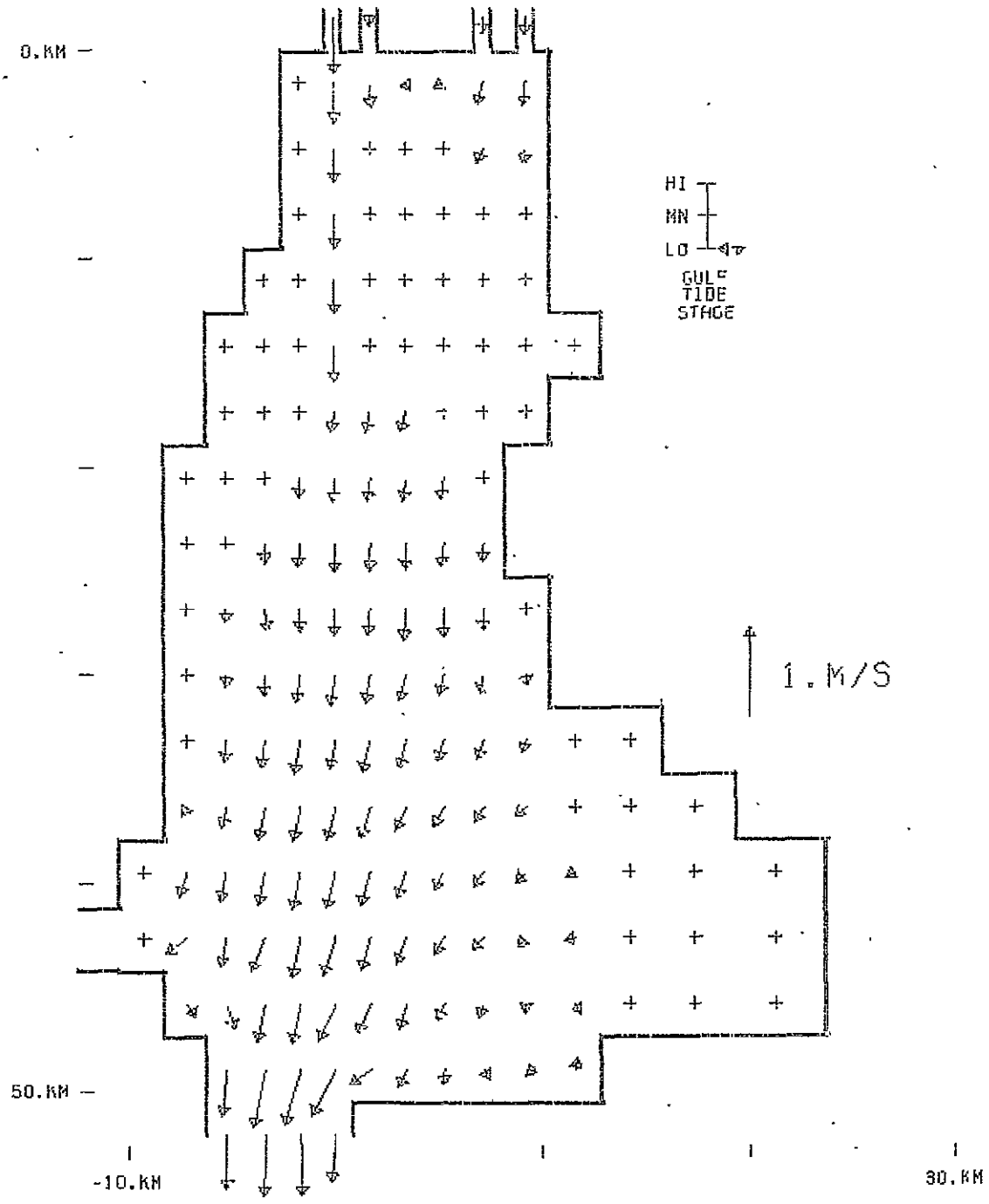


Figure A.96 -

Case 3

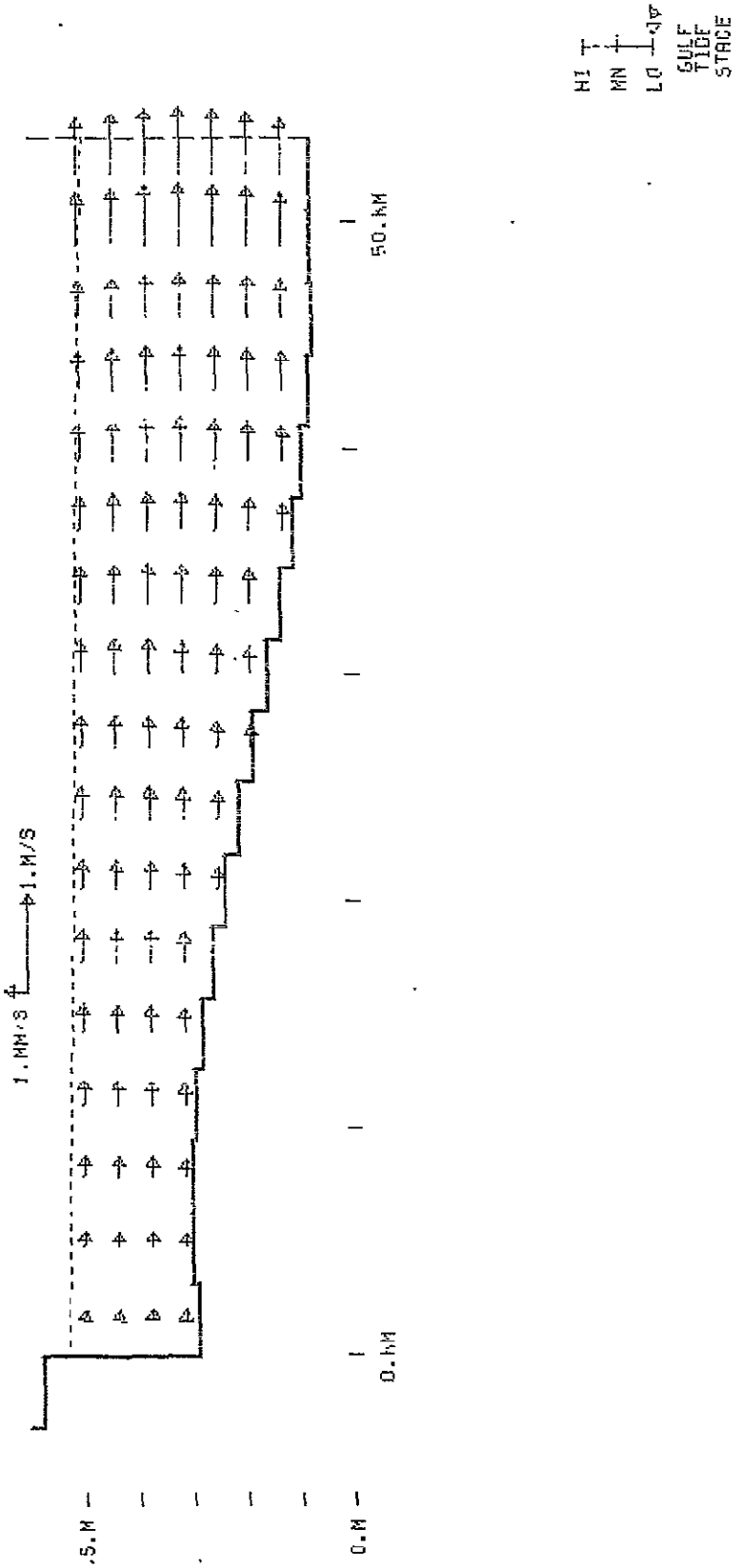


Figure A.97 - Case 3

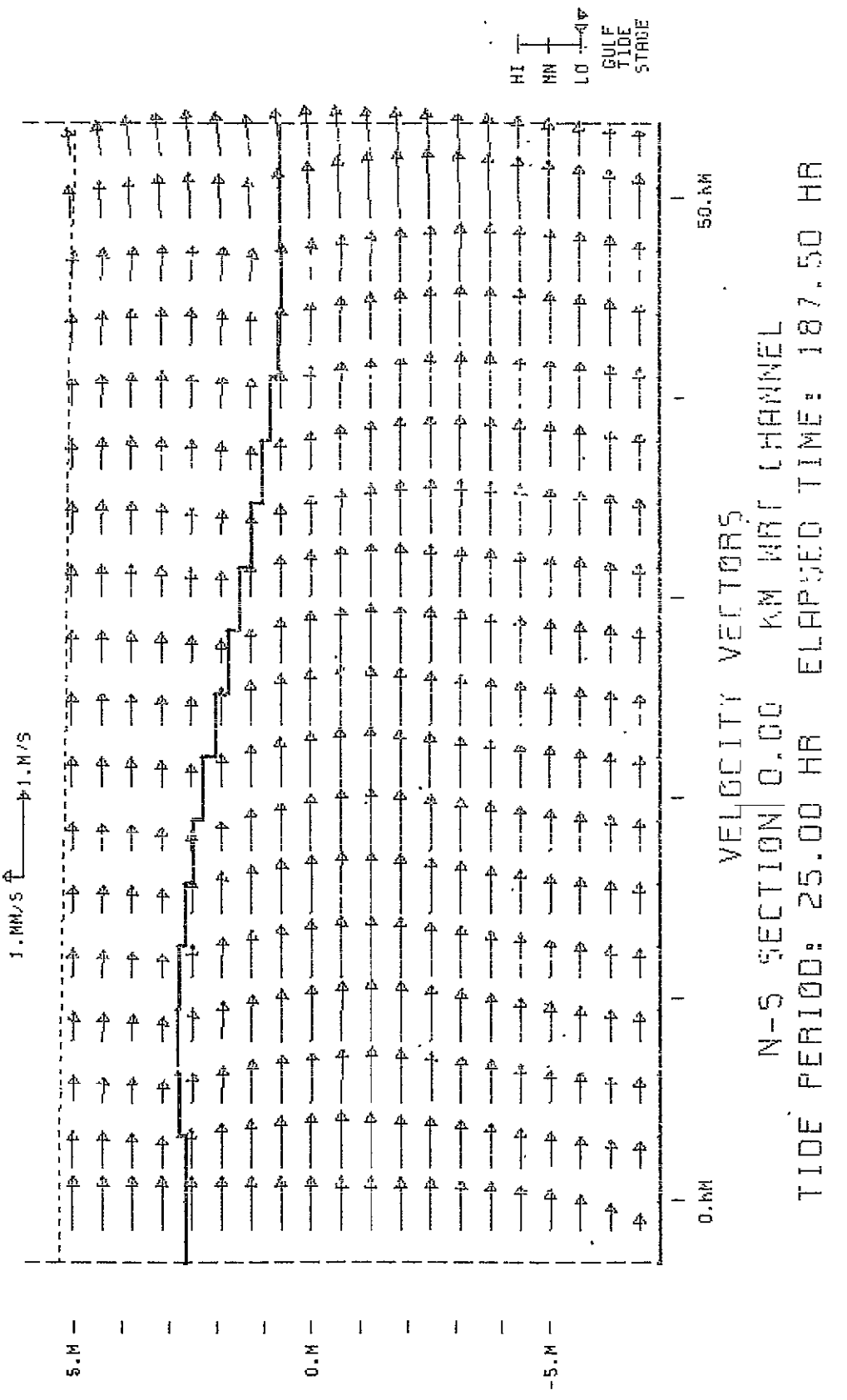


Figure A.98 - Case 3

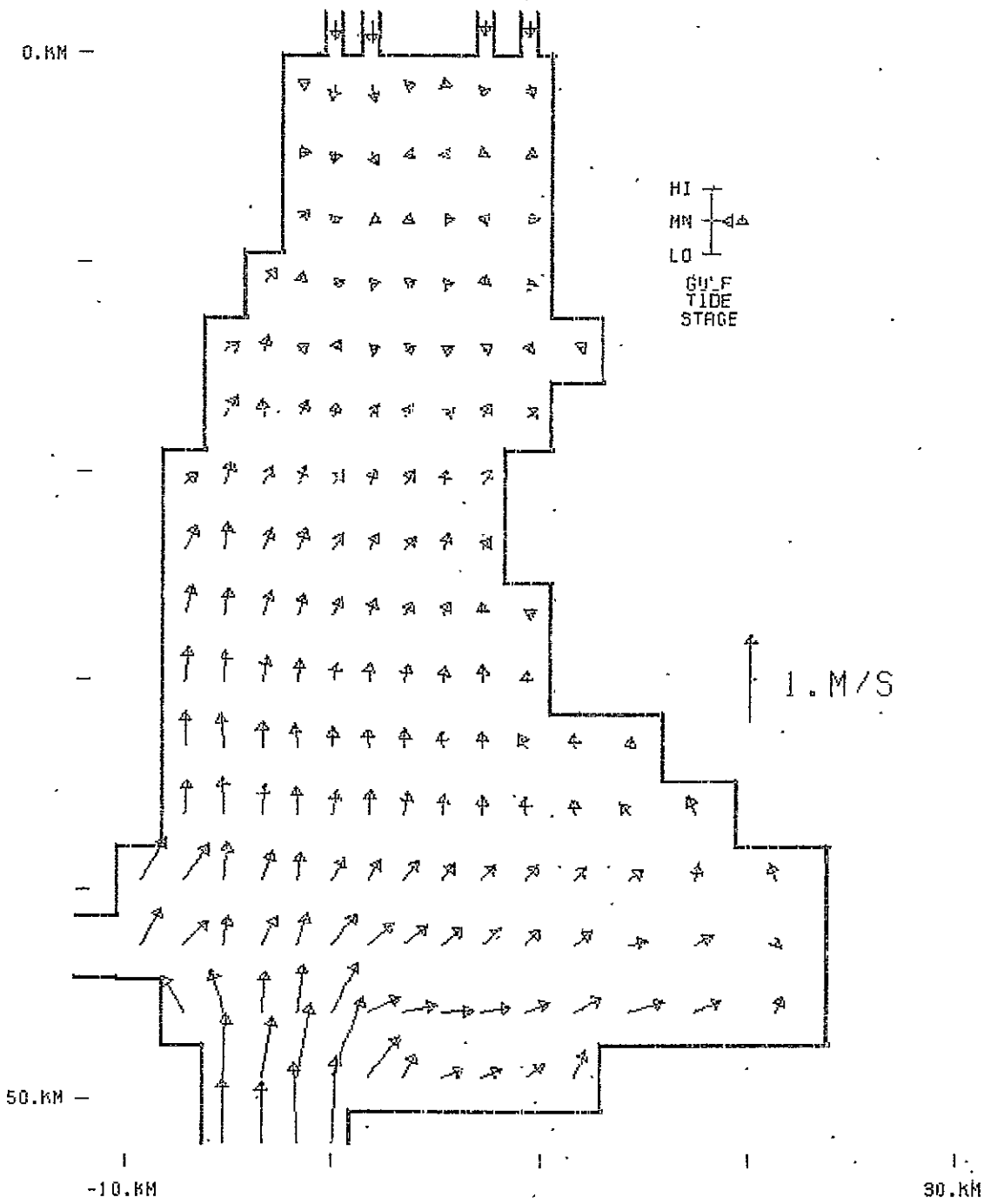


Figure A.99 -

Case 3

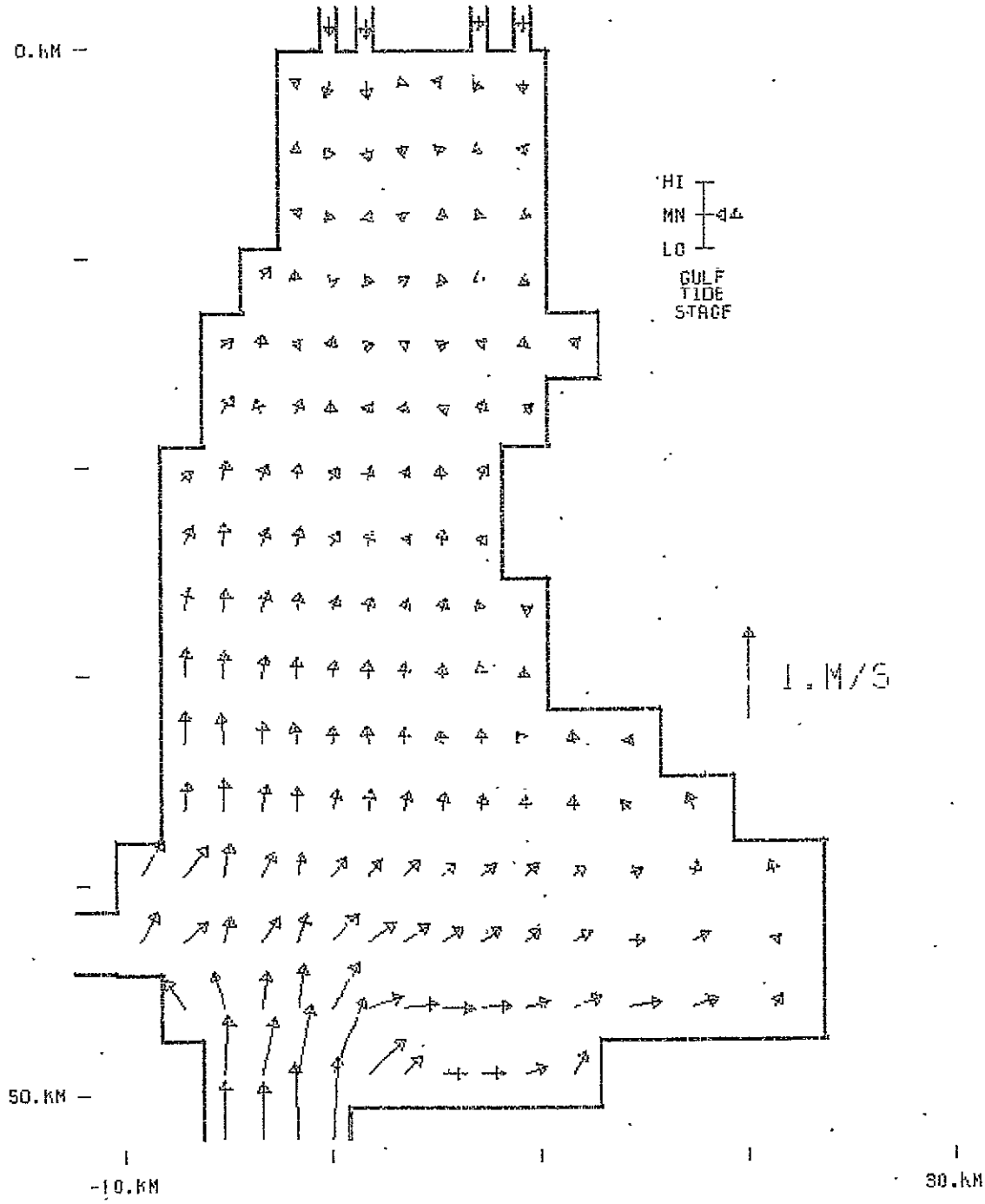


Figure A.100 -

Case 3

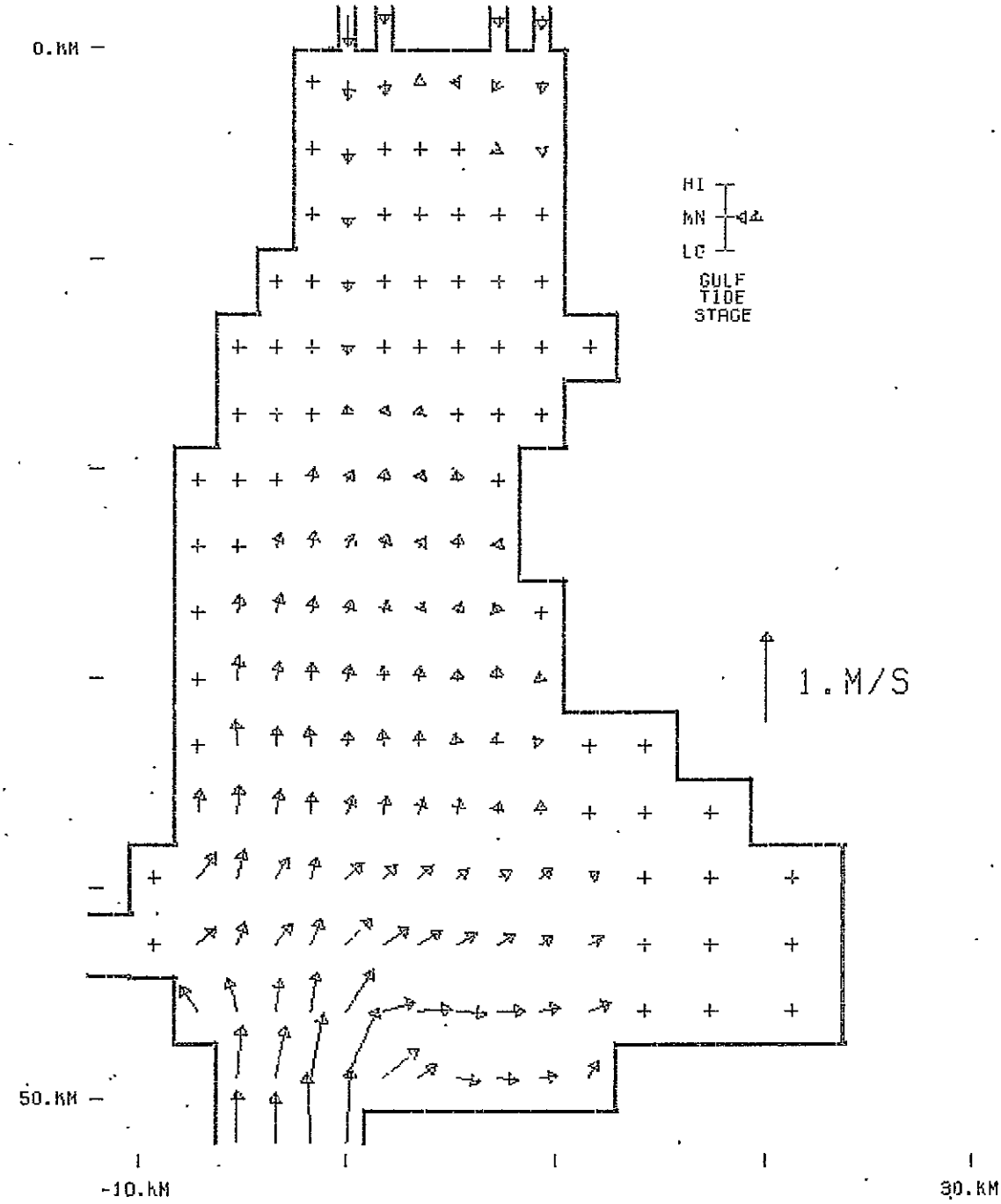
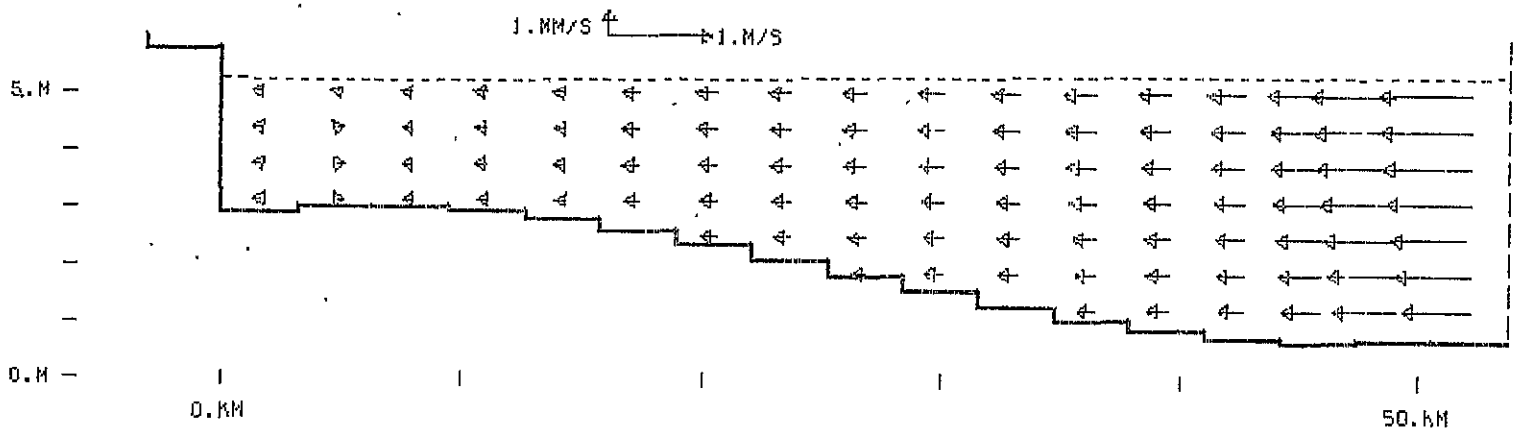
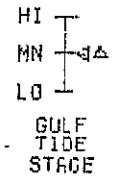


Figure A.101 -

Case 3





VELOCITY VECTORS
N-S SECTION -1.70 KM WRT CHANNEL
TIDE PERIOD: 25.00 HR ELAPSED TIME: 193.75 HR

Figure A.102 - Case 3

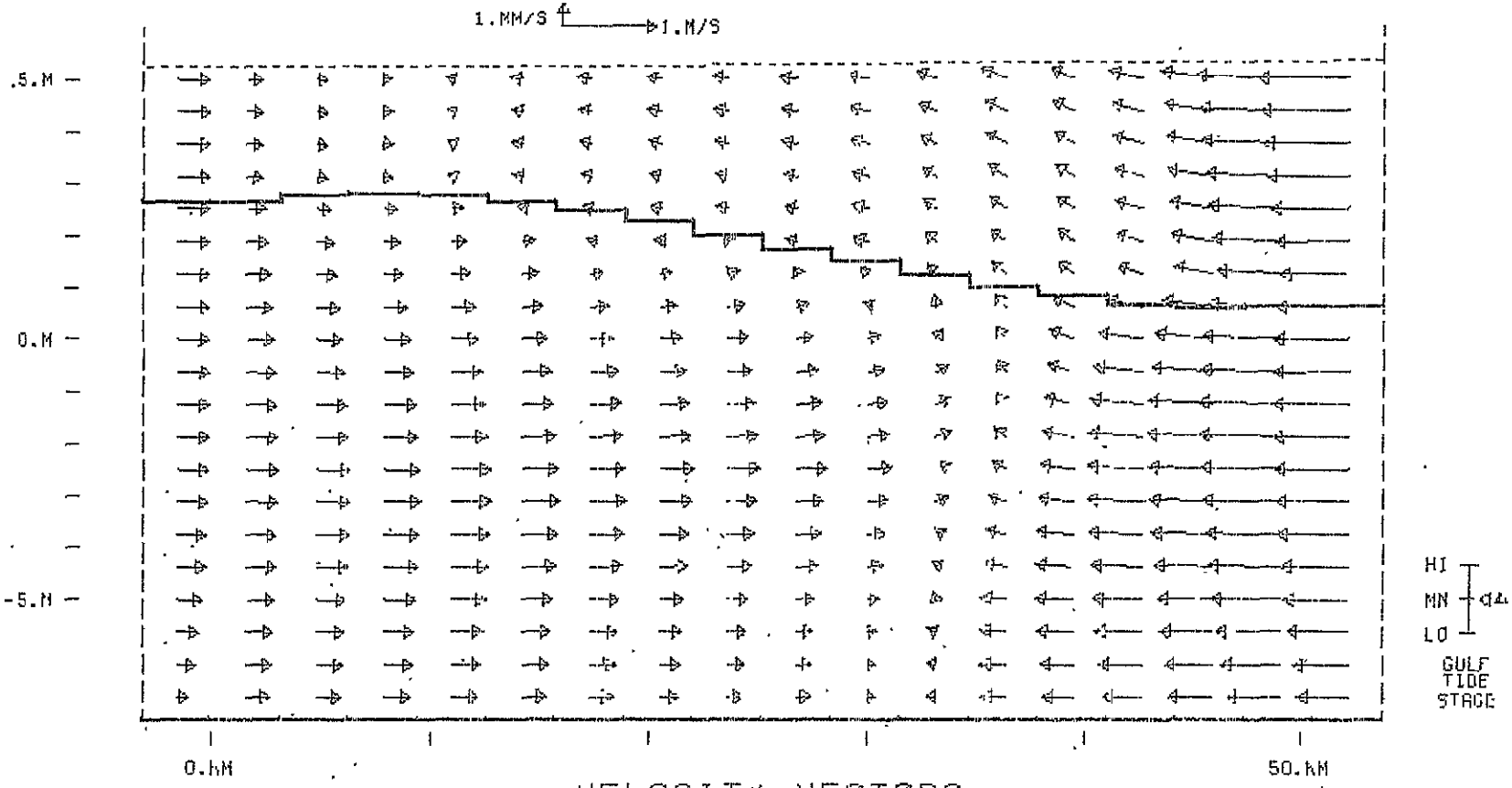


Figure A.103 - Case 3

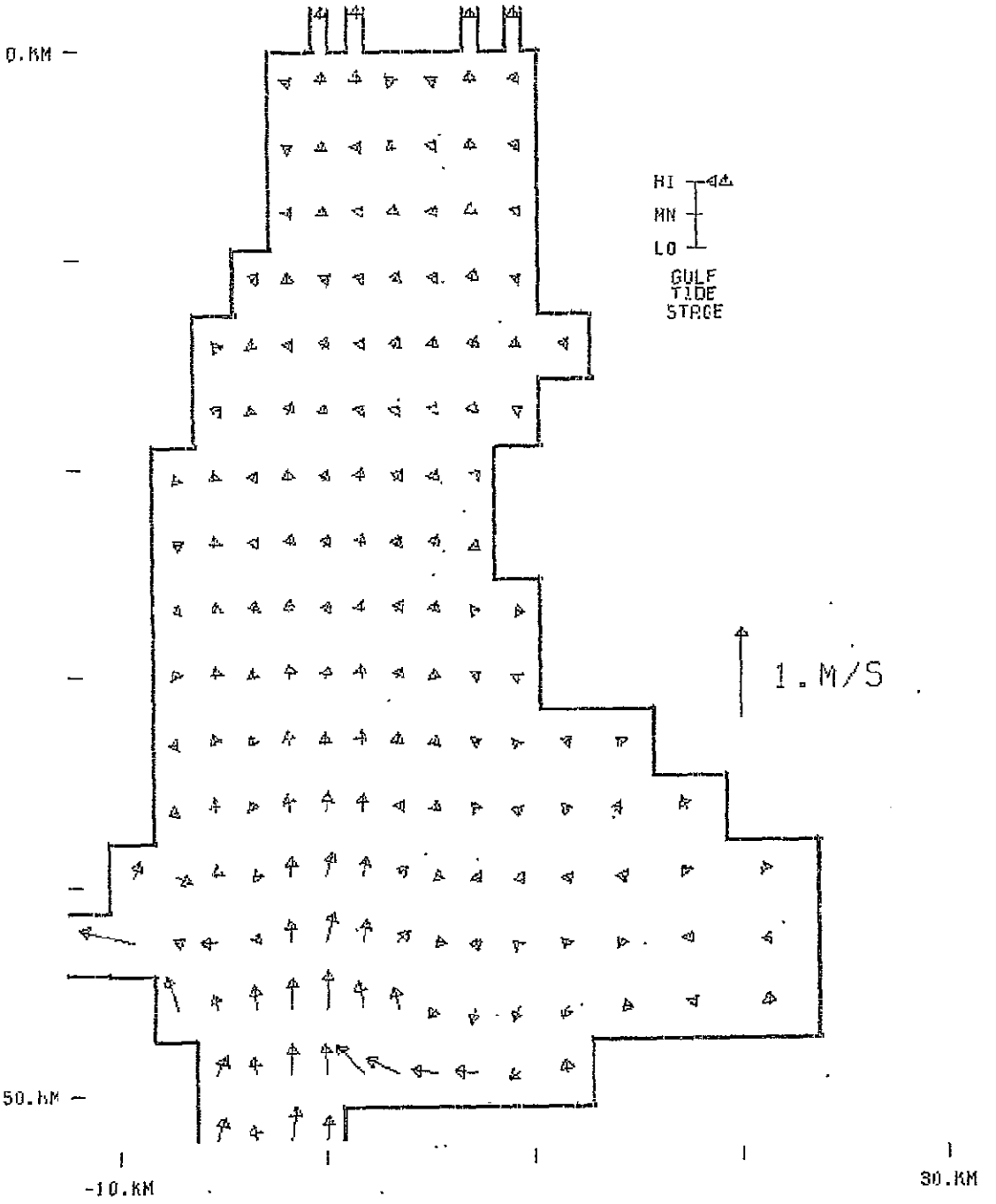


Figure A.104 -

Case 3

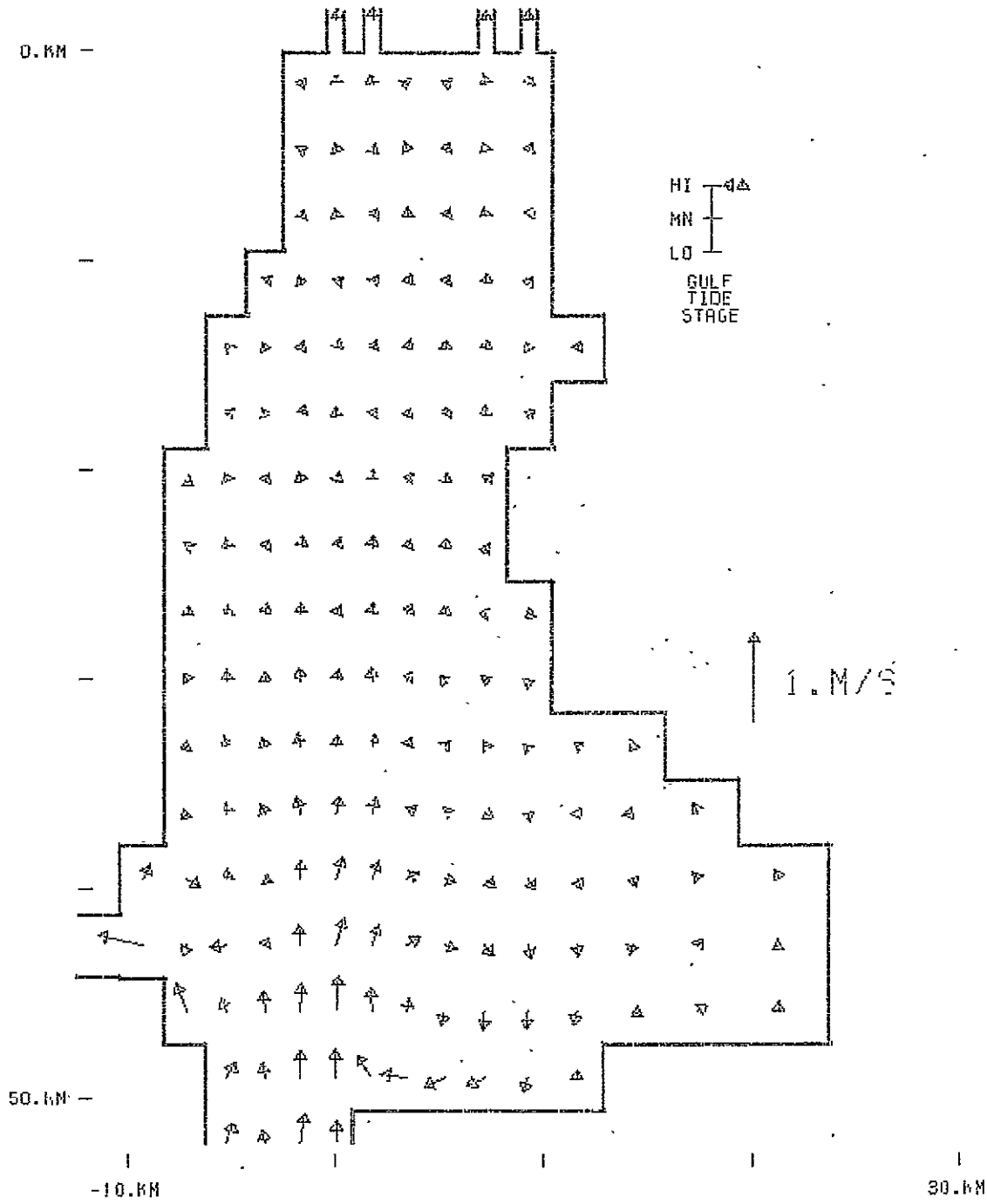


Figure A.105 -

Case 3

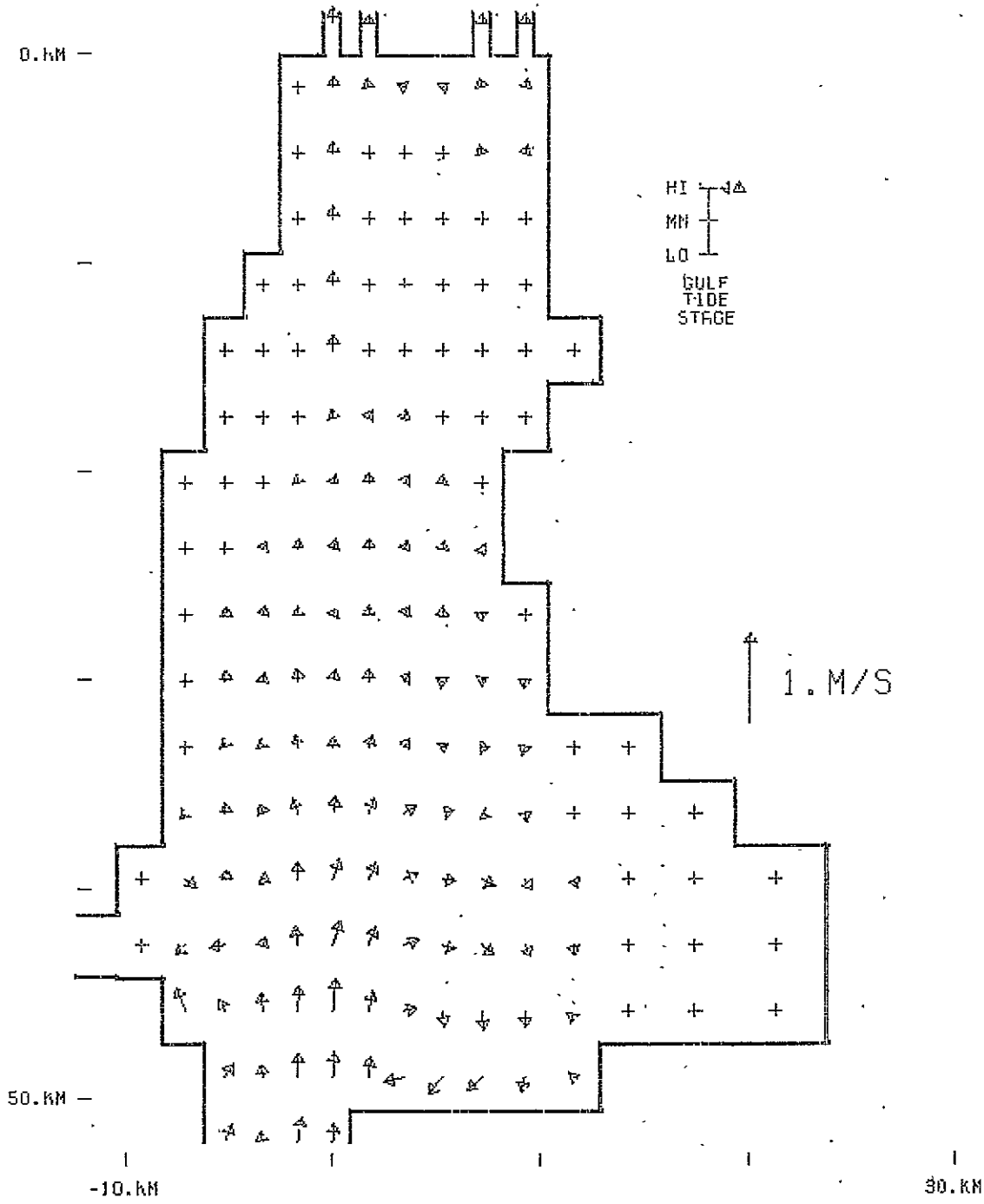


Figure A.106 -

Case 3

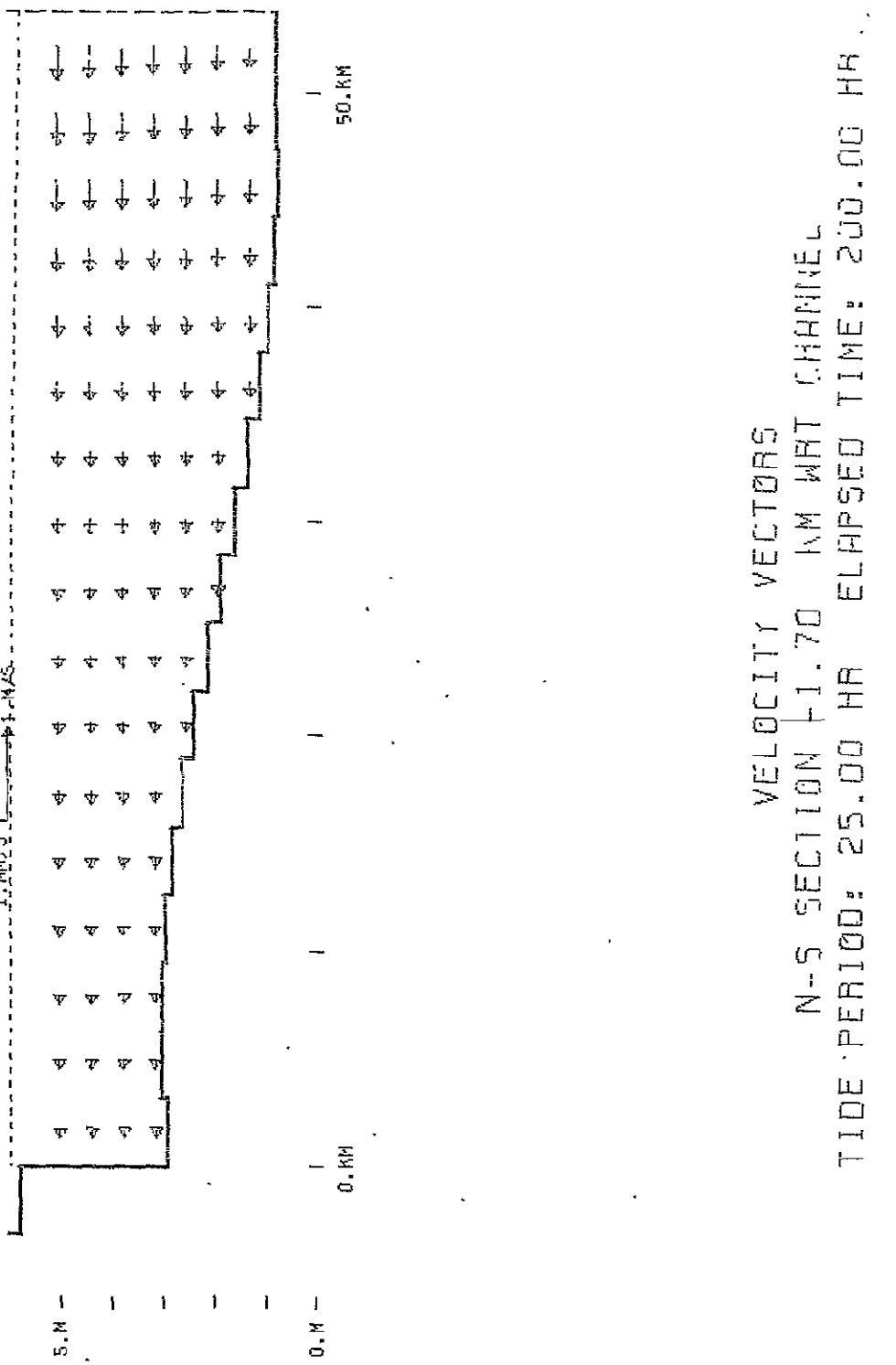


Figure A.107 - Case 3

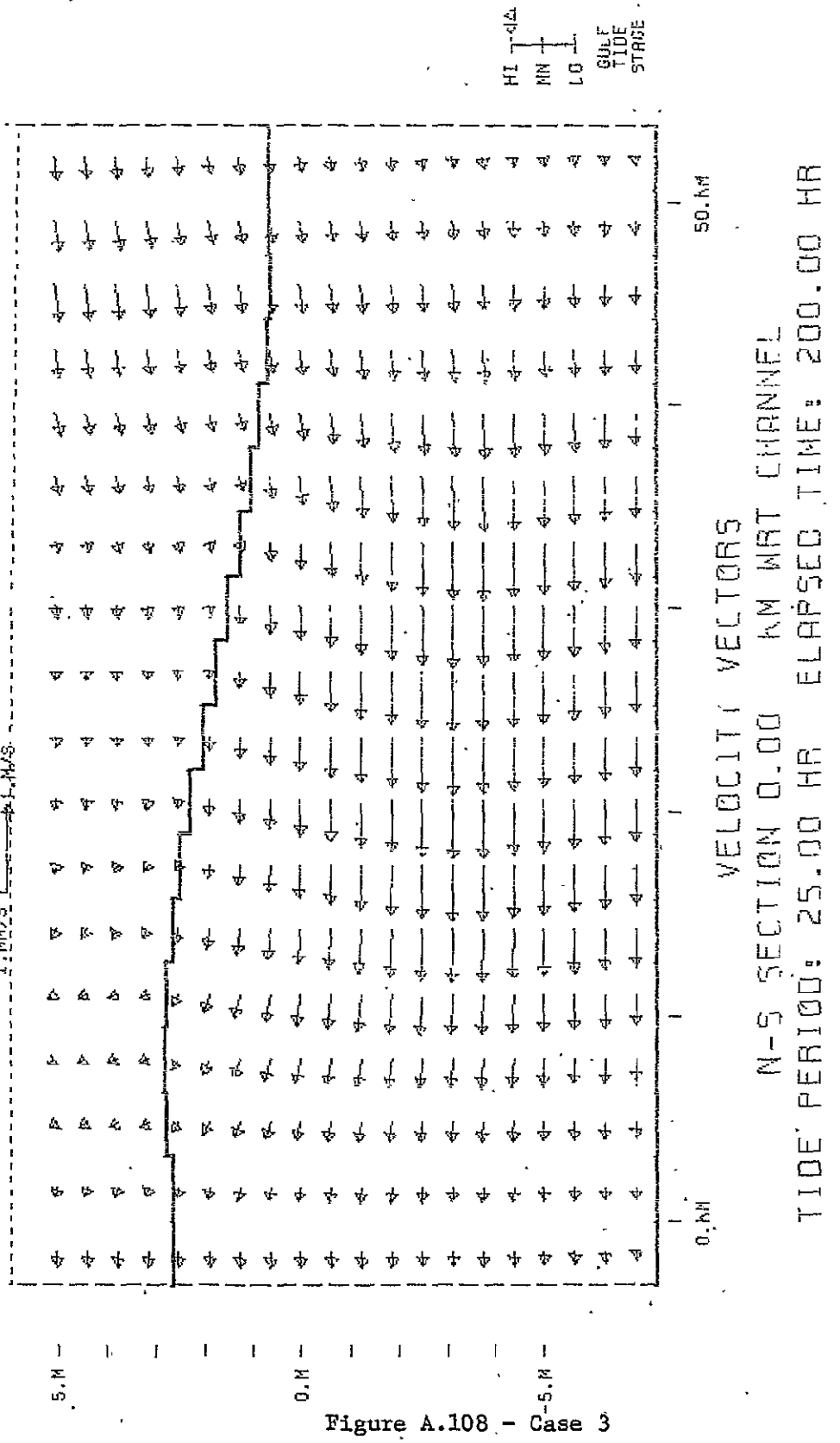


Figure A.108 - Case 3

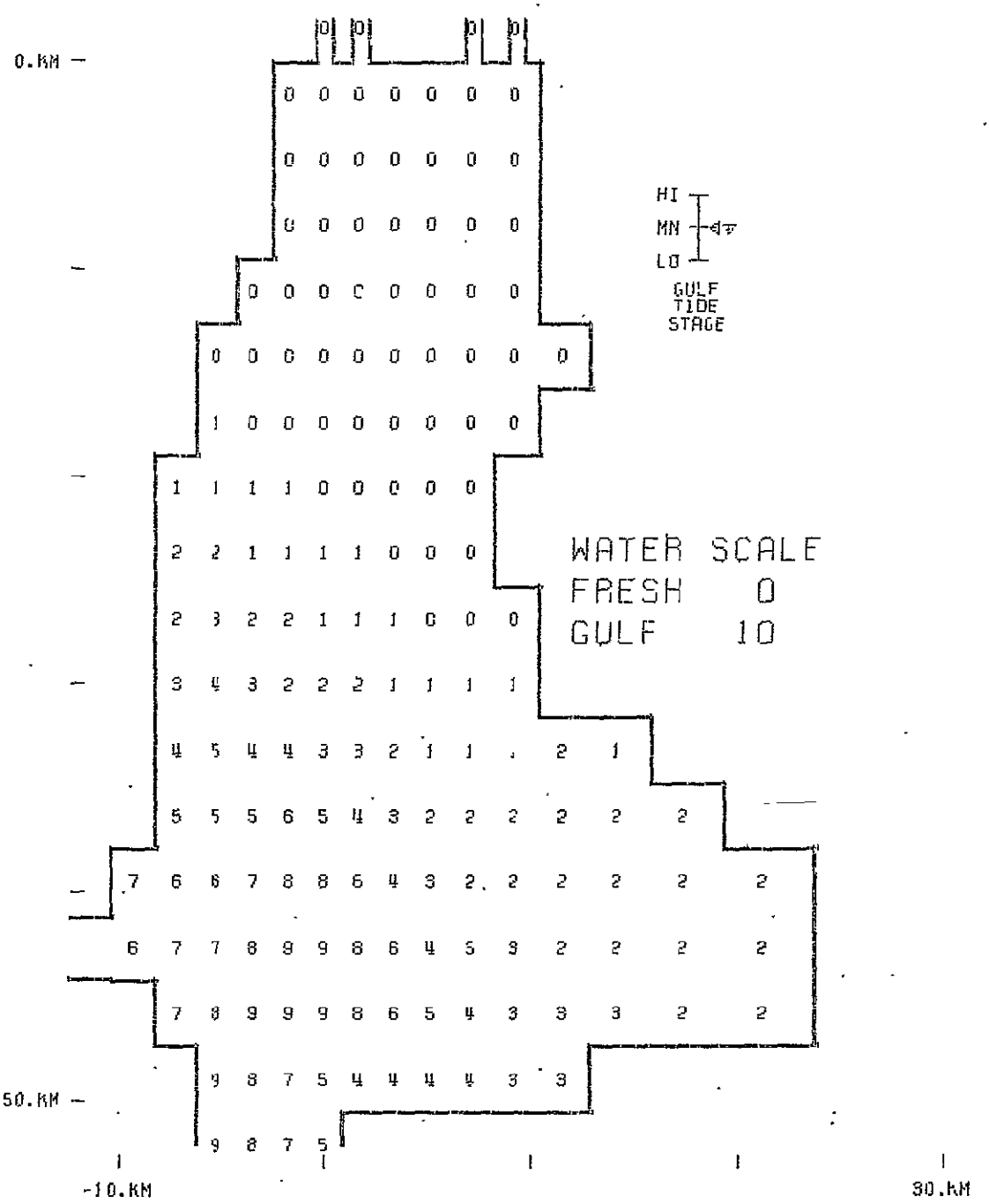


Figure A.109 -

Case 3

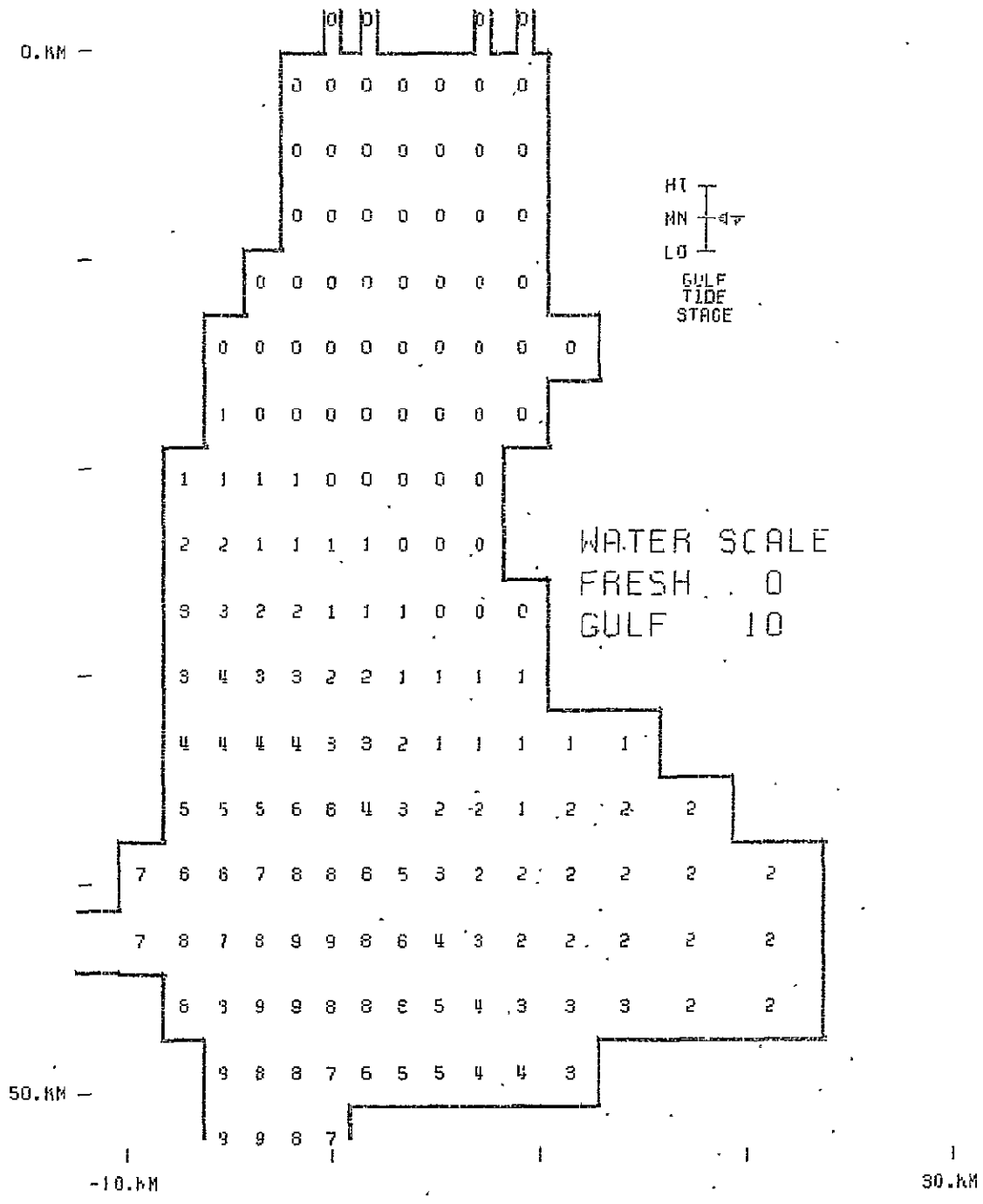


Figure A.110 -

Case 3

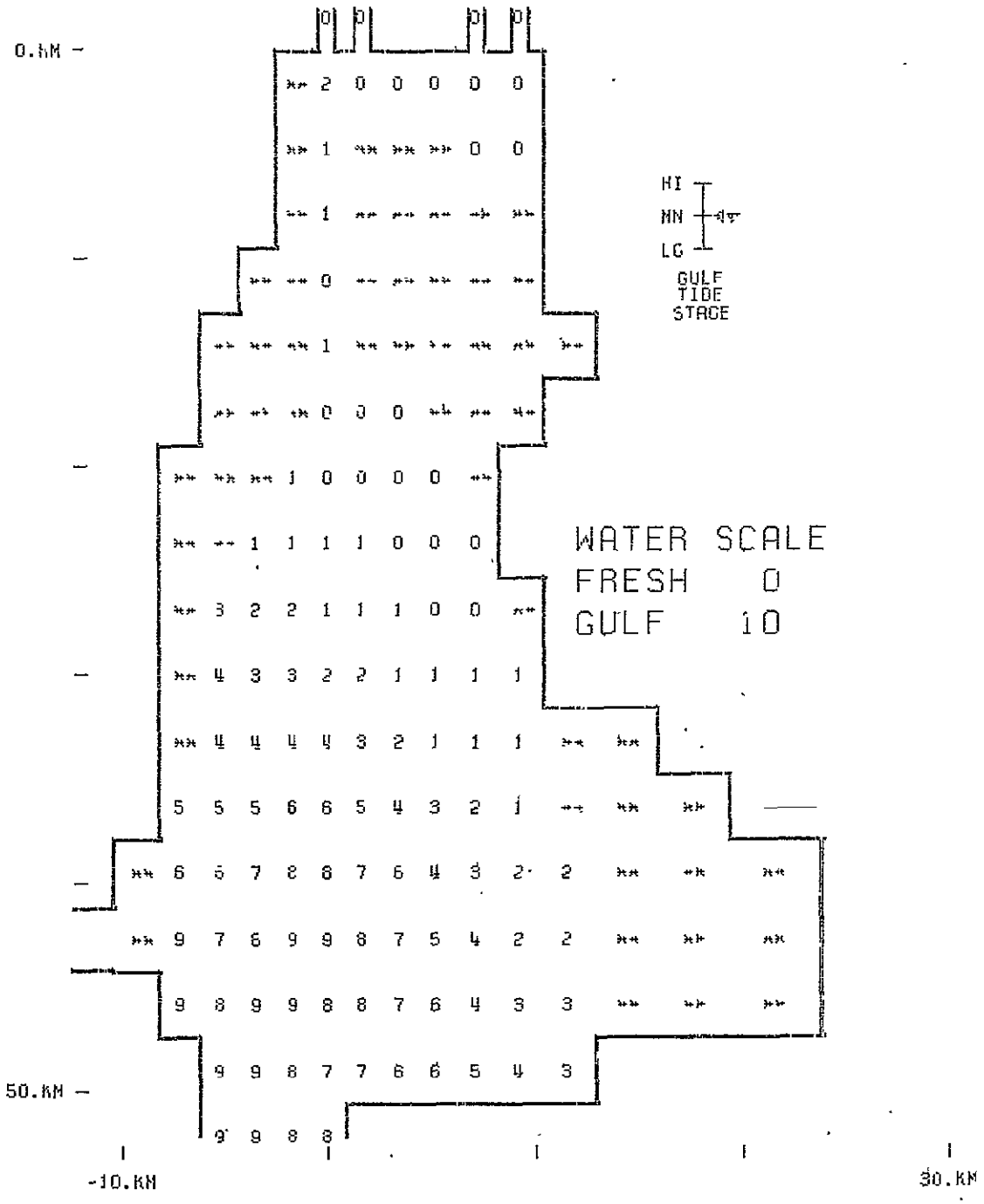


Figure A.111 -

Case 3

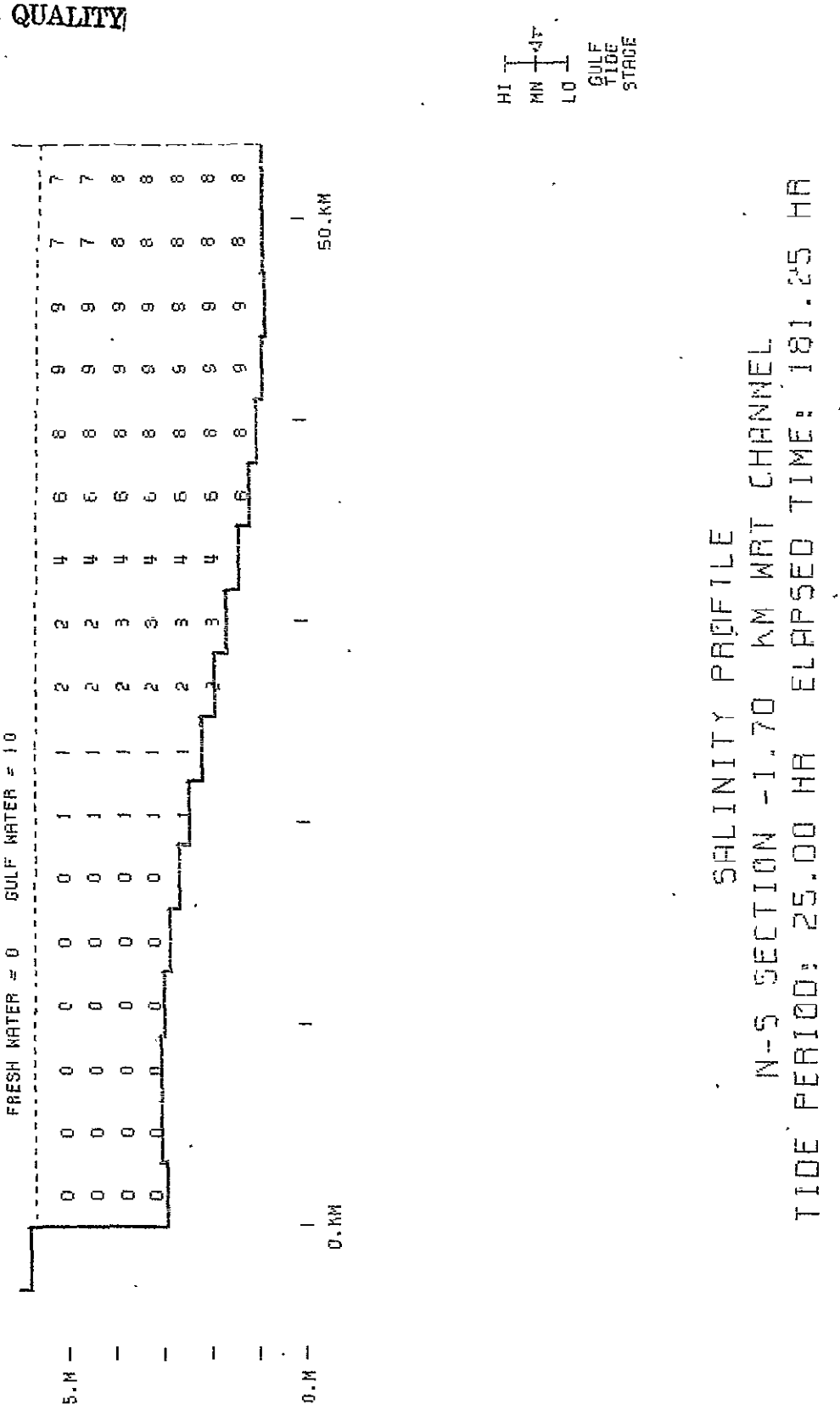


Figure A.112 - Case 3

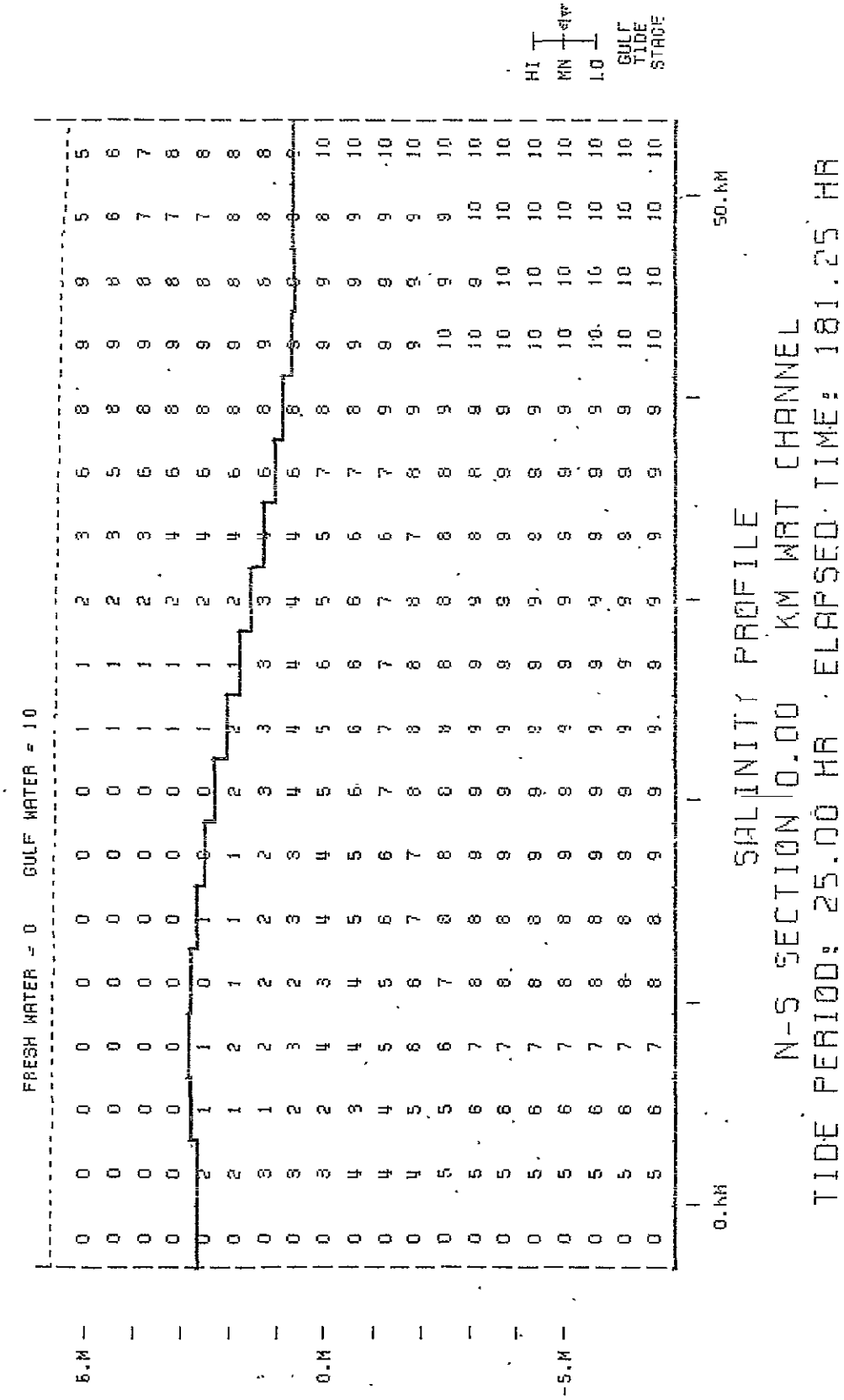


Figure A.113 - Case 3

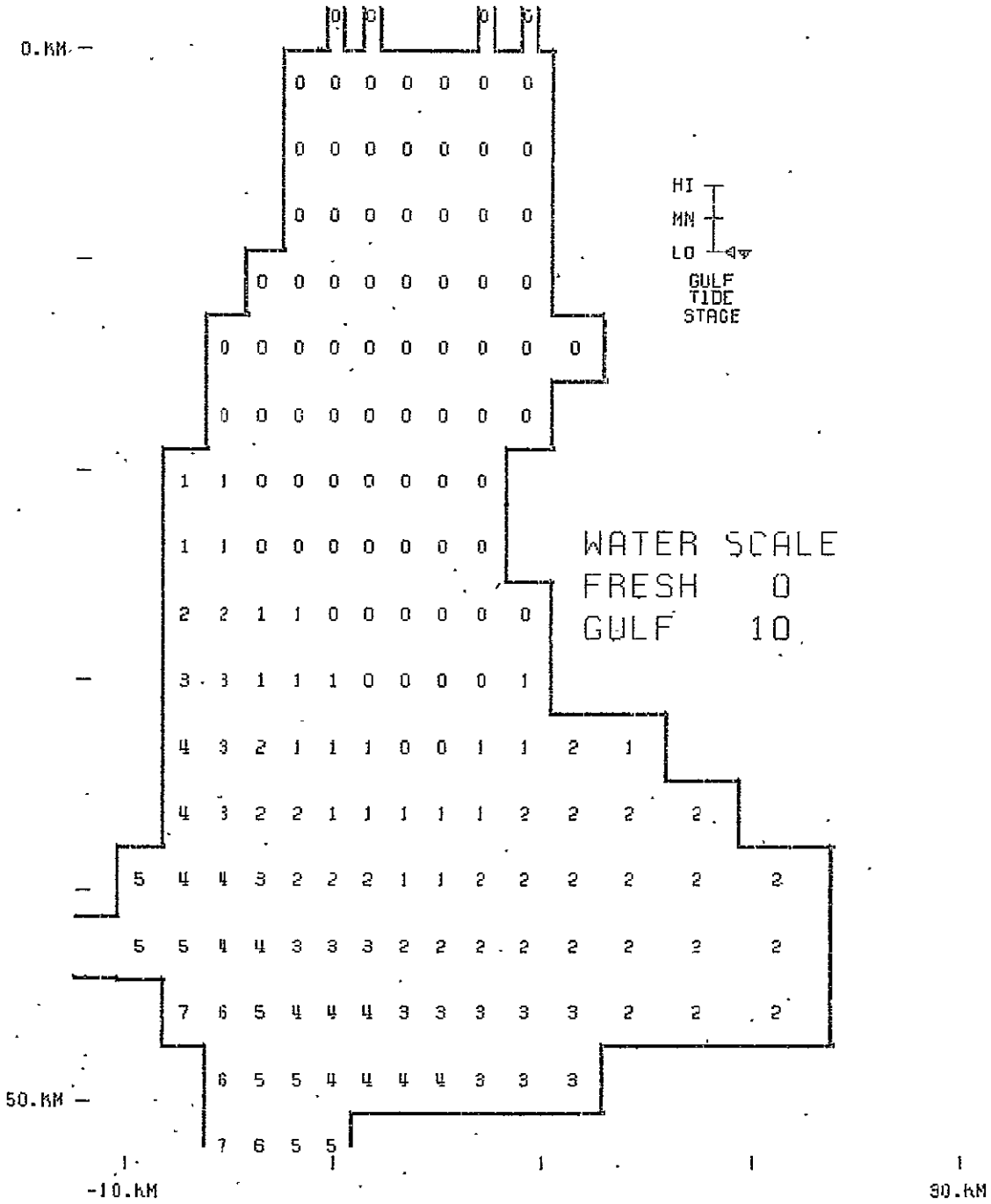


Figure A.114 -

Case 3

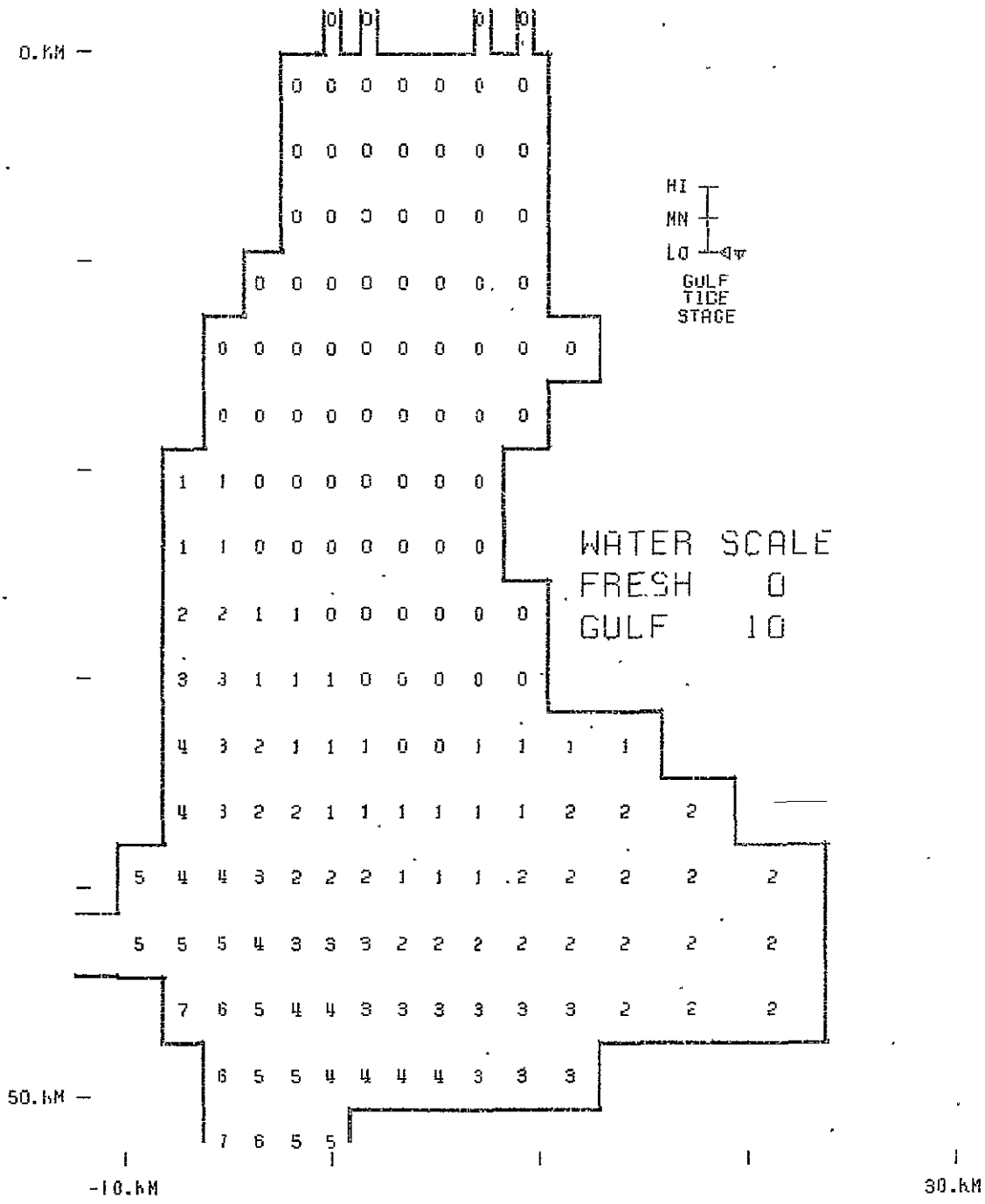


Figure A.115 -

Case 3

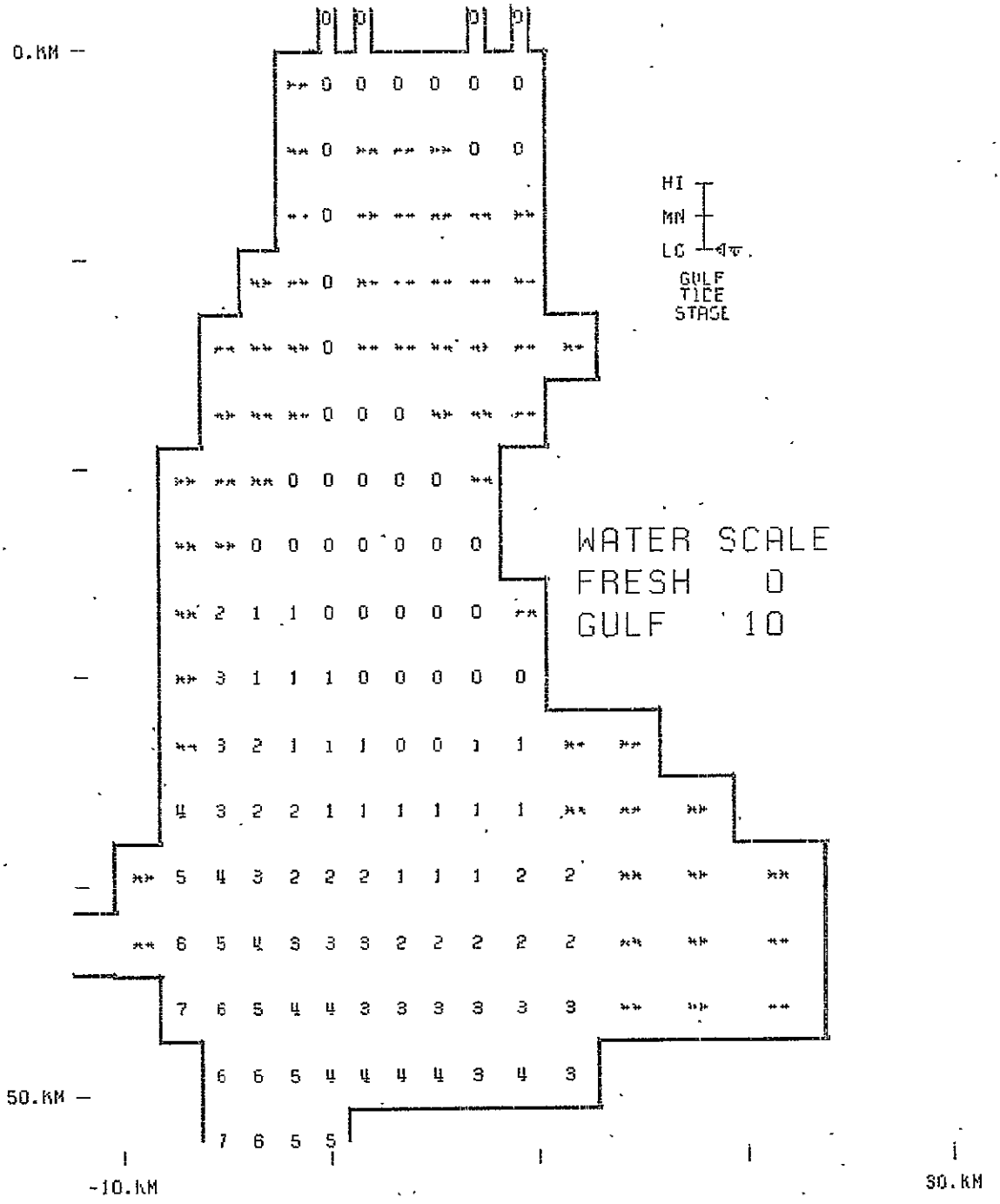
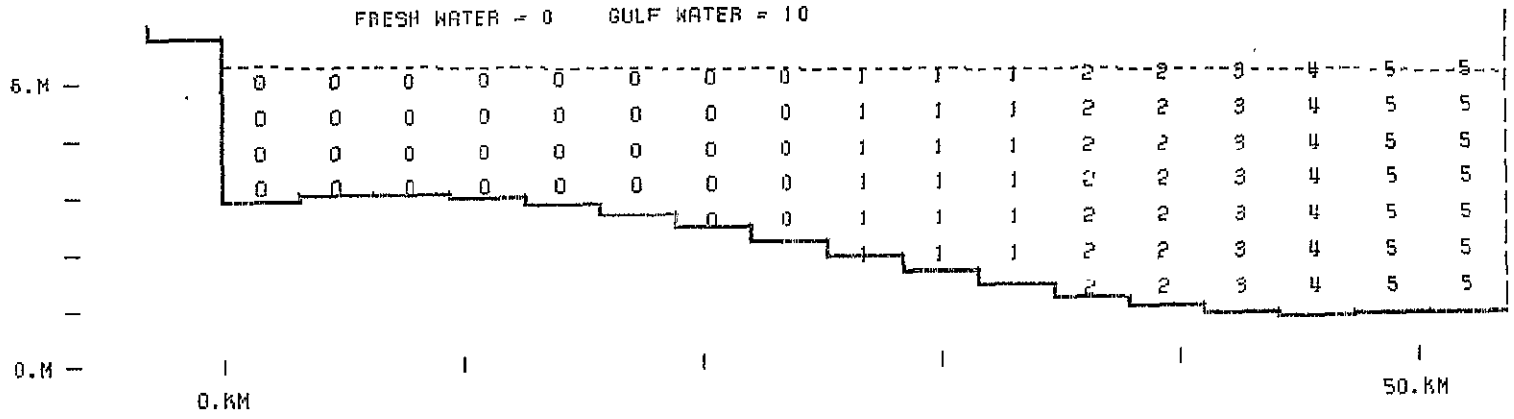
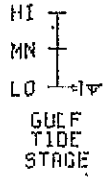


Figure A.116 -

Case 3





SALINITY PROFILE
N-S SECTION -1.70 KM WRT CHANNEL
TIDE PERIOD: 25.00 HR ELAPSED TIME: 187.50 HR

Figure A.117 - Case 3

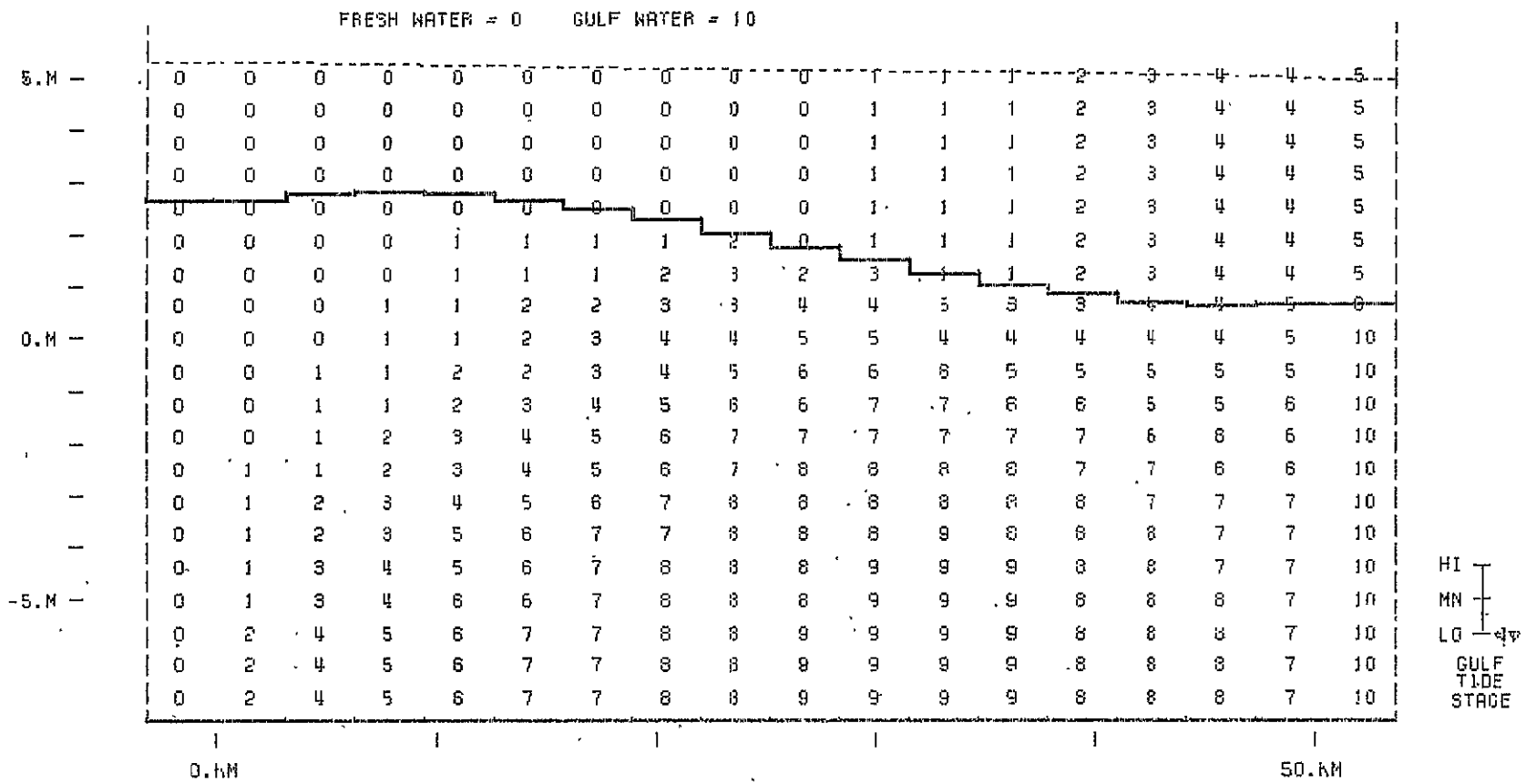


Figure A.118 - Case 3

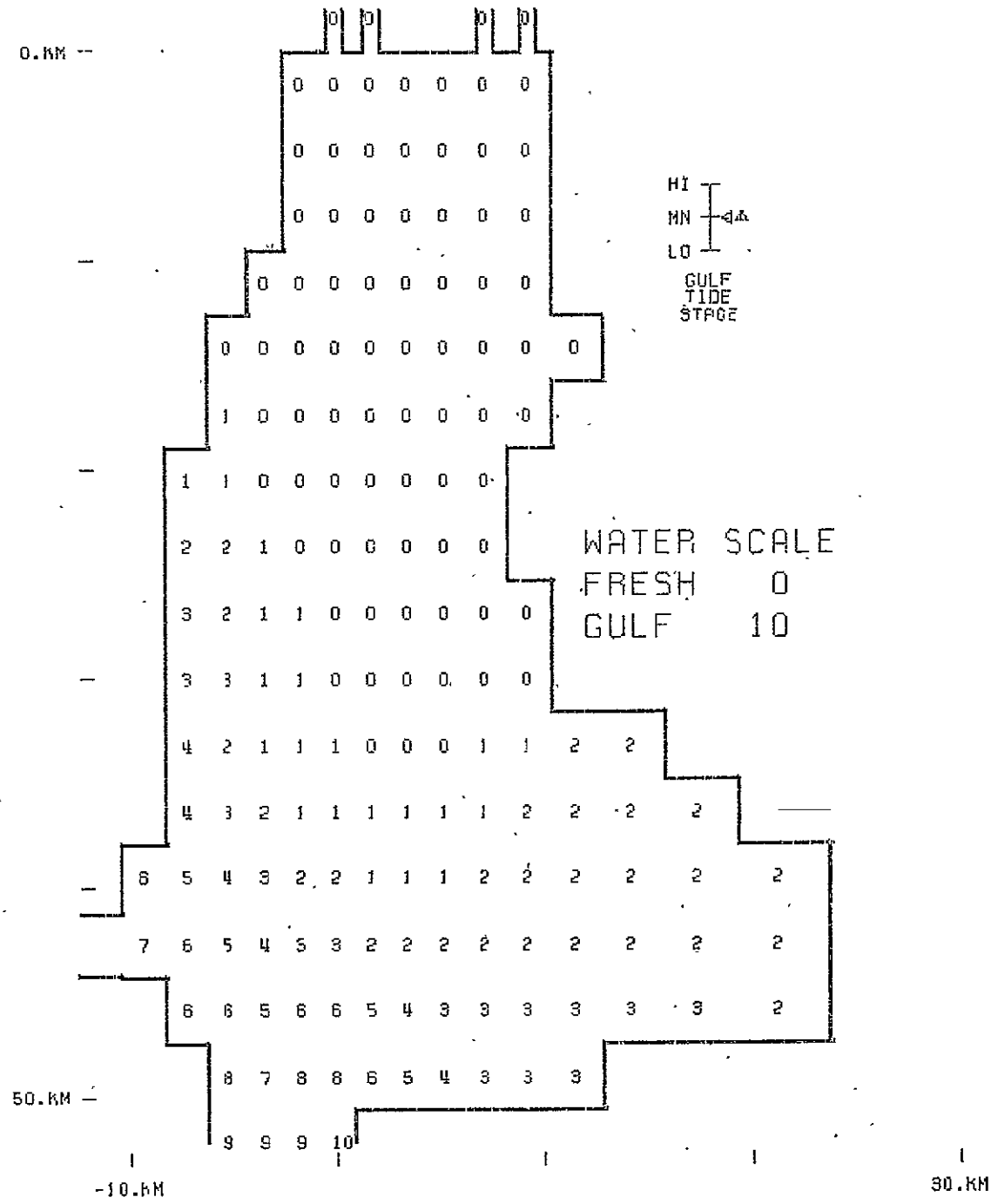


Figure A.119 -

Case 3

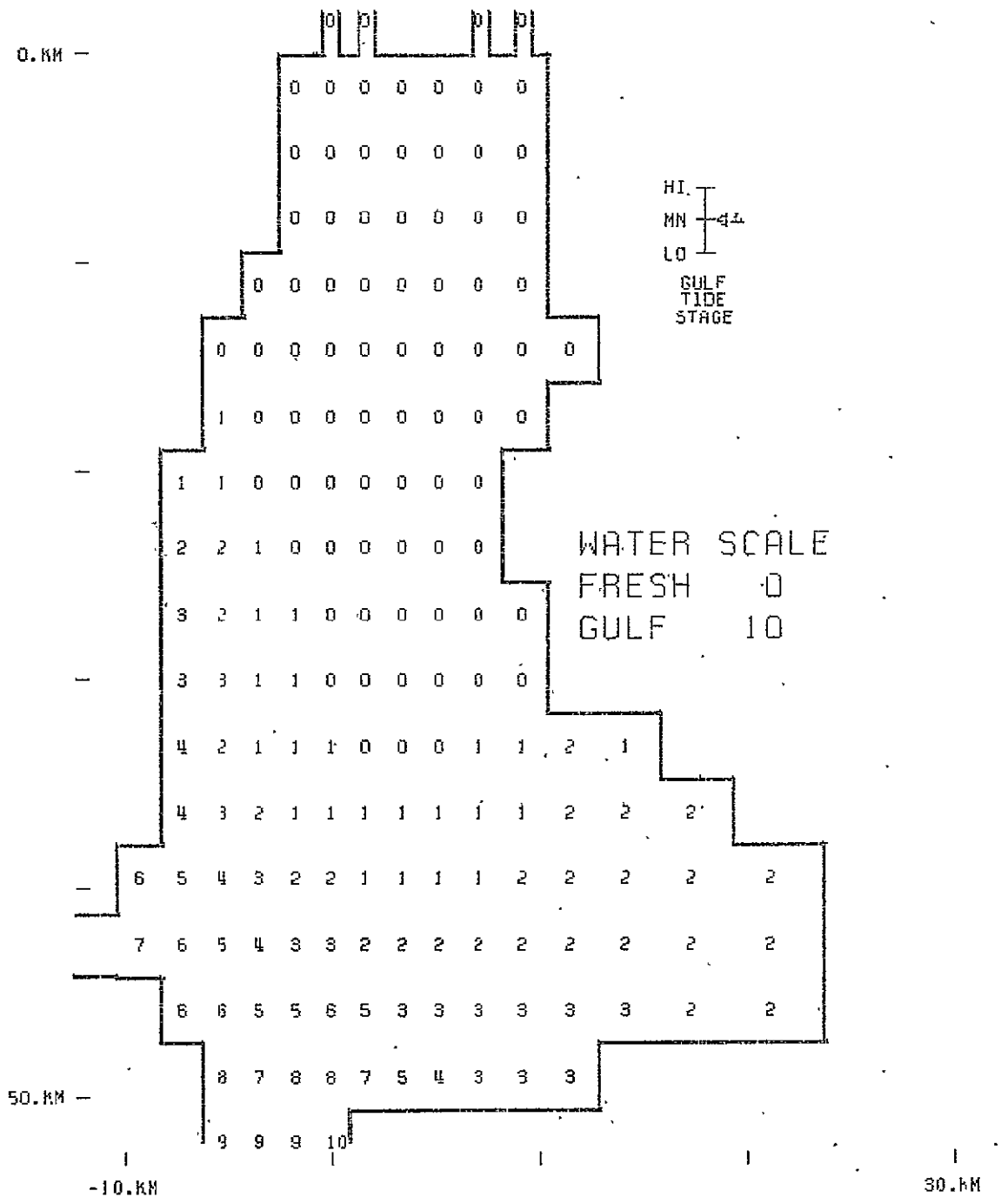


Figure A.120 -

Case 3

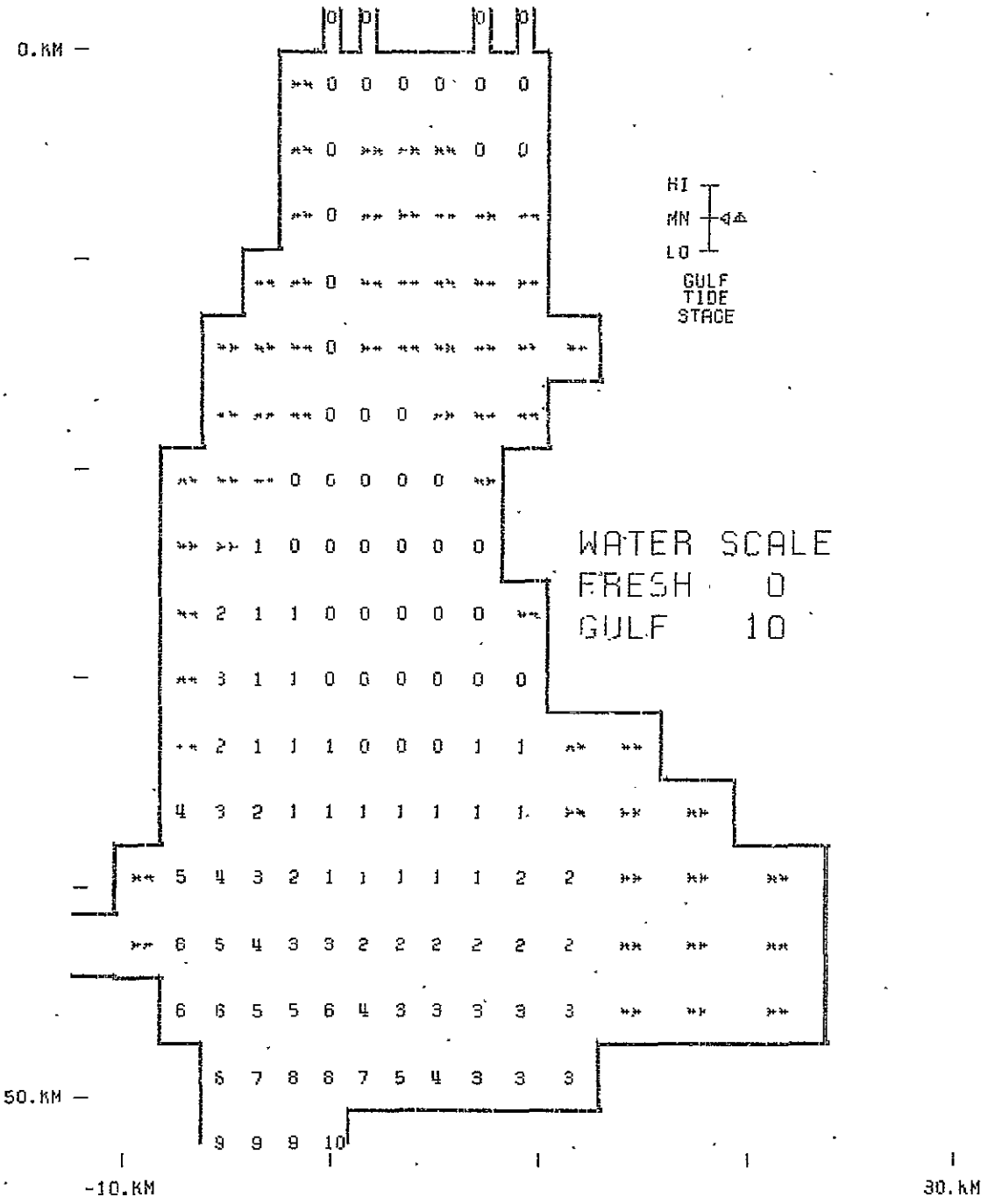


Figure A.121 - Case 3

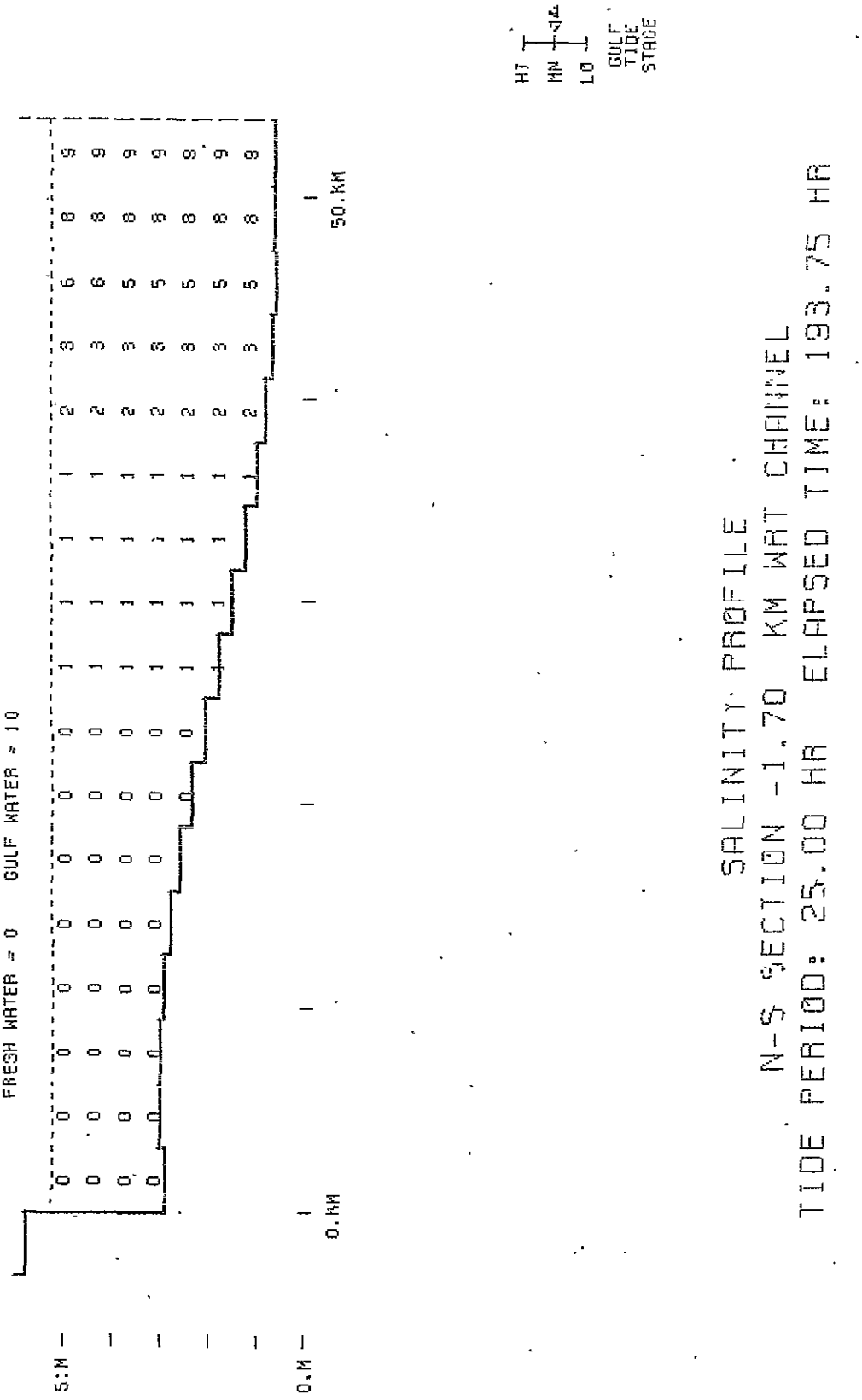


Figure A.122 - Case 3

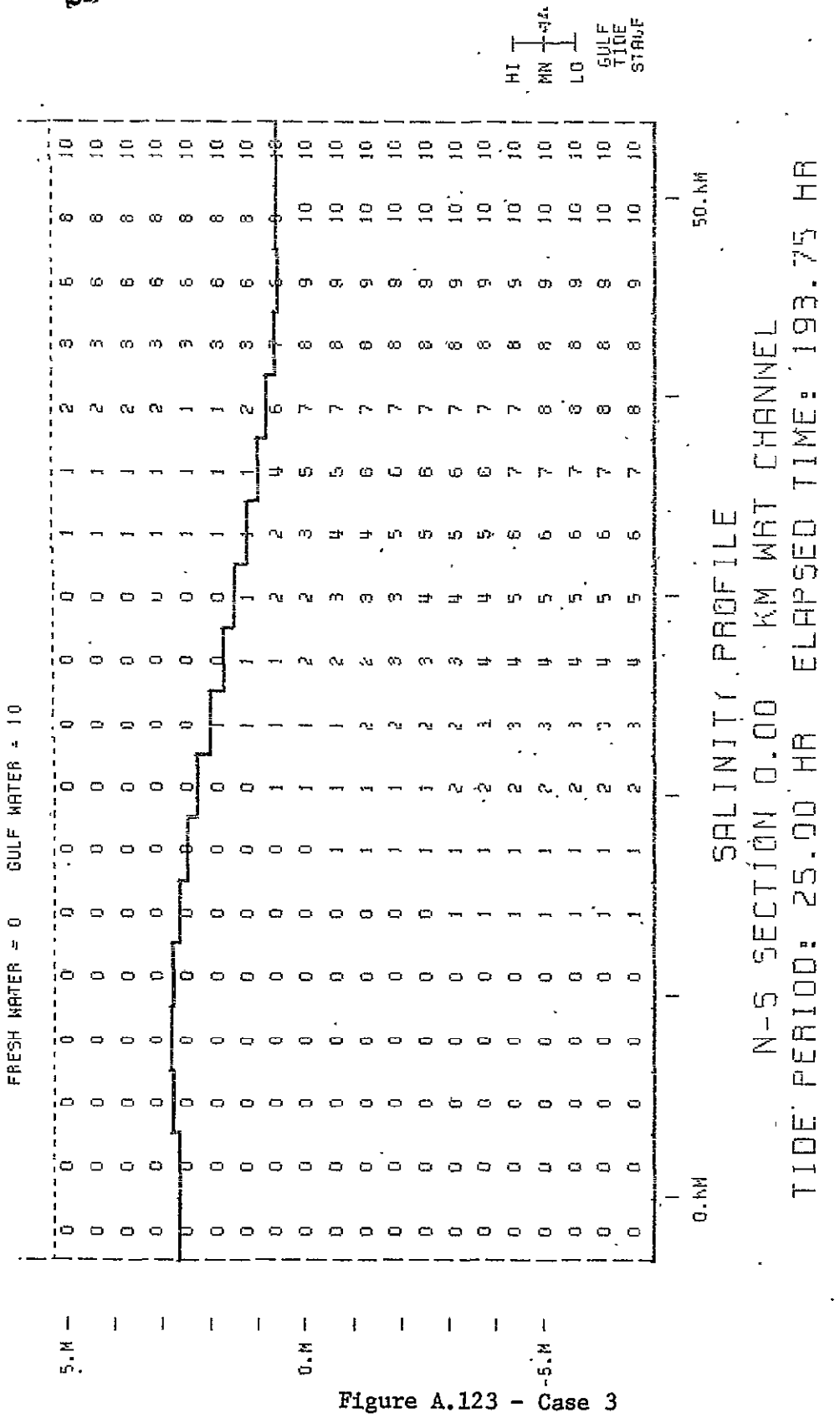


Figure A.123 - Case 3

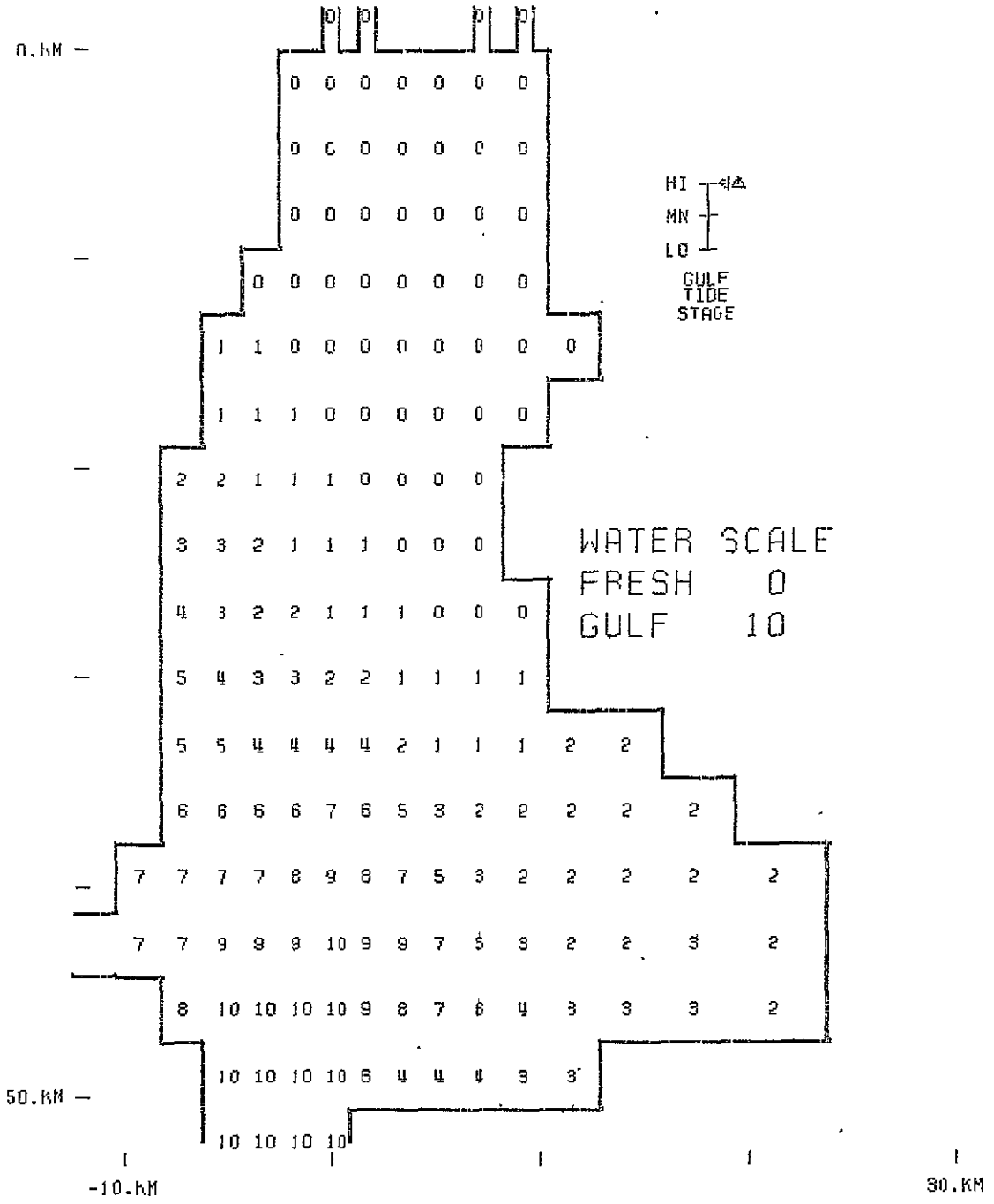


Figure A.124 -

Case 3

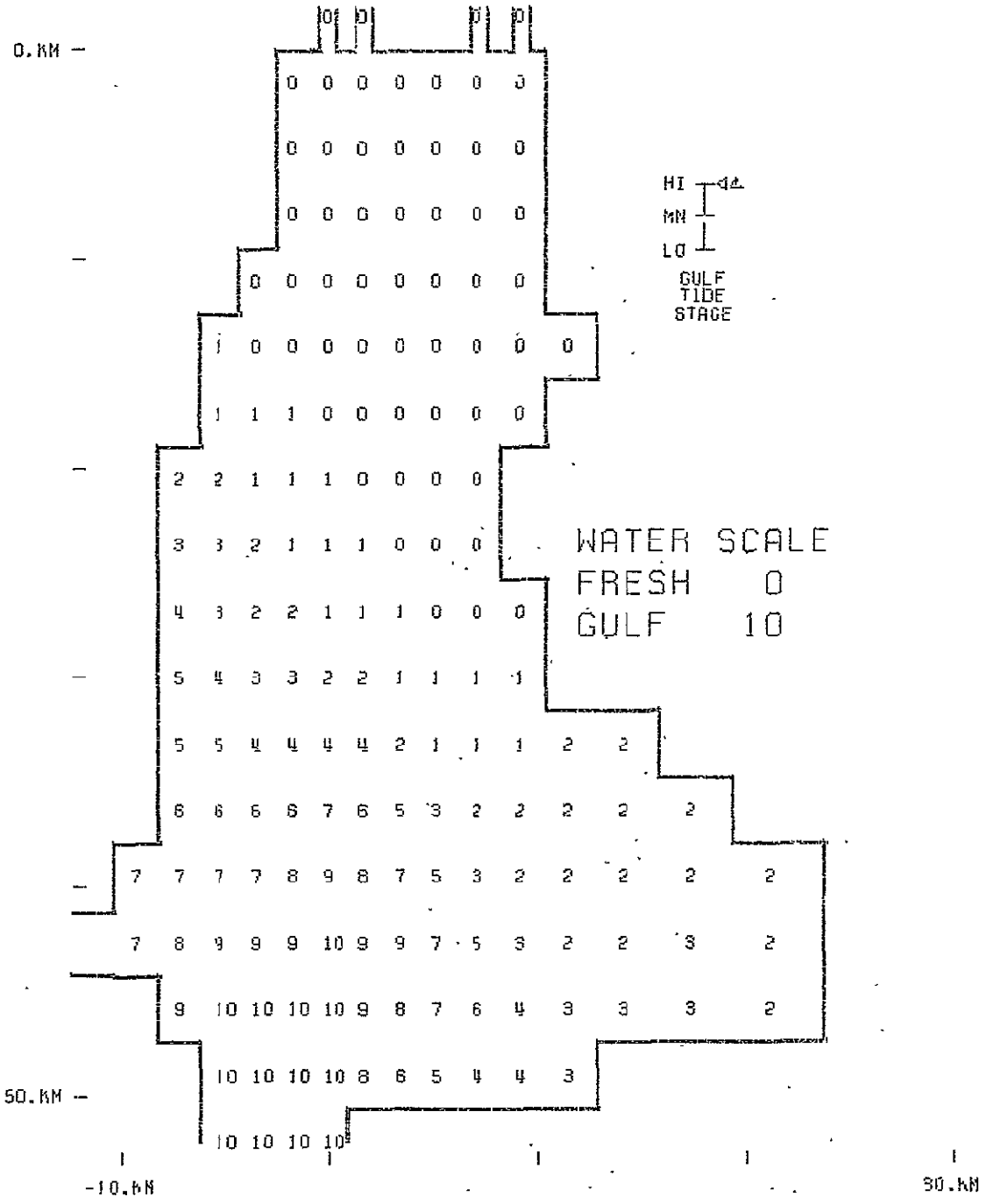


Figure A.125 -

Case 3

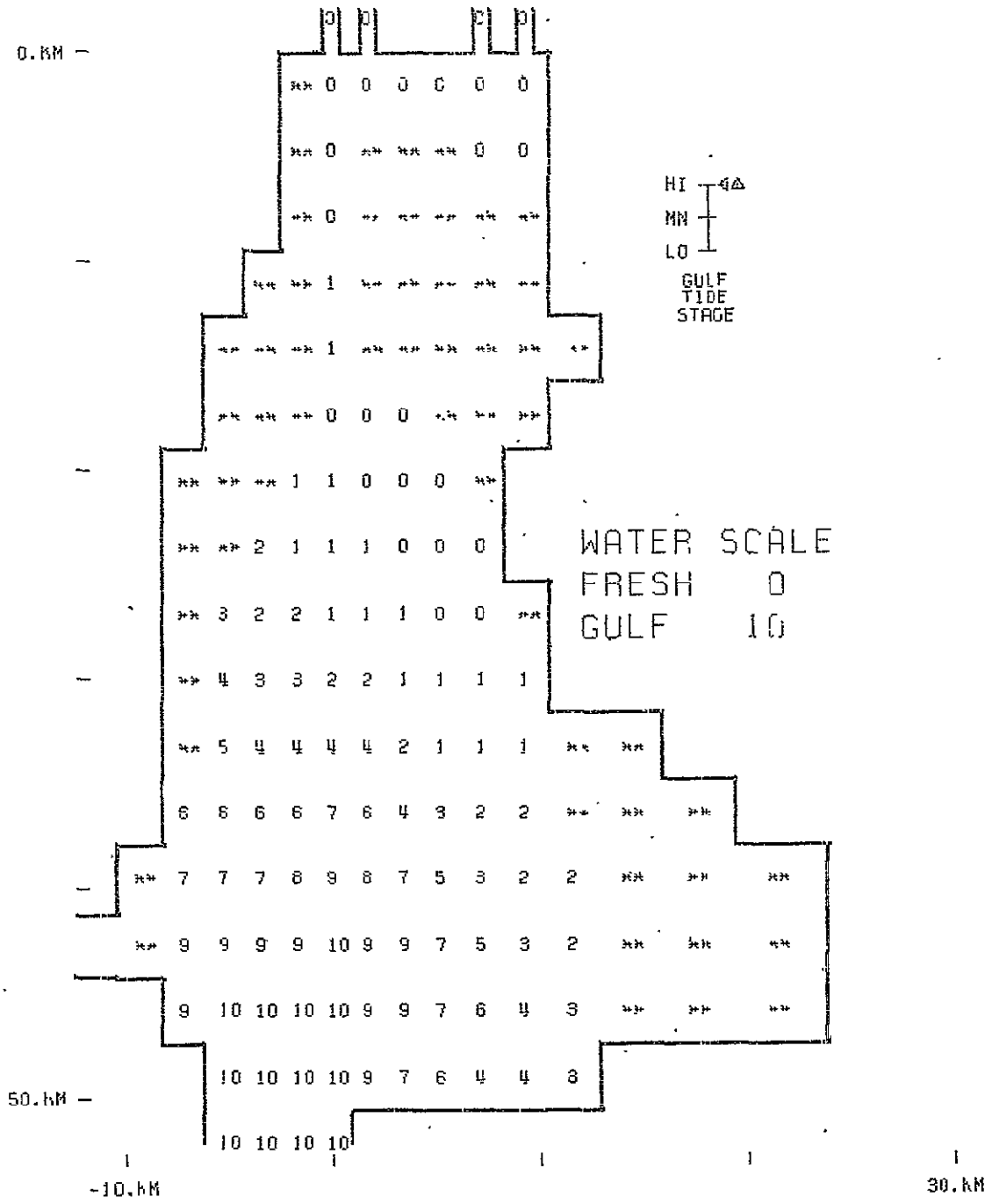
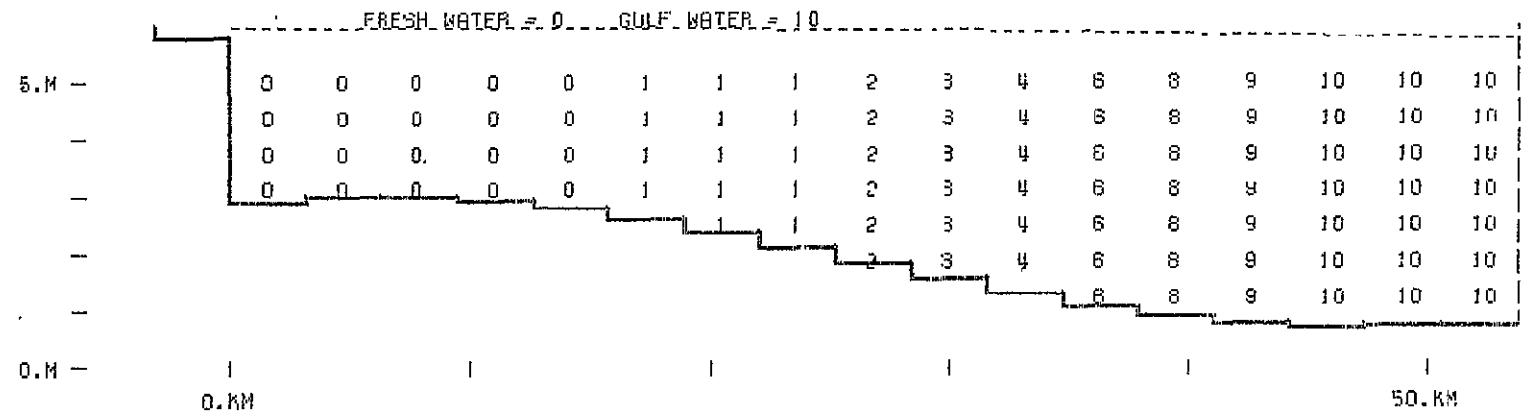


Figure A.126 - Case 3



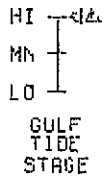


Figure A.127 - Case 3

SALINITY PROFILE
N-S SECTION -1.70 KM WRT CHANNEL
TIDE PERIOD: 25.00 HR ELAPSED TIME: 200.00 HR

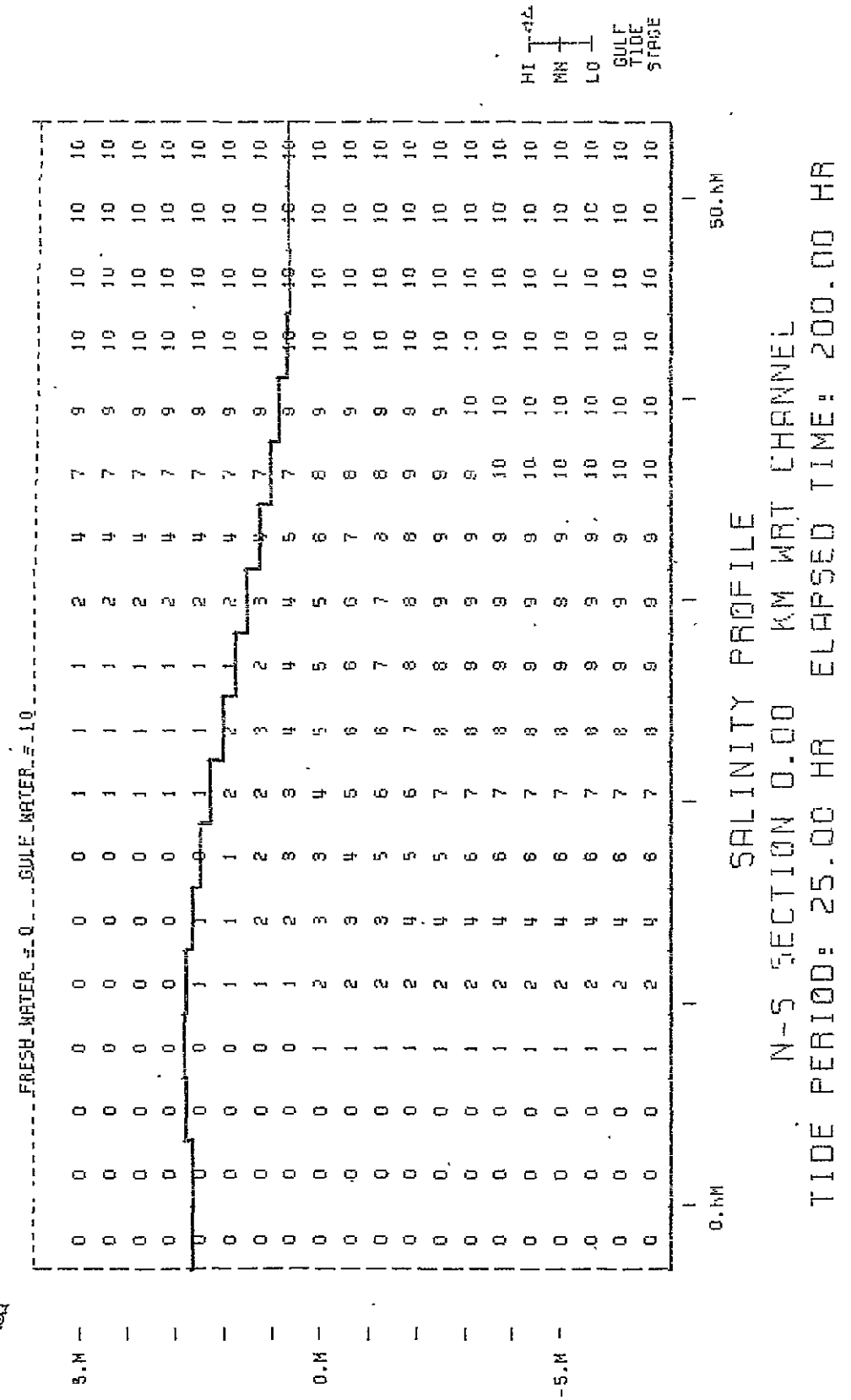


Figure A.128 - Case 3

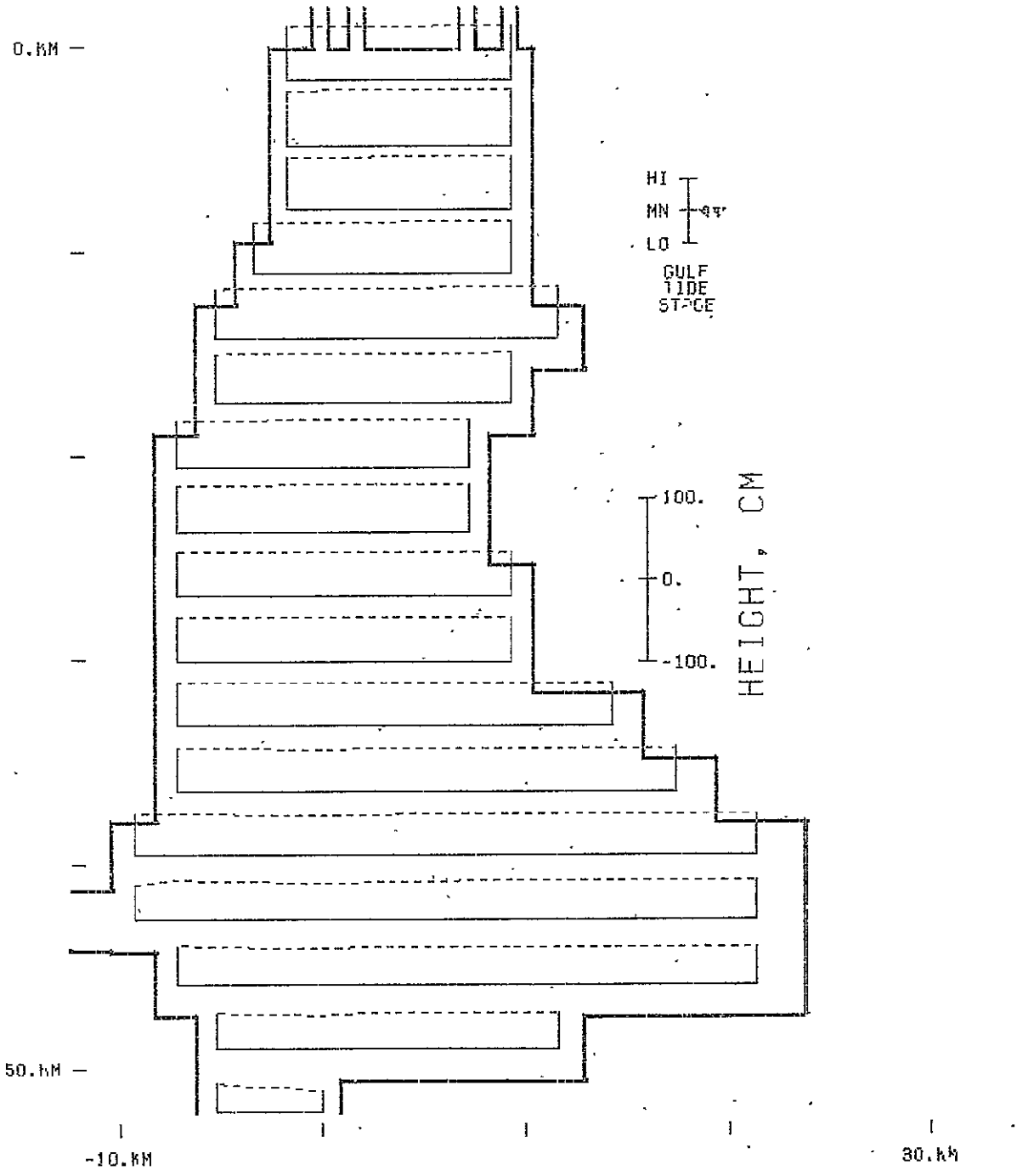


Figure A.129 -

Case 3

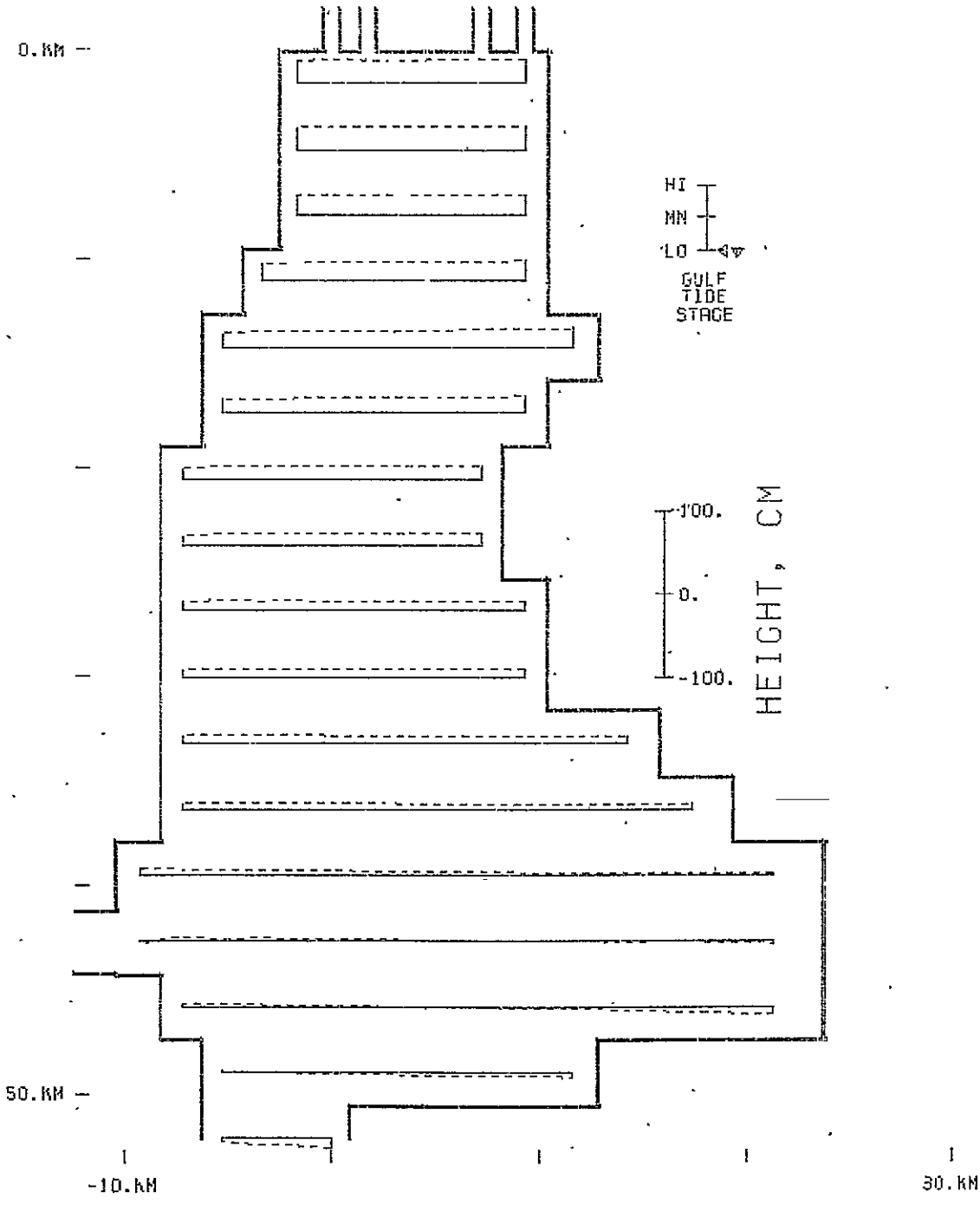


Figure A.130 -

Case 3

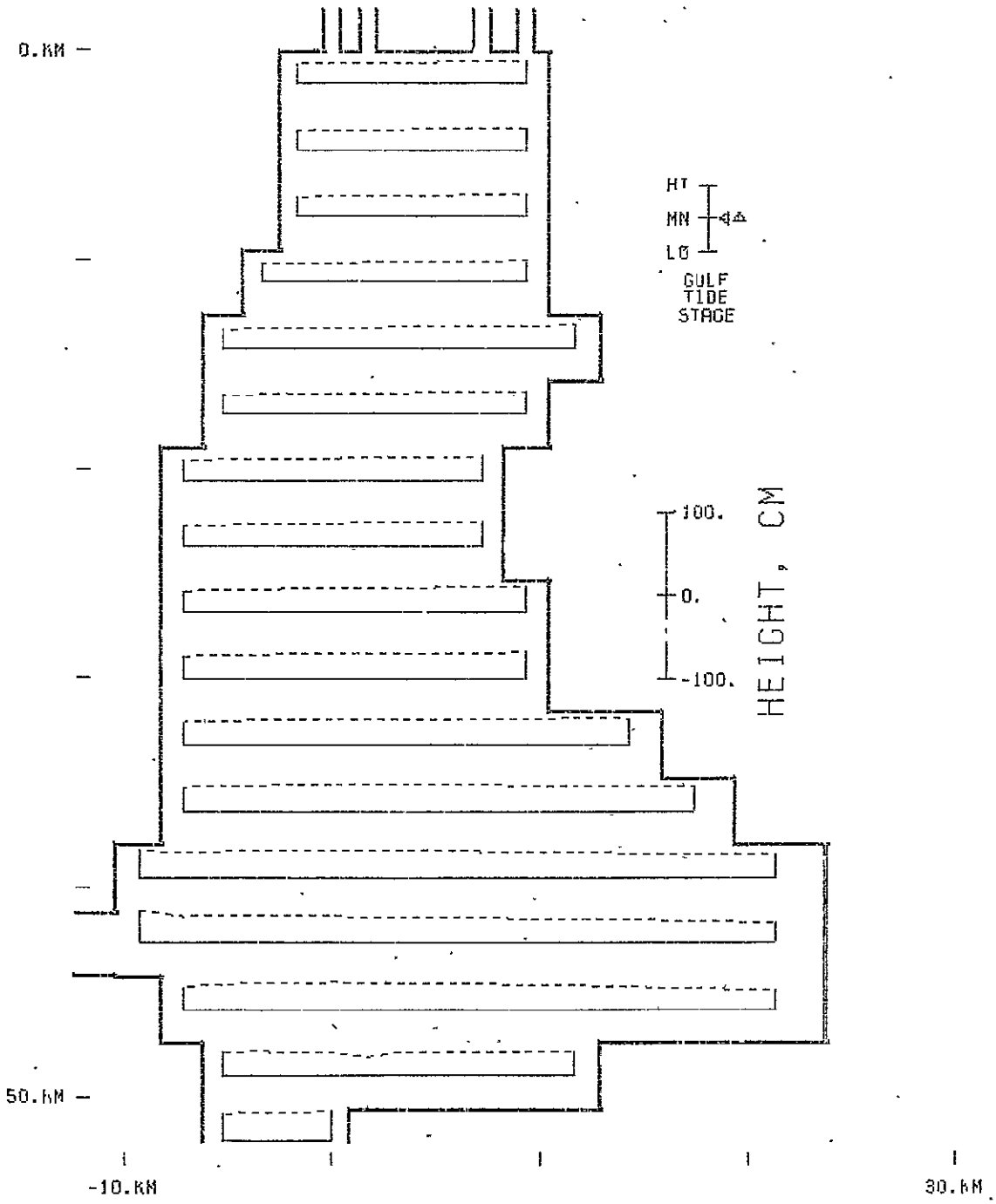


Figure A.131 - Case 3

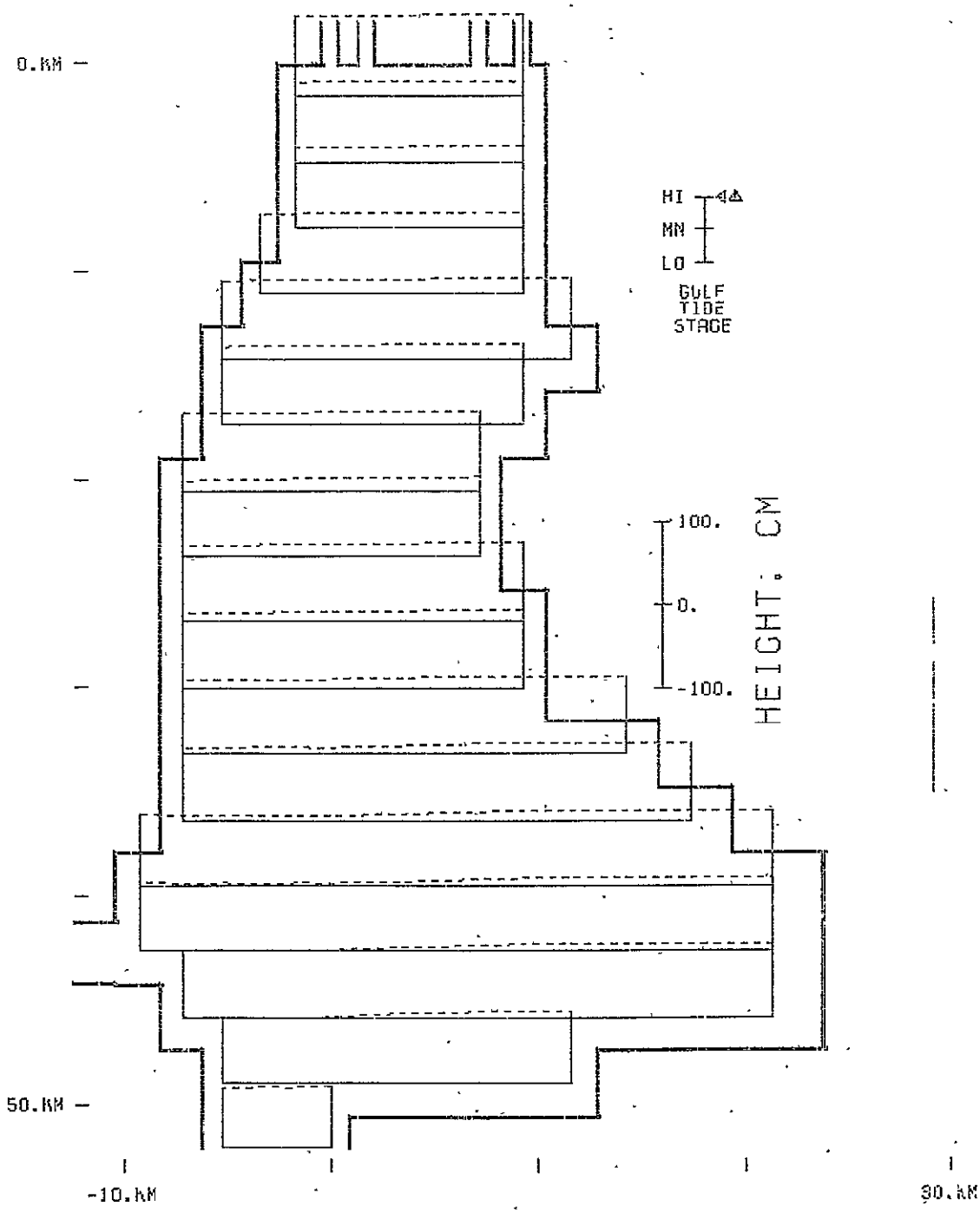


Figure A.132 -

Case 3

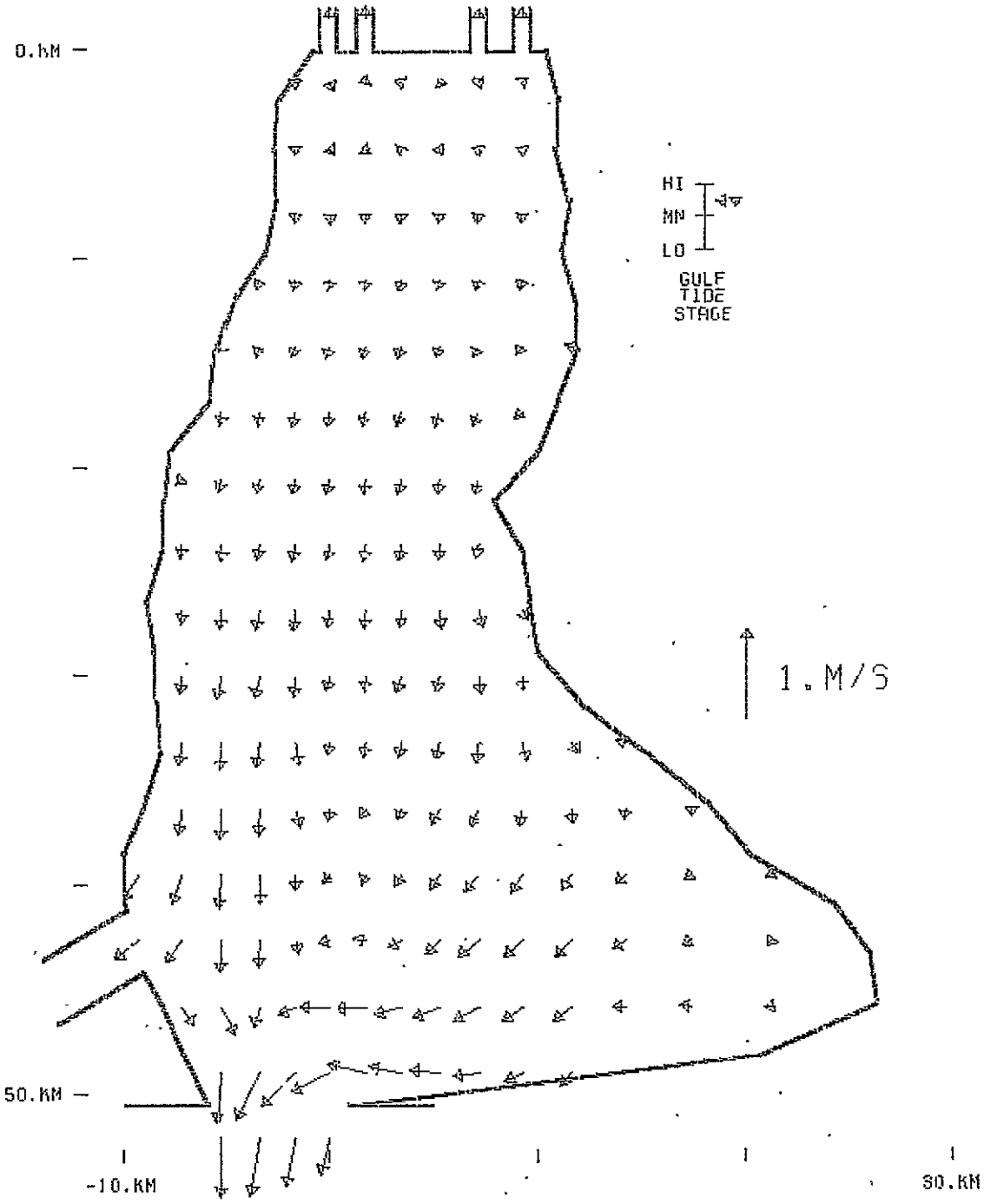


Figure A.133 - Case 4

C.4

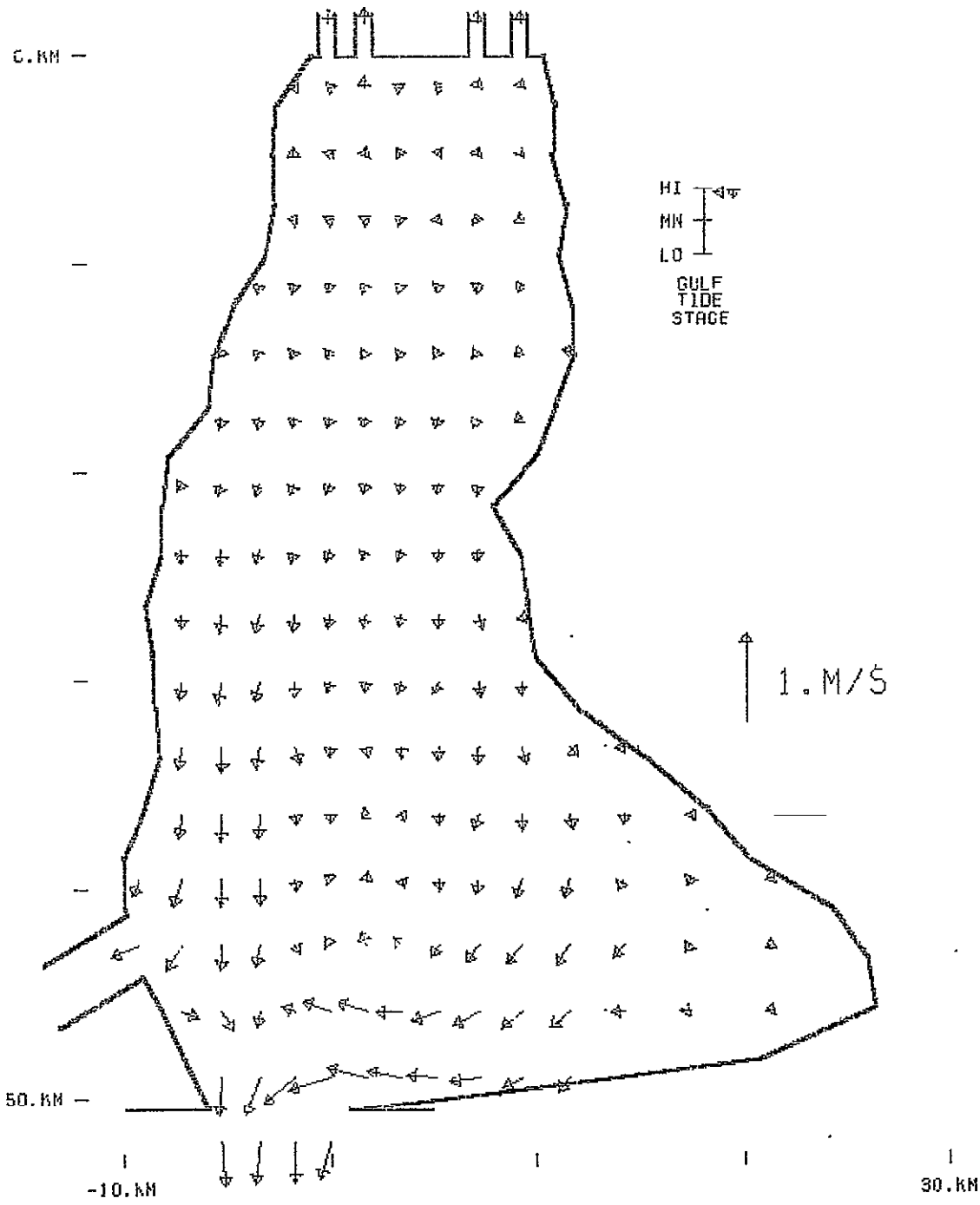


Figure A.134 -

Case 4

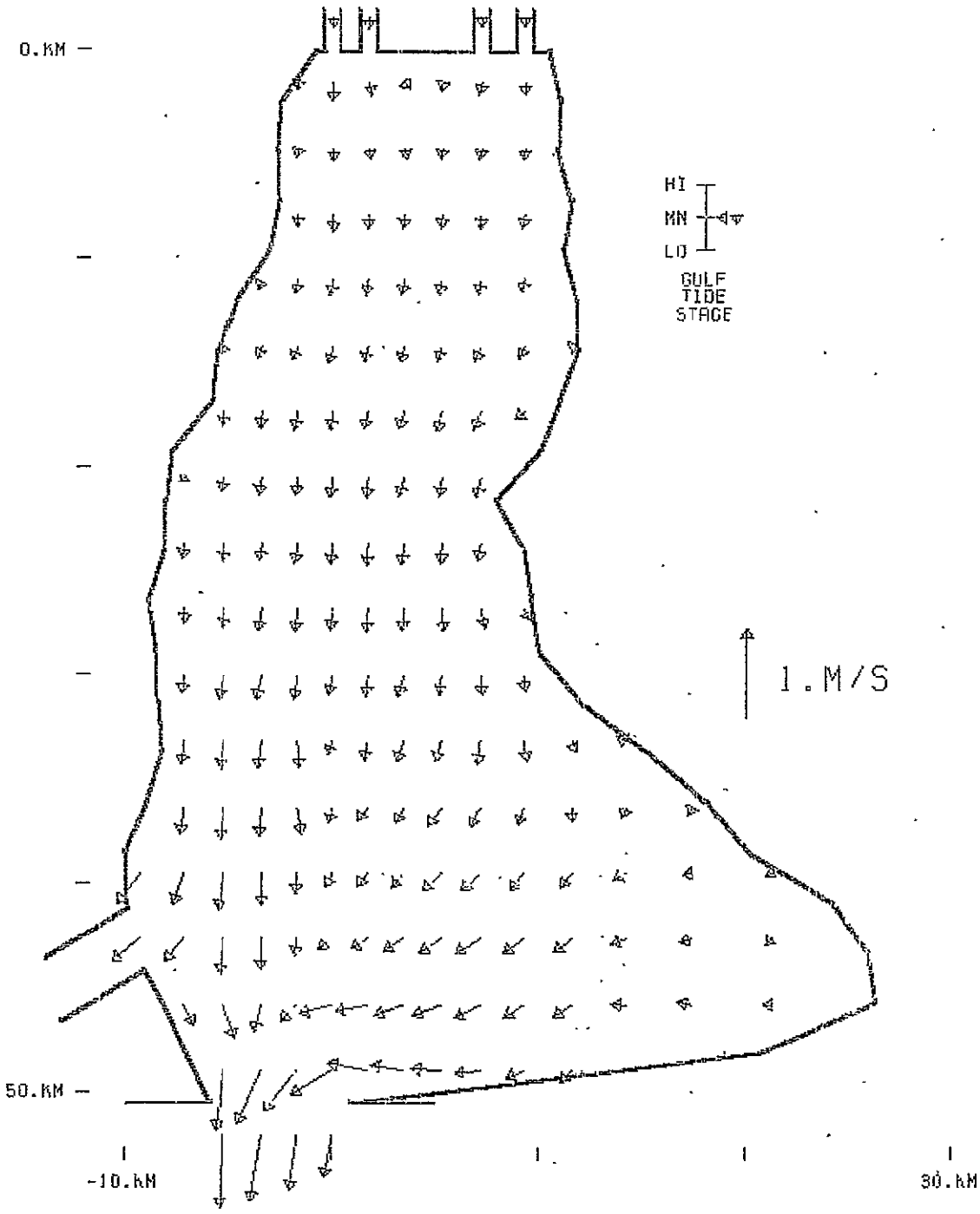


Figure A.135 -

Case 4

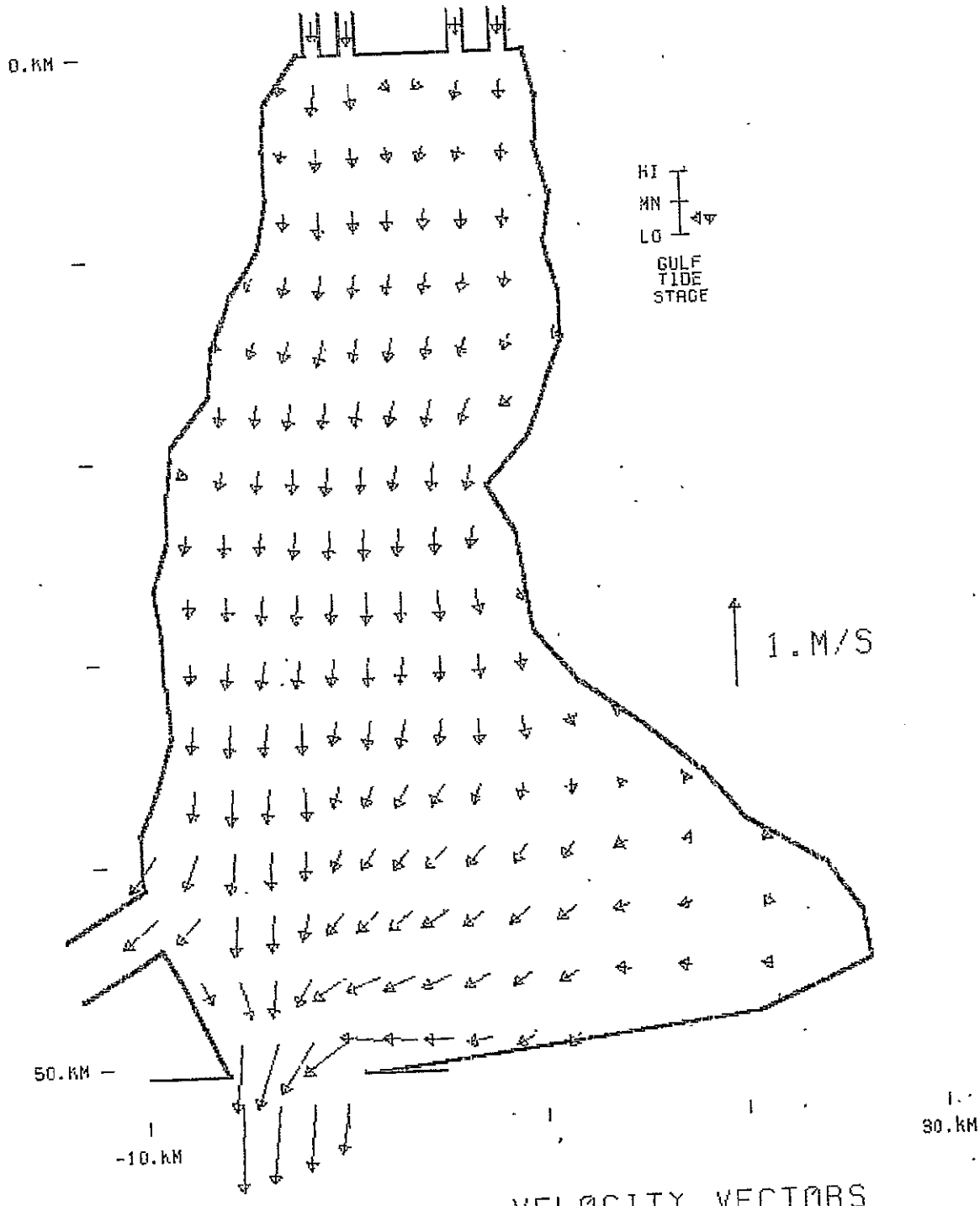


Figure A.136 -

Case 4

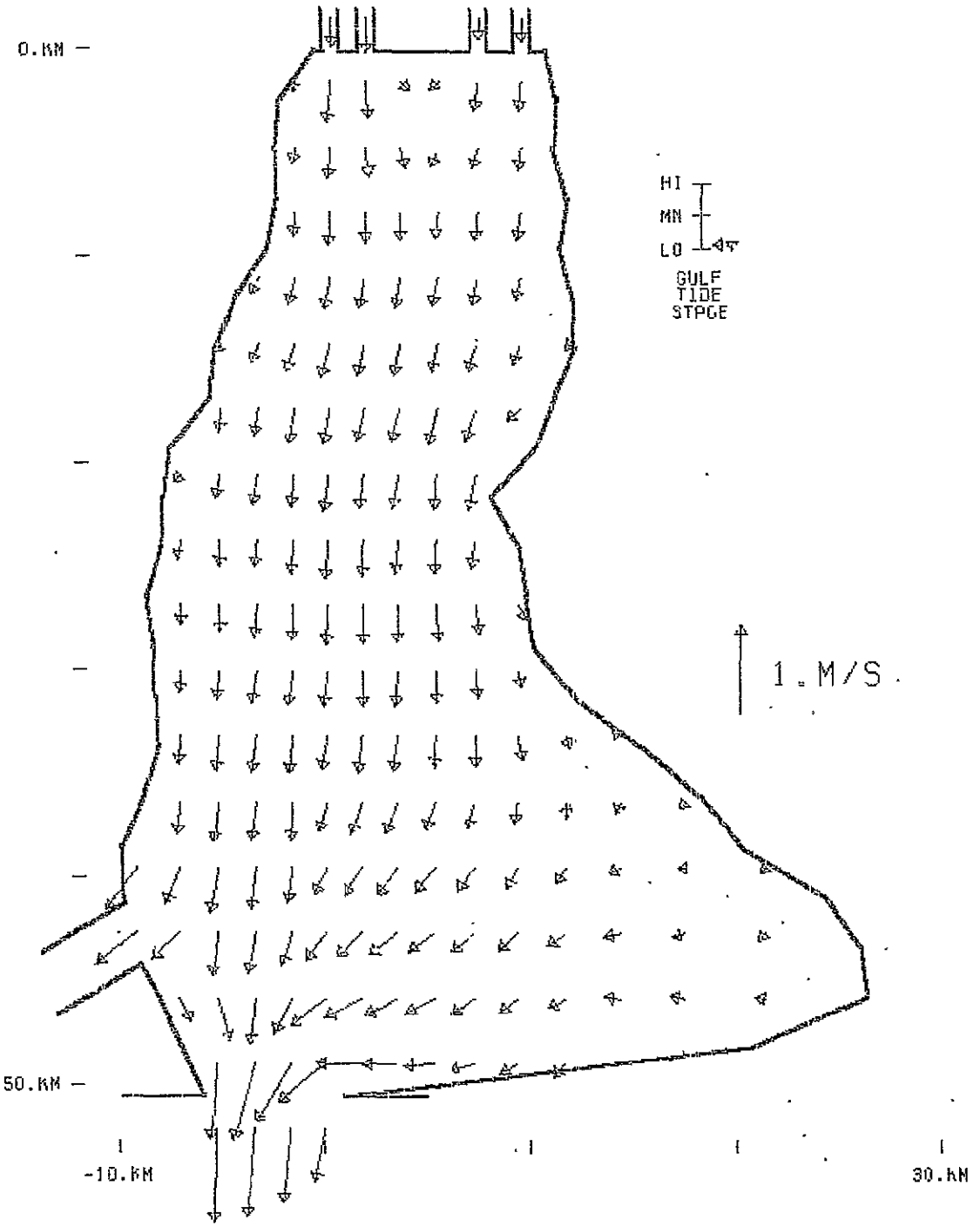


Figure A.137 -

Case 4

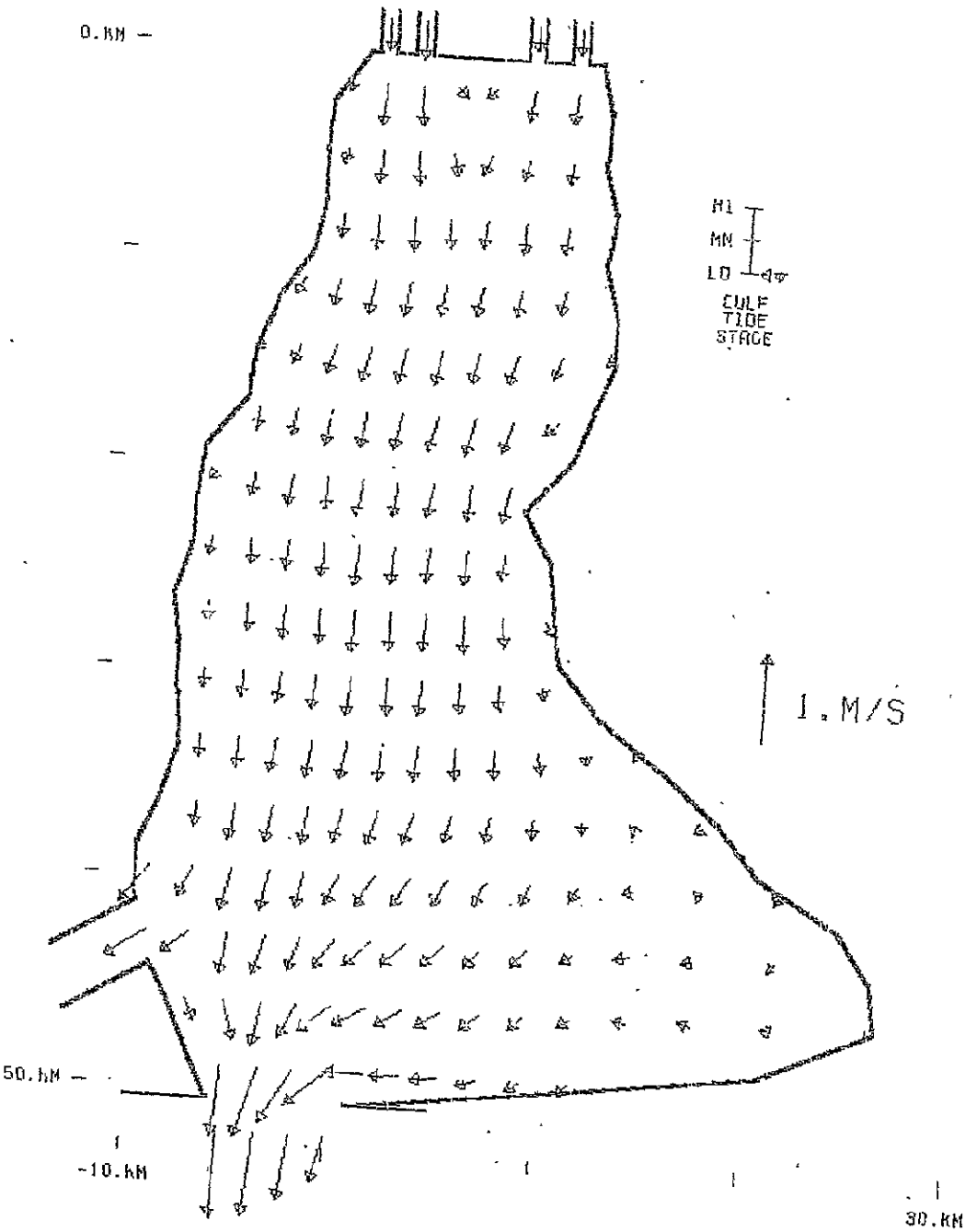


Figure A.138 -

Case 4

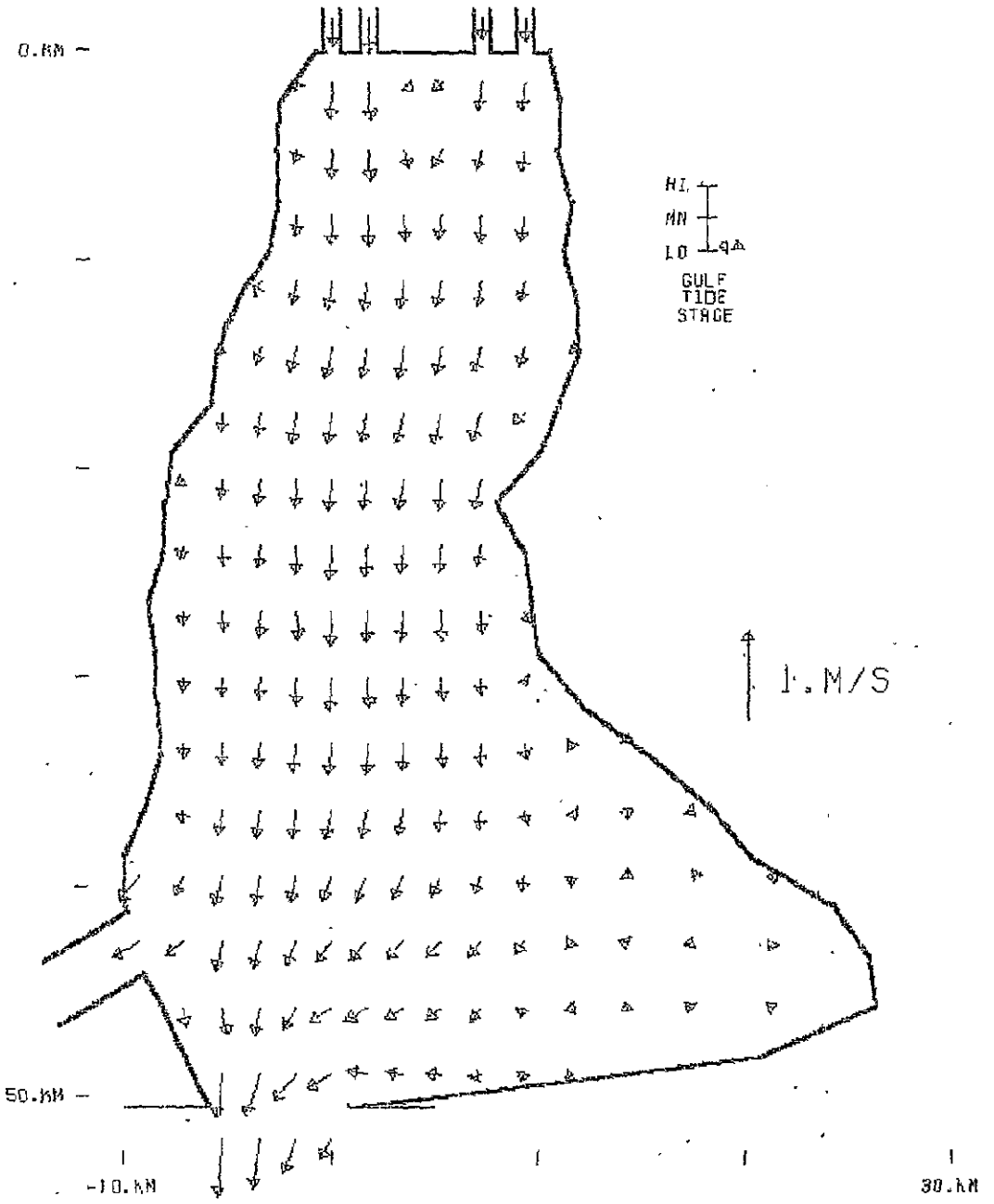


Figure A.139 -

Case 4

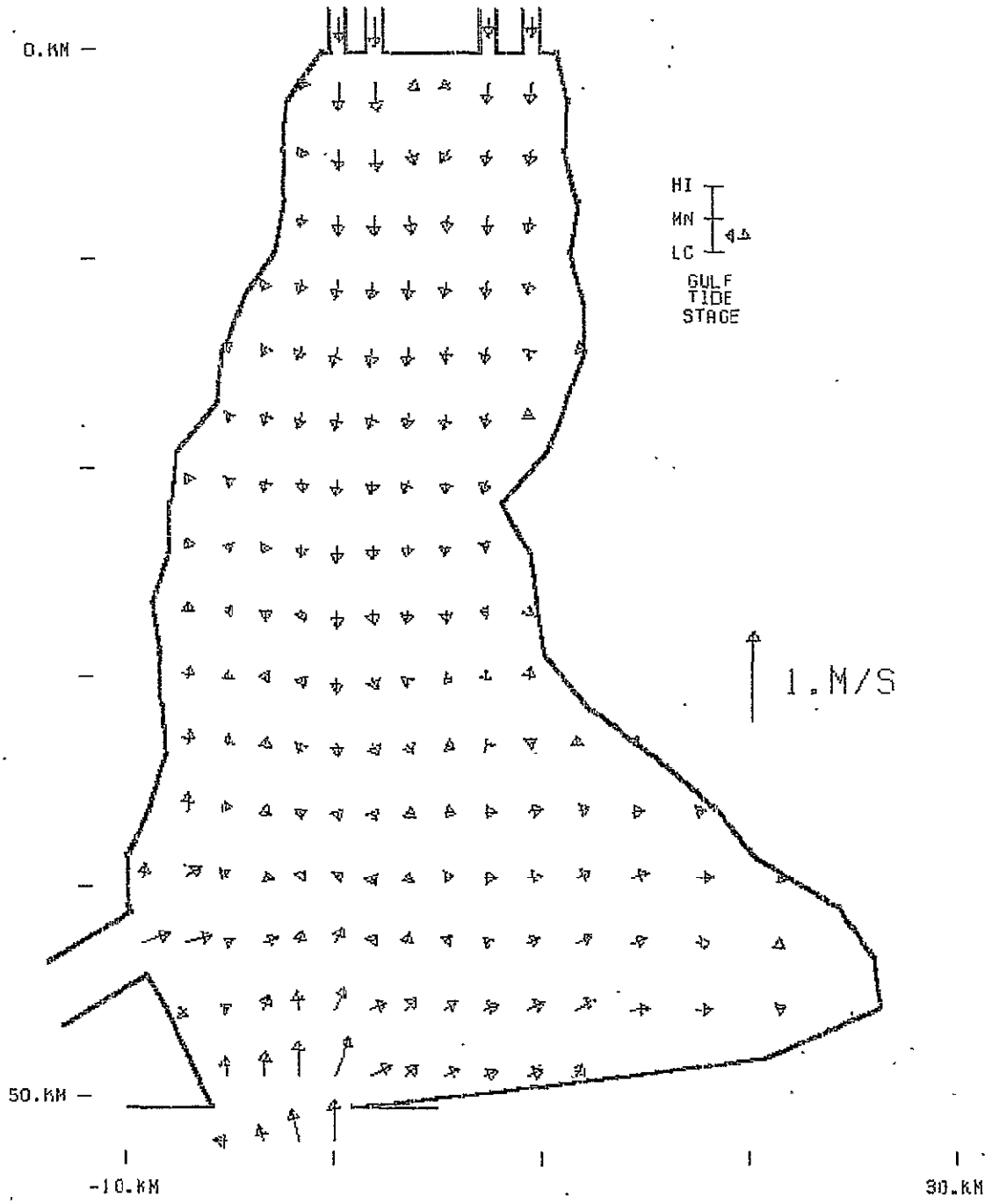


Figure A.140 -

Case 4

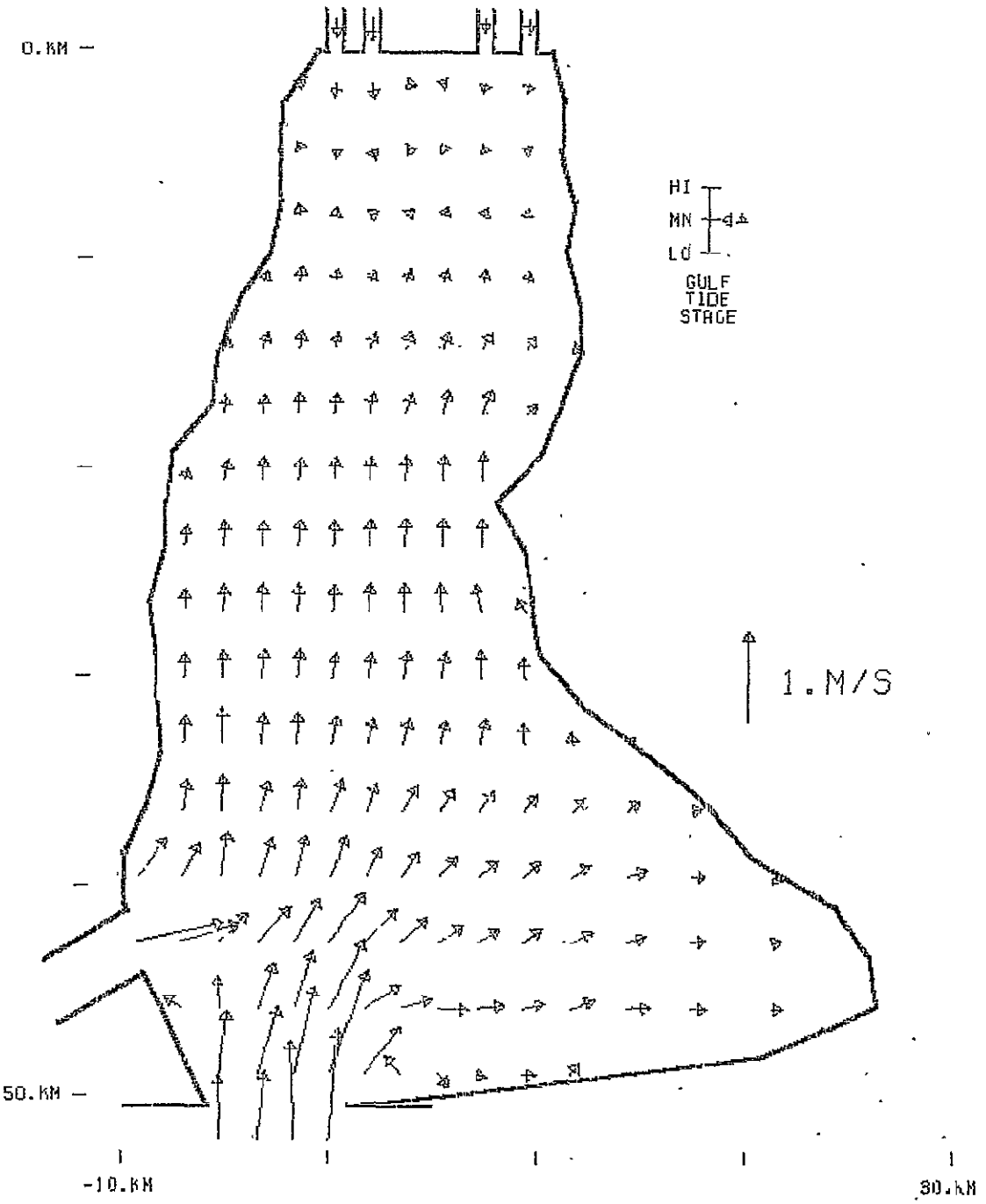


Figure A.141 -

Case 4

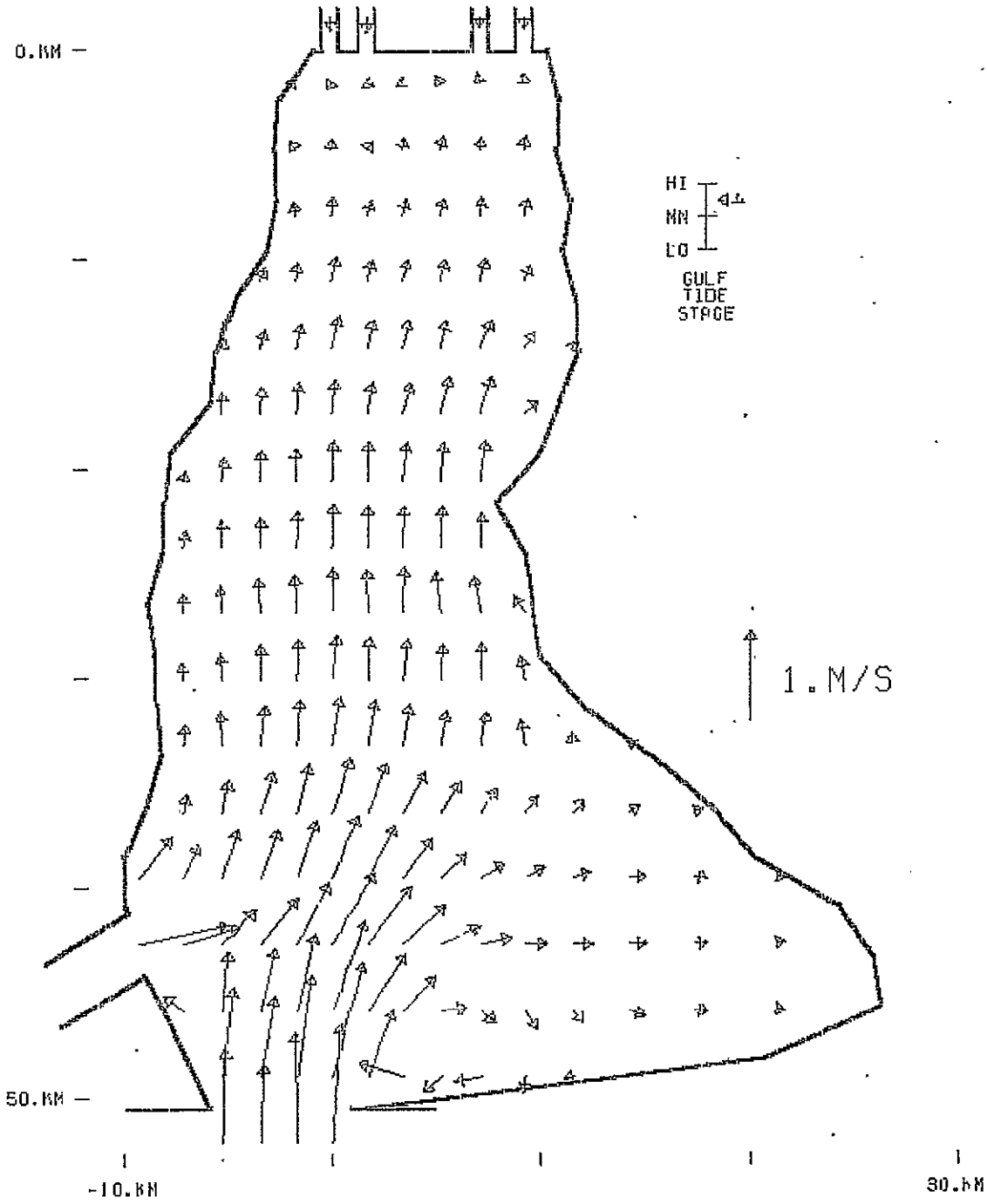


Figure A.142 -

Case 4

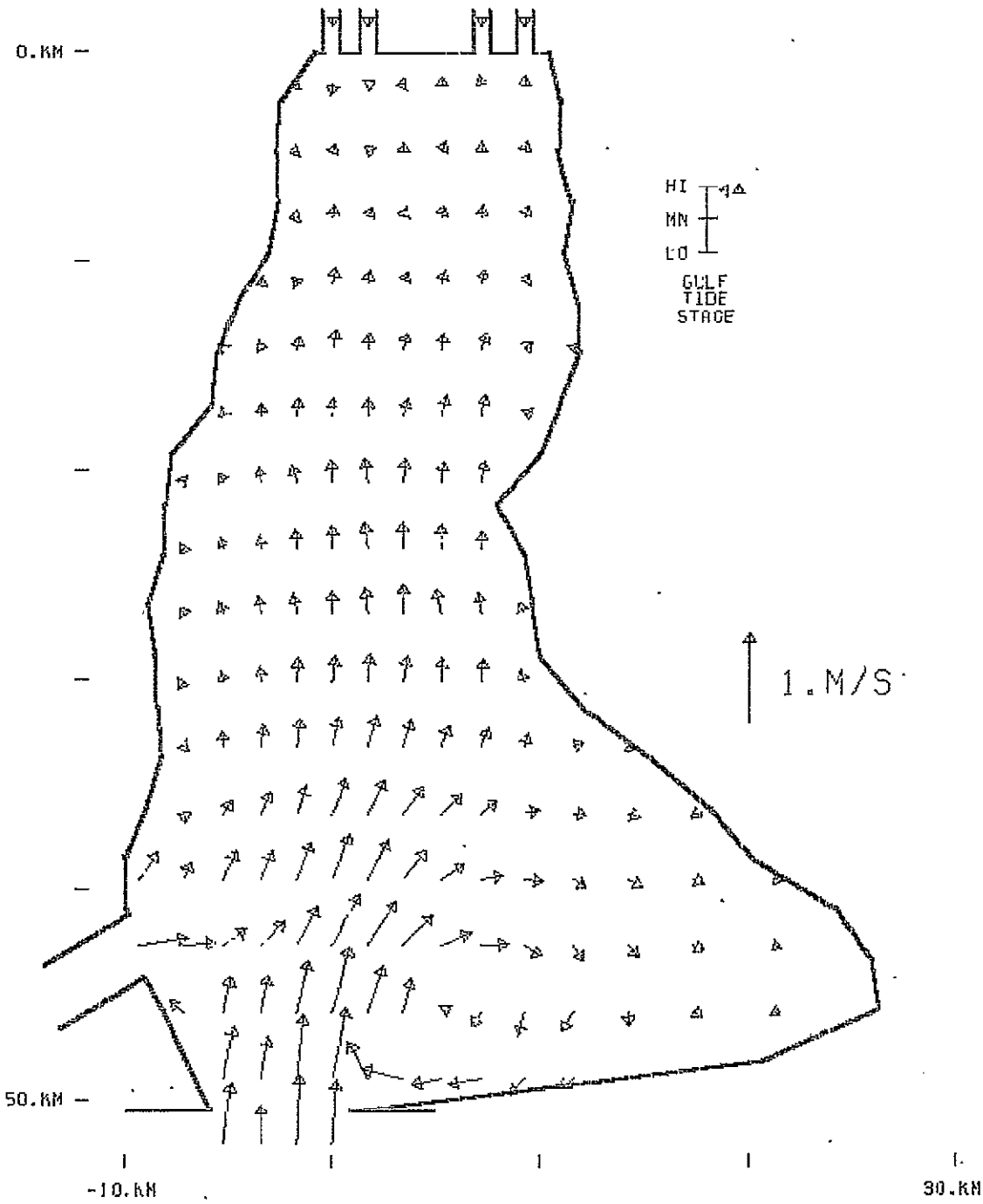


Figure A.143 -

Case 4

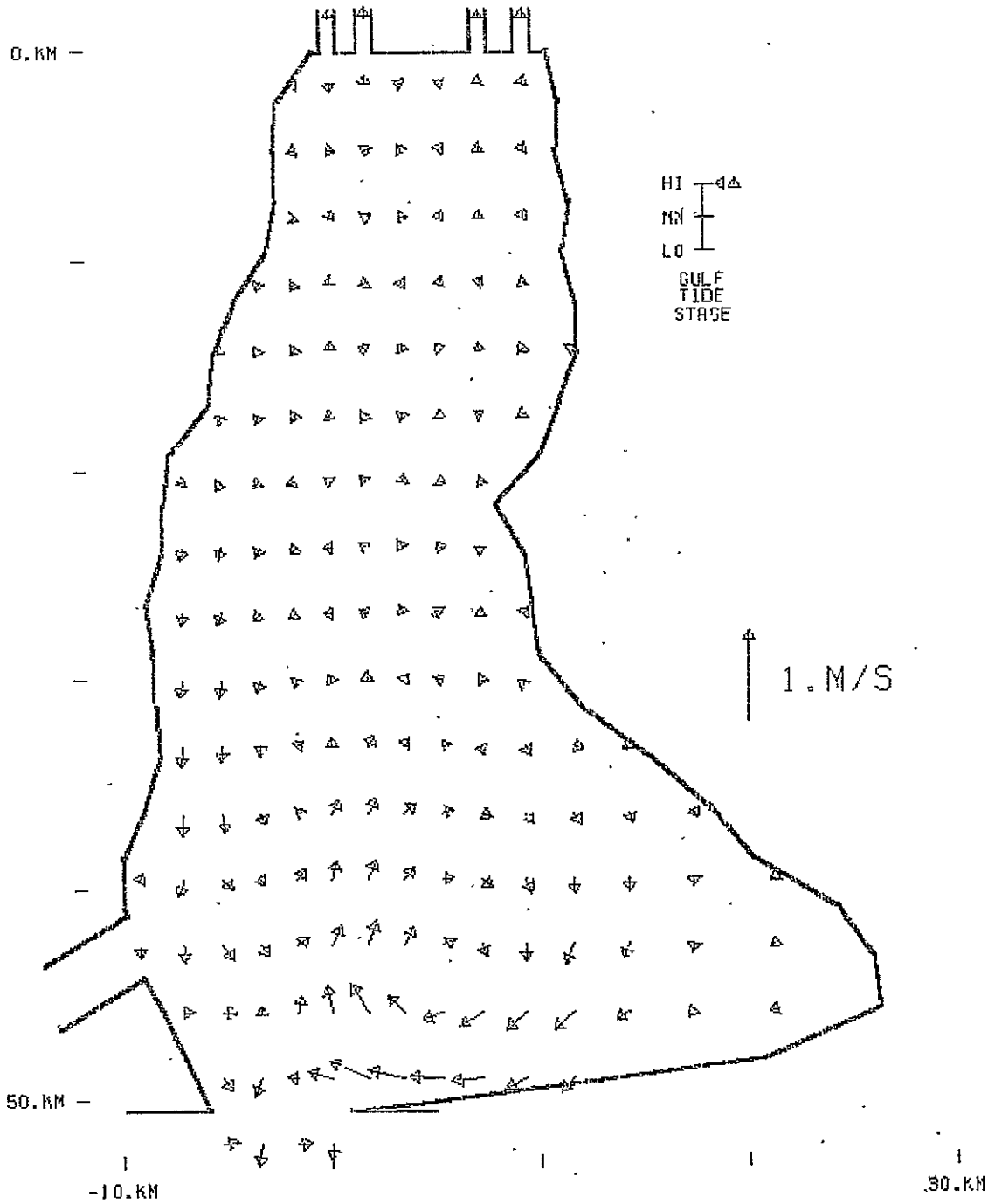


Figure A.144 -

Case 4

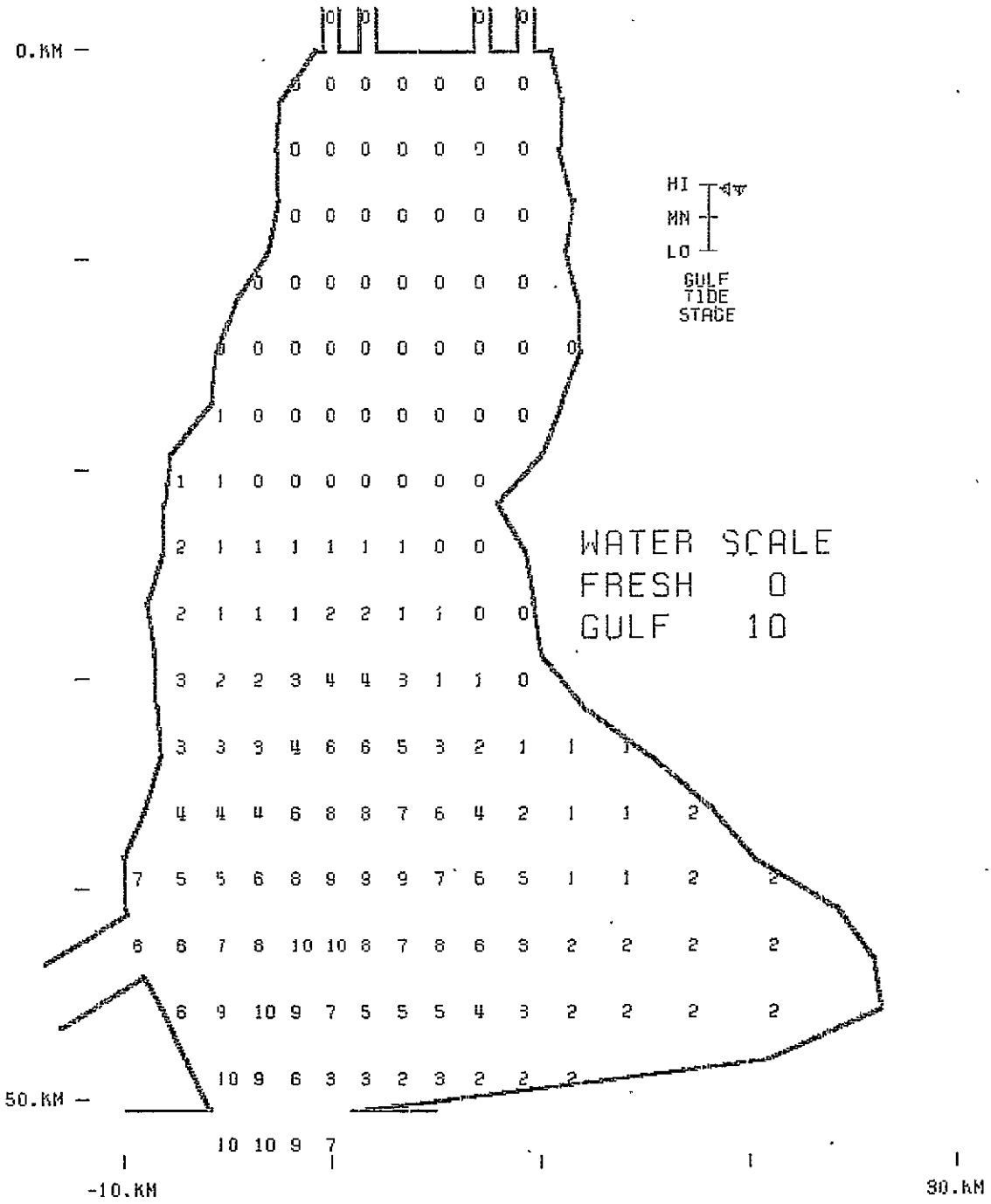


Figure A.145 -

Case 4

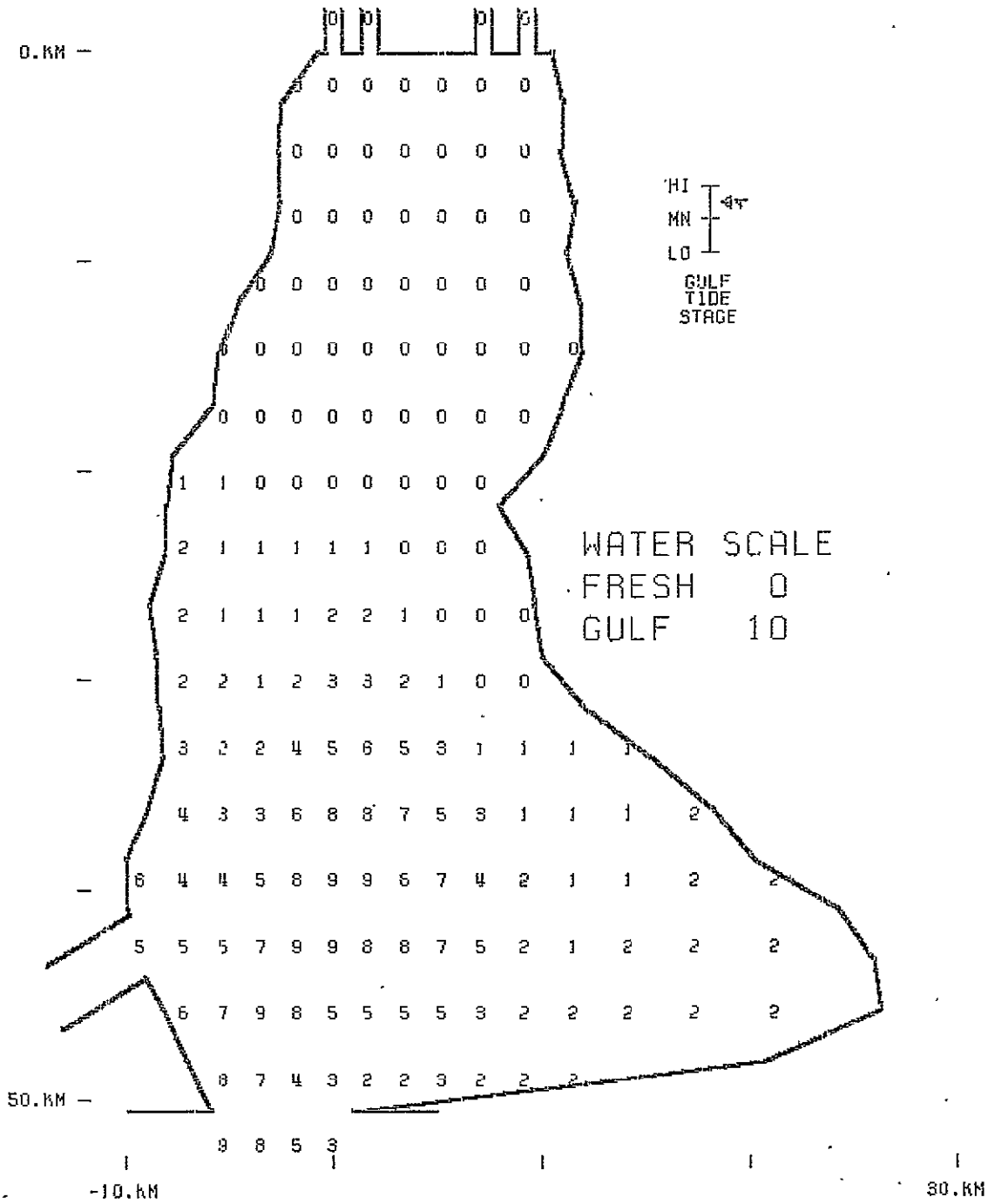


Figure A.146 -

Case 4

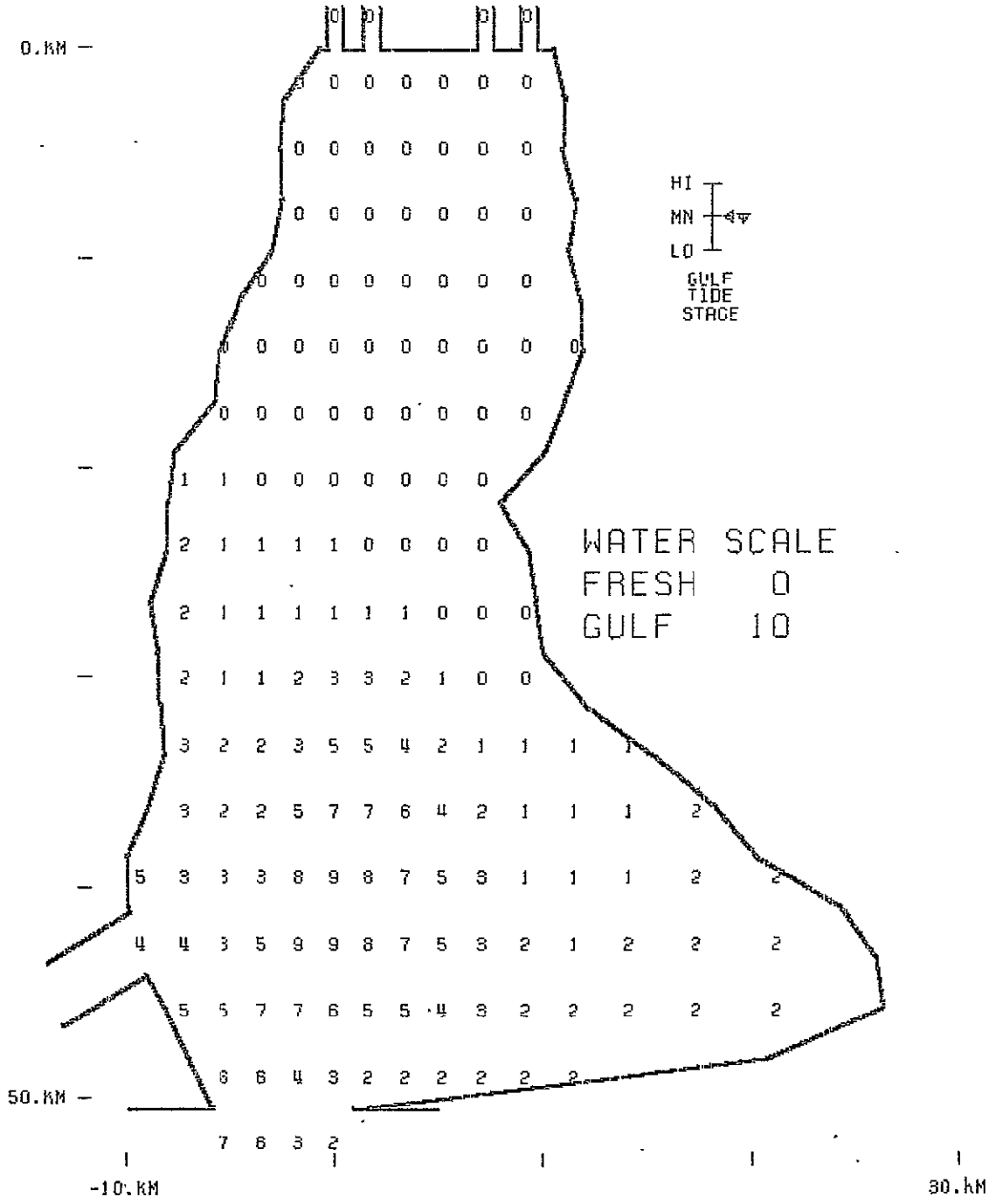


Figure A.147 -

Case 4

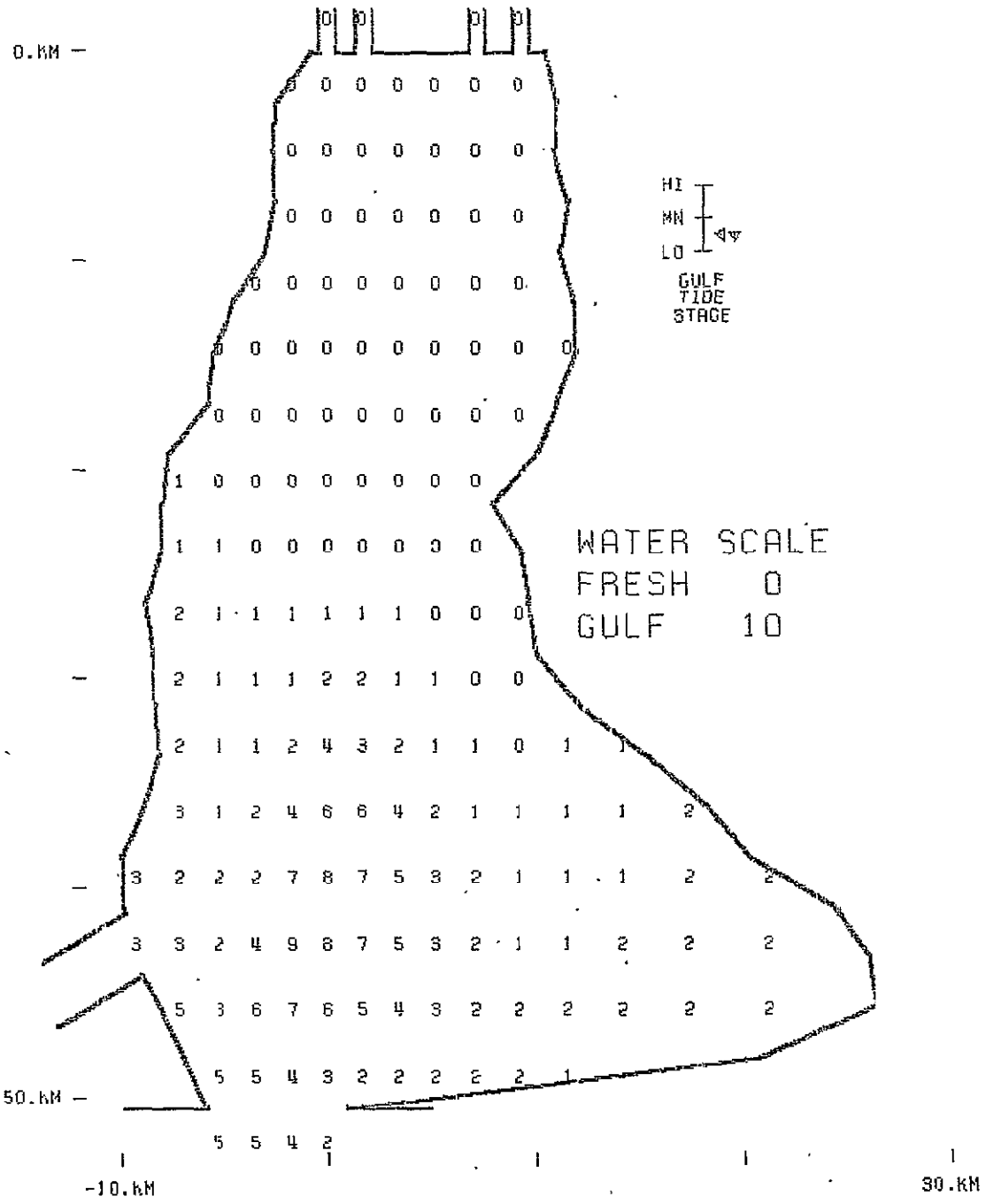


Figure A.148 -

Case 4

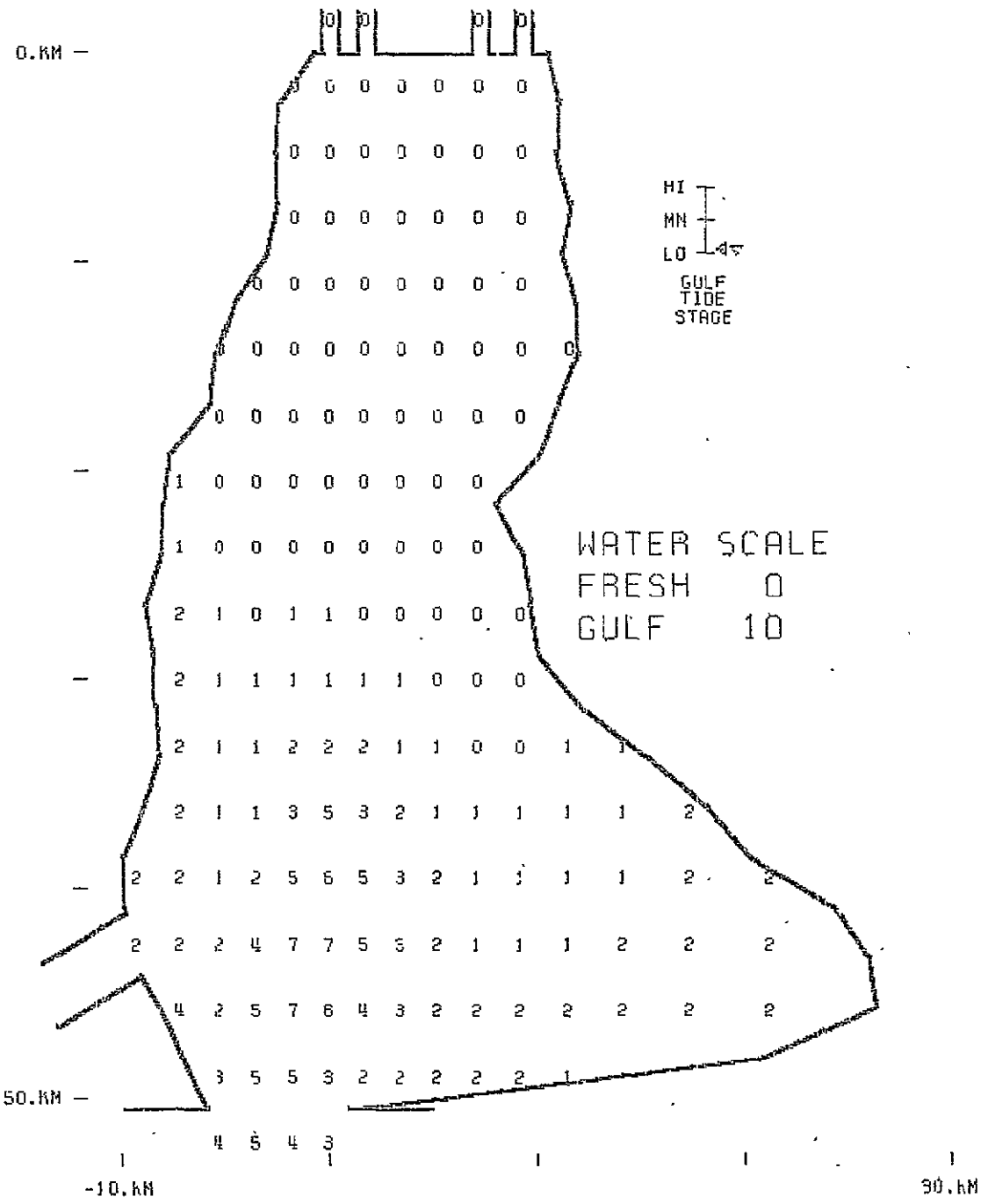


Figure A.149 -

Case 4

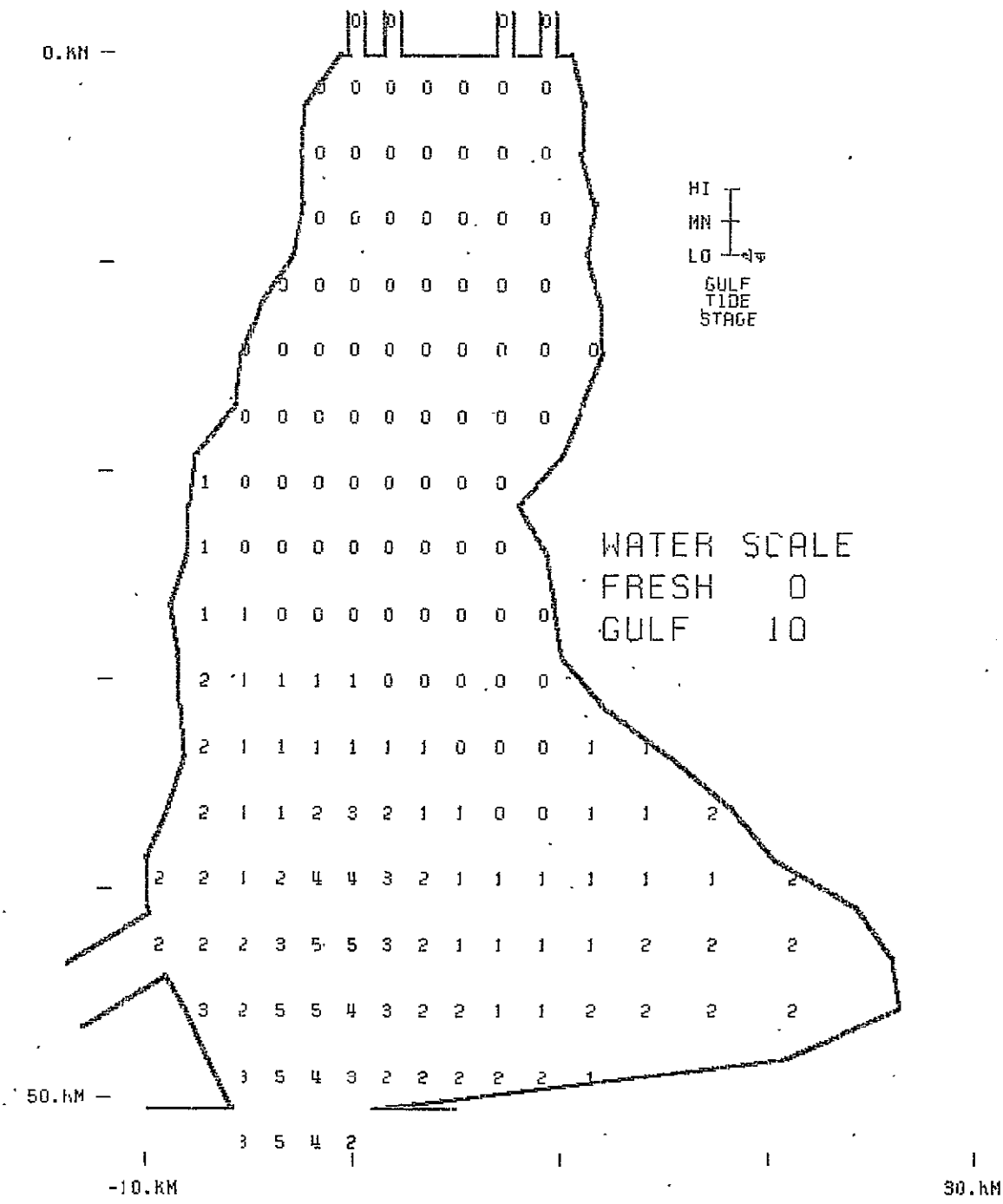


Figure A.150 -

Case 4

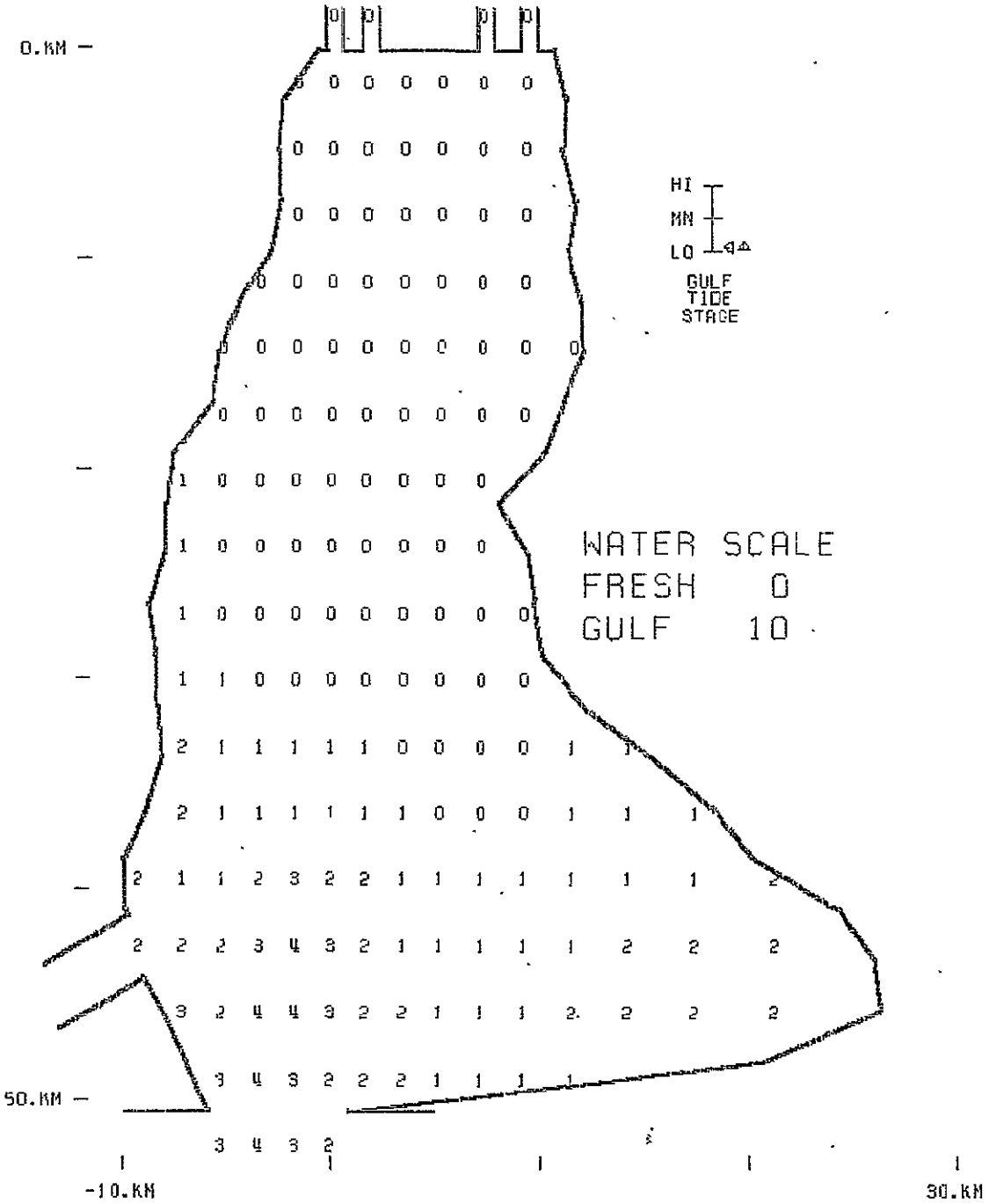


Figure A.151 -

Case 4

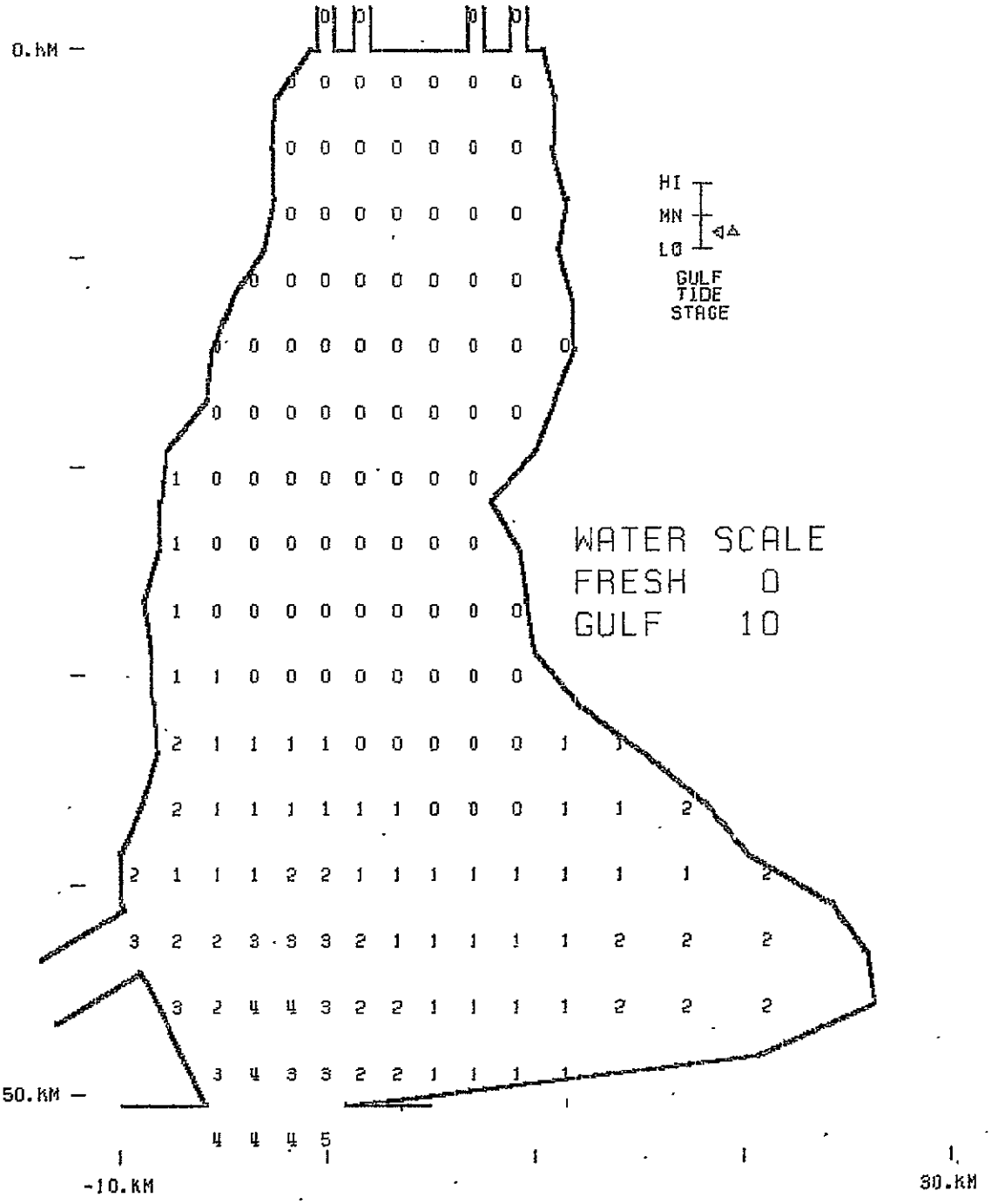


Figure A.152 -

Case 4

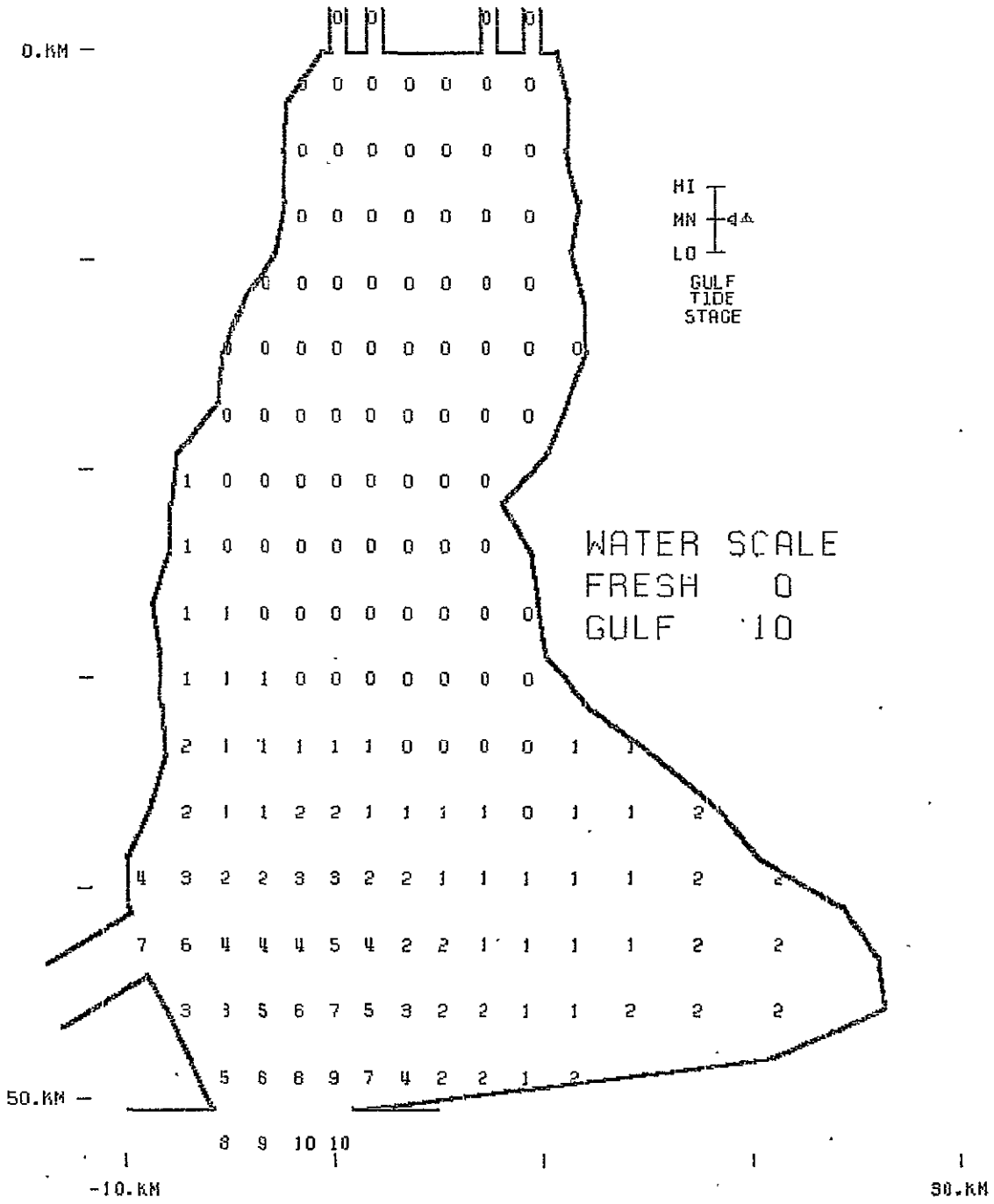


Figure A.153 -

Case 4

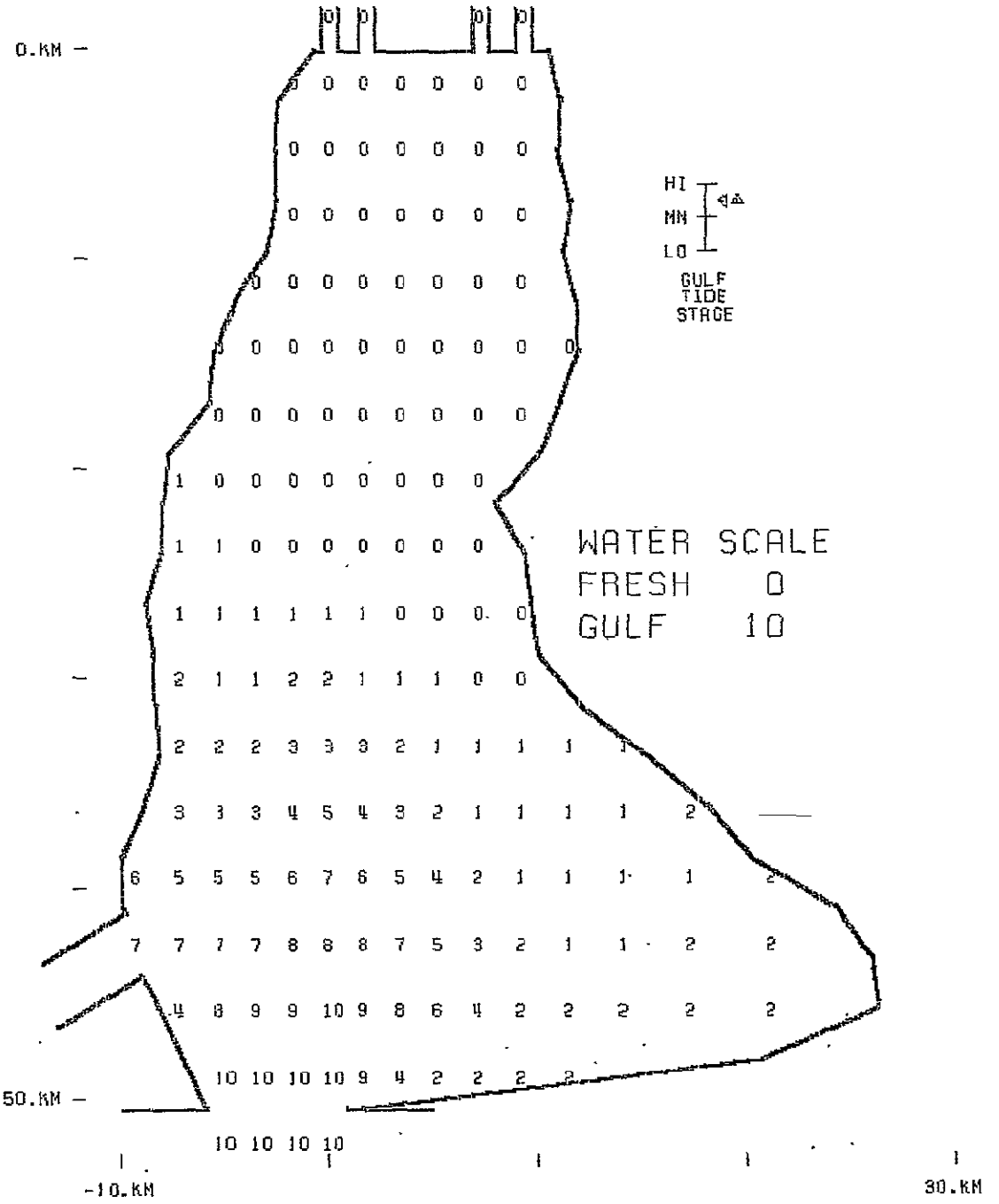


Figure A.154 - Case 4

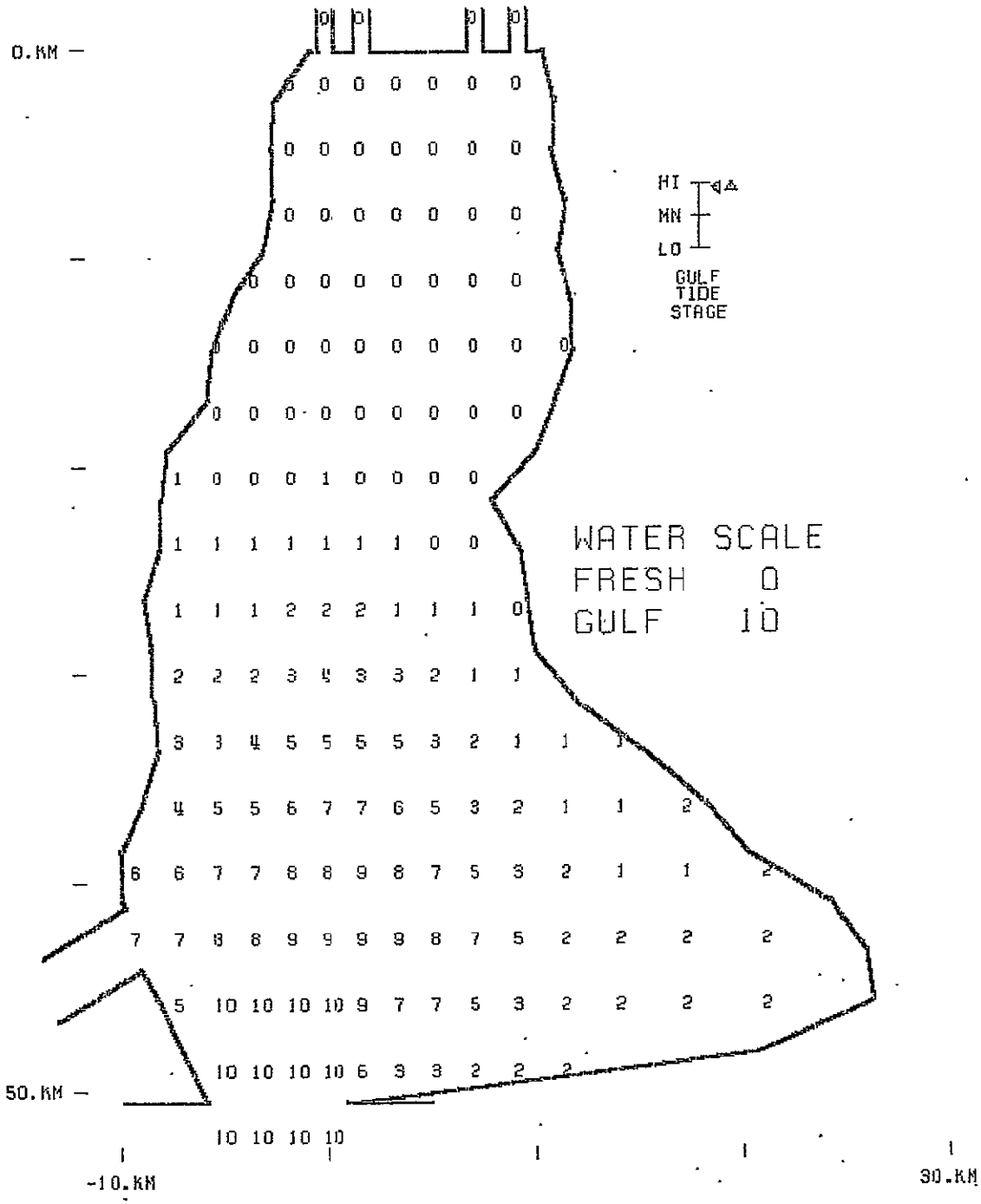


Figure A.155 -

Case 4

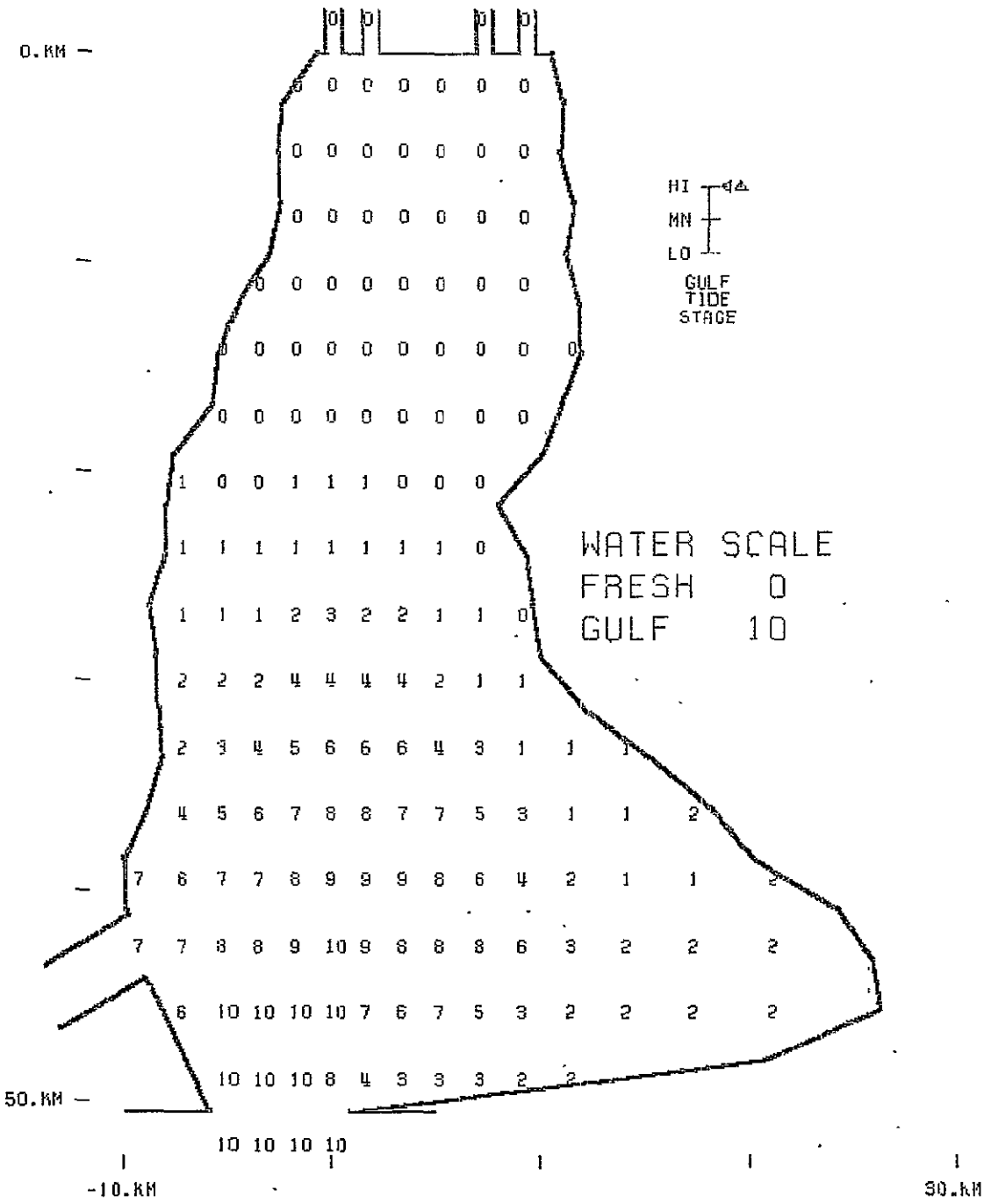


Figure A.156 -

Case 4

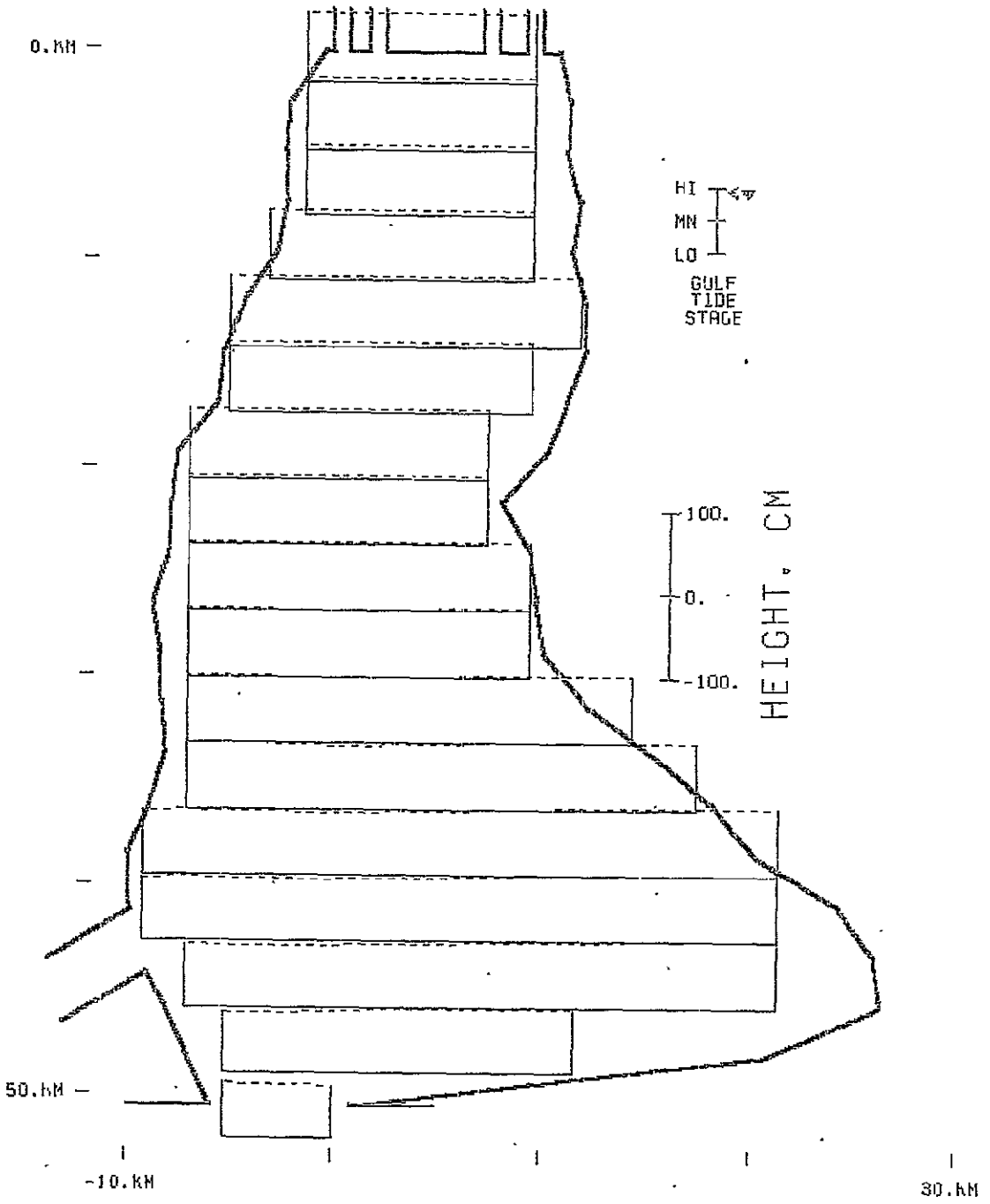


Figure A.157 -

Case 4

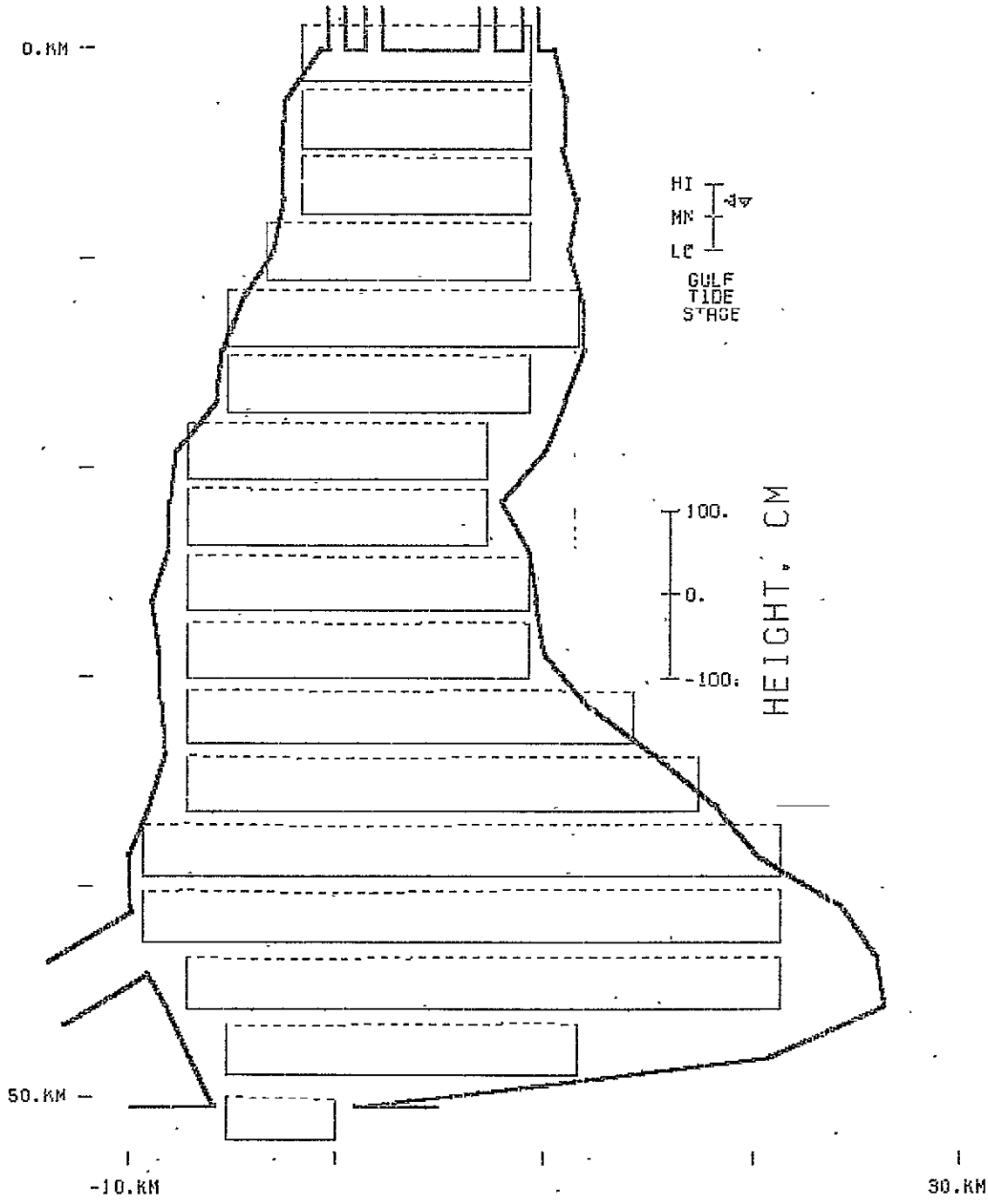


Figure A.158 -

Case 4

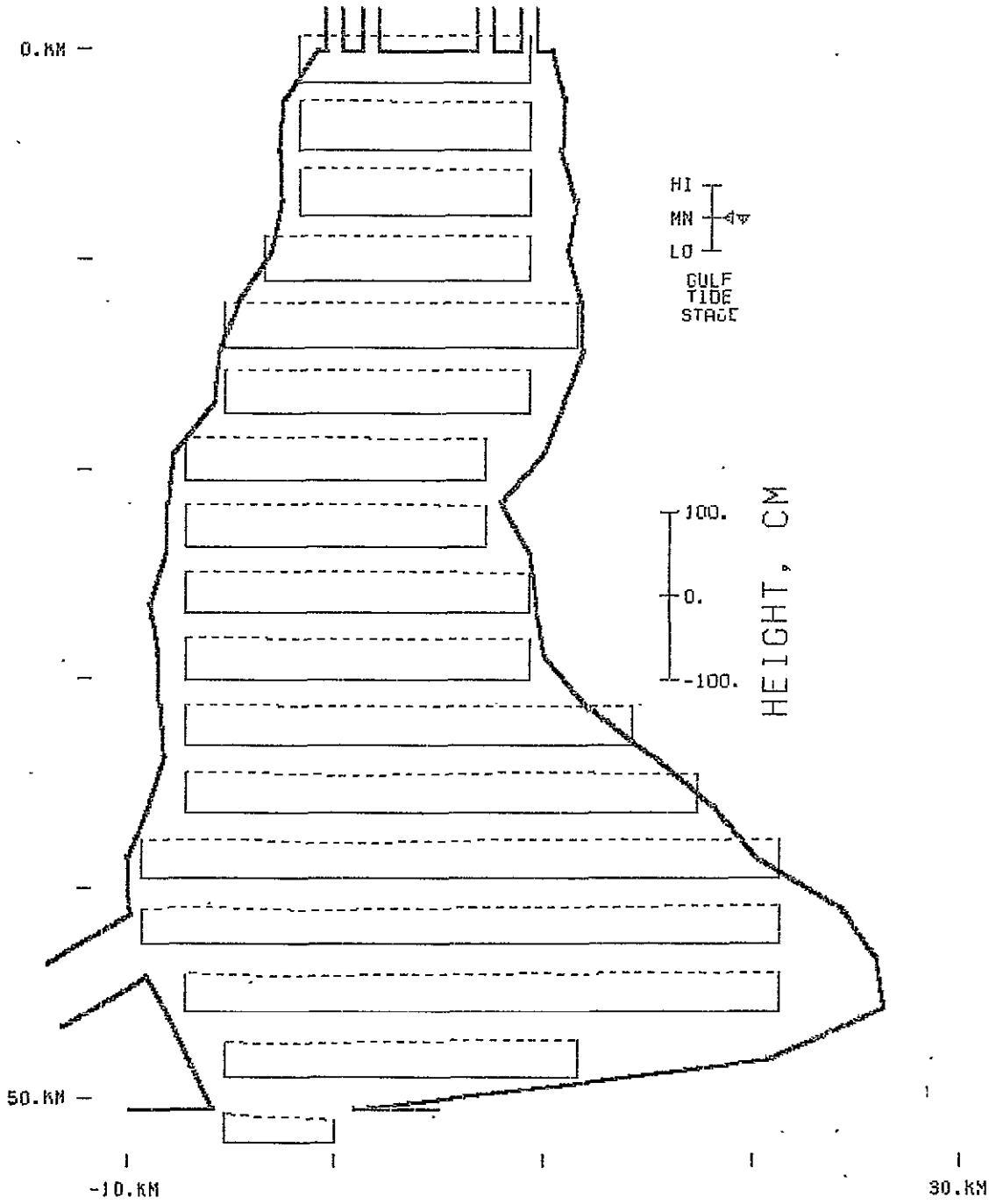


Figure A.159 -

Case 4

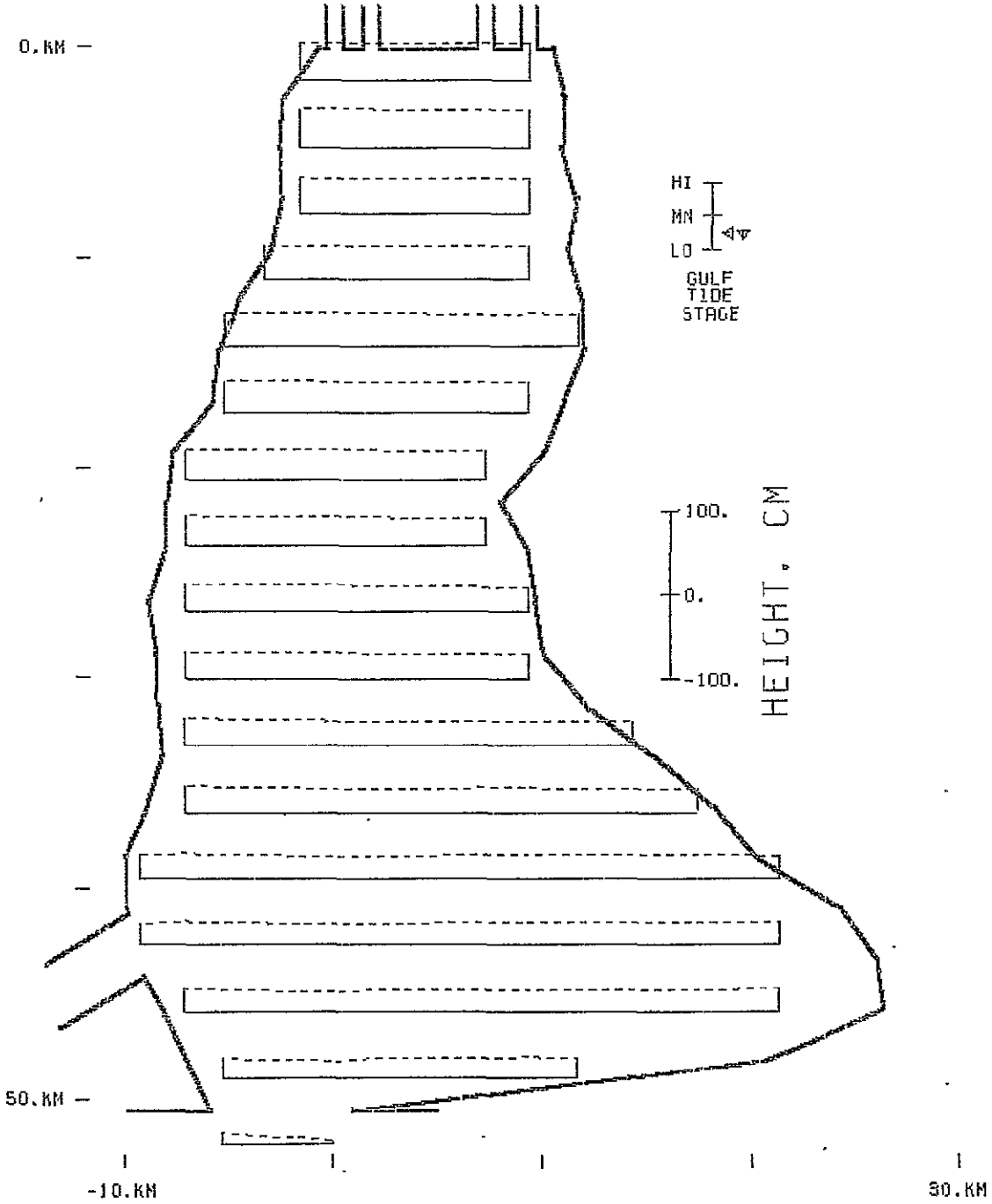


Figure A.160 -

Case 4

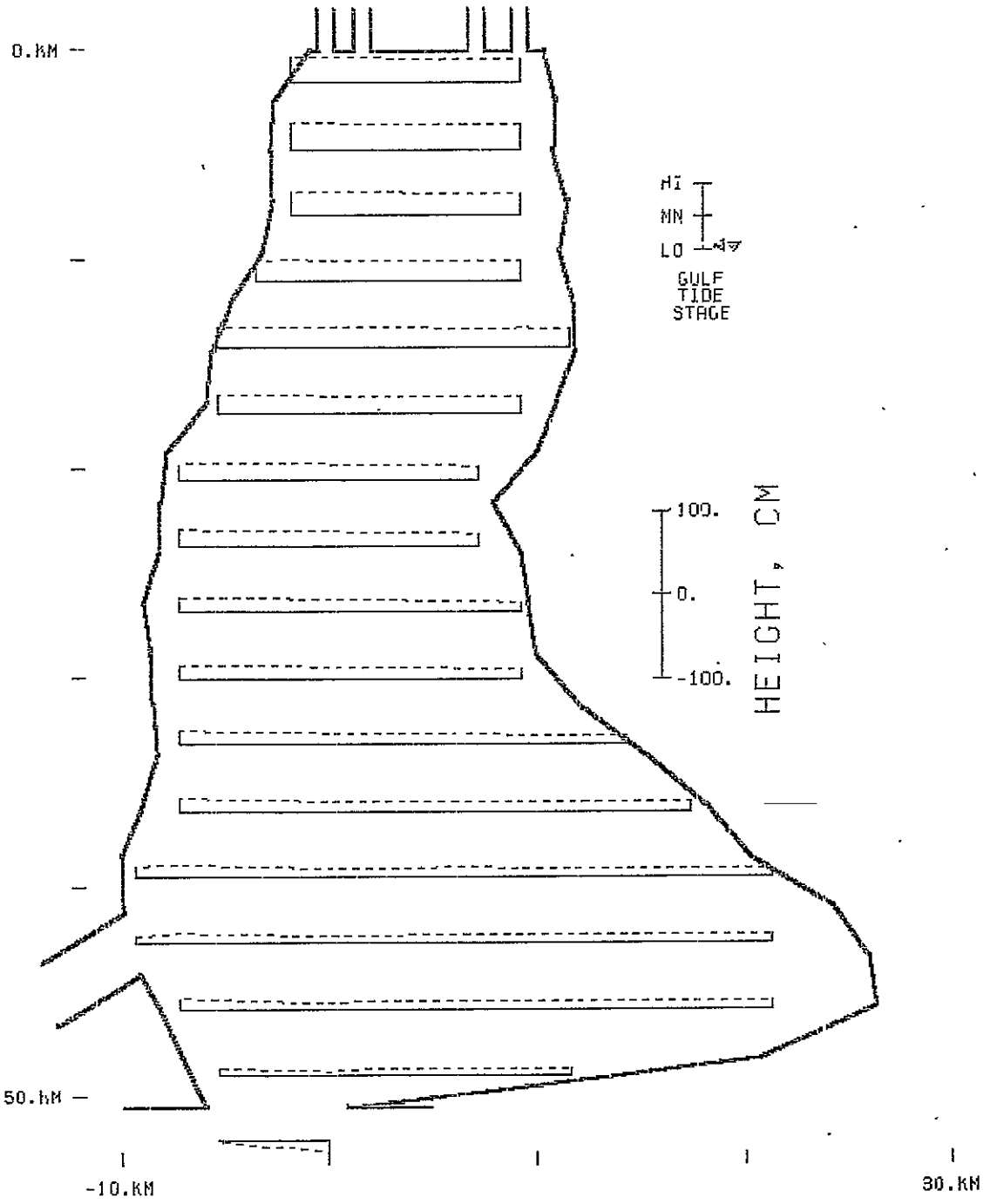


Figure A.161 -

Case 4

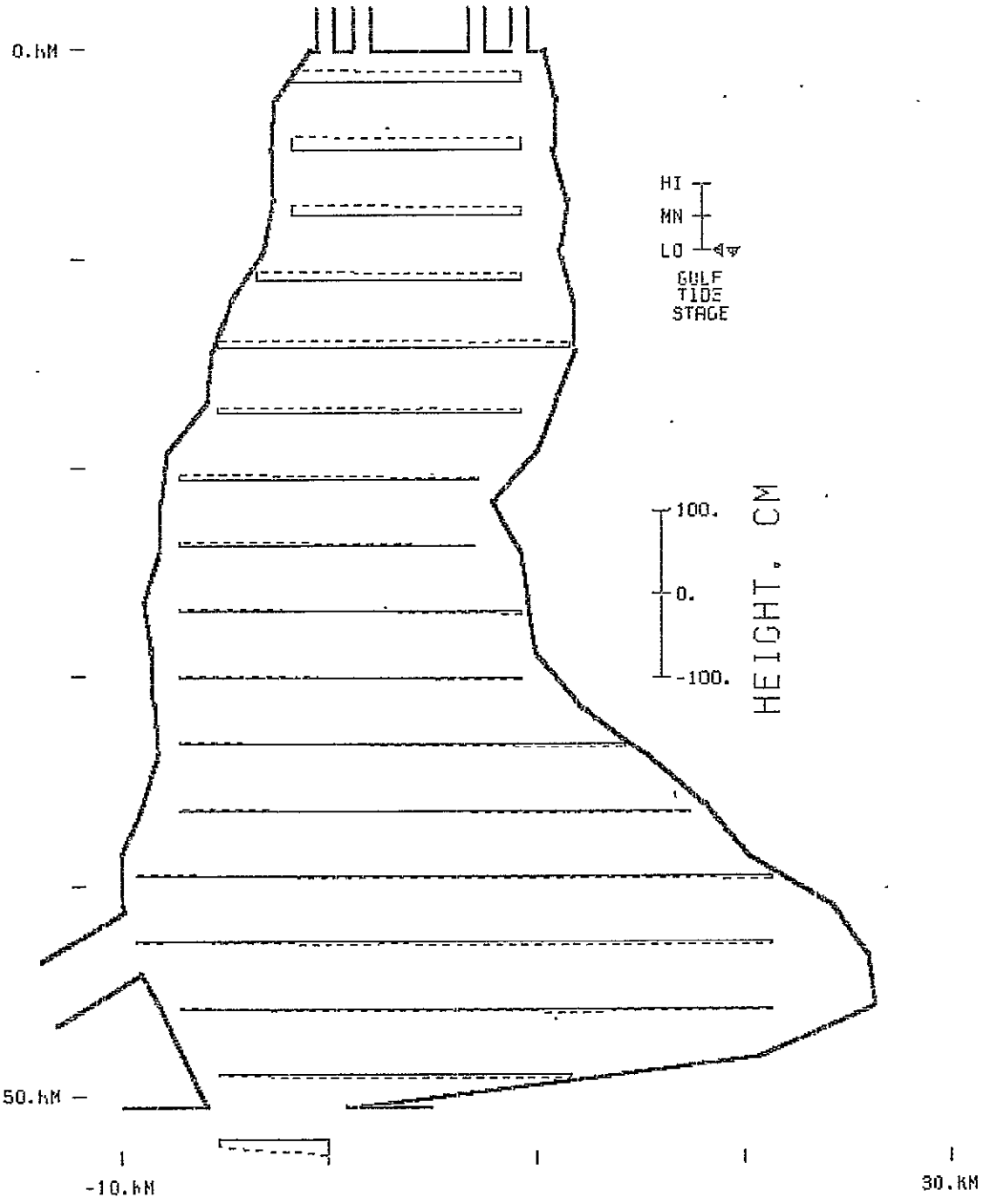


Figure A.162 -

Case 4

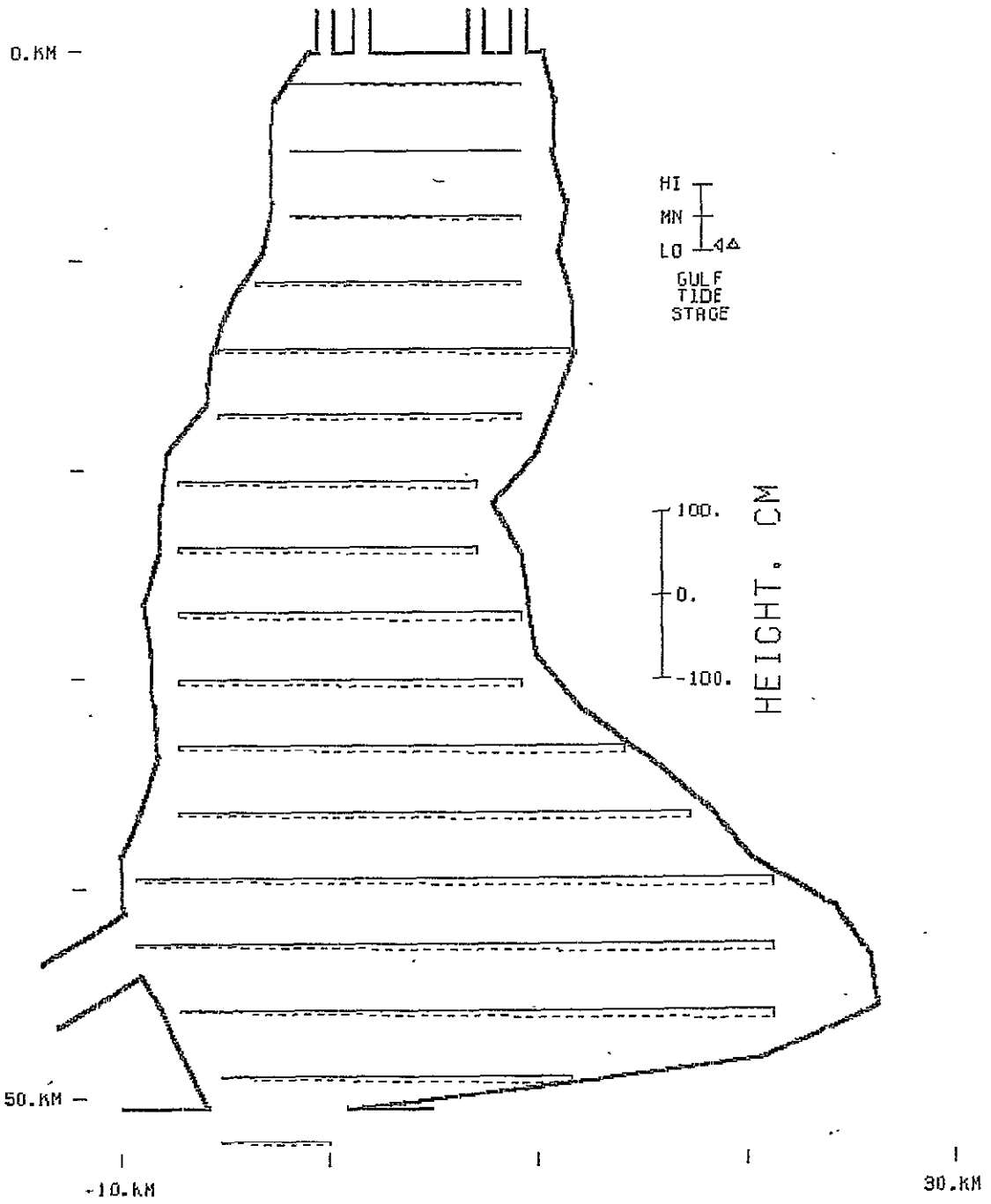


Figure A.163 -

Case 4

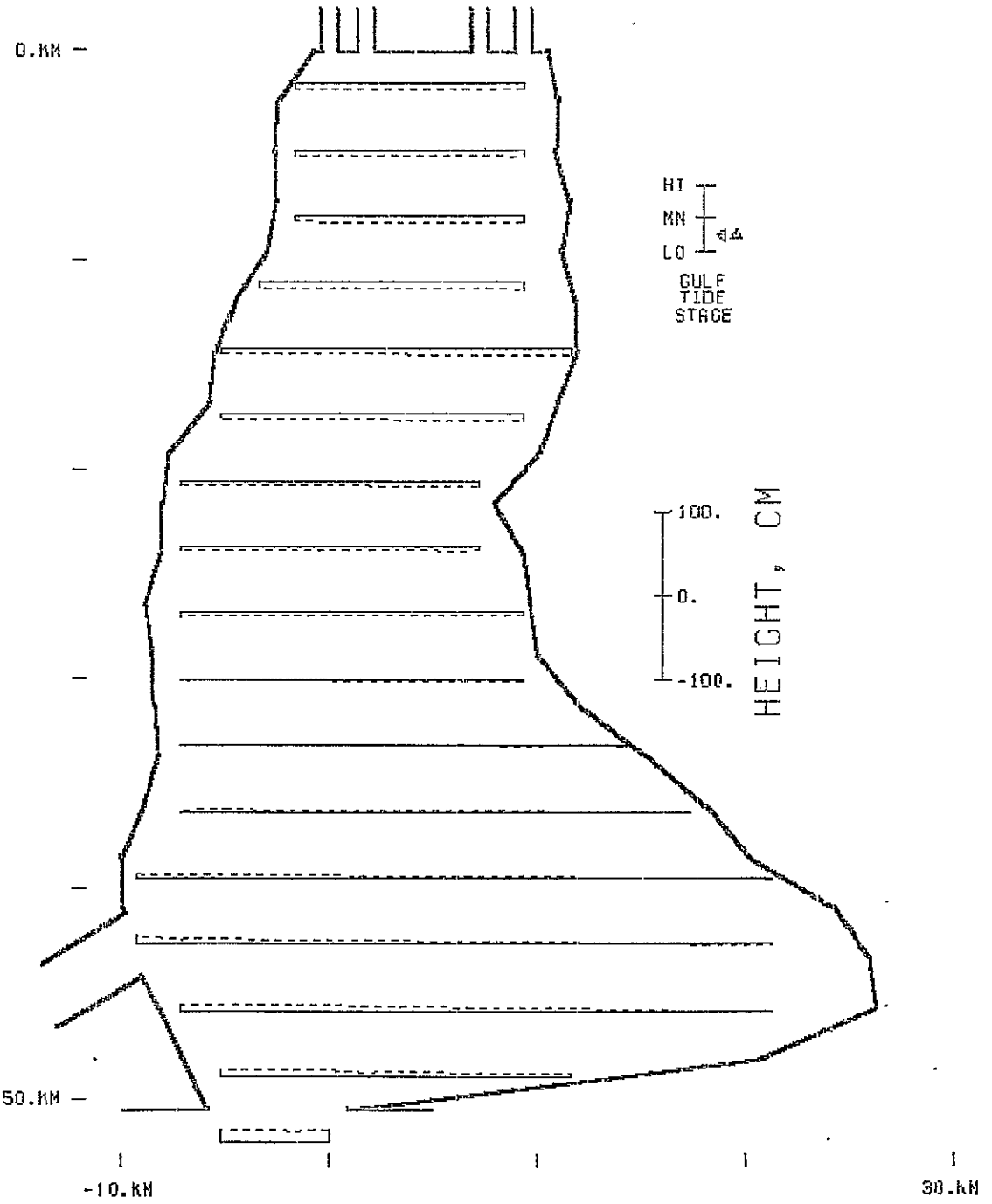


Figure A.164 -

Case 4

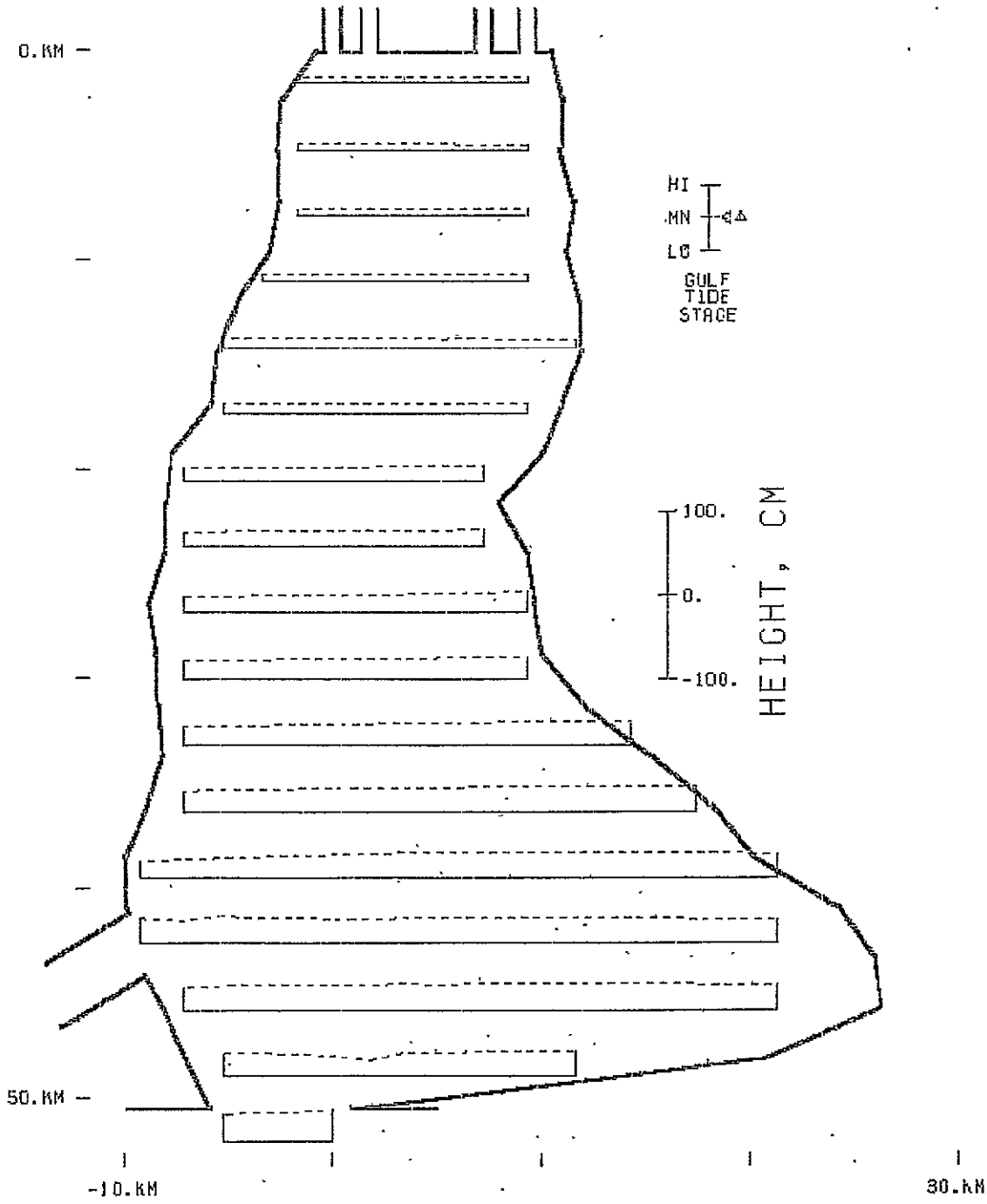


Figure A.165 -

Case 4

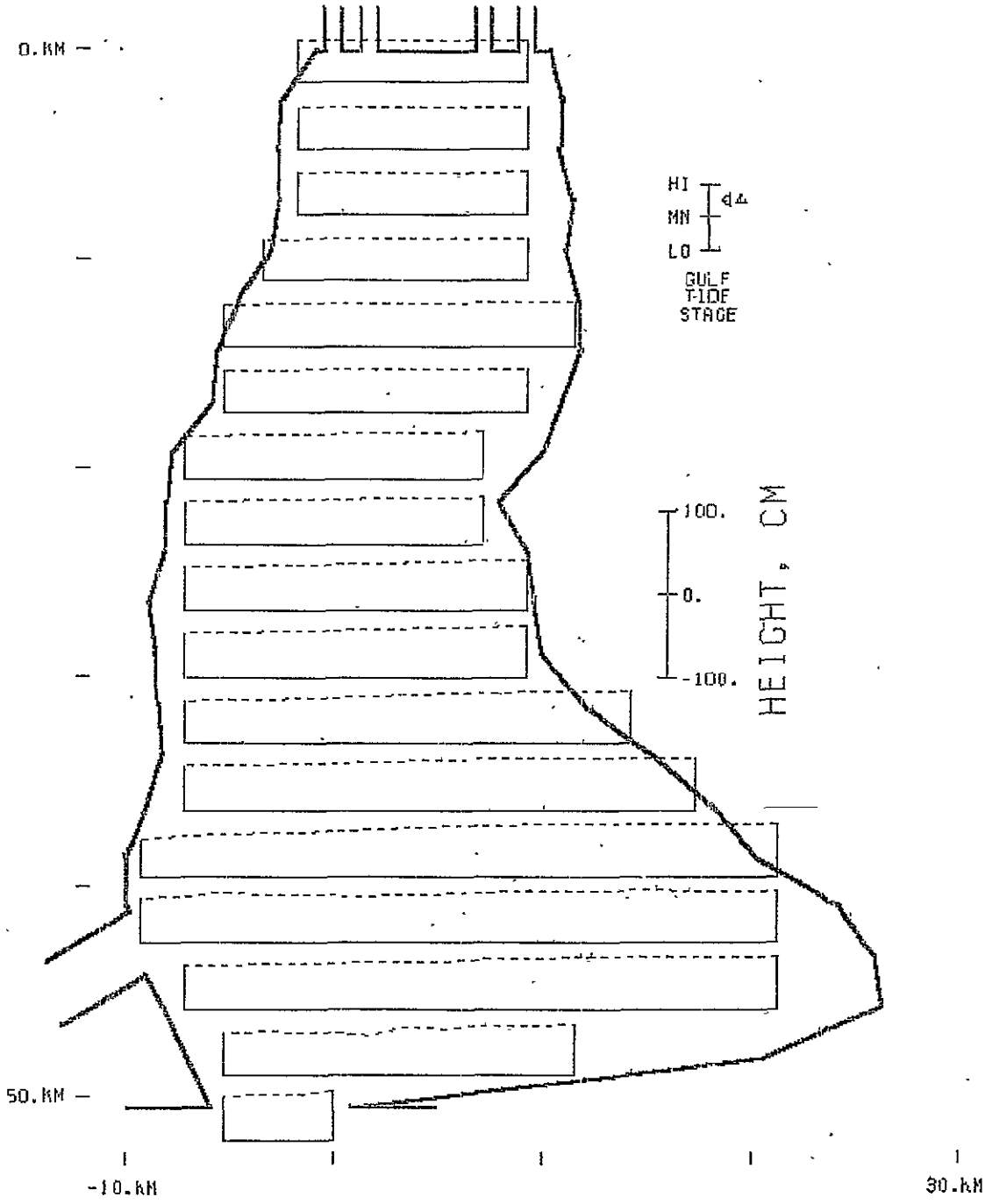


Figure A.166 - Case 4.

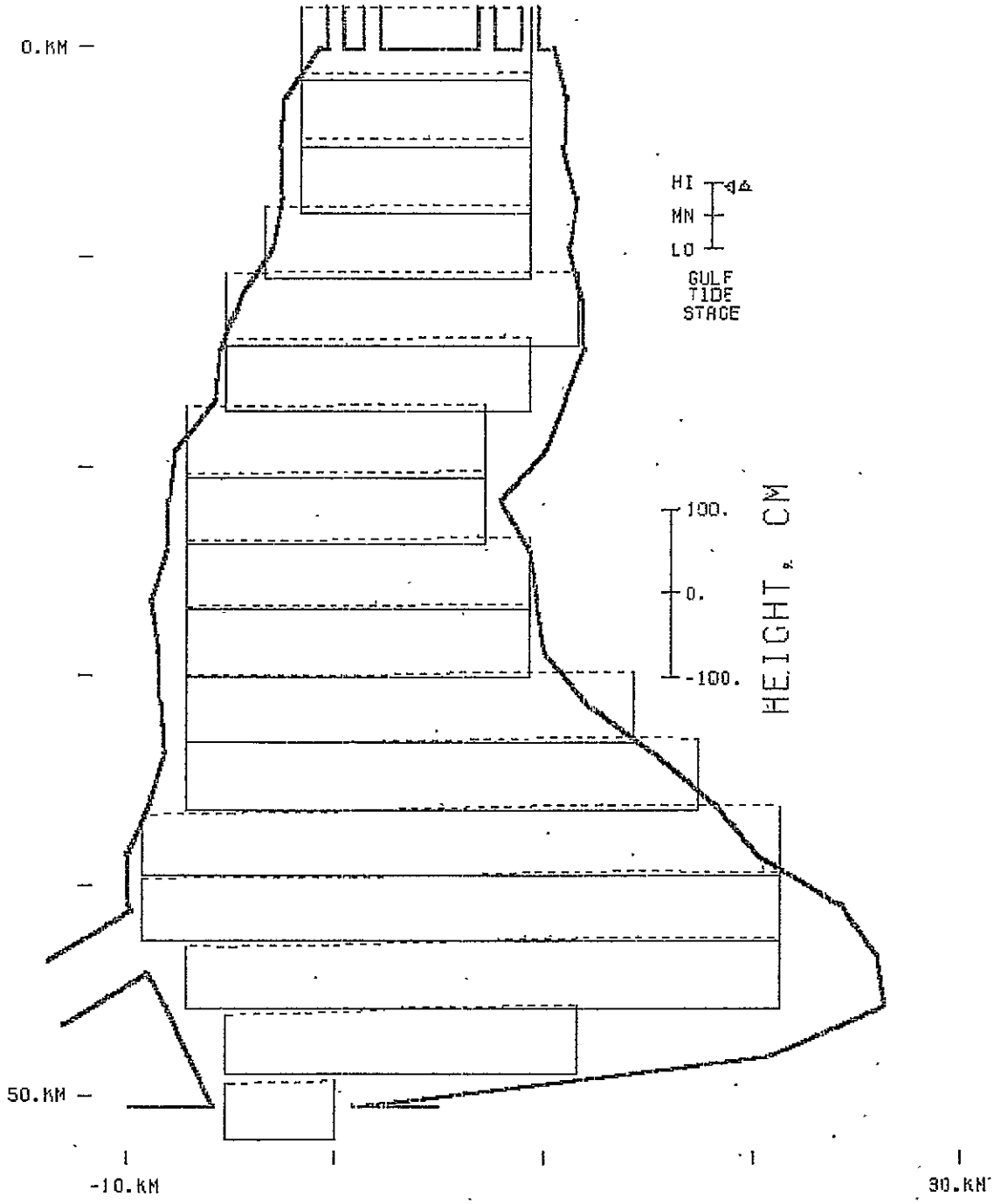


Figure A.167 -

Case 4

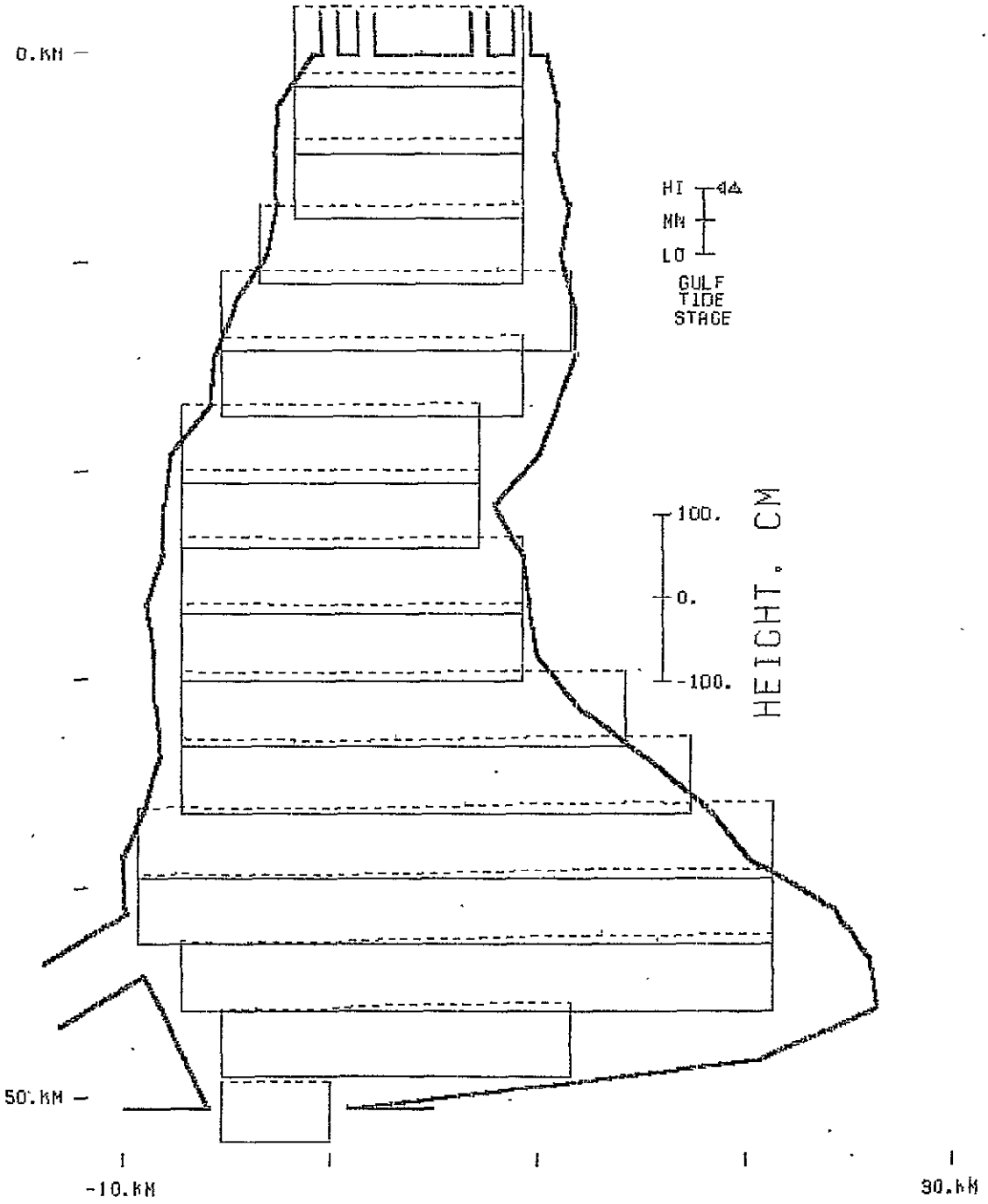


Figure A.168 -

Case 4

APPENDIX B

MOBILE BAY MODEL PROGRAM

This appendix presents a description and listing of the computer program that implements the model of Mobile Bay developed in this research. The description material is devoted to the program organization, input and output.

Discussion of the Program

The Mobile Bay model program is composed of fourteen subroutines driven by a main program. The names of these routines and their primary tasks are listed in Table B.1. The sequence of calculations and the subroutine where the calculations are performed are shown in the flow diagram of Figure B.1. Since the program was written to allow restarting a simulation from data stored on magnetic tape as well as performing several different simulations in succession, the user is referred to the comment statements in the subroutine PRELIM for a detailed explanation of the options. The following information should facilitate interpretation of Figure B.1. At the beginning of a new calculation, whether it be a cold start or a restart, the calling sequence is controlled primarily by PRELIM until the preliminary calculations have been completed and then by MAIN while the time integration is being performed.

At predetermined intervals and at the normal completion of a run, data are written on magnetic tape from RITE, an entry point in REED.

TABLE B.1

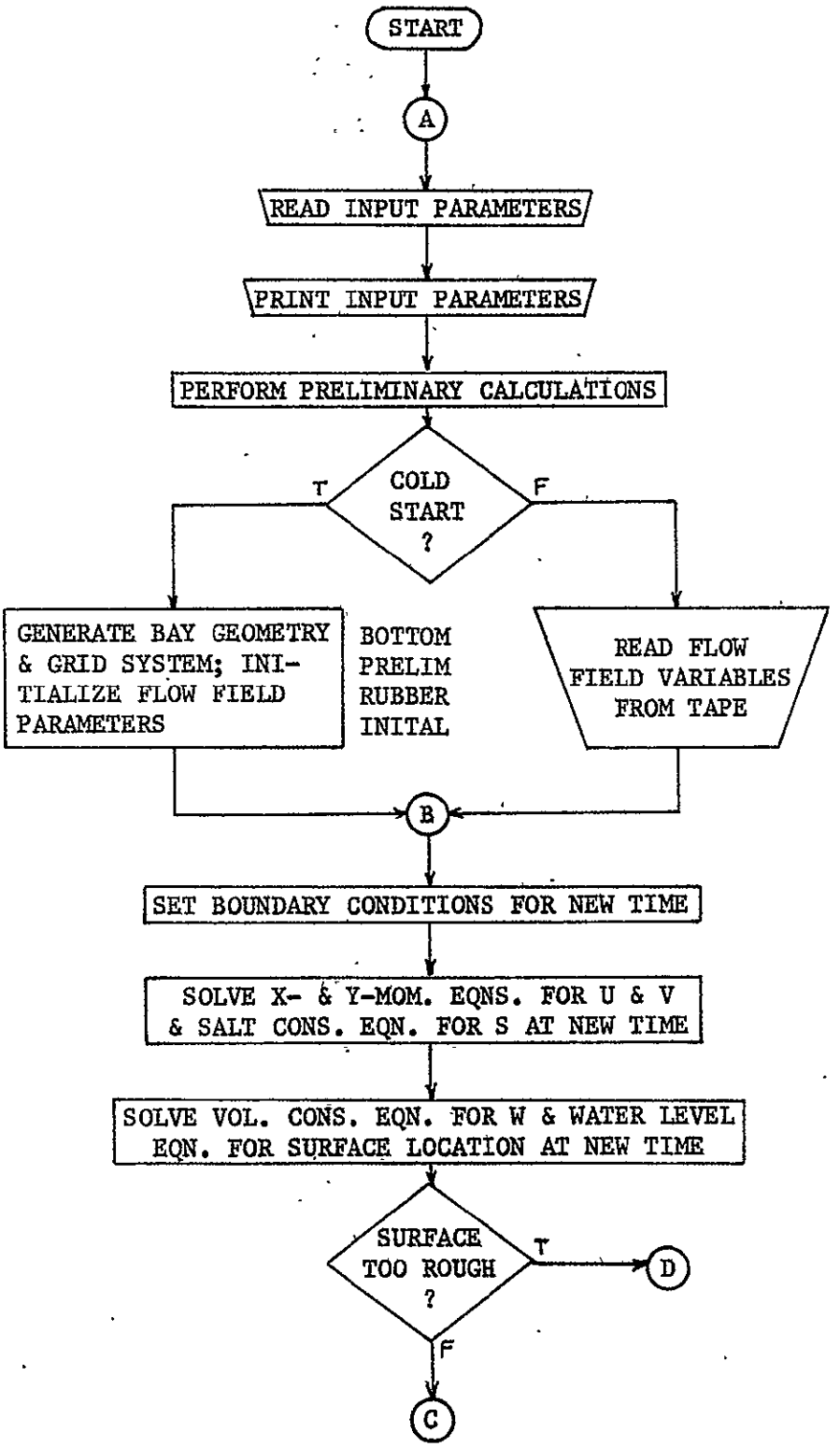
Mobile Bay Model Program Routines

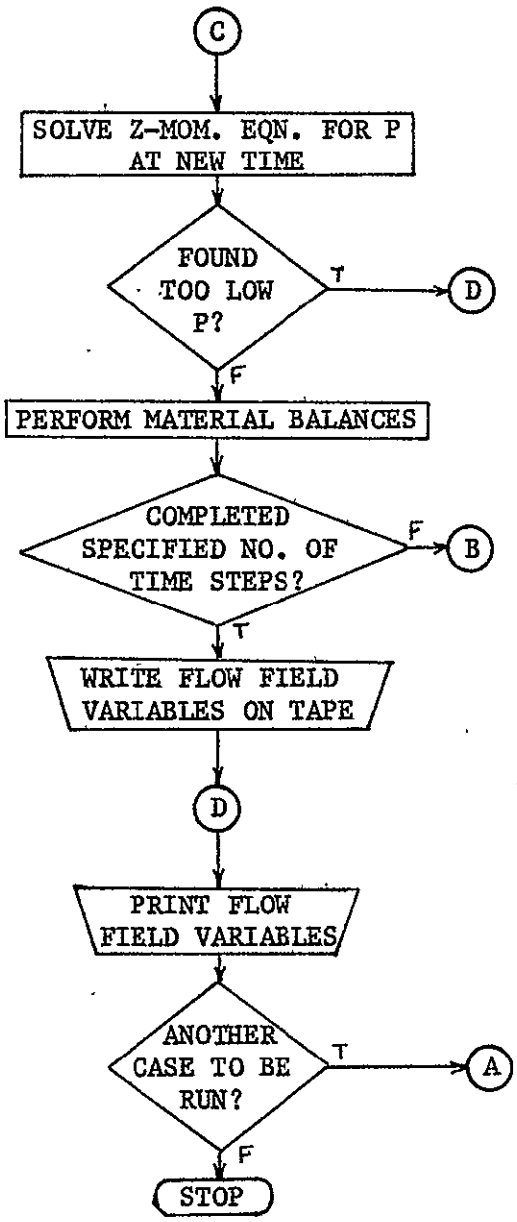
Name	Primary Task
MAIN	Serves as a driver for subroutines.
BALNCE	Performs cumulative and unsteady total and salt water balances at each time level.
BONDRY	Adjusts time dependent boundary conditions at each time level.
BOTTOM	Computes the elevations of the bay and channel bottoms at start of new simulation.
INITAL	Initializes flow field and material balance variables at start of new simulation.
PRELIM	Reads input data from cards, performs or invokes routines to perform preliminary calculations and writes input data to paper.
PRESS	Calculates the pressure field at each time level.
PRNT	Writes flow field variables to paper.
REED	Reads and writes (through entry point RITE) complete problem description on magnetic tape.
RUBBER	Generates bay and channel grid systems at start of new simulation.
SETUP	Retrieves old values of flow field variables from arrays and assigns them to working variables, invokes UVSNEW, and assigns new values to storage locations in arrays.
UVSNEW	Calculates horizontal velocity components and the salinity at a grid point at each time level.
VISCUS	Evaluates the turbulent diffusion terms in the horizontal momentum and species continuity equation.
VOLDIL	Calculates the vertical velocity components at each time level.
BLOCK DATA	Assigns numerical values to constants.

FIGURE B.1

SIMPLIFIED FLOW DIAGRAM FOR

MOBILE BAY MODEL PROGRAM




C
SOLVE Z-MOM. EQN. FOR P AT NEW TIME
PRESS
FOUND TOO LOW P?
T
D
F
PERFORM MATERIAL BALANCES
BALNCE
COMPLETED SPECIFIED NO. OF TIME STEPS?
F
B
MAIN
T
WRITE FLOW FIELD VARIABLES ON TAPE
REED
D
PRINT FLOW FIELD VARIABLES
PRNT
ANOTHER CASE TO BE RUN?
T
A
PRELIM
F
STOP

These data are used by a plot program presented in Appendix C and for restarting if a simulation is to be continued or a change in boundary conditions made.

When the restart capability is utilized, data are read from the magnetic tape by REED. The option of calling this subroutine is invoked by a flag described in PRELIM.

The program was coded to be as general and flexible as possible so that it can be applied to other bays and estuaries with a minimum of program modifications. For instance, the location of the banks and bottom of the bay are specified on input data cards and can be easily modified. At present, one to four rivers with their discharges directed along a x-z grid plane can be located along the upper edge of the bay. Also, a tidal inlet or river with a discharge in the y direction can be positioned anywhere along the right side of the bay when looking down the positive x axis and a tidal inlet can be positioned at will along the lower edge of the bay. While the depth of the two-dimensional channel may vary with x and its y location adjusted, the channel must coincide with a x-z grid plane. With additional programming effort, the channel and rivers can be made to flow at angles other than 0° and 90° to the vertical grid planes.

Program Input

All data cards used by the program are read by PRELIM. Preceding each READ statement are comment cards which define and specify the units for each variable read. These data are nondimensionalized before using them in the simulation. Three basic formats are used for input from data cards 8I10, 9F10.5 and 6E12.0. The first format is used for

integers right adjusted in as field of ten columns, the second for floating point numbers without exponents but with a decimal point punched in a field of ten columns and the third for floating point numbers with right adjusted exponents and decimal points punched in a field of twelve columns. The input variable names and corresponding formats are listed in Table B.2 for easy reference.

Program Output

The bulk of the output from the Mobile Bay model program is to magnetic tape; the rest is printed. The output to tape, which is accomplished with RITE, consists of a complete problem description (i.e. boundary locations, boundary conditions, grid plane locations, flow field variables, etc.) at the beginning and end of a run and at predetermined intervals in between. Because large volumes of data are involved, it is necessary to conserve tape by using unformated WRITE statements. The output to paper, which occurs in MAIN, PRELIM and PRNT, is primarily for the purposes of problem identification, verification of card input data and monitoring the progress of the simulation. Also, all the flow field variables at each grid point are printed at the normal or abnormal end of a run to allow an inspection of the state of the flow field.

Listing of Program

A listing of the FORTRAN IV statements for the Mobile Bay model program and a typical set of input data follow. The large volume of printed output is not presented here.

TABLE B.2

Input Variables for Mobile Bay Model Program

Name	Format	Name	Format	Name	Format
LTAPE	I10	CVIS	F10.0	UVAR2	F10.0
LABELI	I10	ROUGH	F10.0	PASS	F10.0
LABELO	I10	VONKAR	F10.0	PDEPTH	F10.0
IMAX	I10	JWAG	I10	PSALT	F10.0
JMAX	I10	KMXC	I10	PWIDTH	F10.0
KMAX	I10	CDEPTH	I10	IIMAX	I10
NBAR	I10	CWIDTH	I10	SOUTH(I)	F10.0
NMAX	I10	DIRCTN	F10.0	WEST(I)	F10.0
NWRITE	I10	DIRCTS	F10.0	EAST(I)	F10.0
CUTPT	F10.0	FETCHN	F10.0	B1	E12.0
DTREAL	F10.0	FETCHS	F10.0	B2	E12.0
FETCH	F10.0	VWINDN	F10.0	B3	E12.0
HREF	F10.0	VWINDS	F10.0	B4	E12.0
PERIOD	F10.0	ACCEL	F10.0	B5	E12.0
PHI	F10.0	ARTVSC	F10.0	B6	E12.0
SIGMAT	F10.0	YK2	F10.0	B7	E12.0
YMAX	F10.0	NRIV	I10	B8	E12.0
PHSEMP	F10.0	JRIV(I)	I10	B9	E12.0
PHSEPH	F10.0	RDEPTH(I)	I10	B10	E12.0
TAMPMP	F10.0	RWIDTH(I)	I10	BB1	E12.0
TAMPPH	F10.0	PHASE1	F10.0	BB2	E12.0
TAVGMP	F10.0	PHASE2	F10.0	BB3	E12.0
TAVGPH	F10.0	UAVG	F10.0	BB4	E12.0
CDIF	F10.0	UVAR1	F10.0		

```
C
C       DRIVER PROGRAM FOR 3-D, TIME-DEPENDENT HYDRODYNAMIC AND
C       SALINITY MODEL OF MOBILE BAY WITH SHIP CHANNEL.
C
C       OCT. 29, 1975
C
      LOGICAL                 ISTEP,              KEYOUT
      DIMENSION               Q(7,4)
      COMMON/ACCT/            QDOT(7),            QNET(7),
     1    QSDOT(7),          QSNET(7)
      COMMON/BARS/            PTCTIN,             SBAR,
     1    UBAR,              VBAR
      COMMON/FORTNO/          LR,                 LT,
     1    LW
      COMMON/GRID/            DT,                 DTD2,
     1    DX,                DXINV,              DXT2IN,
     2    DXINSQ,            DY,                 DYINV,
     3    DYT2IN,            DYINSQ,             DZ,
     4    DZD2,              DZINV,              DZT2IN,
     5    DZINSQ
      COMMON/INDEX/           I,                  J,
     1    K,                 N
      COMMON/LIMITS/          IMAX,               IMAXM1,
     1    JMAX,              JMAXM1,             KMAX,
     2    KMAXM1,            KMAXM2,             NMAX,
     3    KEYOUT
      COMMON/PULL/            SY(17),             SYY(17),
     1    TIME,              X(18),              Y(17),
     2    YK2,               Z(8)
      COMMON/RYTE/            IO,                 LABELI,
     1    LABELO
      COMMON/STEP/            MN,                 MO,
     1    NBAR,              NWRITE,             ISTEP
      COMMON/UNITS/           BETA,               BETAD2,
     1    FETCH,             GRAV,               HREF,
     2    OMEGA,             PI,                 TREF,
     3    VREF,              YMAX
C
C       READ DATA AND PERFORM PRELIMINARY CALCULATIONS.
C
    5 CALL PRELIM
      TIMEO   = TIME
      CTM     = TREF / 3600.
      CSB     = PTCTIN / TREF
      CUB     = CSB * VREF
      CFR     = HREF * HREF * VREF
      CVL     = HREF * HREF * HREF
```

```
      KEYOUT = .FALSE.
      WRITE(LW,1000) (I, I = 1, 7)
C
CCC      STEP FORWARD IN TIME.
C
   10 N       = N + 1
      TIME    = TIMEO + FLOAT(N) * DT
      MN      = 3 - MN
      MO      = 3 - MO
C
CCC      ADJUST TIME DEPENDENT BOUNDARY CONDITIONS.
C
      CALL BONDRY
C
CCC      ESTIMATE NEW HORIZONTAL VELOCITY COMPONENTS AND SALINITIES.
C
      ISTEP  = .TRUE.
      CALL SETUP
C
CCC      ADJUST VERTICAL VELOCITY COMPONENTS AND SURFACE HEIGHTS.
C
      CALL VOLDIL
      IF(KEYOUT) GO TO 100
C
CCC      COMPUTE NEW PRESSURE FIELD.
C
      CALL PRESS
      IF(KEYOUT) GO TO 100
C
CCC      PERFORM MATERIAL BALANCE AROUND SYSTEM.
C
      CALL BALNCE
C
CCC      DEPENDENT VARIABLES HAVE BEEN COMPUTED FOR THIS TIME STEP.
CCC
CCC      WRITE MONITOR VARIABLES EVERY NBAR TIME STEPS TO TAPE AND
CCC      PRINTER.
C
      IF(N.NE.NBAR*(N/NBAR)) GO TO 30
      IO      = LT
      CALL RITE2
      TIMERL = TIME * CTM
      SBAR1  = SBAR * CSB
      UBAR1  = UBAR * CUB
      VBAR1  = VBAR * CUB
      WRITE(LW,1001) N, TIMERL, SBAR1, UBAR1, VBAR1
      DO 20 L = 1, 7
```

```
      Q(L,1) = QDOT(L) * CFR
      Q(L,2) = QSDOT(L) * CFR
      Q(L,3) = QNET(L) * CVL
   20 Q(L,4) = QSNET(L) * CVL
      WRITE(LW,1002) ((Q(L,M),L=1,7),M=1,4)
C
C        WRITE FLOW FIELD VARIABLES TO TAPE AT SELECTED INTERVALS.
C
   30 IF(N.NE.NWRITE*(N/NWRITE)) GO TO 40
      IO     = LT
      CALL RITE
C
C        TEST FOR END OF RUN.  IF END IS REACHED PRINT RESULTS.
C        OTHERWISE, CONTINUE.
C
   40 IF(N.LT.NMAX) GO TO 10
  100 CALL PRNT
      LABELI = LABELO
      IO     = LT
      CALL RITE
      IF(KEYOUT) STOP
C
C        RESET TIME AND TIME INDEX TO ZERO FOR NEXT CASE.
C
      N      = 0
      GO TO 5
 1000 FORMAT(//,5X,'RATES ARE IN M**3/S, VOLUMES ARE IN M**3',/,5X,
     1 7('    TOTAL/SW'),/,5X,7('    RATE/VOL'),/,1X,7(11X,I1))
 1001 FORMAT(5X,' N = ',I5,', TIME = ',F6.2,' HR., SBAR = ',1PE12.4,
     1', UBAR = ',E12.4,', VBAR = ',E12.4)
 1002 FORMAT(6X,1P7E12.4,/,3(6X,7E12.4,/))
      END
```

```
      SUBROUTINE BALNCE
C
C     PERFORM CUMULATIVE AND UNSTEADY TOTAL AND GULF WATER
C     BALANCES.
C
      LOGICAL                 ISTEP,                  KEYOUT
      DIMENSION               QDOT0(6),               QSDOT0(6)
      DOUBLE PRECISION        SURF(2,18,17),          SURFC(18)
      COMMON/ACCT/            QDOT(7),                QNET(7),
     1    QSDOT(7),           QSNET(7)
      COMMON/BNKCRD/          EAST(22),               IIMAX,
     1    JE,                 JEAST(22),              JW,
     2    JWEST(22),          SOUTH(22),              WEST(22)
      COMMON/CHNNEL/          CDEPTH,                 CWIDTH,
     1    JWAG,               KMXC,                   KMXCM1,
     2    KMXCM2
      COMMON/FLOOR/           KFLOOR(18,17),          ZB(18,17)
      COMMON/FLOORC/          KFLORC(18),             ZBC(18)
      COMMON/FLOW1/           P(18,17,08),            S(2,18,17,08),
     1    U(2,18,17,08),      V(2,18,17,08),          W(18,17,08)
      COMMON/FLOW2/           SURF,                   DHDT(18,17),
     1    DUDT(18,17,08),     DVDT(18,17,08)
      COMMON/FLOWC1/          PC(18,20),              SC(2,18,20),
     1    UC(2,18,20),        WC(18,20)
      COMMON/FLOWC2/          SURFC,                  DHCDT(18),
     1    DUCDT(18,20)
      COMMON/GRID/            DT,                     DTD2,
     1    DX,                 DXINV,                  DXT2IN,
     2    DXINSQ,             DY,                     DYINV,
     3    DYT2IN,             DYINSQ,                 DZ,
     4    DZD2,               DZINV,                  DZT2IN,
     5    DZINSQ
      COMMON/LIMITS/          IMAX,                   IMAXM1,
     1    JMAX,               JMAXM1,                 KMAX,
     2    KMAXM1,             KMAXM2,                 NMAX,
     3    KEYOUT
      COMMON/PASS/            IPASS,                  PDEPTH,
     1    PMOMAF,             PSALT,                  PWIDTH
      COMMON/PULL/            SY(17),                 SYY(17),
     1    TIME,               X(18),                  Y(17),
     2    YK2,                Z(8)
      COMMON/PULLC/           ZC(20)
      COMMON/RIVERS/          JRIV(4),                NRIV,
     1    PHASE1,             PHASE2,                 RDEPTH(4),
     2    RMOMAF(4),          RWIDTH(4),              UAVG,
     3    UVAR1,              UVAR2
      COMMON/STEP/            MN,                     MO,
```

```
     1   NBAR,                    NWRITE,                  ISTEP
      COMMON/TIDE/                PHSEMP,                  PHSEPH,
     1   TAMPMP,                  TAMPPH,                  TAVGMP,
     2   TAVGPH,                  THGTMP,                  THGTPH
      DO 5 NN = 1, 6
      QDOT0(NN)   = QDOT(NN)
    5 QSDOT0(NN) = QSDOT(NN)
C
CC        EVALUATE FLUXES THROUGH RIVER MOUTHS.
C
      DO 20 NN = 1, NRIV
      J       = JRIV(NN)
      KBOT    = KFLOOR(1,J)
      KBOTP1  = KBOT + 1
      UKBOT   = U(MN,1,J,KBOT)
      UKMXM1  = U(MN,1,J,KMAXM1)
      USKBOT  = UKBOT * S(MN,1,J,KBOT)
      SUMA    = 0.5 * (UKBOT + UKMXM1)
      SUMB    = 0.5 * (USKBOT + UKMXM1 * S(MN,1,J,KMAXM1))
      DO 10 K = KBOTP1, KMAXM2
      UK      = U(MN,1,J,K)
      SUMA    = SUMA + UK
   10 SUMB    = SUMB + UK * S(MN,1,J,K)
      DZKBOT  = Z(KBOT) - ZB(1,J)
      DZKMAX  = SURF(MN,1,J) + DZ
      UKBOT   = 0.5 * UKBOT
      USKBOT  = 0.5 * USKBOT
      UKMAX   = U(MN,1,J,KMAX)
      UK      = 0.5 * (UKMAX + UKMXM1)
      USK     = 0.5 * (UKMAX * S(MN,1,J,KMAX) +
     1          UKMXM1 * S(MN,1,J,KMAXM1))
      PREFIX  = DY / SY(J)
      IF(J.EQ.JWAG) PREFIX = PREFIX - CWIDTH
      QDOT(NN)  = (DZ * SUMA + DZKMAX * UK + DZKBOT * UKBOT) * PREFIX
   20 QSDOT(NN) = (DZ * SUMB + DZKMAX * USK + DZKBOT * USKBOT) * PREFIX
C
CC        EVALUATE FLUX THROUGH PAS AUX HERONS.
C
      I       = IPASS
      J       = JWEST(I) - 1
      JP1     = J + 1
      KBOT    = KFLOOR(I,JP1)
      KBOTP1  = KBOT + 1
      UKBOT   = V(MN,I,J,KBOT)
      UKMXM1  = V(MN,I,J,KMAXM1)
      USKBOT  = UKBOT * S(MN,I,J,KBOT)
      SUMA    = 0.5 * (UKBOT + UKMXM1)
```

```
      SUMB    = 0.5 * (USKBOT + UKMXM1  * S(MN,I,J,KMAXM1))
      IF(KBOTP1.GT.KMAXM2) GO TO 35
      DO 30 K = KBOTP1, KMAXM2
      UK      = V(MN,I,J,K)
      SUMA    = SUMA + UK
   30 SUMB    = SUMB + UK * S(MN,I,J,K)
   35 DZKBOT = Z(KBOT) - ZB(I,JP1)
      DZKMAX = THGTPH + DZ
      UKBOT   = 0.5 * UKBOT
      USKBOT = 0.5 * USKBOT
      UKMAX   = V(MN,I,J,KMAX)
      UK      = 0.5 * (UKMAX + UKMXM1)
      USK     = 0.5 * (UKMAX * S(MN,I,J,KMAX) +
     1          UKMXM1 * S(MN,I,J,KMAXM1))
      QDOT(5)  = (DZ * SUMA + DZKMAX * UK + DZKBOT * UKBOT) * DX
      QSDOT(5) = (DZ * SUMB + DZKMAX * USK + DZKBOT * USKBOT) * DX
C
CC       EVALUATE FLUX THROUGH THE MAIN PASS.
C
      JW      = JWEST(IMAX)
      JE      = JEAST(IMAX)
      SURF1   = SURF(MN,IMAX,JW)
      SURFJP  = SURF1
      KBOT    = KMAX + 1
      KBOTJP  = KFLOOR(IMAX,JW)
      QDOT(6)  = 0.0
      QSDOT(6) = 0.0
      DO 210 J = JW, JE
      JP1     = J + 1
      JM1     = J - 1
      PREFIX = DY / SY(J)
      IF(J.EQ.JWAG) PREFIX = PREFIX - CWIDTH
      SURFJM = SURF1
      SURF1   = SURFJP
      SURFJP = SURF(MN,IMAX,JP1)
      IF(J.EQ.JE) SURFJP = SURF1
      KBOTJM = KBOT
      KBOT    = KBOTJP
      KBOTP1 = KBOT + 1
      KBOTJP = KFLOOR(IMAX,JP1)
      IF(KBOT.GT.KMAX) GO TO 210
      UJK     = U(MN,IMAX,J,KBOT)
      USJK    = UJK * S(MN,IMAX,J,KBOT)
      IF(KBOT.GT.KBOTJM) GO TO 40
      UJMK    = UJK
      USJMK   = USJK
      GO TO 50
```

```
 40 UJMK    = U(MN,IMAX,JM1,KBOT)
    USJMK   = UJMK * S(MN,IMAX,JM1,KBOT)
 50 IF(KBOT.GT.KBOTJP) GO TO 60
    UJPK    = UJK
    USJPK   = USJK
    GO TO 70
 60 UJPK    = U(MN,IMAX,JP1,KBOT)
    USJPK   = UJPK * S(MN,IMAX,JP1,KBOT)
 70 UKBOT   = 0.25 * (UJMK + UJK + UJK + UJPK)
    USKBOT  = 0.25 * (USJMK + USJK + USJK + USJPK)
    UJK     = U(MN,IMAX,J,KMAXM1)
    USJK    = UJK * S(MN,IMAX,J,KMAXM1)
    IF(KMAXM1.GT.KBOTJM) GO TO 80
    UJMK    = UJK
    USJMK   = USJK
    GO TO 90
 80 UJMK    = U(MN,IMAX,JM1,KMAXM1)
    USJMK   = UJMK * S(MN,IMAX,JM1,KMAXM1)
 90 IF(KMAXM1.GT.KBOTJP) GO TO 100
    UJPK    = UJK
    USJPK   = USJK
    GO TO 110
100 UJPK    = U(MN,IMAX,JP1,KMAXM1)
    USJPK   = UJPK * S(MN,IMAX,JP1,KMAXM1)
110 UKXM1   = 0.25 * (UJMK + UJK + UJK + UJPK)
    USKXM1  = 0.25 * (USJMK + USJK + USJK + USJPK)
    SUMA    = 0.5 * (UKBOT + UKXM1)
    SUMB    = 0.5 * (USKBOT + USKXM1)
    DO 160 K = KBOTP1, KMAXM2
    UJK     = U(MN,IMAX,J,K)
    USJK    = UJK * S(MN,IMAX,J,K)
    IF(K.GT.KBOTJM) GO TO 120
    UJMK    = UJK
    USJMK   = USJK
    GO TO 130
120 UJMK    = U(MN,IMAX,JM1,K)
    USJMK   = UJMK * S(MN,IMAX,JM1,K)
130 IF(K.GT.KBOTJP) GO TO 140
    UJPK    = UJK
    USJPK   = USJK
    GO TO 150
140 UJPK    = U(MN,IMAX,JP1,K)
    USJPK   = UJPK * S(MN,IMAX,JP1,K)
150 SUMA    = SUMA + 0.25 * (UJMK + UJK + UJK + UJPK)
160 SUMB    = SUMB + 0.25 * (USJMK + USJK + USJK + USJPK)
    DZKBOT  = Z(KBOT) - Z8(IMAX,J)
    DZKMAX  = 0.25 * (SURFJM + SURF1 + SURF1 + SURFJP) + DZ
```

```
      UKBOT   = 0.5 * UKBOT
      USKBOT = 0.5 * USKBOT
      UJK     = U(MN,IMAX,J,KMAX)
      USJK    = UJK * S(MN,IMAX,J,KMAX)
      IF(KMAX.GT.KBOTJM) GO TO 170
      UJMK    = UJK
      USJMK   = USJK
      GO TO 180
  170 UJMK    = U(MN,IMAX,JM1,KMAX)
      USJMK   = UJMK * S(MN,IMAX,JM1,KMAX)
  180 IF(KMAX.GT.KBOTJP) GO TO 190
      UJPK    = UJK
      USJPK   = USJK
      GO TO 200
  190 UJPK    = U(MN,IMAX,JP1,KMAX)
      USJPK   = UJPK * S(MN,IMAX,JP1,KMAX)
  200 UKMAX   = 0.25 * (UJMK + UJK + UJK + UJPK)
      USKMAX = 0.25 * (USJMK + USJK + USJK + USJPK)
      UJK     = 0.5 * (UKXM1 + UKMAX)
      USJK    = 0.5 * (USKXM1 + USKMAX)
      QDOT(6)   = (DZ * SUMA + DZKMAX * UJK + DZKBOT * UKBOT) *
     1            PREFIX + QDOT(6)
      QSDOT(6)  = (DZ * SUMB + DZKMAX * USJK + DZKBOT * USKBOT) *
     1            PREFIX + QSDOT(6)
  210 CONTINUE
C
C        EVALUATE FLUXES THROUGH NORTH AND SOUTH ENDS OF THE CHANNEL
C        AND ADD TO MOBILE RIVER AND MAIN PASS FLUXES, RESPECTIVELY.
C
      I       = 1
  220 KBOT    = KFLORC(I)
      KBOTP1 = KBOT + 1
      UKBOT   = UC(MN,I,KBOT)
      UKMXM1 = UC(MN,I,KMXCM1)
      USKBOT = UKBOT * SC(MN,I,KBOT)
      SUMA    = 0.5 * (UKBOT + UKMXM1)
      SUMB    = 0.5 * (USKBOT + UKMXM1 * SC(MN,I,KMXCM1))
      DO 230 K = KBOTP1, KMXCM2
      UK      = UC(MN,I,K)
      SUMA    = SUMA + UK
  230 SUMB    = SUMB + UK * SC(MN,I,K)
      DZKBOT = ZC(KBOT) - ZBC(I)
      DZKMAX = SURFC(I) + DZ
      UKBOT   = 0.5 * UKBOT
      USKBOT = 0.5 * USKBOT
      UKMAX   = UC(MN,I,KMXC)
      UK      = 0.5 * (UKMAX + UKMXM1)
```

```
      USK     = 0.5 * (UKMAX * SC(MN,I,KMXC) +
     1          UKMXM1 * SC(MN,I,KMXCM1))
      IF(I.NE.1) I = 6
      QDOT(I)  = (DZ * SUMA + DZKMAX * UK + DZKBOT * UKBOT) * CWIDTH +
     1           QDOT(I)
      QSDOT(I) = (DZ * SUMB + DZKMAX * USK +
     1           DZKBOT * USKBOT) * CWIDTH + QSDOT(I)
      IF(I.NE.1) GO TO 240
      I       = IMAX
      GO TO 220
  240 CONTINUE
C
CC        DETERMINE TOTAL AND GULF WATER VOLUME IN BAY.
C
      SUMA    = 0.0
      SUMB    = 0.0
      DO 260 I = 2, IMAX
      IP1     = I + 1
      IF(I.EQ.IMAX) IP1 = IMAX
      IM1     = I - 1
      JW      = JWEST(I)
      JE      = JEAST(I)
      SURF1   = SURF(MN,I,JW)
      SURFJP  = SURF1
      IF(I.EQ.IPASS) SURF1 = THGTPH
      KBOT    = KMAX + 1
      KBOTJP  = KFLOOR(I,JW)
      DO 260 J = JW, JE
      JP1     = J + 1
      JM1     = J - 1
      PREFIX = DY / SY(J)
      IF(J.EQ.JWAG) PREFIX = PREFIX - CWIDTH
      SURFJM = SURF1
      SURF1   = SURFJP
      SURFJP  = SURF(MN,I,JP1)
      IF(J.EQ.JE) SURFJP = SURF1
      KBOTJM = KBOT
      KBOT    = KBOTJP
      KBOTP1  = KBOT + 1
      KBOTJP  = KFLOOR(I,JP1)
      SURFIM  = SURF(MN,IM1,J)
      IF(KFLOOR(IM1,J).GT.KMAX) SURFIM = SURF1
      SURFIP  = SURF(MN,IP1,J)
      IF(KFLOOR(IP1,J).GT.KMAX) SURFIP = SURF1
      IF(I.EQ.IMAX) SURFIP = THGTMP
      SUMC    = 0.0
      DO 250 K = KBOTP1, KMAXM1
```

```
  250 SUMC   = SUMC + S(MN,I,J,K)
      SURFAV = 0.125 * (4.0 * SURF1 + SURFJP + SURFJM + SURFIM +
     1         SURFIP)
      DZKMAX = SURFAV + DZD2
      ZBOT   = ZB(I,J)
      DZKBOT = DZD2 + Z(KBOT) - ZBOT
      SUMB   = (SUMC * DZ + DZKMAX * S(MN,I,J,KMAX) +
     1         DZKBOT * S(MN,I,J,KBOT)) * PREFIX + SUMB
  260 SUMA   = (SURFAV + 1.0 - ZBOT) * PREFIX + SUMA
      SURF1  = SURFC(1)
      SURFIP = SURFC(2)
      DO 280 I = 2, IMAX
      IP1    = I + 1
      IF(I.EQ.IMAX) IP1 = IMAX
      IM1    = I - 1
      KBOT   = KFLORC(I)
      KBOTP1 = KBOT + 1
      SURFIM = SURF1
      SURF1  = SURFIP
      SURFIP = SURFC(IP1)
      IF(I.EQ.IMAX) SURFIP = THGTMP
      SUMC   = 0.0
      DO 270 K = KBOTP1, KMXCM1
  270 SUMC   = SUMC + SC(MN,I,K)
      SURFAV = 0.25 * (SURFIM + SURF1 + SURF1 + SURFIP)
      DZKMAX = SURFAV + DZD2
      ZBOT   = ZBC(I)
      DZKBOT = DZD2 + ZC(KBOT) - ZBOT
      SUMB   = (SUMC * DZ + DZKMAX * SC(MN,I,KMAX) +
     1         DZKBOT * SC(MN,I,KBCT)) * CWIDTH + SUMB
  280 SUMA   = (SURFAV + 1.0 - ZBCT) * CWIDTH + SUMA
      SUMA   = SUMA * DX
      SUMB   = SUMB * DX
      QDOT(7)  = (SUMA - QNET(7)) / DT
      QSDOT(7) = (SUMB - QSNET(7)) / DT
C
CC        CALCULATE CUMULATIVE FLOWS.
C
      QNET(7)  = SUMA
      QSNET(7) = SUMB
      DO 290 NN = 1, 6
      QNET(NN)  = QNET(NN) + (QDOTO(NN) + QDOT(NN)) * DTD2
  290 QSNET(NN) = QSNET(NN) + (QSDOTO(NN) + QSDOT(NN)) * DTD2
      RETURN
      END
```

```
      SUBROUTINE BONDRY
C
C        ADJUST TIME DEPENDENT BOUNDARY CONDITIONS.
C
      LOGICAL                 ISTEP,                  KEYOUT
      DOUBLE PRECISION        SURF(2,18,17),          SURFC(18)
      COMMON/ACCT/            QDOT(7),                QNET(7),
     1    QSDOT(7),          QSNET(7)
      COMMON/BNKCRD/          EAST(22),               IIMAX,
     1    JE,                JEAST(22),              JW,
     2    JWEST(22),         SOUTH(22),              WEST(22)
      COMMON/CHNNEL/          CDEPTH,                 CWIDTH,
     1    JWAG,              KMXC,                   KMXCM1,
     2    KMXCM2
      COMMON/FLOOR/           KFLOOR(18,17),          ZB(18,17)
      COMMON/FLOORC/          KFLORC(18),             ZBC(18)
      COMMON/FLOW1/           P(18,17,08),            S(2,18,17,08),
     1    U(2,18,17,08),     V(2,18,17,08),          W(18,17,08)
      COMMON/FLOW2/           SURF,                   DHDT(18,17),
     1    DUDT(18,17,08),    DVDT(18,17,08)
      COMMON/FLOWC1/          PC(18,20),              SC(2,18,20),
     1    UC(2,18,20),       WC(18,20)
      COMMON/FLOWC2/          SURFC,                  DHCDT(18),
     1    DUCDT(18,20)
      COMMON/FORCES/          F,                      FWINDX(18,17),
     1    FWINDY(18,17),     TOPLYR
      COMMON/GRID/            DT,                     DTD2,
     1    DX,                DXINV,                  DXT2IN,
     2    DXINSQ,            DY,                     DYINV,
     3    DYT2IN,            DYINSQ,                 DZ,
     4    DZD2,              DZINV,                  DZT2IN,
     5    DZINSQ
      COMMON/GULF/            CUTPT,                  KCUT,
     1    PERIDI
      COMMON/LIMITS/          IMAX,                   IMAXM1,
     1    JMAX,              JMAXM1,                 KMAX,
     2    KMAXM1,            KMAXM2,                 NMAX,
     3    KEYOUT
      COMMON/PASS/            IPASS,                  PDEPTH,
     1    PMOMAF,            PSALI,                  PWIDTH
      COMMON/PULL/            SY(17),                 SYY(17),
     1    TIME,              X(18),                  Y(17),
     2    YK2,               Z(8)
      COMMON/PULLC/           ZC(20)
      COMMON/RIVERS/          JRIV(4),                NRIV,
     1    PHASE1,            PHASE2,                 RDEPTH(4),
     2    RMOMAF(4),         RWIDTH(4),              UAVG,
```

```
     3    UVAR1,                    UVAR2
      COMMON/STEP/                  MN,                       MO,
     1    NBAR,                     NWRITE,                   ISTEP
      COMMON/TIDE/                  PHSEMP,                   PHSEPH,
     1    TAMPMP,                   TAMPPH,                   TAVGMP,
     2    TAVGPH,                   THGTMP,                   THGTPH
      COMMON/UNITS/                 BETA,                     BETAD2,
     1    FETCH,                    GRAV,                     HREF,
     2    OMEGA,                    PI,                       TREF,
     3    VREF,                     YMAX
      TNM1    = TIME - DT
C
C         SET VELOCITIES IN RIVER MOUTHS AND NORTH END OF CHANNEL.
C
      URIV    = 1.142857 * (UAVG + UVAR1 * COS((TNM1 - PHASE1) * PERIDI)
     1          + UVAR2 * COS((TNM1 - PHASE2) * 2.0 * PERIDI))
      DO 30 NN = 1, NRIV
      URIV1   = URIV / RMOMAF(NN)
      J       = JRIV(NN)
      ZBOT    = ZB(1,J)
      KBOT    = KFLOOR(1,J)
      PREFIX = 1.0 / (SURF(MO,1,J) + 1.0 - ZBOT)
      DO 10 K = KBOT, KMAX
      VERT    = (Z(K) - ZBOT) * PREFIX
   10 U(MO,1,J,K) = URIV1 * (1.0 - (1.0 - VERT)**7)
      IF(J.NE.JWAG) GO TO 30
      ZBOT    = ZBC(1)
      KBOT    = KFLORC(1)
      URIV1   = URIV1 * RMOMAF(1)
      PREFIX = 1.0 / (SURFC(1) + 1.0 - ZBOT)
      DO 20 K = KBOT, KMXC
      VERT    = (ZC(K) - ZBOT) * PREFIX
   20 UC(MO,1,K) = URIV1 * (1.0 - (1.0 - VERT)**7)
   30 CONTINUE
C
C         SET FREE SURFACE LOCATION IN PASS AUX HERONS AND MAIN PASS.
C
      THGTMP = TAVGMP + TAMPMP * COS((TNM1 - PHSEMP) * PERIDI)
      JW      = JWEST(IPASS)
      THGTPH = SURF(MO,IPASS,JW)
C
C         SET SALINITY BOUNDARY CONDITION IN THE MAIN PASS.
C
      IF(KCUT.LT.1) GO TO 60
      JW      = JWEST(IMAX)
      JE      = JEAST(IMAX)
      DO 50 J = JW, JE
```

```
      KBOT    = KFLOOR(IMAX,J)
      IF(KCUT.LT.KBOT) GO TO 50
      DO 40 K = KBOT, KCUT
   40 S(MO,IMAX,J,K) = 1.0
   50 CONTINUE
   60 KCUTC   = KCUT + KMXC - KMAX
      IF(KCUTC.LT.0) GO TO 80
      KBOT    = KFLORC(IMAX)
      DO 70 K = KBOT, KCUTC
   70 SC(MO,IMAX,K) = 1.0
   80 CONTINUE
C
CC         SET VELOCITIES IN PASS AUX HERONS.
C
      JW      = JWEST(IPASS)
      JWM1    = JW - 1
      ZBOT    = ZB(IPASS,JW)
      KBOT    = KFLOOR(IPASS,JW)
      PREFIX = 1.0 / (THGTPH + 1.0 - ZBOT)
      VPASS   = - 0.228571 * QDOT(6) * PREFIX / PWIDTH
      DO 90 K = KBOT, KMAX
      VERT    = (Z(K) - ZBOT) * PREFIX
      S(MO,IPASS,JWM1,K) = PSALT
   90 V(MO,IPASS,JWM1,K) = VPASS * (1.0 - (1.0 - VERT)**7)
      RETURN
      END
```

```
      SUBROUTINE BOTTOM(C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,CC1,CC2,CC3,CC4)
C
C        CALCULATE BAY AND CHANNEL BOTTOM LOCATION RELATIVE TO A DATUM
C        PLANE LOCATED HREF METERS BELOW MEAN SEA LEVEL.  A POLYNOMIAL
C        IN X AND Y THAT HAS BEEN FITTED TO SOUNDING DATA TAKEN FROM
C        NOAA TOPOGRAPHICAL MAP C&GS 1266 IS USED.
C
       LOGICAL                  KEYOUT
       COMMON/BNKCRD/           EAST(22),              IIMAX,
      1   JE,                   JEAST(22),             JW,
      2   JWEST(22),            SOUTH(22),             WEST(22)
       COMMON/CHNNEL/           CDEPTH,                CWIDTH,
      1   JWAG,                 KMXC,                  KMXCM1,
      2   KMXCM2
       COMMON/FLOOR/            KFLOOR(18,17),         ZB(18,17)
       COMMON/FLOORC/           KFLORC(18),            ZBC(18)
       COMMON/GRID/             DT,                    DTD2,
      1   DX,                   DXINV,                 DXT2IN,
      2   DXINSQ,               DY,                    DYINV,
      3   DYT2IN,               DYINSQ,                DZ,
      4   DZD2,                 DZINV,                 DZT2IN,
      5   DZINSQ
       COMMON/LIMITS/           IMAX,                  IMAXM1,
      1   JMAX,                 JMAXM1,                KMAX,
      2   KMAXM1,               KMAXM2,                NMAX,
      3   KEYOUT
       COMMON/PASS/             IPASS,                 PDEPTH,
      1   PMOMAF,               PSALT,                 PWIDTH
       COMMON/PULL/             SY(17),                SYY(17),
      1   TIME,                 X(18),                 Y(17),
      2   YK2,                  Z(8)
       COMMON/PULLC/            ZC(20)
       COMMON/RIVERS/           JRIV(4),               NRIV,
      1   PHASE1,               PHASE2,                RDEPTH(4),
      2   RMOMAF(4),            RWIDTH(4),             UAVG,
      3   UVAR1,                UVAR2
       COMMON/UNITS/            BETA,                  BETAD2,
      1   FETCH,                GRAV,                  HREF,
      2   OMEGA,                PI,                    TREF,
      3   VREF,                 YMAX
C
       KMAXP1 = KMAX + 1
       KMXCP1 = KMXC + 1
       DO 10 I = 1, IMAX
       KFLORC(I) = KMXCP1
       ZBC(I)    = 1.01
       DO 10 J = 1, JMAX
```

```
      KFLOOR(I,J) = KMAXP1
   10 ZB(I,J)     = 1.01
      DO 20 I = 2, IMAXM1
      JW      = JWEST(I)
      JE      = JEAST(I)
      X1      = X(I)
      CONST1 = C1 + X1 * (C2 + X1 * (C4 + X1 * C7))
      CONST2 = C3 + X1 * (C5 + X1 * C8)
      CONST3 = C6 + X1 * C9
      DO 20 J = JW, JE
      Y1      = Y(J)
   20 ZB(I,J) = CONST1 + Y1 * (CONST2 + Y1 * (CONST3 + Y1 * C10))
      DO 40 I = 2, IMAXM1
      JW      = JWEST(I)
      JE      = JEAST(I)
      DO 40 J = JW, JE
      ZB1     = ZB(I,J)
      IF(ZB1.LT.1.0) GO TO 35
      K       = KMAXM2
      ZB(I,J) = Z(K) - 0.1 * DZ
      GO TO 40
   35 K       = 0
   30 K       = K + 1
      IF(Z(K).LT.ZB1) GO TO 30
      IF(K.LE.KMAXM2) GO TO 40
      K       = KMAXM2
      ZB(I,J) = Z(K) - 0.1 * DZ
   40 KFLOOR(I,J) = K
      DO 50 NN = 1, NRIV
      J       = JRIV(NN)
      ZB(1,J) = ZB(2,J)
   50 KFLOOR(1,J) = KFLOOR(2,J)
      JW      = JWEST(IMAX)
      JE      = JEAST(IMAX)
      DO 60 J = JW, JE
      ZB(IMAX,J) = ZB(IMAXM1,J)
   60 KFLOOR(IMAX,J) = KFLOOR(IMAXM1,J)
      JW      = JWEST(IPASS)
      ZB1     = 1.0 - PDEPTH
      ZB(IPASS,JW) = ZB1
      K       = 0
   70 K       = K + 1
      IF(Z(K).LT.ZB1) GO TO 70
      KFLOOR(IPASS,JW) = K
      DO 80 I = 1, IMAX
      X1      = X(I)
   80 ZBC(I) = CC1 + X1 * (CC2 + X1 * (CC3 + X1 * CC4))
```

```
      DO 100 I = 1, IMAX
      ZB1      = ZBC(I)
      K        = 0
   90 K        = K + 1
      IF(ZC(K).LT.ZB1) GO TO 90
      IF(K.LE.KMXCM2) GO TO 100
      K        = KMXCM2
      ZBC(I) = ZC(K) - 0.1 * DZ
  100 KFLORC(I) = K
      RETURN
      END
```

```
      SUBROUTINE INITAL
C
CCC      INITIALIZE FLOW FIELD AND MATERIAL BALANCE VARIABLES AND
CCC      DETERMINE INITIAL TOTAL AND GULF WATER CONTENT.
C
      LOGICAL                 ISTEP,                  KEYOUT
      DOUBLE PRECISION        SURF(2,18,17),          SURFC(18)
      COMMON/ACCT/            QDOT(7),                QNET(7),
     1    QSDOT(7),          QSNET(7)
      COMMON/BARS/            PTCTIN,                 SBAR,
     1    UBAR,              VBAR
      COMMON/BNKCRD/          EAST(22),               IIMAX,
     1    JE,                JEAST(22),              JW,
     2    JWEST(22),         SOUTH(22),              WEST(22)
      COMMON/CHNNEL/          CDEPTH,                 CWIDTH,
     1    JWAG,              KMXC,                   KMXCM1,
     2    KMXCM2
      COMMON/FLOOR/           KFLOOR(18,17),          ZB(18,17)
      COMMON/FLOORC/          KFLORC(18),             ZBC(18)
      COMMON/FLOW1/           P(18,17,08),            S(2,18,17,08),
     1    U(2,18,17,08),     V(2,18,17,08),          W(18,17,08)
      COMMON/FLOW2/           SURF,                   DHDT(18,17),
     1    DUDT(18,17,08),    DVDT(18,17,08)
      COMMON/FLOWC1/          PC(18,20),              SC(2,18,20),
     1    UC(2,18,20),       WC(18,20)
      COMMON/FLOWC2/          SURFC,                  DHCDT(18),
     1    DUCDT(18,20)
      COMMON/FORTNO/          LR,                     LT,
     1    LW
      COMMON/GRID/            DT,                     DTD2,
     1    DX,                DXINV,                  DXT2IN,
     2    DXINSQ,            DY,                     DYINV,
     3    DYT2IN,            DYINSQ,                 DZ,
     4    DZD2,              DZINV,                  DZT2IN,
     5    DZINSQ
      COMMON/GULF/            CUTPT,                  KCUT,
     1    PERIDI
      COMMON/LIMITS/          IMAX,                   IMAXM1,
     1    JMAX,              JMAXM1,                 KMAX,
     2    KMAXM1,            KMAXM2,                 NMAX,
     3    KEYOUT
      COMMON/PASS/            IPASS,                  PDEPTH,
     1    PMOMAF,            PSALT,                  PWIDTH
      COMMON/PULL/            SY(17),                 SYY(17),
     1    TIME,              X(18),                  Y(17),
     2    YK2,               Z(8)
      COMMON/PULLC/           ZC(20)
```

```
      COMMON/RIVERS/          JRIV(4),          NRIV,
     1   PHASE1,              PHASE2,           RDEPTH(4),
     2   RMOMAF(4),           RWIDTH(4),        UAVG,
     3   UVAR1,               UVAR2
      COMMON/STEP/            MN,               MO,
     1   NBAR,                NWRITE,           ISTEP
      COMMON/TIDE/            PHSEMP,           PHSEPH,
     1   TAMPMP,              TAMPPH,           TAVGMP,
     2   TAVGPH,              THGTMP,           THGTPH
      COMMON/UNITS/           BETA,             BETAD2,
     1   FETCH,               GRAV,             HREF,
     2   OMEGA,               PI,               TREF,
     3   VREF,                YMAX
C
CC        SET ALL FLOW FIELD VARIABLES TO AN EXTREME VALUE TO INSURE
CC        THAT NOTHING IS MISSED.
C
      TIME    = 0.0
      DO 20 I = 1, IMAX
      DO 10 J = 1, JMAX
      SURF(MN,I,J)= 1.E70
      DHDT(I,J)   = 1.E70
      DO 10 K = 1, KMAX
      P(I,J,K)    = 1.E70
      S(MN,I,J,K) = 1.E70
      U(MN,I,J,K) = 1.E70
      DUDT(I,J,K) = 1.E70
      V(MN,I,J,K) = 1.E70
      DVDT(I,J,K) = 1.E70
   10 W(I,J,K)    = 1.E70
      SURFC(I)    = 1.E70
      DHCDT(I)    = 1.E70
      DO 20 K = 1, KMXC
      PC(I,K)     = 1.E70
      SC(MN,I,K)  = 1.E70
      UC(MN,I,K)  = 1.E70
      DUCDT(I,K)  = 1.E70
   20 WC(I,K)     = 1.E70
C
CC        SPECIFY INITIAL CONDITIONS IN RIVER MOUTHS.
C
      TIDAMP = TAVGMP + TAMPMP
      URIV1  = 1.142857 * (UAVG + UVAR1 * COS((TIME - PHASE1) * PERIDI)
     1         + UVAR2 * COS((TIME - PHASE2) * 2.0 * PERIDI))
      DO 30 NN = 1, NRIV
      J        = JRIV(NN)
      KBOT     = KFLOOR(1,J)
```

```
      ZBOT    = ZB(1,J)
      SURF(MN,1,J) = TIDAMP
      DHDT(1,J)    = 0.0
      URIV1   = URIV1 / RMOMAF(NN)
      PREFIX = 1.0 / (TIDAMP + 1.0 - ZBOT)
      DO 30 K = KBOT, KMAX
      DUDT(1,J,K)  = 0.0
      DVDT(1,J,K)  = 0.0
      VERT    = (Z(K) - ZBOT) * PREFIX
      PROFIL = 1.0 - (1.0 - VERT)**7
      U(MN,1,J,K)  = URIV1 * PROFIL
      V(MN,1,J,K)  = 0.0
      W(1,J,K)     = 0.0
   30 S(MN,1,J,K)  = 0.0
C
CC       SPECIFY INITIAL CONDITIONS IN PASS AUX HERONS.
C
      JW      = JWEST(IPASS)
      JWM1    = JW - 1
      KBOT    = KFLOOR(IPASS,JW)
      DO 35 K = KBOT, KMAX
      S(MN,IPASS,JWM1,K) = PSALT
   35 V(MN,IPASS,JWM1,K) = 0.0
C
CC       SPECIFY INITIAL CONDITIONS IN BAY PROPER.
C
      XMAX    = X(IMAX)
      DO 40 I = 2, IMAX
      SALT    = X(I) / XMAX
      IF(I.LE.11) SALT = 0.01
      JW      = JWEST(I)
      JE      = JEAST(I)
      DO 40 J = JW, JE
      SURF(MN,I,J) = TIDAMP
      DHDT(I,J)    = 0.0
      KBOT    = KFLOOR(I,J)
      DO 40 K = KBOT, KMAX
      DUDT(I,J,K)  = 0.0
      DVDT(I,J,K)  = 0.0
      U(MN,I,J,K)  = 0.0
      V(MN,I,J,K)  = 0.0
      W(I,J,K)     = 0.0
   40 S(MN,I,J,K)  = SALT
C
CC       SPECIFY CONDITIONS IN SHIP CHANNEL.
C
      KBOT    = KFLORC(1)
```

```
      ZBOT    = ZBC(1)
      URIV1   = URIV1 * RMOMAF(NRIV)
      SURFC(1)    = TIDAMP
      DHCDT(1)    = 0.0
      PREFIX = 1.0 / (TIDAMP + 1.0 - ZBOT)
      DO 50 K = KBOT, KMXC
      DUCDT(1,K) = 0.0
      VERT    = (ZC(K) - ZBOT) * PREFIX
      PROFIL = 1.0 - (1.0 - VERT)**7
      UC(MN,1,K) = URIV1 * PROFIL
      WC(1,K)     = 0.0
   50 SC(MN,1,K) = 0.0
      DO 60 I = 2, IMAX
      SALT    = X(I) / XMAX
      SURFC(I)    = TIDAMP
      DHCDT(I)    = 0.0
      KBOT    = KFLORC(I)
      DO 60 K = KBOT, KMXC
      DUCDT(I,K) = 0.0
      UC(MN,I,K) = 0.0
      WC(I,K)     = 0.0
   60 SC(MN,I,K) = SALT
      CALL PRESS
C
CC       DETERMINE INITIAL GULF AND TOTAL WATER VOLUMES.
C
      NPT     = 0
      VOL     = 0.0
      KMAXP1 = KMAX + 1
      DO 70 I = 2, IMAXM1
      JW      = JWEST(I)
      JE      = JEAST(I)
      DO 70 J = JW, JE
   70 NPT     = NPT + KMAXP1 - KFLOOR(I,J)
      DO 80 I = 2, IMAXM1
   80 NPT     = NPT + KFLOOR(I,JWAG) - KFLORC(I)
      DO 85 NN = 1, 7
      QDOT(NN)   = 0.0
      QSDOT(NN) = 0.0
      QNET(NN)   = 0.0
   85 QSNET(NN) = 0.0
      CALL BALNCE
      PTCTIN = 1.0 / (FLOAT(NPT) * DT)
      HREF3   = HREF * HREF * HREF
      VOL     = QNET(7) * HREF3
      VOLSLT = QSNET(7) * HREF3
      WRITE(LW,1000) NPT, VOL, VOLSLT
```

```
C
C         ZERO MATERIAL BALANCE VARIABLES.
C
      DO 90 NN = 1, 6
      QNET(NN)  = 0.0
   90 QSNET(NN) = 0.0
      RETURN
C
 1000 FORMAT(///,5X,'NO. OF POINTS IN FLOW FIELD  = ',I4,//,5X,'INITIAL
     1TOTAL WATER VOLUME = ',1PE10.3,' M**3',//,5X,'INITIAL GULF WATER V
     2OLUME  = ',E10.3,' M**3')
      END
```

```
      SUBROUTINE PRELIM
C
CCC      READ DATA TO SET UP PROBLEM, PERFORM PRELIMINARY CALCULATIONS,
CCC      AND WRITE DESCRIPTIVE INFORMATION.
C
      LOGICAL                 ISTEP,                  KEYOUT
      DOUBLE PRECISION        SURF(2,18,17),          SURFC(18)
      COMMON/BNKCRD/          EAST(22),               IIMAX,
     1    JE,                 JEAST(22),              JW,
     2    JWEST(22),          SOUTH(22),              WEST(22)
      COMMON/CHNNEL/          CDEPTH,                 CWIDTH,
     1    JWAG,               KMXC,                   KMXCM1,
     2    KMXCM2
      COMMON/FLOOR/           KFLOOR(18,17),          ZB(18,17)
      COMMON/FLOORC/          KFLORC(18),             ZBC(18)
      COMMON/FLOWI/           P(18,17,08),            S(2,18,17,08),
     1    U(2,18,17,08),      V(2,18,17,08),          W(18,17,08)
      COMMON/FLOW2/           SURF,                   DHDT(18,17),
     1    DUDT(18,17,08),     DVDT(18,17,08)
      COMMON/FLOWC1/          PC(18,20),              SC(2,18,20),
     1    UC(2,18,20),        WC(18,20)
      COMMON/FLOWC2/          SURFC,                  DHCDT(18),
     1    DUCDT(18,20)
      COMMON/FORCES/          F,                      FWINDX(18,17),
     1    FWINDY(18,17),      TOPLYR
      COMMON/FORTNO/          LR,                     LT,
     1    LW
      COMMON/GRID/            DT,                     DTD2,
     1    DX,                 DXINV,                  DXT2IN,
     2    DXINSQ,             DY,                     DYINV,
     3    DYT2IN,             DYINSQ,                 DZ,
     4    DZD2,               DZINV,                  DZT2IN,
     5    DZINSQ
      COMMON/GULF/            CUTPT,                  KCUT,
     1    PERIDI
      COMMON/LIMITS/          IMAX,                   IMAXM1,
     1    JMAX,               JMAXM1,                 KMAX,
     2    KMAXM1,             KMAXM2,                 NMAX,
     3    KEYOUT
      COMMON/LN/              FUNLN1,                 FUNKAP
      COMMON/PASS/            IPASS,                  PDEPTH,
     1    PMOMAF,             PSALT,                  PWIDTH
      COMMON/PULL/            SY(17),                 SYY(17),
     1    TIME,               X(18),                  Y(17),
     2    YK2,                Z(8)
      COMMON/PULLC/           ZC(20)
      COMMON/RIVERS/          JRIV(4),                NRIV,
```

```
     1    PHASE1,              PHASE2,              RDEPTH(4),
     2    RMOMAF(4),           RWIDTH(4),           UAVG,
     3    UVAR1,               UVAR2
      COMMON/RYTE/             IO,                  LABELI,
     1    LABELO
      COMMON/STEP/             MN,                  MO,
     1    NBAR,                NWRITE,              ISTEP
      COMMON/TIDE/             PHSEMP,              PHSEPH,
     1    TAMPMP,              TAMPPH,              TAVGMP,
     2    TAVGPH,              THGTMP,              THGTPH
      COMMON/TUNE/             ACCEL,               ARTVSC,
     1    AVCOMP
      COMMON/TURB/             CDIF,                CRICH,
     1    CVIS,                DIFFUS,              VISC
      COMMON/UNITS/            BETA,                BETAD2,
     1    FETCH,               GRAV,                HREF,
     2    OMEGA,               PI,                  TREF,
     3    VREF,                YMAX
C
CCC     *LTAPE = 1 IF A RUN IS TO BE CONTINUED WITH STARTING DATA READ
CCC      FROM TAPE OR FROM PREVIOUS CASE.
CCC     *LTAPE = 0 IF STARTING DATA IS TO BE GENERATED INTERNALLY.
CCC     *LTAPE = -1 IF LAST CASE HAS BEEN RUN.
CCC
CCC     *LABELI = 0 IF STARTING DATA GENERATED INTERNALLY OR LEFT FROM
CCC      PREVIOUS CASE.  LABELI IS THE LABEL NUMBER OF THE DATA SET USED
CCC      FOR STARTING DATA AND TO WHICH RESULTS ARE WRITTEN, OTHERWISE.
CCC     *LABELO IS THE LABEL NUMBER OF THE DATA SET TO WHICH DUPLICATE
CCC      COPY OF LAST RESULTS FOR A GIVEN CASE ARE WRITTEN.
C
      READ(LR,1000) LTAPE, LABELI, LABELO
      IF(LTAPE.EQ.-1) STOP
      IF(LABELI.EQ.0) GO TO 5
      IO     = LT
      IF(LTAPE.EQ.1) CALL REED
    5 CONTINUE
C
CCC     *IMAX, JMAX, KMAX ARE THE MAXIMUM NUMBER OF GRID POINTS IN THE
CCC      X, Y, Z DIRECTIONS, EXCLUDING THE SHIP CHANNEL.
CCC     *NBAR IS THE NUMBER OF TIME STEPS BETWEEN OUTPUTS OF DESCRIPTIVE
CCC      PARAMETERS.  NBAR MAY BE ADJUSTED AS REQUIRED AS THE CALCULA-
CCC      TION PROCEEDS.
CCC     *NMAX IS THE MAXIMUM NUMBER OF TIME STEPS TO BE TAKEN.
CCC     *NWRITE IS THE NUMBER OF TIME STEPS BETWEEN OUTPUTS OF FLOW
CCC      FIELD VARIABLES AND GEOMETRY TO TAPE.  IT SHOULD DIVIDE NMAX
CCC      AN INTEGER NUMBER OF TIMES.
C
```

```
      READ(LR,1000) IMAX, JMAX, KMAX, NBAR, NMAX, NWRITE
C
CC      *CUTPT IS THE HEIGHT IN METERS OF THE SALT WEDGE AT THE MAIN
CC       PASS RELATIVE TO THE Z = 0 REFERENCE PLANE.
CC      *DTREAL IS THE SIZE OF THE TIME STEP IN SECONDS.
CC      *FETCH IS THE LENGTH IN KILOMETERS FROM NORTH TO SOUTH.
CC      *HREF IS THE LOCATION IN METERS OF THE Z = 0 REFERENCE PLANE
CC       BELOW MEAN SEA LEVEL.  Z = 0 IS SLIGHTLY BELOW THE DEEPEST
CC       POINT IN THE BAY EXCLUDING THE SHIP CHANNEL.
CC      *PERIOD IS TIME IN HOURS OF A COMPLETE TIDE CYCLE.
CC      *PHI IS THE AVERAGE LATITUDE IN DEGREES OF THE BAY.
CC      *SIGMAT SPECIFIES THE DENSITY OF SEA WATER RELATIVE TO THAT OF
CC       FRESH WATER, 1.0 GM/CC.  SIGMAT = (SEA WATER DENSITY-1.0)*1000.
C
      READ(LR,1001) CUTPT, DTREAL, FETCH, HREF, PERIOD, PHI, SIGMAT
C
CC      *YMAX IS SLIGHTLY GREATER THAN THE MAXIMUM DISTANCE IN METERS
CC       FROM THE SHIP CHANNEL TO THE EAST BANK.
C
      READ(LR,1001) YMAX
C
CC      *PHSEMP, PHSEPH ARE PHASE LAGS IN HOURS AT THE PASS AUX HERONS
CC       AND MAIN PASS, RESPECTIVELY.
CC      *TAVGMP, TAVGPH ARE AVERAGE TIDE HEIGHTS IN CENTIMETERS RELATIVE
CC       TO MEAN SEA LEVEL.
CC      *TAMPMP, TAMPPH ARE TIDAL AMPLITUDES IN CENTIMETERS.
C
      READ(LR,1001) PHSEMP, PHSEPH, TAMPMP, TAMPPH, TAVGMP, TAVGPH
C
CC      *CDIF IS THE COEFFICIENT OF EDDY DIFFUSION IN SQUARE METERS
CC       PER SECOND.
CC      *CVIS IS THE COEFFICIENT OF EDDY VISCOSITY IN SQUARE METERS
CC       PER SECOND.
CC      *ROUGH IS THE DIAMETER IN MILLIMETERS OF THE AVERAGE PARTICLE
CC       WHICH GENERATES TURBULENCE ON THE BOTTOM.
CC      *VONKAR IS THE VON KARMAN UNIVERSAL BOUNDARY LAYER CONSTANT.
C
      READ(LR,1001) CDIF, CVIS, ROUGH, VONKAR
C
CC      *JWAG IS THE VALUE OF THE INDEX J CORRESPONDING TO THE Y-COORD-
CC       INATE LINE ALONG WHICH THE SHIP CHANNEL LIES.
CC      *KMXC IS THE MAXIMUM NUMBER OF GRID POINTS IN THE Z-DIRECTION
CC       IN THE SHIP CHANNEL.
C
      READ(LR,1000) JWAG, KMXC
C
CC      *CDEPTH IS THE DEPTH IN METERS OF THE SHIP CHANNEL.
```

```
C       *CWIDTH IS THE WIDTH IN METERS OF THE SHIP CHANNEL.
C
      READ(LR,1001) CDEPTH, CWIDTH
C
C       *DIRCTN, DIRCTS ARE THE DIRECTIONS IN DEGREES TO WHICH THE WIND
C        IS BLOWING IN THE NORTH AND SOUTH PORTIONS OF THE BAY.  ANGLES
C        ARE MEASURED IN AN ANTICLOCKWISE DIRECTION FROM THE POSITIVE
C        X-AXIS.
C       *FETCHN, FETCHS ARE THE NORTH AND SOUTH LIMITS IN KILOMETERS
C        OF A REGION SEPARATING THE REGIONS ABOVE.
C       *VWINDN, VWINDS ARE THE WIND VELOCITIES IN CM/SEC 10 METERS
C        ABOVE THE WATER SURFACE.
C
      READ(LR,1001) DIRCTN, DIRCTS, FETCHN, FETCHS, VWINDN, VWINDS
C
C       *ACCEL IS AN ACCELERATION FACTOR EXPEDITING CONVERGENCE OF THE
C        SALT CONSERVATION EQUATION.  IT IS NORMALLY UNITY.
C       *ARTVSC IS AN ARTIFICIAL VISCOSITY PARAMETER IN THE KINEMATIC
C        EQUATION FOR THE FREE SURFACE.  IT NORMALLY VARIES FROM 0.5 TO
C        1.0.
C       *YK2 IS A COORDINATE STRETCHING PARAMETER IN THE Y-DIRECTION.
C        SEE SUBROUTINE RUBBER FOR DEFINITIONS.  ALWAYS MAKE YK2 LESS
C        THAN PI/2.0.
C
      READ(LR,1001) ACCEL, ARTVSC, YK2
C
C       *NRIV IS THE NUMBER OF RIVERS DEBOUCHING AT THE NORTH SHORE OF
C        THE BAY.
C       *JRIV ARE THE INDEX VALUES OF THE Y COORDINATES OF RIVER MOUTH
C        LOCATIONS.
C       *RDEPTH ARE THE RIVER DEPTHS IN METERS.
C       *RWIDTH ARE THE RIVER WIDTHS IN METERS.
C
      READ(LR,1000) NRIV
      DO 10 NN = 1, NRIV
      READ(LR,1001) RIV, RDEPTH(NN), RWIDTH(NN)
   10 JRIV(NN) = IFIX(RIV + 0.5)
C
C       *PHASE1, PHASE2 ARE PHASE LAGS IN HOURS IN A TRIGONOMETRIC
C        FUNCTION APPROXIMATING THE TIME VARIATION OF THE RIVER DIS-
C        CHARGE VELOCITIES.
C       *UAVG IS THE TIME AVERAGE DISCHARGE VELOCITY IN METER/SECOND.
C       *UVAR1, UVAR2 ARE AMPLITUDES IN METER/SECOND OF THE PERIODIC
C        TERMS.
C
      READ(LR,1001) PHASE1, PHASE2, UAVG, UVAR1, UVAR2
C
```

```
C       *IPASS IS THE INDEX VALUE OF THE X-CCORDINATE OF THE LOCATION
C        OF THE PASS AUX HERONS ON THE WESTERN SHORE.
C       *PDEPTH IS THE PASS DEPTH IN METERS.
C       *PSALT IS THE SALINITY IN THE MISSISSIPPI SOUND.
C       *PWIDTH IS THE PASS WIDTH IN METERS.
C
      READ(LR,1001) PASS, PDEPTH, PSALT, PWIDTH
      IPASS  = IFIX(PASS + 0.5)
C
C        SKIP READING CARDS SPECIFYING BAY GEOMETRY IF CONTINUING A RUN.
C
      IF(LTAPE.EQ.1) GO TO 22
C
C        READ IIMAX DATA CARDS TO SPECIFY THE GEOMETRY OF THE BAY AS
C        FOLLOWSO
C        1.  THE DISTANCE SOUTH OF THE NORTHERN BOUNDARY OF THE BAY.
C            BEGIN WITH 0.0 & END WITH FETCH.
C        2.  THE DISTANCE FROM THE SHIP CHANNEL TO THE WESTERN BOUNDARY
C            OF THE BAY (NOTE...ALWAYS NEGATIVE).
C        3.  THE DISTANCE FROM THE SHIP CHANNEL TO THE EASTERN BOUNDARY
C            OF THE BAY.
C        ALL DISTANCES ARE IN NAUTICAL MILES.
C
      READ(LR,1000) IIMAX
      DO 20 II = 1, IIMAX
      READ(LR,1001) SOUTH1, WEST1, EAST1
      SOUTH(II) = SOUTH1 * 1852.
      WEST(II)  = WEST1 * 1852.
   20 EAST(II)  = EAST1 * 1852.
C
C        INPUT CONSTANTS OF POLYNOMIAL FITS TO BAY AND SHIP CHANNEL
C        BOTTOMS.  THESE BOUNDARIES ARE SPECIFIED RELATIVE TO THE Z = 0
C        REFERENCE PLANE.  CONSTANTS HAVE BEEN NORMALIZED WITH HREF.
C
      READ(LR,1018) B1, B2, B3, B4, B5, B6
      READ(LR,1018) B7, B8, B9, B10
      READ(LR,1018) BB1, BB2, BB3, BB4
   22 CONTINUE
C
C       PRINT INPUT DATA TO IDENTIFY CASE BEING RUN.
C
      WRITE(LW,1002) LTAPE, LABELI, LABELO
      WRITE(LW,1003) IMAX, JMAX, KMAX, NBAR, NMAX, NWRITE
      WRITE(LW,1004) CUTPT, DTREAL, FETCH, HREF, PERIOD, PHI, SIGMAT,
     1 YMAX
      WRITE(LW,1023) PHSEMP, PHSEPH, TAMPMP, TAMPPH, TAVGMP, TAVGPH
      WRITE(LW,1005) CDIF, CVIS, ROUGH, VONKAR
```

```
      WRITE(LW,1006) JWAG, KMXC
      WRITE(LW,1007) CDEPTH, CWIDTH
      WRITE(LW,1008) DIRCTN, DIRCTS, FETCHN, FETCHS, VWINDN, VWINDS
      WRITE(LW,1009) ACCEL, ARTVSC, YK2
      WRITE(LW,1010) (NN, JRIV(NN), RDEPTH(NN), RWIDTH(NN), NN = 1,
     1 NRIV)
      WRITE(LW,1024) PHASE1, PHASE2, UAVG, UVAR1, UVAR2
      WRITE(LW,1011) IPASS, PDEPTH, PSALT, PWIDTH
C
CC       SKIP PRINTING BAY GEOMETRY IF CONTINUING A RUN.
C
      IF(LTAPE.EQ.1) GO TO 24
      WRITE(LW,1012)
      WRITE(LW,1013) (II, SOUTH(II), WEST(II), EAST(II), II = 1, IIMAX)
      WRITE(LW,1019) B1, B2, B3, B4, B5, B6, B7, B8, B9, B10,
     1               BB1, BB2, BB3, BB4
   24 CONTINUE
C
CC       CALCULATE CONSTANTS AND CONVERT UNITS.
C
      IMAXM1 = IMAX - 1
      JMAXM1 = JMAX - 1
      KMAXM1 = KMAX - 1
      KMAXM2 = KMAX - 2
      KMXCM1 = KMXC - 1
      KMXCM2 = KMXC - 2
      VREF   = SQRT(GRAV * HREF)
      TREF   = HREF / VREF
      CUTPT  = CUTPT / HREF
      DT     = DTREAL / TREF
      DTD2   = 0.5 * DT
      FETCH  = FETCH * 1000.0 / HREF
      PERIOD = PERIOD * 3600.0 / TREF
      PERIDI = 2.0 * PI / PERIOD
      PHI    = PHI * PI / 180.0
      PHSEMP = PHSEMP * 3600.0 / TREF
      PHSEPH = PHSEPH * 3600.0 / TREF
      TAMPMP = 0.01 * TAMPMP / HREF
      TAMPPH = 0.01 * TAMPPH / HREF
      TAVGMP = 0.01 * TAVGMP / HREF
      TAVGPH = 0.01 * TAVGPH / HREF
      YMAX   = YMAX * 1000.0 / HREF
      ARTVSC = ARTVSC / (HREF * VREF)
      CDIF   = CDIF / (HREF * VREF)
      CVIS   = CVIS / (HREF * VREF)
      ROUGH  = 0.001 * ROUGH / HREF
      CDEPTH = CDEPTH / HREF
```

```
      CWIDTH = CWIDTH / HREF
      DIRCTN = DIRCTN * PI / 180.0
      DIRCTS = DIRCTS * PI / 180.0
      FETCHN = FETCHN * 1000.0 / HREF
      FETCHS = FETCHS * 1000.0 / HREF
      DO 30 NN = 1, NRIV
      RDEPTH(NN) = RDEPTH(NN) / HREF
   30 RWIDTH(NN) = RWIDTH(NN) / HREF
      PHASE1 = PHASE1 * 3600. / TREF
      PHASE2 = PHASE2 * 3600. / TREF
      UAVG   = UAVG / VREF
      UVAR1  = UVAR1 / VREF
      UVAR2  = UVAR2 / VREF
      PDEPTH = PDEPTH / HREF
      PWIDTH = PWIDTH / HREF
C
CC       GENERATE GRID SYSTEM.
C
      CALL RUBBER
C
CC       SKIP CALCULATING BOUNDARIES IF CONTINUING A RUN.
C
      IF(LTAPE.EQ.1) GO TO 137
      DO 40 II = 1, IIMAX
      SOUTH(II) = SOUTH(II) / HREF
      WEST(II)  = WEST(II) / HREF
   40 EAST(II)  = EAST(II) / HREF
C
CC       ESTABLISH BAY LIMITS FOR GRID SYSTEM.
C
      I       = 1
      II      = 1
   50 J       = JWAG
   60 J       = J + 1
      IF(Y(J).LT.EAST(II)) GO TC 60
      JEAST(I) = J - 1
      J       = JWAG
   70 J       = J - 1
      IF(Y(J).GT.WEST(II)) GO TC 70
      JWEST(I) = J + 1
      IF(I.NE.1) GO TO 80
      I       = IMAX
      II      = IIMAX
      GO TO 50
   80 DO 120 I = 2, IMAXM1
      II      = 1
   90 II       = II + 1
```

```
      IF(SOUTH(II).LT.X(I)) GO TO 90
      WATE   = (X(I) - SOUTH(II-1))/(SOUTH(II) - SOUTH(II-1))
      YEAST  = EAST(II) * WATE + EAST(II-1) * (1.0 - WATE)
      YWEST  = WEST(II) * WATE + WEST(II-1) * (1.0 - WATE)
      J      = JWAG
  100 J      = J + 1
      IF(Y(J).LT.YEAST) GO TO 100
      JEAST(I) = J - 1
      J      = JWAG
  110 J      = J - 1
      IF(Y(J).GT.YWEST) GO TO 110
  120 JWEST(I) = J + 1
      WRITE(LW,1014)
      WRITE(LW,1015) (I, JWEST(I), JEAST(I), I = 1, IMAX)
C
C        GENERATE BOTTOM CONFIGURATION.
C
      CALL BOTTOM(B1,B2,B3,B4,B5,B6,B7,B8,B9,B10,BB1,BB2,BB3,BB4)
      WRITE(LW,1016)
      DO 130 I = 1, IMAX
      DO 130 J = 1, JMAX
      XREAL  = X(I) * HREF
      YREAL  = Y(J) * HREF
      ZBREAL = ZB(I,J) * HREF
  130 WRITE(LW,1017) I, J, KFLCOR(I,J), XREAL, YREAL, ZBREAL
      WRITE(LW,1021)
      DO 135 I = 1, IMAX
      XREAL  = X(I) * HREF
      ZBREAL = ZBC(I) * HREF
  135 WRITE(LW,1022) I, KFLORC(I), XREAL, ZBREAL
  137 CONTINUE
C
C        COMPLETE CALCULATION OF CONSTANTS AND WIND SHEAR FORCES.
C
      BETA   = 0.001 * SIGMAT
      BETAD2 = 0.5 * BETA
      CRICH  = -2.0 * BETA * DZ
      F      = 2.0 * OMEGA * SIN(PHI) * TREF
      FUNLN1 = ALOG(0.2/ROUGH)
      FUNKAP = VONKAR * 8.5
      FWINDN = 1.64E-4 * (VWINDN**1.333)
      FWINDS = 1.64E-4 * (VWINDS**1.333)
      CXN    = COS(DIRCTN) * FWINDN
      CYN    = SIN(DIRCTN) * FWINDN
      CXS    = COS(DIRCTS) * FWINDS
      CYS    = SIN(DIRCTS) * FWINDS
      WRITE(LW,1020) FWINDN, CXN, CYN
```

```
      WRITE(LW,1020) FWINDS, CXS, CYS
      PREFIX = 1.0 / ((100.0 * VREF)**2)
      CXN    = PREFIX * CXN
      CYN    = PREFIX * CYN
      CXS    = PREFIX * CXS
      CYS    = PREFIX * CYS
      DO 149 I = 1, IMAX
      X1     = X(I)
      IF(X1.GT.FETCHN) GO TO 141
      CX     = CXN
      CY     = CYN
      GO TO 144
  141 IF(X1.LT.FETCHS) GO TO 143
      CX     = CXS
      CY     = CYS
      GO TO 144
  143 CX     = (X1 - FETCHN) / (FETCHS - FETCHN) * (CXS - CXN) + CXN
      CY     = (X1 - FETCHN) / (FETCHS - FETCHN) * (CYS - CYN) + CYN
  144 DO 148 J = 1, JMAX
      FWINDX(I,J) = CX
  148 FWINDY(I,J) = CY
  149 CONTINUE
      C1     = CUTPT / DZ
      IF(C1.LT.0.0) GO TO 150
      KCUT   = IFIX(C1 + 0.5)
      GO TO 160
  150 KCUT   = IFIX(C1 - 0.5)
  160 DO 170 NN = 1, NRIV
      J      = JRIV(NN)
      AREA1  = RWIDTH(NN) * RDEPTH(NN)
      AREA2  = 0.5 * (Y(J+1) - Y(J-1)) * (1.0 - ZB(1,J))
      RATIO  = AREA2 / AREA1
  170 RMOMAF(NN) = RATIO
      PMOMAF = 0.5 * (X(IPASS+1) - X(IPASS-1)) / PWIDTH
C
CC       IF LTAPE = 0, MAKE COLD START BY CALLING SUBROUTINE INITAL.
C
      IF(LTAPE.EQ.0) CALL INITAL
      DO 190 I = 1, IMAX
      DO 180 J = 1, JMAX
      SURF(MO,I,J)= SURF(MN,I,J)
      DO 180 K = 1, KMAX
      U(MO,I,J,K) = U(MN,I,J,K)
      V(MO,I,J,K) = V(MN,I,J,K)
      S(MO,I,J,K) = S(MN,I,J,K)
  180 CONTINUE
      DO 190 K = 1, KMXC
```

```
      UC(MO,I,K)   = UC(MN,I,K)
      SC(MO,I,K)   = SC(MN,I,K)
  190 CONTINUE
      IF(LABELI.EQ.0.AND.LTAPE.EQ.1) GO TO 200
      IO      = LT
      CALL RITE3
  200 CONTINUE
      RETURN
C
 1000 FORMAT(8I10)
 1001 FORMAT(9F10.5)
 1002 FORMAT(/,5X,'LTAPE  = ',I4,/,5X,'LABELI = ',I4,/,5X,'LABELO = ',
     1 I4)
 1003 FORMAT(/,5X,'IMAX   = ',I4,/,5X,'JMAX   = ',I4,/,5X,'KMAX   = ',
     1 I4,/,5X,'NBAR   = ',I4,/,5X,'NMAX   = ',I4,/,5X,'NWRITE = ',I4)
 1004 FORMAT(/,5X,'CUTPT  = ',F10.4,' M',/,5X,'DTREAL = ',F10.4,' SEC',
     1 /,5X,'FETCH  = ',F10.4,' KM',/,5X,'HREF   = ',F10.4,' M',/,5X,
     2 'PERIOD = ',F10.4,' HR',/,5X,'PHI    = ',F10.4,' DEG',/,5X,
     3 'SIGMAT = ',F10.4,/,5X,'YMAX   = ',F10.4,' KM')
 1005 FORMAT(/,5X,'CDIF   = ',F10.4,' M**2/SEC',/,5X,'CVIS   = ',F10.4,
     1 ' M**2/SEC',/,5X,'ROUGH  = ',F10.4,' MM',/,5X,'VONKAR = ',F10.4)
 1006 FORMAT(/,5X,'JWAG   = ',I4,/,5X,'KMXC   = ',I4)
 1007 FORMAT(/,5X,'CDEPTH = ',F10.4,' M',/,5X,'CWIDTH = ',F10.4,' M')
 1008 FORMAT(/,5X,'DIRCTN = ',F10.4,' DEG,     DIRCTS = ',F10.4,' DEG',/
     1,5X,'FETCHN = ',F10.4,' KM,      FETCHS = ',F10.4,' KM',/
     2,5X,'VWINDN = ',F10.4,' CM/SEC,  VWINDS = ',F10.4,' CM/SEC')
 1009 FORMAT(/,5X,'ACCEL  = ',F10.4,/,5X,'ARTVSC = ',F10.4,/,5X,
     1 'YK2    = ',F10.4)
 1010 FORMAT(/,5X,'RIVER ',I1,'   J = ',I2,'   DEPTH = ',F5.1,
     1 ' M   WIDTH = ',F6.0,' M')
 1011 FORMAT(/,5X,'IPASS  = ',I4,/,5X,'PDEPTH = ',F10.4,' M',/,5X,
     1 'PSALT  = ',F10.4,/,5X,'PWIDTH = ',F10.4,' M')
 1012 FORMAT(/,15X,'II',15X,'SOUTH (M)',15X,'WEST (M)',15X,'EAST (M)',/)
 1013 FORMAT((13X,I4,13X,1PE13.6,2(10X,E13.6)))
 1014 FORMAT(/,15X,'I',15X,'JWEST',15X,'JEAST')
 1015 FORMAT(13X,I3,16X,I3,17X,I3)
 1016 FORMAT(/,10X,'I',10X,'J',10X,'KFLOOR(I,J)',15X,'XREAL',14X,
     1 'YREAL',15X,'ZBREAL',/)
 1017 FORMAT(8X,I3,8X,I3,10X,I4,20X,1PE13.6,7X,E13.6,5X,E13.6)
 1018 FORMAT(6E12.0)
 1019 FORMAT(//,5X,'B1 THRU B10 AND BB1 THRU BB4 ARE AS FOLLOWS0',/,
     1 5X,1P5E16.6,/,(5X,5E16.6))
 1020 FORMAT(/,5X,'FWIND = ',1PE12.5,' DYNES/CM**2,   FWINDX = ',E12.5,
     1 ' DYNES/CM**2,   FWINDY = ',E12.5,' DYNES/CM**2')
 1021 FORMAT(/,10X,'I',11X,'KFLORC(I)',16X,'XREAL',14X,'ZBCREAL',/)
 1022 FORMAT(8X,I3,10X,I4,20X,1PE13.6,5X,E13.6)
 1023 FORMAT(/,5X,'PHSEMP = ',F10.4,' HR,   PHSEPH = ',F10.4,' HR',
```

```
     1 /,5X,'TAMPMP = ',F10.4,' CM,    TAMPPH = ',F10.4,' CM',/,5X,
     2 'TAVGMP = ',F10.4,' CM,    TAVGPH = ',F10.4,' CM')
1024 FORMAT(/,5X,'PHASE1 = ',F10.4,' HR',/,5X,'PHASE2 = ',F10.4,' HR',
     1 /,5X,'UAVG'    = ',F10.4,' M/S',/,5X,'UVAR1   = ',F10.4,' M/S',
     2 /,5X,'UVAR2    = ',F10.4,' M/S')
      END
```

```
      SUBROUTINE PRESS
C
C        CALCULATE PRESSURE FIELD USING STATIC FLUID APPROXIMATION.
C        INTEGRATE FROM FREE SURFACE DOWN TO BAY BOTTOM.  NOTE THAT P IS
C        THE DEVIATION FROM THE HYDROSTATIC PRESSURE FOR FRESH WATER
C        HAVING A FREE SURFACE AT MEAN SEA LEVEL.
C
      LOGICAL                  ISTEP,              KEYOUT
      DOUBLE PRECISION         SURF(2,18,17),      SURFC(18)
      COMMON/BNKCRD/           EAST(22),           IIMAX,
     1    JE,                  JEAST(22),          JW,
     2    JWEST(22),           SOUTH(22),          WEST(22)
      COMMON/CHNNEL/           CDEPTH,             CWIDTH,
     1    JWAG,                KMXC,               KMXCM1,
     2    KMXCM2
      COMMON/FLOOR/            KFLOOR(18,17),      ZB(18,17)
      COMMON/FLOORC/           KFLORC(18),         ZBC(18)
      COMMON/FLOW1/            P(18,17,08),        S(2,18,17,08),
     1    U(2,18,17,08),       V(2,18,17,08),      W(18,17,08)
      COMMON/FLOW2/            SURF,               DHDT(18,17),
     1    DUDT(18,17,08),      DVDT(18,17,08)
      COMMON/FLOWC1/           PC(18,20),          SC(2,18,20),
     1    UC(2,18,20),         WC(18,20)
      COMMON/FLOWC2/           SURFC,              DHCDT(18),
     1    DUCDT(18,20)
      COMMON/FORTNO/           LR,                 LT,
     1    LW
      COMMON/GRID/             DT,                 DTD2,
     1    DX,                  DXINV,              DXT2IN,
     2    DXINSQ,              DY,                 DYINV,
     3    DYT2IN,              DYINSQ,             DZ,
     4    DZD2,                DZINV,              DZT2IN,
     5    DZINSQ
      COMMON/LIMITS/           IMAX,               IMAXM1,
     1    JMAX,                JMAXM1,             KMAX,
     2    KMAXM1,              KMAXM2,             NMAX,
     3    KEYOUT
      COMMON/INDEX/            I,                  J,
     1    K,                   N
      COMMON/RIVERS/           JRIV(4),            NRIV,
     1    PHASE1,              PHASE2,             RDEPTH(4),
     2    RMOMAF(4),           RWIDTH(4),          UAVG,
     3    UVAR1,               UVAR2
      COMMON/STEP/             MN,                 MO,
     1    NBAR,                NWRITE,             ISTEP
      COMMON/UNITS/            BETA,               BETAD2,
     1    FETCH,               GRAV,               HREF,
```

```
     2   OMEGA,                  PI,                     TREF,
     3   VREF,                   YMAX
      TEST   = 3.0 * DZ
      DO 10 I = 1, IMAX
      JW     = JWEST(I)
      JE     = JEAST(I)
      DO 10 J = JW, JE
      KBOT   = KFLOOR(I,J)
      IF(KBOT.GT.KMAX) GO TO 10
      SK     = S(MN,I,J,KMAX)
      RHO    = 1.0 + BETA * SK
      PK     = RHO * SURF(MN,I,J)
      P(I,J,KMAX) = PK
      KKMAX  = KMAX - KBOT
      DO 5 KK = 1, KKMAX
      K      = KMAX - KK
      SKP1   = SK
      PKP1   = PK
      SK     = S(MN,I,J,K)
      DELRHO = BETAD2 * (SK + SKP1)
      PK     = PKP1 + DELRHO * DZ
      P(I,J,K) = PK
C
C        DETERMINE IF PRESSURE HAS DIVERGED.
C
      IF(ABS(PK).GT.TEST) GO TO 100
    5 CONTINUE
   10 CONTINUE
      DO 20 I = 1, IMAX
      KBOT   = KFLORC(I)
      SK     = SC(MN,I,KMXC)
      RHO    = 1.0 + BETA * SK
      PK     = RHO * SURFC(I)
      PC(I,KMXC) = PK
      KKMAX  = KMXC - KBOT
      DO 20 KK = 1, KKMAX
      K      = KMXC - KK
      SKP1   = SK
      PKP1   = PK
      SK     = SC(MN,I,K)
      DELRHO = BETAD2 * (SK + SKP1)
      PK     = PKP1 + DELRHO * DZ
      PC(I,K) = PK
C
C        DETERMINE IF PRESSURE HAS DIVERGED.
C
      IF(ABS(PK).GT.TEST) GO TO 110
```

```
   20 CONTINUE
      RETURN
  100 WRITE(LW,1000) N, I, J, K
      GO TO 120
  110 WRITE(LW,1001) N, I, K
  120 KEYOUT = .TRUE.
      RETURN
C
 1000 FORMAT(/,5X,'PRESSURE DIVERGED AT N = ',I4,', I = ',I4,', J = ',
     1 I4,', K = ',I4)
 1001 FORMAT(/,5X,'PRESSURE DIVERGED AT N = ',I4,', I = ',I4,', K = ',
     1 I4)
      END
```

```
      SUBROUTINE PRNT
C
CC        OUTPUT FLOW FIELD VARIABLES TO LINE PRINTER.
C
      LOGICAL                 ISTEP,                  KEYOUT
      DIMENSION               SURF1(18)
      DOUBLE PRECISION        SURF(2,18,17),          SURFC(18)
      COMMON/BARS/            PTCTIN,                 SBAR,
     1    UBAR,                VBAR
      COMMON/BNKCRD/          EAST(22),               IIMAX,
     1    JE,                  JEAST(22),              JW,
     2    JWEST(22),           SOUTH(22),              WEST(22)
      COMMON/CHNNEL/          CDEPTH,                 CWIDTH,
     1    JWAG,                KMXC,                   KMXCM1,
     2    KMXCM2
      COMMON/FLOOR/           KFLOOR(18,17),          ZB(18,17)
      COMMON/FLOORC/          KFLORC(18),             ZBC(18)
      COMMON/FLOW1/           P(18,17,08),            S(2,18,17,08),
     1    U(2,18,17,08),       V(2,18,17,08),          W(18,17,08)
      COMMON/FLOW2/           SURF,                   DHDT(18,17),
     1    DUDT(18,17,08),      DVDT(18,17,08)
      COMMON/FLOWC1/          PC(18,20),              SC(2,18,20),
     1    UC(2,18,20),         WC(18,20)
      COMMON/FLOWC2/          SURFC,                  DHCDT(18),
     1    DUCDT(18,20)
      COMMON/FORTNO/          LR,                     LT,
     1    LW
      COMMON/INDEX/           I,                      J,
     1    K,                   N
      COMMON/LIMITS/          IMAX,                   IMAXM1,
     1    JMAX,                JMAXM1,                 KMAX,
     2    KMAXM1,              KMAXM2,                 NMAX,
     3    KEYOUT
      COMMON/PULL/            SY(17),                 SYY(17),
     1    TIME,                X(18),                  Y(17),
     2    YK2,                 Z(8)
      COMMON/STEP/            MN,                     MO,
     1    NBAR,                NWRITE,                 ISTEP
      COMMON/UNITS/           BETA,                   BETAD2,
     1    FETCH,               GRAV,                   HREF,
     2    OMEGA,               PI,                     TREF,
     3    VREF,                YMAX
C
      CTM     = TREF / 3600.
      CSB     = PTCTIN / TREF
      CUB     = CSB * VREF
      CSH     = HREF * 100.
```

```
      CPR     = GRAV * HREF * 0.01
      KMAXP1 = KMAX + 1
      KMXCP1 = KMXC + 1
      JMAXP1 = JMAX + 1
      TIMERL = TIME * CTM
      SBAR    = SBAR * CSB
      UBAR    = UBAR * CUB
      VBAR    = VBAR * CUB
      WRITE(LW,1000) N, TIMERL, SBAR, UBAR, VBAR
      WRITE(LW,1001)
      DO 10 I = 1, IMAX
      JW      = JWEST(I)
      JE      = JEAST(I)
      DO 10 J = JW, JE
      WRITE(LW,1002)
      KBOT    = KFLOOR(I,J)
      IF(KBOT.GT.KMAX) GO TO 10
      KKMAX   = KMAXP1 - KBOT
      DO 5 KK = 1, KKMAX
      K       = KMAXP1 - KK
      UNEW    = U(MN,I,J,K) * VREF
      VNEW    = V(MN,I,J,K) * VREF
      WNEW    = W(I,J,K) * VREF
      SNEW    = S(MN,I,J,K)
      PNEW    = P(I,J,K) * CPR
C
CC       VELOCITIES ARE IN M/S, SALINITIES ARE GULF WATER FRACTIONS,
CC       AND PRESSURES ARE IN BARS.
C
      WRITE(LW,1003) UNEW, VNEW, WNEW, SNEW, PNEW, I, J, K
    5 CONTINUE
   10 CONTINUE
      WRITE(LW,1004)
      DO 20 I = 1, IMAX
      WRITE(LW,1002)
      KKMXC   = KMXCP1 - KFLORC(I)
      DO 20 KK = 1, KKMXC
      K       = KMXCP1 - KK
      UNEW    = UC(MN,I,K) * VREF
      WNEW    = WC(I,K) * VREF
      SNEW    = SC(MN,I,K)
      PNEW    = PC(I,K) * CPR
      WRITE(LW,1005) UNEW, WNEW, SNEW, PNEW, I, K
   20 CONTINUE
C
CC       GENERATE SURFACE HEIGHT MAP.
C
```

```
      WRITE(LW,1006)
      WRITE(LW,1007) (I, I = 1, IMAX)
      DO 40 JJ = 1, JMAX
      J       = JMAXP1 - JJ
      DO 30 I = 1, IMAX
   30 SURF1(I) = SURF(MN,I,J) * CSH
      WRITE(LW,1008) J, (SURF1(I), I = 1, IMAX)
   40 CONTINUE
C
      WRITE(LW,1009)
      DO 50 I = 1, IMAX
   50 SURF1(I) = SURFC(I) * CSH
      WRITE(LW,1008) JWAG, (SURF1(I), I = 1, IMAX)
      RETURN
 1000 FORMAT(5X,' N = ',I5,', TIME = ',F6.2,' HR., SBAR = ',1PE12.4,
     1', UBAR = ',E12.4,', VBAR = ',E12.4)
 1001 FORMAT(//,7X,'U,M/S       V,M/S       W,M/S    FRAC G W      P,BAR
     1 I    J    K',/)
 1002 FORMAT(/)
 1003 FORMAT(5X,1PE9.2,4(2X,E9.2),3(2X,I3))
 1004 FORMAT(//,7X,'UC,M/S      WC,M/S    FRAC G W      PC,BAR     I    K',/)
 1005 FORMAT(5X,1PE9.2,3(2X,E9.2),2(2X,I2))
 1006 FORMAT(///,48X,'SURFACE HEIGHTS, CM.',//)
 1007 FORMAT(4X,'I=',2016,//)
 1008 FORMAT(' J=',I2,3X,F5.1,20(1X,F5.1))
 1009 FORMAT(///,52X,'SHIP CHANNEL',/)
      END
```

```
      SUBROUTINE REED
C
CC        SUBROUTINE FOR I/O TO SEQUENTIAL STORAGE DEVICE.
C
      LOGICAL                 ISTEP,              KEYOUT
      DOUBLE PRECISION        SURF(2,18,17),      SURFC(18)
      COMMON/ACCT/            QDOT(7),            QNET(7),
     1    QSDOT(7),           QSNET(7)
      COMMON/BARS/            PTCTIN,             SBAR,
     1    UBAR,               VBAR
      COMMON/BNKCRD/          EAST(22),           IIMAX,
     1    JE,                 JEAST(22),          JW,
     2    JWEST(22),          SOUTH(22),          WEST(22)
      COMMON/CHNNEL/          CDEPTH,             CWIDTH,
     1    JWAG,               KMXC,               KMXCM1,
     2    KMXCM2
      COMMON/FLOOR/           KFLOOR(18,17),      ZB(18,17)
      COMMON/FLOORC/          KFLORC(18),         ZBC(18)
      COMMON/FLOW1/           P(18,17,08),        S(2,18,17,08),
     1    U(2,18,17,08),      V(2,18,17,08),      W(18,17,08)
      COMMON/FLOW2/           SURF,               DHDT(18,17),
     1    DUDT(18,17,08),     DVDT(18,17,08)
      COMMON/FLOWC1/          PC(18,20),          SC(2,18,20),
     1    UC(2,18,20),        WC(18,20)
      COMMON/FLOWC2/          SURFC,              DHCDT(18),
     1    DUCDT(18,20)
      COMMON/FORCES/          F,                  FWINDX(18,17),
     1    FWINDY(18,17),      TOPLYR
      COMMON/GRID/            DT,                 DTD2,
     1    DX,                 DXINV,              DXT2IN,
     2    DXINSQ,             DY,                 DYINV,
     3    DYT2IN,             DYINSQ,             DZ,
     4    DZD2,               DZINV,              DZT2IN,
     5    DZINSQ
      COMMON/GULF/            CUTPT,              KCUT,
     1    PERIDI
      COMMON/LIMITS/          IMAX,               IMAXM1,
     1    JMAX,               JMAXM1,             KMAX,
     2    KMAXM1,             KMAXM2,             NMAX,
     3    KEYOUT
      COMMON/PASS/            IPASS,              PDEPTH,
     1    PMOMAF,             PSALT,              PWIDTH
      COMMON/PULL/            SY(17),             SYY(17),
     1    TIME,               X(18),              Y(17),
     2    YK2,                Z(8)
      COMMON/PULLC/           ZC(20)
      COMMON/RIVERS/          JRIV(4),            NRIV,
```

```
1    PHASE1,                PHASE2,                RDEPTH(4),
2    RMOMAF(4),             RWIDTH(4),             UAVG,
3    UVAR1,                 UVAR2
 COMMON/RYTE/               IO,                    LABELI,
1    LABELO
 COMMON/STEP/               MN,                    MO,
1    NBAR,                  NWRITE,                ISTEP
 COMMON/TIDE/               PHSEMP,                PHSEPH,
1    TAMPMP,                TAMPPH,                TAVGMP,
2    TAVGPH,                THGTMP,                THGTPH
 COMMON/TUNE/               ACCEL,                 ARTVSC,
1    AVCOMP
 COMMON/TURB/               CDIF,                  CRICH,
1    CVIS,                  DIFFUS,                VISC
 COMMON/UNITS/              BETA,                  BETAD2,
1    FETCH,                 GRAV,                  HREF,
2    OMEGA,                 PI,                    TREF,
3    VREF,                  YMAX
 REWIND IO
 READ(IO) ACCEL, ARTVSC, BETA,   CDEPTH, CDIF,   CUTPT,  CVIS,
1 CWIDTH, DT,    DX,     DY,     DZ,     FETCH,  HREF,   IIMAX,
2 IMAX,   IPASS, JMAX,   JWAG,   KCUT,   KMAX,   KMXC,   DUM,
3 DUM,    NBAR,  NRIV,   NMAX,   NWRITE, PDEPTH, PERIDI, PMOMAF,
4 PTCTIN, PWIDTH, TREF,  VREF,   YMAX
 READ(IO) (EAST(I), SOUTH(I), WEST(I), I = 1, IIMAX)
 READ(IO) (JEAST(I), JWEST(I), I = 1, IMAX)
 READ(IO) ((KFLOOR(I,J), ZB(I,J), I = 1, IMAX), J = 1, JMAX)
 READ(IO) (KFLORC(I), ZBC(I), I = 1, IMAX)
 READ(IO) (((P(I,J,K),     I = 1, IMAX), J = 1, JMAX), K = 1, KMAX)
 READ(IO) (((S(MN,I,J,K), I = 1, IMAX), J = 1, JMAX), K = 1, KMAX)
 READ(IO) (((U(MN,I,J,K), I = 1, IMAX), J = 1, JMAX), K = 1, KMAX)
 READ(IO) (((V(MN,I,J,K), I = 1, IMAX), J = 1, JMAX), K = 1, KMAX)
 READ(IO) (((W(I,J,K),    I = 1, IMAX), J = 1, JMAX), K = 1, KMAX)
 READ(IO) ((DHDT(I,J),    I = 1, IMAX), J = 1, JMAX)
 READ(IO) (((DUDT(I,J,K), I = 1, IMAX), J = 1, JMAX), K = 1, KMAX)
 READ(IO) (((DVDT(I,J,K), I = 1, IMAX), J = 1, JMAX), K = 1, KMAX)
 READ(IO) ((PC(I,K), SC(MN,I,K), UC(MN,I,K), WC(I,K), I = 1, IMAX),
1 K = 1, KMXC)
 READ(IO) (DHCDT(I), (DUCDT(I,K), K = 1, KMXC), SURFC(I), I = 1,
1 IMAX)
 READ(IO) ((FWINDX(I,J), FWINDY(I,J), I = 1, IMAX), J = 1, JMAX)
 READ(IO) (SY(J), SYY(J), Y(J), J = 1, JMAX), (X(I), I = 1, IMAX),
1 (Z(K), K = 1, KMAX), (ZC(K), K = 1, KMXC)
 READ(IO) (JRIV(I), RDEPTH(I), RMOMAF(I), RWIDTH(I), I = 1, NRIV),
1 PHASE1, PHASE2, UAVG, UVAR1, UVAR2
 READ(IO) PHSEMP, PHSEPH, TAMPMP, TAMPPH, TAVGMP, TAVGPH
 ENTRY REED2
```

```
      READ(IO) (QDOT(L), QNET(L), QSDOT(L), QSNET(L), L = 1, 7), TIME
      READ(IO) SBAR, UBAR, VBAR
      READ(IO) ((SURF(MN,I,J), I = 1, IMAX), J = 1, JMAX)
      RETURN
C
CC       A ENTRY TO WRITE TO STORAGE DEVICE.
C
      ENTRY RITE
      IF(LABELI.EQ.LABELO) END FILE IO
      GO TO 10
      ENTRY RITE3
      REWIND IO
   10 CONTINUE
      WRITE(IO) ACCEL,  ARTVSC, BETA,   CDEPTH, CDIF,   CUTPT,  CVIS,
     1  CWIDTH, DT,     DX,     DY,     DZ,     FETCH,  HREF,   IIMAX,
     2  IMAX,   IPASS,  JMAX,   JWAG,   KCUT,   KMAX,   KMXC,   LABELI,
     3  LABELO, NBAR,   NRIV,   NMAX,   NWRITE, PDEPTH, PERIDI, PMOMAF,
     4  PTCTIN, PWIDTH, TREF,   VREF,   YMAX
      WRITE(IO) (EAST(I), SOUTH(I), WEST(I), I = 1, IIMAX)
      WRITE(IO) (JEAST(I), JWEST(I), I = 1, IMAX)
      WRITE(IO) ((KFLOOR(I,J), ZB(I,J), I = 1, IMAX), J = 1, JMAX)
      WRITE(IO) (KFLORC(I), ZBC(I), I = 1, IMAX)
      WRITE(IO) (((P(I,J,K),     I = 1, IMAX), J = 1, JMAX), K = 1, KMAX)
      WRITE(IO) (((S(MN,I,J,K), I = 1, IMAX), J = 1, JMAX), K = 1, KMAX)
      WRITE(IO) (((U(MN,I,J,K), I = 1, IMAX), J = 1, JMAX), K = 1, KMAX)
      WRITE(IO) (((V(MN,I,J,K), I = 1, IMAX), J = 1, JMAX), K = 1, KMAX)
      WRITE(IO) (((W(I,J,K),    I = 1, IMAX), J = 1, JMAX), K = 1, KMAX)
      WRITE(IO) ((DHDT(I,J),    I = 1, IMAX), J = 1, JMAX)
      WRITE(IO)(((DUDT(I,J,K),  I = 1, IMAX), J = 1, JMAX), K = 1, KMAX)
      WRITE(IO)(((DVDT(I,J,K),  I = 1, IMAX), J = 1, JMAX), K = 1, KMAX)
      WRITE(IO) ((PC(I,K), SC(MN,I,K), UC(MN,I,K), WC(I,K), I = 1,
     1 IMAX), K = 1, KMXC)
      WRITE(IO) (DHCDT(I), (DUCDT(I,K), K = 1, KMXC), SURFC(I), I = 1,
     1 IMAX)
      WRITE(IO) ((FWINDX(I,J), FWINDY(I,J), I = 1, IMAX), J = 1, JMAX)
      WRITE(IO) (SY(J), SYY(J), Y(J), J = 1, JMAX), (X(I), I = 1, IMAX),
     1 (Z(K), K = 1, KMAX), (ZC(K), K = 1, KMXC)
      WRITE(IO) (JRIV(I), RDEPTH(I), RMOMAF(I), RWIDTH(I), I = 1, NRIV),
     1 PHASE1, PHASE2, UAVG, UVAR1, UVAR2
      WRITE(IO) PHSEMP, PHSEPH, TAMPMP, TAMPPH, TAVGMP, TAVGPH
      ENTRY RITE2
      WRITE(IO) (QDOT(L), QNET(L), QSDOT(L), QSNET(L), L = 1, 7), TIME
      WRITE(IO) SBAR, UBAR, VBAR
      WRITE(IO) ((SURF(MN,I,J), I = 1, IMAX), J = 1, JMAX)
      RETURN
      END
```

```
      SUBROUTINE RUBBER
C
CC        GENERATE GRID SYSTEM.
C
      LOGICAL                    KEYOUT
      COMMON/CHNNEL/             CDEPTH,          CWIDTH,
     1    JWAG,                 KMXC,            KMXCM1,
     2    KMXCM2
      COMMON/FORTNO/             LR,              LT,
     1    LW
      COMMON/GRID/               DT,              DTD2,
     1    DX,                   DXINV,           DXT2IN,
     2    DXINSQ,               DY,              DYINV,
     3    DYT2IN,               DYINSQ,          DZ,
     4    DZD2,                 DZINV,           DZT2IN,
     5    DZINSQ
      COMMON/LIMITS/             IMAX,            IMAXM1,
     1    JMAX,                 JMAXM1,          KMAX,
     2    KMAXM1,               KMAXM2,          NMAX,
     3    KEYOUT
      COMMON/PULL/               SY(17),          SYY(17),
     1    TIME,                 X(18),           Y(17),
     2    YK2,                  Z(8)
      COMMON/PULLC/              ZC(20)
      COMMON/UNITS/              BETA,            BETAD2,
     1    FETCH,                GRAV,            HREF,
     2    OMEGA,                PI,              TREF,
     3    VREF,                 YMAX
C
      WRITE(LW,1000)
      DX     = FETCH / FLOAT(IMAX - 2)
      DXINV  = 1.0 / DX
      DXT2IN = 0.5 * DXINV
      DXINSQ = DXINV * DXINV
      X1     = - 1.5 * DX
      DO 10 I = 1, IMAX
      X2     = X1 + DX * FLOAT(I)
      X(I)   = X2
      XREAL  = X2 * HREF
   10 WRITE(LW,1001) I, X2, XREAL
      WRITE(LW,1002)
      DY     = 1.0 / FLOAT(JMAX - JWAG)
      DYINV  = 1.0 / DY
      DYT2IN = 0.5 * DYINV
      DYINSQ = DYINV * DYINV
      YK1    = YMAX / TAN(YK2)
      CONSQ  = YK1 * YK1
```

```
      DO 20 J = 1, JMAX
      CAPY   = DY * FLOAT(J - JWAG)
      Y1     = YK1 * TAN(YK2 * CAPY)
      Y(J)   = Y1
      YREAL  = Y1 * HREF
      YSQ    = Y1 * Y1
      DENOM  = CONSQ + YSQ
      SY(J)  = YK1 / (YK2 * DENOM)
      SYY(J) = -2.0 * YK1 * Y1 / (YK2 * DENOM * DENOM)
   20 WRITE(LW,1003) J, Y1, SY(J), SYY(J), YREAL
      WRITE(LW,1004)
      DZ     = 1.0 / FLOAT(KMAX)
      DZD2   = 0.5 * DZ
      DZINV  = 1.0 / DZ
      DZT2IN = 0.5 * DZINV
      DZINSQ = DZINV * DZINV
      Z1     = 0.0
      DO 30 K = 1, KMAX
      Z2     = Z1 + DZ * FLOAT(K)
      Z(K)   = Z2
      ZREAL  = Z2 * HREF
   30 WRITE(LW,1001) K, Z2, ZREAL
      WRITE(LW,1005)
      Z1     = 1.0 - DZ * FLOAT(KMXC)
      DO 40 K = 1, KMXC
      Z2     = Z1 + DZ * FLOAT(K)
      ZC(K)  = Z2
      ZREAL  = Z2 * HREF
   40 WRITE(LW,1001) K, Z2, ZREAL
      RETURN
C
 1000 FORMAT(//,13X,'I',18X,'X',15X,'XREAL(M)',/)
 1001 FORMAT(10X,I4,11X,1PE13.6,5X,E13.6)
 1002 FORMAT(//,13X,'J',18X,'Y',15X,'SY',15X,'SYY',15X,'YREAL(M)',/)
 1003 FORMAT(10X,I4,11X,1PE13.6,3(5X,E13.6))
 1004 FORMAT(//,13X,'K',18X,'Z',15X,'ZREAL(M)',/)
 1005 FORMAT(//,13X,'K',17X,'ZC',13X,'ZCREAL(M)',/)
      END
```

```
      SUBROUTINE SETUP
C
CCC
CCC      RETRIEVE VALUES FROM ARRAYS AND ASSIGN THEM TO SCALARS IN ORDER
CCC      TO CALCULATE NEW VALUES OF U, V, AND S.
C
      LOGICAL                 ISTEP,                  KEYOUT
      DOUBLE PRECISION        SURF(2,18,17),          SURFC(18)
      COMMON/BARS/            PTCTIN,                 SBAR,
     1    UBAR,              VBAR
      COMMON/BNKCRD/          EAST(22),               IIMAX,
     1    JE,                JEAST(22),              JW,
     2    JWEST(22),         SOUTH(22),              WEST(22)
      COMMON/CHNNEL/          CDEPTH,                 CWIDTH,
     1    JWAG,              KMXC,                   KMXCM1,
     2    KMXCM2
      COMMON/CONC/            SIM1,                   SIP1,
     1    SJM1,              SJP1,                   SKM1,
     2    SKP1,              SNEW,                   SOLD
      COMMON/FLOOR/           KFLOOR(18,17),          ZB(18,17)
      COMMON/FLOORC/          KFLORC(18),             ZBC(18)
      COMMON/FLOW1/           P(18,17,08),            S(2,18,17,08),
     1    U(2,18,17,08),     V(2,18,17,08),          W(18,17,08)
      COMMON/FLOW2/           SURF,                   DHDT(18,17),
     1    DUDT(18,17,08),    DVDT(18,17,08)
      COMMON/FLOWC1/          PC(18,20),              SC(2,18,20),
     1    UC(2,18,20),       WC(18,20)
      COMMON/FLOWC2/          SURFC,                  DHCDT(18),
     1    DUCDT(18,20)
      COMMON/FORCES/          F,                      FWINDX(18,17),
     1    FWINDY(18,17),     TOPLYR
      COMMON/GRID/            DT,                     DTD2,
     1    DX,                DXINV,                  DXT2IN,
     2    DXINSQ,            DY,                     DYINV,
     3    DYT2IN,            DYINSQ,                 DZ,
     4    DZD2,              DZINV,                  DZT2IN,
     5    DZINSQ
      COMMON/INDEX/           I,                      J,
     1    K,                 N
      COMMON/LIMITS/          IMAX,                   IMAXM1,
     1    JMAX,              JMAXM1,                 KMAX,
     2    KMAXM1,            KMAXM2,                 NMAX,
     3    KEYOUT
      COMMON/LN/              FUNLN1,                 FUNKAP
      COMMON/PASS/            IPASS,                  PDEPTH,
     1    PMOMAF,            PSALT,                  PWIDTH
      COMMON/PRES/            PIM1,                   PIP1,
     1    PJM1,              PJP1
```

```
      COMMON/PULL/          SY(17),         SYY(17),
     1    TIME,            X(18),          Y(17),
     2    YK2,             Z(8)
      COMMON/PULLC/         ZC(20)
      COMMON/RIVERS/        JRIV(4),        NRIV,
     1    PHASE1,          PHASE2,         RDEPTH(4),
     2    RMOMAF(4),       RWIDTH(4),      UAVG,
     3    UVAR1,           UVAR2
      COMMON/STEP/          MN,             MO,
     1    NBAR,            NWRITE,         ISTEP
      COMMON/STRTCH/        PREFIX,         SYJM1,
     1    SYJ,             SYJP1,          SY2
      COMMON/TIDE/          PHSEMP,         PHSEPH,
     1    TAMPMP,          TAMPPH,         TAVGMP,
     2    TAVGPH,          THGTMP,         THGTPH
      COMMON/UNITS/         BETA,           BETAD2,
     1    FETCH,           GRAV,           HREF,
     2    OMEGA,           PI,             TREF,
     3    VREF,            YMAX
      COMMON/VELCTY/        UIM1,           UIP1,
     1    UJM1,            UJP1,           UKM1,
     2    UKP1,            UNEW,           UOLD,
     3    VIM1,            VIP1,           VJM1,
     4    VJP1,            VKM1,           VKP1,
     5    VNEW,            VOLD,           WKM1,
     6    WKP1,            WNEW
C
      SBAR   = 0.0
      UBAR   = 0.0
      VBAR   = 0.0
      KDELTA = KMXC - KMAX
      FRAC   = CWIDTH * SY(JWAG) * DYINV
C
C         3-D CALCULATION FOR BAY PROPER.
C
      DO 290 I = 2, IMAX
      JW     = JWEST(I)
      JE     = JEAST(I)
      IP1    = I + 1
      IF(I.EQ.IMAX) IP1 = IMAX
      IM1    = I - 1
      JWM1   = JW - 1
      SYJ    = SY(JWM1)
      SYJP1  = SY(JW)
      KBOT   = KFLOOR(I,JWM1)
      KBOTJP = KFLOOR(I,JW)
      DO 290 J = JW, JE
```

```
      JP1     = J + 1
      JM1     = J - 1
      SY2     = SYY(J)
      KBOTIM  = KFLOOR(IM1,J)
      KBOTIP  = KFLOOR(IP1,J)
      KBOTJM  = KBOT
      KBOT    = KBOTJP
      K       = KBOT
      KBOTJP  = KFLOOR(I,JP1)
      KBOTP1  = KBOT + 1
      SYJM1   = SYJ
      SYJ     = SYJP1
      SYJP1   = SY(JP1)
      PREFIX  = SYJ * DYT2IN
      IF(KBOT.GT.KMAX) GO TO 290
      TOPLYR  = SURF(MN,I,J) + DZD2
      ZB1     = ZB(I,J)
      ALPHA   = 5.0 * (Z(KBOTP1) - ZB1) - 1.0
      FUNLN2  = ALPHA * (1.0 - ALPHA * (0.5 - ALPHA * (0.333333 -
     1          ALPHA * (0.25 - 0.2 * ALPHA))))
      FUNLN   = 1.0 - DZ / ((Z(KBOT) - ZB1 + DZD2) * (FUNLN1 + FUNLN2 +
     1          FUNKAP))
      SOLD    = S(MO,I,J,KBOT)
      UOLD    = U(MO,I,J,KBOT)
      VOLD    = V(MO,I,J,KBOT)
      WNEW    = W(I,J,KBOT)
      WKM1    = - WNEW
      SKM1    = SOLD
      IF(J.NE.JWAG) GO TO 5
      NTRFCC  = KFLOOR(I,JWAG) + KDELTA - 1
      WKM1    = WC(I,NTRFCC) * FRAC
      SKM1    = SC(MN,I,NTRFCC)
    5 CONTINUE
      SKP1    = S(MO,I,J,KBOTP1)
      UKP1    = U(MO,I,J,KBOTP1)
      VKP1    = V(MO,I,J,KBOTP1)
      WKP1    = W(I,J,KBOTP1)
      IF(KBOT.GE.KBOTJM) GO TO 10
      SJM1    = SOLD
      VJM1    = -VOLD
      IF(I.NE.IPASS.OR.J.NE.JW) GO TO 20
      SJM1    = PSALT
      VJM1    = V(MO,I,JM1,KBOT)
      GO TO 20
   10 SJM1    = S(MO,I,JM1,KBOT)
      VJM1    = V(MO,I,JM1,KBOT)
   20 IF(KBOT.GE.KBOTJP) GO TO 30
```

```
      SJP1    = SOLD
      VJP1    = -VOLD
      GO TO 40
   30 SJP1    = S(MO,I,JP1,KBOT)
      VJP1    = V(MO,I,JP1,KBOT)
   40 IF(KBOT.GE.KBOTIM) GO TO 50
      SIM1    = SOLD
      UIM1    = -UOLD
      GO TO 60
   50 SIM1    = S(MO,IM1,J,KBOT)
      UIM1    = U(MO,IM1,J,KBOT)
   60 IF(I.EQ.IMAX) GO TO 80
      IF(KBOT.GE.KBOTIP) GO TO 70
      SIP1    = SOLD
      UIP1    = -UOLD
      GO TO 90
   70 SIP1    = S(MO,IP1,J,KBOT)
      UIP1    = U(MO,IP1,J,KBOT)
      GO TO 90
   80 SIP1    = 1.0
      UIP1    = UOLD
   90 CALL BOTGRD
      S(MN,I,J,KBOT) = SNEW
      SBAR    = SBAR + (SNEW * SNEW - SOLD * SOLD)
      DO 190 K = KBOTP1, KMAXM1
      KP1     = K + 1
      SKM1    = SOLD
      SOLD    = SKP1
      SKP1    = S(MO,I,J,KP1)
      UKM1    = UOLD
      UOLD    = UKP1
      UKP1    = U(MO,I,J,KP1)
      VKM1    = VOLD
      VOLD    = VKP1
      VKP1    = V(MO,I,J,KP1)
      WKM1    = WNEW
      WNEW    = WKP1
      WKP1    = W(I,J,KP1)
      IF(K.GE.KBOTJM) GO TO 100
      SJM1    = SOLD
      UJM1    = UOLD
      VJM1    = -VOLD
      PJM1    = P(I,J,K)
      IF(I.NE.IPASS.OR.J.NE.JW) GO TO 110
      SJM1    = PSALT
      VJM1    = V(MO,I,JM1,K)
      PJM1    = THGTPH + BETA * PSALT * (THGTPH + 1.0 - Z(K))
```

```
      GO TO 110
  100 SJM1    = S(MO,I,JM1,K)
      UJM1    = U(MO,I,JM1,K)
      VJM1    = V(MO,I,JM1,K)
      PJM1    = P(I,JM1,K)
  110 IF(K.GE.KBOTJP) GO TO 120
      SJP1    = SOLD
      UJP1    = UOLD
      VJP1    = -VOLD
      PJP1    = P(I,J,K)
      GO TO 130
  120 SJP1    = S(MO,I,JP1,K)
      UJP1    = U(MO,I,JP1,K)
      VJP1    = V(MO,I,JP1,K)
      PJP1    = P(I,JP1,K)
  130 IF(K.GE.KBOTIM) GO TO 140
      SIM1    = SOLD
      UIM1    = -UOLD
      VIM1    = VOLD
      PIM1    = P(I,J,K)
      GO TO 150
  140 SIM1    = S(MO,IM1,J,K)
      UIM1    = U(MO,IM1,J,K)
      VIM1    = V(MO,IM1,J,K)
      PIM1    = P(IM1,J,K)
  150 IF(I.EQ.IMAX) GO TO 170
      IF(K.GE.KBOTIP) GO TO 160
      SIP1    = SOLD
      UIP1    = -UOLD
      VIP1    = VOLD
      PIP1    = P(I,J,K)
      GO TO 180
  160 SIP1    = S(MO,IP1,J,K)
      UIP1    = U(MO,IP1,J,K)
      VIP1    = V(MO,IP1,J,K)
      PIP1    = P(IP1,J,K)
      GO TO 180
  170 SIP1    = 1.0
      UIP1    = UOLD
      VIP1    = VOLD
      PIP1    = THGTMP + BETA * (THGTMP + 1.0 - Z(K))
  180 CALL UVSNEW
      S(MN,I,J,K) = SNEW
      SBAR    = SBAR + (SNEW * SNEW - SOLD * SOLD)
      U(MN,I,J,K) = UNEW
      UBAR    = UBAR + (UNEW * UNEW - UOLD * UOLD)
      V(MN,I,J,K) = VNEW
```

```
      VBAR    = VBAR + (VNEW * VNEW - VOLD * VOLD)
  190 CONTINUE
C
CC        INTERPOLATE LOGARITHMICALLY FOR VELOCITIES AT BOTTOM GRID
CC        POINT.
C
      U1        = U(MO,I,J,KBOT)
      U2        = U(MN,I,J,KBOTP1) * FUNLN
      U(MN,I,J,KBOT) = U2
      UBAR    = UBAR + (U2 * U2 - U1 * U1)
      U1      = V(MO,I,J,KBOT)
      U2      = V(MN,I,J,KBOTP1) * FUNLN
      V(MN,I,J,KBOT) = U2
      VBAR    = VBAR + (U2 * U2 - U1 * U1)
C
CC        EVALUATE NEW S, U, AND V AT FREE SURFACE.  VALUES OF S, U, V,
CC        AND W JUST ABOVE THE SURFACE ARE ASSUMED TO BE EQUAL TO THOSE
CC        AT THE SURFACE.
C
      K       = KMAX
      SKM1    = SOLD
      SOLD    = SKP1
      UKM1    = UOLD
      UOLD    = UKP1
      VKM1    = VOLD
      VOLD    = VKP1
      WKM1    = WNEW
      WNEW    = WKP1
      IF(KMAX.GE.KBOTJM) GO TO 200
      SJM1    = SOLD
      UJM1    = UOLD
      VJM1    = -VOLD
      PJM1    = P(I,J,KMAX)
      IF(I.NE.IPASS.OR.J.NE.JW) GO TO 210
      SJM1    = PSALT
      VJM1    = V(MO,I,JM1,KMAX)
      PJM1    = THGTPH * (1.0 + BETA * PSALT)
      GO TO 210
  200 SJM1    = S(MO,I,JM1,KMAX)
      UJM1    = U(MO,I,JM1,KMAX)
      VJM1    = V(MO,I,JM1,KMAX)
      PJM1    = P(I,JM1,KMAX)
  210 IF(KMAX.GE.KBOTJP) GO TO 220
      SJP1    = SOLD
      UJP1    = UOLD
      VJP1    = -VOLD
      PJP1    = P(I,J,KMAX)
```

```
      GO TO 230
  220 SJP1    = S(MO,I,JP1,KMAX)
      UJP1    = U(MO,I,JP1,KMAX)
      VJP1    = V(MO,I,JP1,KMAX)
      PJP1    = P(I,JP1,KMAX)
  230 IF(KMAX.GE.KBOTIM) GO TO 240
      SIM1    = SOLD
      UIM1    = -UOLD
      VIM1    = VOLD
      PIM1    = P(I,J,KMAX)
      GO TO 250
  240 SIM1    = S(MO,IM1,J,KMAX)
      UIM1    = U(MO,IM1,J,KMAX)
      VIM1    = V(MO,IM1,J,KMAX)
      PIM1    = P(IM1,J,KMAX)
  250 IF(I.EQ.IMAX) GO TO 270
      IF(KMAX.GE.KBOTIP) GO TO 260
      SIP1    = SOLD
      UIP1    = -UOLD
      VIP1    = VOLD
      PIP1    = P(I,J,KMAX)
      GO TO 280
  260 SIP1    = S(MO,IP1,J,KMAX)
      UIP1    = U(MO,IP1,J,KMAX)
      VIP1    = V(MO,IP1,J,KMAX)
      PIP1    = P(IP1,J,KMAX)
      GO TO 280
  270 SIP1    = 1.0
      UIP1    = UOLD
      VIP1    = VOLD
      PIP1    = THGTMP * (1.0 + BETA)
  280 CALL SURFAC
      S(MN,I,J,KMAX) = SNEW
      SBAR    = SBAR + (SNEW * SNEW - SOLD * SOLD)
      U(MN,I,J,KMAX) = UNEW
      UBAR    = UBAR + (UNEW * UNEW - UOLD * UOLD)
      V(MN,I,J,KMAX) = VNEW
      VBAR    = VBAR + (VNEW * VNEW - VOLD * VOLD)
  290 CONTINUE
C
C         2-D CALCULATION FOR SHIP CHANNEL.
C
      KBOT    = KFLORC(1)
      KBOTIP  = KFLORC(2)
      DO 460 I = 2, IMAX
      IP1     = I + 1
      IF(I.EQ.IMAX) IP1 = IMAX
```

```
      IM1     = I - 1
      KBOTIM  = KBOT
      KBOT    = KBOTIP
      K       = KBOT
      KBOTIP  = KFLORC(IP1)
      KBOTP1  = KBOT + 1
      NTRFCE  = KFLOOR(I,JWAG)
      NTRFCC  = NTRFCE + KDELTA - 1
      TOPLYR  = SURFC(I) + DZD2
      ZB1     = ZBC(I)
      ALPHA   = 5.0 * (ZC(KBOTP1) - ZB1) - 1.0
      FUNLN2  = ALPHA * (1.0 - ALPHA * (0.5 - ALPHA * (0.333333 -
     1          ALPHA * (0.25 - 0.2 * ALPHA))))
      FUNLN   = 1.0 - DZ / ((ZC(KBOT) - ZB1 + DZD2) * (FUNLN1 + FUNLN2 +
     1          FUNKAP))
      SOLD    = SC(MO,I,KBOT)
      UOLD    = UC(MO,I,KBOT)
      WNEW    = WC(I,KBOT)
      SKM1    = SOLD
      SKP1    = SC(MO,I,KBOTP1)
      UKP1    = UC(MO,I,KBOTP1)
      WKP1    = WC(I,KBOTP1)
      IF(KBOT.GE.KBOTIM) GO TO 300
      SIM1    = SOLD
      UIM1    = -UOLD
      GO TO 310
  300 SIM1    = SC(MO,IM1,KBOT)
      UIM1    = UC(MO,IM1,KBOT)
  310 IF(I.EQ.IMAX) GO TO 330
      IF(KBOT.GE.KBOTIP) GO TO 320
      SIP1    = SOLD
      UIP1    = -UOLD
      GO TO 340
  320 SIP1    = SC(MO,IP1,KBOT)
      UIP1    = UC(MO,IP1,KBOT)
      GO TO 340
  330 SIP1    = 1.0
      UIP1    = UOLD
  340 CALL BTGRDC
      SC(MN,I,KBOT) = SNEW
      SBAR    = SBAR + (SNEW * SNEW - SOLD * SOLD)
      DO 400 K = KBOTP1, NTRFCC
      KP1     = K + 1
      SKM1    = SOLD
      SOLD    = SKP1
      SKP1    = SC(MO,I,KP1)
      UKM1    = UOLD
```

```
      UOLD    = UKP1
      UKP1    = UC(MO,I,KP1)
      WKM1    = WNEW
      WNEW    = WKP1
      WKP1    = WC(I,KP1)
      IF(K.GE.KBOTIM) GO TO 350
      SIM1    = SOLD
      UIM1    = -UOLD
      PIM1    = PC(I,K)
      GO TO 360
  350 SIM1    = SC(MO,IM1,K)
      UIM1    = UC(MO,IM1,K)
      PIM1    = PC(IM1,K)
  360 IF(I.EQ.IMAX) GO TO 380
      IF(K.GE.KBOTIP) GO TO 370
      SIP1    = SOLD
      UIP1    = -UOLD
      PIP1    = PC(I,K)
      GO TO 390
  370 SIP1    = SC(MO,IP1,K)
      UIP1    = UC(MO,IP1,K)
      PIP1    = PC(IP1,K)
      GO TO 390
  380 SIP1    = 1.0
      UIP1    = UOLD
      PIP1    = THGTMP + BETA * (THGTMP + 1.0 - ZC(K))
  390 CALL USNEW
      SC(MN,I,K) = SNEW
      SBAR    = SBAR + (SNEW * SNEW - SOLD * SOLD)
      UC(MN,I,K) = UNEW
      UBAR    = UBAR + (UNEW * UNEW - UOLD * UOLD)
  400 CONTINUE
      U1      = UC(MO,I,KBOT)
      U2      = UC(MN,I,KBOTP1) * FUNLN
      UC(MN,I,KBOT) = U2
      UBAR    = UBAR + (U2 * U2 - U1 * U1)
      DO 460 K = NTRFCE, KMAX
      KC      = K + KDELTA
      UC(MN,I,KC) = U(MN,I,JWAG,K)
  450 SC(MN,I,KC) = S(MN,I,JWAG,K)
  460 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE UVSNEW
C
C        EVALUATE HORIZONTAL MOMENTUM AND SALT WATER TRANSPORT EQUATIONS
C        USING ROACHE'S SECOND WINDWARD DIFFERENCES FOR THE CONVECTIVE
C        TERMS.
C
      LOGICAL                 ISTEP
      DOUBLE PRECISION        SURF(2,18,17),      SURFC(18)
      COMMON/BNKCRD/          EAST(22),           IIMAX,
     1    JE,                 JEAST(22),          JW,
     2    JWEST(22),          SOUTH(22),          WEST(22)
      COMMON/CHNNEL/          CDEPTH,             CWIDTH,
     1    JWAG,               KMXC,               KMXCM1,
     2    KMXCM2
      COMMON/CONC/            SIM1,               SIP1,
     1    SJM1,               SJP1,               SKM1,
     2    SKP1,               SNEW,               SOLD
      COMMON/FLOW2/           SURF,               DHDT(18,17),
     1    DUDT(18,17,08),     DVDT(18,17,08)
      COMMON/FLOWC2/          SURFC,              DHCDT(18),
     1    DUCDT(18,20)
      COMMON/FORCES/          F,                  FWINDX(18,17),
     1    FWINDY(18,17),      TOPLYR
      COMMON/GRID/            DT,                 DTD2,
     1    DX,                 DXINV,              DXT2IN,
     2    DXINSQ,             DY,                 DYINV,
     3    DYT2IN,             DYINSQ,             DZ,
     4    DZD2,               DZINV,              DZT2IN,
     5    DZINSQ
      COMMON/INDEX/           I,                  J,
     1    K,                  N
      COMMON/PASS/            IPASS,              PDEPTH,
     1    PMOMAF,             PSALT,              PWIDTH
      COMMON/PRES/            PIM1,               PIP1,
     1    PJM1,               PJP1
      COMMON/RIVERS/          JRIV(4),            NRIV,
     1    PHASE1,             PHASE2,             RDEPTH(4),
     2    RMOMAF(4),          RWIDTH(4),          UAVG,
     3    UVAR1,              UVAR2
      COMMON/STEP/            MN,                 MO,
     1    NBAR,               NWRITE,             ISTEP
      COMMON/STRTCH/          PREFIX,             SYJM1,
     1    SYJ,                SYJP1,              SY2
      COMMON/TUNE/            ACCEL,              ARTVSC,
     1    AVCOMP
      COMMON/TURB/            CDIF,               CRICH,
     1    CVIS,               DIFFUS,             VISC
```

```
      COMMON/UNITS/          BETA,                BETAD2,
     1   FETCH,              GRAV,                HREF,
     2   OMEGA,              PI,                  TREF,
     3   VREF,               YMAX
      COMMON/VELCTY/         UIM1,                UIP1,
     1   UJM1,               UJP1,                UKM1,
     2   UKP1,               UNEW,                UOLD,
     3   VIM1,               VIP1,                VJM1,
     4   VJP1,               VKM1,                VKP1,
     5   VNEW,               VOLD,                WKM1,
     6   WKP1,               WNEW
C
C         3-D EQUATIONS FOR BAY.
C
      RHOINV = 1.0 - BETA * SOLD
      WINDX  = 0.0
      WINDY  = 0.0
      GO TO 10
C
C         SPECIAL ENTRY FOR SURFACE CALCULATIONS.
C
      ENTRY SURFAC
      RHOINV = 1.0 - BETA * SOLD
      COMP   = RHOINV / TOPLYR
      WINDX  = FWINDX(I,J) * COMP
      WINDY  = FWINDY(I,J) * COMP
   10 CONTINUE
      UF     = UIP1 + UOLD
      UB     = UOLD + UIM1
      IF(UF.GT.0.0) GO TO 20
      COMPU1 = UF * UIP1
      COMPV1 = UF * VIP1
      COMPS1 = UF * SIP1
      GO TO 30
   20 COMPU1 = UF * UOLD
      COMPV1 = UF * VOLD
      COMPS1 = UF * SOLD
   30 IF(UB.GT.0.0) GO TO 40
      COMPU1 = COMPU1 - UB * UOLD
      COMPV1 = COMPV1 - UB * VOLD
      COMPS1 = COMPS1 - UB * SOLD
      GO TO 90
   40 IF(I.NE.2) GO TO 70
      DO 50 NN = 1, NRIV
      IF(JRIV(NN).EQ.J) GO TO 60
   50 CONTINUE
      GO TO 70
```

C.S1

ORIGINAL PAGE IS
OF POOR QUALITY

```
 60 COMPU1 = COMPU1 - UB * UIM1 * RMOMAF(NN)
    GO TO 80
 70 COMPU1 = COMPU1 - UB * UIM1
 80 COMPV1 = COMPV1 - UB * VIM1
    COMPS1 = COMPS1 - UB * SIM1
 90 COMPU2 = RHOINV * (PIP1 - PIM1)
    COMPV2 = RHOINV * (PJP1 - PJM1)
    VF     = VJP1 + VOLD
    VB     = VOLD + VJM1
    IF(VF.GT.0.0) GO TO 100
    COMPU3 = VF * UJP1
    COMPV3 = VF * VJP1
    COMPS3 = VF * SJP1
    GO TO 110
100 COMPU3 = VF * UOLD
    COMPV3 = VF * VOLD
    COMPS3 = VF * SOLD
110 IF(VB.GT.0.0) GO TO 120
    COMPU3 = COMPU3 - VB * UOLD
    COMPV3 = COMPV3 - VB * VOLD
    COMPS3 = COMPS3 - VB * SOLD
    GO TO 150
120 COMPU3 = COMPU3 - VB * UJM1
    IF(I.NE.IPASS.OR.J.NE.JW) GO TO 130
    COMPV3 = COMPV3 - VB * VJM1 * PMOMAF
    GO TO 140
130 COMPV3 = COMPV3 - VB * VJM1
140 COMPS3 = COMPS3 - VB * SJM1
150 WF     = WKP1 + WNEW
    WB     = WNEW + WKM1
    IF(WF.GT.0.0) GO TO 160
    COMPU4 = WF * UKP1
    COMPV4 = WF * VKP1
    COMPS4 = WF * SKP1
    GO TO 170
160 COMPU4 = WF * UOLD
    COMPV4 = WF * VOLD
    COMPS4 = WF * SOLD
170 IF(WB.GT.0.0) GO TO 180
    COMPU4 = COMPU4 - WB * UOLD
    COMPV4 = COMPV4 - WB * VOLD
    COMPS4 = COMPS4 - WB * SOLD
    GO TO 190
180 COMPU4 = COMPU4 - WB * UKM1
    COMPV4 = COMPV4 - WB * VKM1
    COMPS4 = COMPS4 - WB * SKM1
190 CONTINUE
```

```
C
C          NOW SOLVING THE X-MOMENTUM EQUATION.
C
      CALL VISCUS
C
C          DU/DT = -(D(U*U)/DX + D(U*V)/DY + D(U*W)/DZ + (1.0/RHO)*DP/DX)
C                  + COEFVZ*(DSQU/DZSQ) + F*VOLD + FWINDX/(RHO*DEPTH)
C
      DUODT  = - DXT2IN * (COMPU1 + COMPU2) - PREFIX * COMPU3 -
     1           DZT2IN * COMPU4 + VISC + F * VOLD + WINDX
      UNEW   = UOLD + (DUODT + DUDT(I,J,K)) * DTD2
      IF(ISTEP) DUDT(I,J,K) = DUODT
C
C          NOW SOLVING THE Y-MOMENTUM EQUATION.
C
      CALL VVISC
C
C          DV/DT = -(D(U*V)/DX + D(V*V)/DY + D(V*W)/DZ + (1.0/RHO)*DP/DY)
C                  + COEFVZ*(DSQU/DZSQ) - F*UOLD + FWINDY/(RHO*DEPTH)
C
      DVODT  = - DXT2IN * COMPV1 - PREFIX * (COMPV2 + COMPV3) -
     1           DZT2IN * COMPV4 + VISC - F * UOLD + WINDY
      VNEW   = VOLD + (DVODT + DVDT(I,J,K)) * DTD2
      IF(ISTEP) DVDT(I,J,K) = DVODT
C
C          NOW SOLVING THE SALT CONSERVATION EQUATION.
C
      CALL DIFUSE
C
C          DS/DT = -(D(U*S)/DX + D(V*S)/DY + D(W*S)/DZ)
C                  + COEFDX*(DSQS/DXSQ) + COEFDY*(DSQS/DYSQ)
C                  + COEFDZ*(DSQS/DZSQ)
C
  200 DSODT  = - DXT2IN * COMPS1 - PREFIX * COMPS3 - DZT2IN * COMPS4
     1         + DIFFUS
      SNEW   = SOLD + DSODT * DT * ACCEL
      IF(SNEW.LT.0.0) SNEW = 0.0
      IF(SNEW.GT.1.0) SNEW = 1.0
      RETURN
C
C          SPECIAL ENTRY FOR SALINITY CALCULATION AT BOTTOM GRID POINT.
C
      ENTRY BOTGRD
      UF     = UIP1 + UOLD
      UB     = UOLD + UIM1
      IF(UF.GT.0.0) GO TO 210
      COMPS1 = UF * SIP1
```

```
      GO TO 220
  210 COMPS1 = UF * SOLD
  220 IF(UB.GT.0.0) GO TO 230
      COMPS1 = COMPS1 - UB * SOLD
      GO TO 240
  230 COMPS1 = COMPS1 - UB * SIM1
  240 VF     = VJP1 + VOLD
      VB     = VOLD + VJM1
      IF(VF.GT.0.0) GO TO 250
      COMPS3 = VF * SJP1
      GO TO 260
  250 COMPS3 = VF * SOLD
  260 IF(VB.GT.0.0) GO TO 270
      COMPS3 = COMPS3 - VB * SOLD
      GO TO 280
  270 COMPS3 = COMPS3 - VB * SJM1
  280 WF     = WKP1 + WNEW
      WB     = WNEW + WKM1
      IF(WF.GT.0.0) GO TO 290
      COMPS4 = WF * SKP1
      GO TO 300
  290 COMPS4 = WF * SOLD
  300 IF(WB.GT.0.0) GO TO 302
      COMPS4 = COMPS4 - WB * SOLD
      GO TO 304
  302 COMPS4 = COMPS4 - WB * SKM1
  304 CONTINUE
      SKM1   = SOLD
      CALL BTDIF
      GO TO 200
C
CC        2-D EQUATIONS FOR SHIP CHANNEL.
C
      ENTRY USNEW
      RHOINV = 1.0 - BETA * SOLD
      WINDX  = 0.0
      GO TO 310
C
      ENTRY SRFACC
      RHOINV = 1.0 - BETA * SOLD
      WINDX  = FWINDX(I,JWAG) * RHOINV / TOPLYR
  310 CONTINUE
      UF     = UIP1 + UOLD
      UB     = UOLD + UIM1
      IF(UF.GT.0.0) GO TO 320
      COMPU1 = UF * UIP1
      COMPS1 = UF * SIP1
```

```
      GO TO 330
  320 COMPU1 = UF * UOLD
      COMPS1 = UF * SOLD
  330 IF(UB.GT.0.0) GO TO 340
      COMPU1 = COMPU1 - UB * UOLD
      COMPS1 = COMPS1 - UB * SOLD
      GO TO 350
  340 COMPU1 = COMPU1 - UB * UIM1
      COMPS1 = COMPS1 - UB * SIM1
  350 COMPU2 = RHOINV * (PIP1 - PIM1)
      WF     = WKP1 + WNEW
      WB     = WNEW + WKM1
      IF(WF.GT.0.0) GO TO 360
      COMPU4 = WF * UKP1
      COMPS4 = WF * SKP1
      GO TO 370
  360 COMPU4 = WF * UOLD
      COMPS4 = WF * SOLD
  370 IF(WB.GT.0.0) GO TO 380
      COMPU4 = COMPU4 - WB * UOLD
      COMPS4 = COMPS4 - WB * SOLD
      GO TO 390
  380 COMPU4 = COMPU4 - WB * UKM1
      COMPS4 = COMPS4 - WB * SKM1
  390 CONTINUE
C
      CALL VISCSC
      DUODT  = - DXT2IN * (COMPU1 + COMPU2) - DZT2IN * COMPU4
     1         + VISC + WINDX
      UNEW   = UOLD + (DUODT + DUCDT(I,K)) * DTD2
      IF(ISTEP) DUCDT(I,K) = DUODT
C
      CALL DIFUSC
  400 DSODT  = - DXT2IN * COMPS1 - DZT2IN * COMPS4 + DIFFUS
      SNEW   = SOLD + DSODT * DT * ACCEL
      IF(SNEW.LT.0.0) SNEW = 0.0
      IF(SNEW.GT.1.0) SNEW = 1.0
      RETURN
C
      ENTRY BTGRDC
      UF     = UIP1 + UOLD
      UB     = UOLD + UIM1
      IF(UF.GT.0.0) GO TO 410
      COMPS1 = UF * SIP1
      GO TO 420
  410 COMPS1 = UF * SOLD
  420 IF(UB.GT.0.0) GO TO 430
```

```
      COMPS1 = COMPS1 - UB * SOLD
      GO TO 440
  430 COMPS1 = COMPS1 - UB * SIM1
  440 WF     = WKP1 + WNEW
      IF(WF.GT.0.0) GO TO 450
      COMPS4 = WF * SKP1
      GO TO 460
  450 COMPS4 = WF * SOLD
  460 CONTINUE
      CALL BTDIFC
      GO TO 400
      END
```

```
      SUBROUTINE VISCUS
C
CC        CALCULATE TURBULENT SHEAR STRESS AND DIFFUSION TERMS.
C
      COMMON/CONC/              SIM1,                   SIP1,
     1    SJM1,                SJP1,                   SKM1,
     2    SKP1,                SNEW,                   SOLD
      COMMON/GRID/              DT,                     DTD2,
     1    DX,                  DXINV,                  DXT2IN,
     2    DXINSQ,              DY,                     DYINV,
     3    DYT2IN,              DYINSQ,                 DZ,
     4    DZD2,                DZINV,                  DZT2IN,
     5    DZINSQ
      COMMON/STRTCH/            PREFIX,                 SYJM1,
     1    SYJ,                 SYJP1,                  SY2
      COMMON/TURB/              CDIF,                   CRICH,
     1    CVIS,                DIFFUS,                 VISC
      COMMON/VELCTY/            UIM1,                   UIP1,
     1    UJM1,                UJP1,                   UKM1,
     2    UKP1,                UNEW,                   UOLD,
     3    VIM1,                VIP1,                   VJM1,
     4    VJP1,                VKM1,                   VKP1,
     5    VNEW,                VOLD,                   WKM1,
     6    WKP1,                WNEW
C
CC        3-D EQUATIONS FOR BAY.
C
      CVIS1  = CVIS
      UDEL   = UKP1 - UKM1
      VDEL   = VKP1 - VKM1
      DQ     = SQRT(UDEL * UDEL + VDEL * VDEL)
   10 RICH   = CRICH * (SKP1 - SKM1) / (DQ * DQ + 1.0E-6)
      IF(RICH.LT.0.0) RICH = 0.0
      DAMPV  = 1.0 / SQRT(1.0 + 10.0 * RICH)
      ZCOMP  = CVIS1 * (UKP1 - UOLD - UOLD + UKM1) * DZINSQ
      GO TO 20
      ENTRY VVISC
      ZCOMP  = CVIS1 * (VKP1 - VOLD - VOLD + VKM1) * DZINSQ
   20 VISC   = DAMPV * ZCOMP
      RETURN
C
      ENTRY BTDIF
      CDIF1  = CDIF
      DAMPD  = 1.0
      GO TO 30
C
      ENTRY DIFUSE
```

```
      DAMPD   = 1.0 / (1.0 + 3.333 * RICH)**1.5
   30 TWOS    = SOLD + SOLD
      YCOMP1  = SY2 * DYT2IN * (SJP1 - SJM1)
      YCOMP2  = 4.0 * PREFIX * PREFIX * (SJP1 - TWOS + SJM1)
   40 XCOMP   = (SIP1 - TWOS + SIM1) * DXINSQ
      ZCOMP   = CDIF1 * (SKP1 - TWOS + SKM1) * DZINSQ
      DIFFUS  = CDIF1 * (XCOMP + YCOMP1 + YCOMP2) + DAMPD * ZCOMP
      RETURN
C
CCC      2-D EQUATIONS FOR SHIP CHANNEL.
C
      ENTRY VISCSC
      DQ      = ABS(UKP1 - UKM1)
      CVIS1   = 2.0 * CVIS
      GO TO 10
C
      ENTRY BTDIFC
      CDIF1   = 2.0 * CDIF
      DAMPD   = 1.0
      GO TO 50
C
      ENTRY DIFUSC
      DAMPD   = 1.0 / (1.0 + 3.333 * RICH)**1.5
   50 TWOS    = SOLD + SOLD
      YCOMP1  = 0.0
      YCOMP2  = 0.0
      GO TO 40
      END
```

```
      SUBROUTINE VOLDIL
C
CCC
CCC      CALCULATE NEW VERTICAL VELOCITIES AND SURFACE HEIGHT LOCATIONS
CCC      USING THE VOLUME DILATATION EQUATION.
C
      LOGICAL                 ISTEP,                  KEYOUT
      DOUBLE PRECISION        SURF(2,18,17),          SURFC(18)
      DOUBLE PRECISION        SURFAV,                 SURFIM,
     1    SURFIP,              SURFJM,                 SURFJP,
     2    SURF1,               SURF2,                  TWOSRF
      COMMON/BNKCRD/          EAST(22),               IIMAX,
     1    JE,                  JEAST(22),              JW,
     2    JWEST(22),           SOUTH(22),              WEST(22)
      COMMON/CHNNEL/          CDEPTH,                 CWIDTH,
     1    JWAG,                KMXC,                   KMXCM1,
     2    KMXCM2
      COMMON/FLOOR/           KFLOOR(18,17),          ZB(18,17)
      COMMON/FLOORC/          KFLORC(18),             ZBC(18)
      COMMON/FLOW1/           P(18,17,08),            S(2,18,17,08),
     1    U(2,18,17,08),       V(2,18,17,08),          W(18,17,08)
      COMMON/FLOW2/           SURF,                   DHDT(18,17),
     1    DUDT(18,17,08),      DVDT(18,17,08)
      COMMON/FLOWC1/          PC(18,20),              SC(2,18,20),
     1    UC(2,18,20),         WC(18,20)
      COMMON/FLOWC2/          SURFC,                  DHCDT(18),
     1    DUCDT(18,20)
      COMMON/FORTNO/          LR,                     LT,
     1    LW
      COMMON/GRID/            DT,                     DTD2,
     1    DX,                  DXINV,                  DXT2IN,
     2    DXINSQ,              DY,                     DYINV,
     3    DYT2IN,              DYINSQ,                 DZ,
     4    DZD2,                DZINV,                  DZT2IN,
     5    DZINSQ
      COMMON/GULF/            CUTPT,                  KCUT,
     1    PERIOD
      COMMON/INDEX/           I,                      J,
     1    K,                   N
      COMMON/LIMITS/          IMAX,                   IMAXM1,
     1    JMAX,                JMAXM1,                 KMAX,
     2    KMAXM1,              KMAXM2,                 NMAX,
     3    KEYOUT
      COMMON/PASS/            IPASS,                  PDEPTH,
     1    PMOMAF,              PSALT,                  PWIDTH
      COMMON/PULL/            SY(17),                 SYY(17),
     1    TIME,                X(18),                  Y(17),
     2    YK2,                 Z(8)
```

```
      COMMON/PULLC/          ZC(20)
      COMMON/RIVERS/         JRIV(4),              NRIV,
     1   PHASE1,             PHASE2,               RDEPTH(4),
     2   RMOMAF(4),          RWIDTH(4),            UAVG,
     3   UVAR1,              UVAR2
      COMMON/STEP/           MN,                   MO,
     1   NBAR,               NWRITE,               ISTEP
      COMMON/TIDE/           PHSEMP,               PHSEPH,
     1   TAMPMP,             TAMPPH,               TAVGMP,
     2   TAVGPH,             THGTMP,               THGTPH
      COMMON/TUNE/           ACCEL,                ARTVSC,
     1   AVCOMP
      KDELTA = KMXC - KMAX
      TEST   = -1.5 * DZ
      FRAC   = CWIDTH * SY(JWAG) * DYINV
C
CC       2-D EQUATIONS FOR SHIP CHANNEL.
C
      KBOT   = KFLORC(1)
      KBOTIP = KBOT
      DO 430 I = 1, IMAX
      IM1    = I - 1
      IF(I.EQ.1) IM1 = 1
      IP1    = I + 1
      IF(I.EQ.IMAX) IP1 = IMAX
      KBOTIM = KBOT
      KBOT   = KBOTIP
      KBOTIP = KFLORC(IP1)
      KBOTP1 = KBOT + 1
      NTRFCE = KFLOOR(I,JWAG)
      NTRFCC = NTRFCE + KDELTA - 1
      IF(KBOT.GE.KBOTIM) GO TO 300
      UIM1   = - UC(MN,I,KBOT)
      GO TO 310
  300 UIM1   = UC(MN,IM1,KBOT)
  310 IF(KBOT.GE.KBOTIP) GO TO 320
      UIP1   = - UC(MN,I,KBOT)
      GO TO 330
  320 UIP1   = UC(MN,IP1,KBOT)
  330 CONTINUE
      WNEW   = 0.25 * (UIM1 - UIP1) * DXINV * (ZC(KBOT) - ZBC(I))
      WC(I,KBOT) = WNEW
      DO 380 K = KBOTP1, NTRFCC
      KM1    = K - 1
      UIMKM1 = UIM1
      IF(K.GE.KBOTIM) GO TO 340
      UIM1   = - UC(MN,I,K)
```

```
      GO TO 350
  340 UIM1    = UC(MN,IM1,K)
  350 UIPKM1  = UIP1
      IF(K.GE.KBOTIP) GO TO 360
      UIP1    = - UC(MN,I,K)
      GO TO 370
  360 UIP1    = UC(MN,IP1,K)
  370 CONTINUE
      UIN     = UIM1 + UIMKM1
      UOUT    = UIP1 + UIPKM1
      WNEW    = WNEW + 0.25 * (UIN - UOUT) * DXINV * DZ
      WC(I,K) = WNEW
  380 CONTINUE
  430 CONTINUE
C
C        3-D EQUATIONS FOR BAY.
C
      DO 280 I = 1, IMAX
      IM1     = I - 1
      IF(I.EQ.1) IM1 = 1
      IP1     = I + 1
      IF(I.EQ.IMAX) IP1 = IMAX
      JW      = JWEST(I)
      JE      = JEAST(I)
      SURF1   = SURF(MO,I,JW)
      SURFJP  = SURF1
      KBOT    = KMAX + 1
      KBOTJP  = KFLOOR(I,JW)
      Y1      = Y(JW-1)
      YJP1    = Y(JW)
      DELY2   = YJP1 - Y1
      DO 280 J = JW, JE
      JP1     = J + 1
      JM1     = J - 1
      YJM1    = Y1
      Y1      = YJP1
      YJP1    = Y(JP1)
      DELY    = 0.5 * (YJP1 - YJM1)
      DELYIN  = 1.0 / DELY
      DELY1   = DELY2
      DELY2   = YJP1 - Y1
      KBOTJM  = KBOT
      KBOT    = KBOTJP
      KBOTJP  = KFLOOR(I,JP1)
      SURFJM  = SURF1
      SURF1   = SURFJP
      SURFJP  = SURF(MO,I,JP1)
```

```
      IF(KBOTJP.GT.KMAX) SURFJP = SURF1
      IF(KBOT.GT.KMAX) GO TO 280
      SURFIM = SURF(MO,IM1,J)
      IF(I.EQ.IMAX) GO TO 10
      SURFIP = SURF(MO,IP1,J)
      GO TO 20
   10 SURFIP = THGTMP
   20 KBOTIM = KFLOOR(IM1,J)
      KBOTIP = KFLOOR(IP1,J)
      KBOTP1 = KBOT + 1
      IF(KBOTIM.GT.KMAX) SURFIM = SURF1
      IF(KBOTIP.GT.KMAX) SURFIP = SURF1
      IF(KBOT.GE.KBOTIM) GO TO 30
      UIM1   = - U(MN,I,J,KBOT)
      GO TO 40
   30 UIM1   = U(MN,IM1,J,KBOT)
   40 IF(KBOT.GE.KBOTIP) GO TO 50
      UIP1   = - U(MN,I,J,KBOT)
      GO TO 60
   50 UIP1   = U(MN,IP1,J,KBOT)
   60 IF(KBOT.GE.KBOTJM) GO TO 70
      VJM1   = - V(MN,I,J,KBOT)
      IF(I.EQ.IPASS.AND.J.EQ.JW) VJM1 = V(MO,I,JM1,KBOT)
      GO TO 80
   70 VJM1   = V(MN,I,JM1,KBOT)
   80 IF(KBOT.GE.KBOTJP) GO TO 90
      VJP1   = - V(MN,I,J,KBOT)
      GO TO 100
   90 VJP1   = V(MN,I,JP1,KBOT)
  100 CONTINUE
      WNEW   = 0.25 * ((UIM1 - UIP1) * DXINV + (VJM1 - VJP1) * DELYIN) *
     1 (Z(KBOT) - ZB(I,J))
      W(I,J,KBOT) = WNEW
      DO 190 K = KBOTP1, KMAX
      KM1    = K - 1
      UIMKM1 = UIM1
      IF(K.GE.KBOTIM) GO TO 110
      UIM1   = - U(MN,I,J,K)
      GO TO 120
  110 UIM1   = U(MN,IM1,J,K)
  120 UIPKM1 = UIP1
      IF(K.GE.KBOTIP) GO TO 130
      UIP1   = - U(MN,I,J,K)
      GO TO 140
  130 UIP1   = U(MN,IP1,J,K)
  140 VJMKM1 = VJM1
      IF(K.GE.KBOTJM) GO TO 150
```

```
      VJM1    = - V(MN,I,J,K)
      IF(I.EQ.IPASS.AND.J.EQ.JW) VJM1 = V(MO,I,JM1,K)
      GO TO 160
  150 VJM1    = V(MN,I,JM1,K)
  160 VJPKM1 = VJP1
      IF(K.GE.KBOTJP) GO TO 170
      VJP1    = - V(MN,I,J,K)
      GO TO 180
  170 VJP1    = V(MN,I,JP1,K)
  180 CONTINUE
      UIN     = UIM1 + UINKM1
      UOUT    = UIP1 + UIPKM1
      VIN     = VJM1 + VJNKM1
      VOUT    = VJP1 + VJPKM1
      WNEW    = WNEW + 0.25 * ((UIN - UOUT) * DXINV +
     1          (VIN - VOUT) * DELYIN) * DZ
  190 W(I,J,K) = WNEW
      IF(J.NE.JWAG) GO TO 210
      WCHN    = WC(I,NTRFCC)
      WBAY    = FRAC * WCHN
      DO 200 K = KBOT, KMAX
  200 W(I,JWAG,K) = W(I,JWAG,K) + WBAY
  210 CONTINUE
C
CCC       CALCULATE FREE SURFACE LOCATION.
C
      UMID    = U(MN,I,J,KMAX) + U(MN,I,J,KMAXM1)
      VMID    = V(MN,I,J,KMAX) + V(MN,I,J,KMAXM1)
      UIN     = (UIN + UMID) * (0.5 * (SURFIM + SURF1) + DZ)
      UOUT    = (UOUT + UMID) * (0.5 * (SURFIP + SURF1) + DZ)
      VIN     = (VIN + VMID) * (0.5 * (SURFJM + SURF1) + DZ)
      VOUT    = (VOUT + VMID) * (0.5 * (SURFJP + SURF1) + DZ)
      TWOSRF = SURF1 + SURF1
      COMPH1 = (SURFIP - TWOSRF + SURFIM) * DXINSQ
      PREFIX = SY(J) * DYINV
      COMPH2 = (SURFJP - TWOSRF + SURFJM) * PREFIX * PREFIX +
     1          (SURFJP - SURFJM) * DYT2IN * SYY(J)
      DHODT   = W(I,J,KMAXM1) + 0.25 * ((UIN - UOUT) * DXINV +
     1          (VIN - VOUT) * DELYIN) + ARTVSC * (COMPH1 + COMPH2)
      SURF2   = SURF1 + (DHODT + DHDT(I,J)) * DTD2
      DHDT(I,J) = DHODT
      IF(SURF2.LT.TEST) GO TO 290
      SURF(MN,I,J) = SURF2
      GO TO 470
  290 WRITE(LW,1000) N, I, J
      KEYOUT = .TRUE.
      RETURN
```

```
  470 CONTINUE
C
      IF(J.NE.JWAG) GO TO 280
      DO 390 K = KBOT, KMAX
  390 WC(I,K+KDELTA) = WCHN + W(I,JWAG,K)
      SURF2  = SURF(MN,I,JWAG) + WCHN * DT
      IF(SURF2.LT.TEST) GO TO 440
      SURFC(I) = SURF2
  280 CONTINUE
      GO TO 450
  440 WRITE(LW,1001) N, I
      KEYOUT = .TRUE.
  450 RETURN
 1000 FORMAT(5X,'SURFACE TOO ROUGH AT N = ',I5,5X,' I = ',I3,5X,' J = ',
     I I3)
 1001 FORMAT(5X,'SURFACE TOO ROUGH AT N = ',I5,5X,' I = ',I3)
      END
```

```
      BLOCK DATA
C
C     VARIOUS CONSTANTS ARE DEFINED.
C
      LOGICAL                   ISTEP
      COMMON/FORTNO/            LR,                     LT,
     1    LW
      COMMON/INDEX/             I,                      J,
     1    K,                    N
      COMMON/STEP/              MN,                     MO,
     1    NBAR,                 NWRITE,                 ISTEP
      COMMON/UNITS/             BETA,                   BETAD2,
     1    FETCH,                GRAV,                   HREF,
     2    OMEGA,                PI,                     TREF,
     3    VREF,                 YMAX
C
C     LW, LR, LT ARE THE LOGICAL I/O UNIT NUMBERS FOR WRITING ON LINE
C     PRINTER, READING FROM CARD READER, USING TAPE, RESPECTIVELY.
C
      DATA LW,LR,LT/6,5,3/
      DATA MN,MO,N/2,1,0/
      DATA GRAV,OMEGA,PI/9.790,7.29E-5,3.14159/
      END
```

```
DATA
          0          0          2
         18         17          8         60       6000        375
          0.        60.    50.5860         5.        25.       30.7         30.
   26.3130
          0.     1.7127       39.5       35.9       33.2       33.2
     0.0015     0.0015       10.0        0.4
          7         20
       12.3      122.0
        0.0        0.0        0.0        0.0        0.0        0.0
        1.0     2400.         1.0
          4
          7.         9.      235.6
          8.         9.      316.6
         11.         6.      335.5
         12.         8.      312.4
     0.9996     3.8256     0.4971    -0.4378    -0.0796
         15.        1.5        0.7     3000.
         22
        0.0      -0.10       6.00
        1.3      -1.35       5.90
        2.6      -1.45       5.85
        3.9      -1.40       6.20
        5.2      -1.65       6.00
        6.5      -2.50       6.35
        7.8      -3.00       6.40
        9.1      -3.15       5.90
       10.4      -4.20       5.40
       11.7      -4.35       4.25
       13.0      -4.40       5.00
       14.3      -4.80       5.20
       15.6      -4.60       5.40
       16.9      -4.55       6.50
       18.2      -4.45       8.25
       19.5      -4.85       9.80
       20.8      -5.40      10.90
       22.1      -5.35      13.10
       23.4      -5.00      14.00
       24.7      -4.40      14.20
       26.0      -3.80      11.20
       27.3      -3.20       0.50
 .5090404E0 .7724584E-4-.1369818E-3-.2940399E-7 .3376983E-7 .6457060E-7
.179078E-11-.223053E-11 .460191E-11-.164306E-10
-.1459126E1          0.E0          0.E0          0.E0
          1          0          4
         18         17          8         60       3000        375
          0.        60.    50.5860         5.        25.       30.7         30.
```

26.3130					
0.	1.7127	39.5	35.9	33.2	33.2
0.0015	0.0015	10.0	0.4		
7	20				
12.3	122.0				
120.	200.	25.0	37.0	474.	608.
1.0	2400.	1.0			
4					
7.	9.	235.6			
8.	9.	316.6			
11.	6.	335.5			
12.	8.	312.4			
0.9996	3.8256	0.2721	-0.4378	-0.0796	
15.	1.5	0.7	3000.		

APPENDIX C

VARIAN PLOT PROGRAM

The Varian plot program generates plots of velocity vectors, salinity numbers and water level contours in cross-sections corresponding to the grid planes in the Mobile Bay model program. The plot program has been written specifically for the output of the Mobile Bay program. In addition, it requires the Varian plot package currently available on Louisiana State University Computer Research Center's IBM 360.

The Varian plot program determines what cross-sections to plot at a given time level for which data are stored on tape on the basis of the values of control variables read from punched card input. These variables are defined with comment statements preceding the READ statements in which they appear, and are listed in Table C.1 along with their input format for easy reference. These READ statements are located in the MAIN program. The model data are read from tape by a subroutine entitled READ, scaled by PREP and plotted by Varian subroutines invoked by the MAIN program. Subroutines DODAD1 and DODAD2 generate reference scales and XYBRD1, XZBRD1 and YZBRD1 generate the bay boundaries corresponding to the xy- xz- and yz-cross-sections. The printed output from the program is nominal.

A listing of the FORTRAN IV statements for the Varian plot program and a typical set of input data follow.

TABLE C.1

Input Variables for Varian Plot Program

Name	Format
NPLOT1	I10
NPLOT2	I10
NXY	I10
NXZ	I10
NYZ	I10
KSECTN(I)	I10
JSECTN(I)	I10
ISECTN(I)	I10

```
C
C       THIS PROGRAM PLOTS VELOCITY, SALINITY, AND SURFACE HEIGHT DATA
C       FOR VARIOUS CROSS-SECTIONS OF MOBILE BAY.
C
      LOGICAL                 ISTEP,                  KEYOUT
      DIMENSION               ISECTN(5),              JSECTN(5),
     1    KSECTN(5)
      DIMENSION               LCASWY(4),              LCHANL(4),
     1    LELA(4),            LEWS(3),                LFRE(3),
     2    LFRGU(9),           LGUL(3),                LHEI(3),
     3    LMMS(2),            LNSS(3),                LPLO(3),
     4    LSAL(4),            LSUR(4),                LTID(4),
     5    LVEL(4),            LWAT(3),                LIMPS(2)
      DOUBLE PRECISION        SURF(18,17),            SURFC(18)
      COMMON/ACCT/            QDOT(7),                QNET(7),
     1    QSDOT(7),           QSNET(7)
      COMMON/BARS/            PTCTIN,                 SBAR,
     1    UBAR,               VBAR
      COMMON/BNKCRD/          EAST(22),               IIMAX,
     1    JE,                 JEAST(22),              JW,
     2    JWEST(22),          SOUTH(22),              WEST(22)
      COMMON/CHNNEL/          CDEPTH,                 CWIDTH,
     1    JWAG,               KMXC,                   KMXCM1,
     2    KMXCM2
      COMMON/FLOOR/           KFLOOR(18,17),          ZB(18,17)
      COMMON/FLOORC/          KFLORC(18),             ZBC(18)
      COMMON/FLOW1/           P(18,17,08),            S(18,17,08),
     1    U(18,17,08),        V(18,17,08),            W(18,17,08)
      COMMON/FLOW2/           SURF,                   DHDT(18,17),
     1    DUDT(18,17,08),     DVDT(18,17,08)
      COMMON/FLOWC1/          PC(18,20),              SC(18,20),
     1    UC(18,20),          WC(18,20)
      COMMON/FLOWC2/          SURFC,                  DHCDT(18),
     1    DUCDT(18,20)
      COMMON/FORCES/          F,                      FWINDX(18,17),
     1    FWINDY(18,17),      TOPLYR
      COMMON/FORTNO/          LR,                     LT,
     1    LW
      COMMON/GRID/            DT,                     DTD2,
     1    DX,                 DXINV,                  DXT2IN,
     2    DXINSQ,             DY,                     DYINV,
     3    DYT2IN,             DYINSQ,                 DZ,
     4    DZD2,               DZINV,                  DZT2IN,
     5    DZINSQ
      COMMON/GULF/            CUTPT,                  KCUT,
     1    PERIDI
      COMMON/INDEX/           I,                      J,
```

```
1    K,                    N
 COMMON/LIMITS/            IMAX,                 IMAXM1,
1    JMAX,                 JMAXM1,               KMAX,
2    KMAXM1,               KMAXM2,               NMAX,
3    KEYOUT
 COMMON/PASS/              IPASS,                PDEPTH,
1    PMOMAF,               PSALT,                PWIDTH
 COMMON/PLOTT/             DXD2,                 L,
1    NPLOT1,               NPLOT2,               PERIOD,
2    ZMAX,                 ZOHSCL
 COMMON/PULL/              SY(17),               SYY(17),
1    TIME,                 X(18),                Y(17),
2    YK2,                  Z(8)
 COMMON/PULLC/             ZC(20)
 COMMON/RIVERS/            JRIV(4),              NRIV,
1    PHASE1,               PHASE2,               RDEPTH(4),
2    RMOMAF(4),            RWIDTH(4),            UAVG,
3    UVAR1,                UVAR2
 COMMON/RYTE/              IO,                   LABELI,
1    LABELO
 COMMON/SCALE/             HSCALE,               SSCALE,
1    USCALE,               WSCALE,               XSCALE,
2    ZSCALE
 COMMON/STEP/              MN,                   MO,
1    NBAR,                 NWRITE,               ISTEP
 COMMON/TIDE/              PHSEMP,               PHSEPH,
1    TAMPMP,               TAMPPH,               TAVGMP,
2    TAVGPH,               THGTMP,               THGTPH
 COMMON/TUNE/              ACCEL,                ARTVSC,
1    AVCOMP
 COMMON/TURB/              CDIF,                 CRICH,
1    CVIS,                 DIFFUS,               VISC
 COMMON/UNITS/             BETA,                 BETAD2,
1    FETCH,                GRAV,                 HREF,
2    OMEGA,                PI,                   TREF,
3    VREF,                 YMAX
 DATA LCASWY/' KM ','WRT ','CAUS','EWAY'/
 DATA LCHANE/' KM ','WRT ','CHAN','NEL '/
 DATA LELA   /'ELAP','SED ','TIME','0   '/
 DATA LEWS   /'E-W ','SECT','ION '/
 DATA LFRE   /'FRES','H   ','0   '/
 DATA LFRGU /'FRES','H WA','TER ','= 0 ','   G','ULF ','WATE',
1      'R = ','10  '/
 DATA LGUL   /'GULF','   1','0   '/
 DATA LHEI   /'HEIG','HT, ','CM  '/
 DATA LHR    /' HR '/
 DATA LMMS   /' M M','SL  '/
```

```
      DATA LNSS  /'N-S ','SECT','ION '/
      DATA LPLO  /'PLOT',' ELE','V0  '/
      DATA LSAL  /'SALI','NITY',' PRO','FILE'/
      DATA LSUR  /'SURF','ACE ','PROF','ILE '/
      DATA LTID  /'TIDE',' PER','IODO','    '/
      DATA LVEL  /'VELO','CITY',' VEC','TORS'/
      DATA LWAT  /'WATE','R SC','ALE '/
      DATA LIMPS /'1.M/','S   '/
      CALL IDENT('1303 88532 FHP')
      IO     = LT
      REWIND IO
      DGORD  = 180.0 / PI
      PID2   = 0.5 * PI
   10 CALL PLOT(  5.00, 12.0,-3)
C
CCC     *NPLOT1, NPLOT2 ARE THE FIRST AND LAST FRAME NUMBERS FOR THE
CCC      SEQUENCE OF FRAMES TO BE PLOTTED.  A FRAME IS A COMPLETE SET
CCC      OF DATA DESCRIBING THE FLOW FIELD AT A GIVEN INSTANT IN TIME.
CCC      THE FRAME NUMBER MULTIPLIED BY NWRITE EQUALS THE TIME INDEX N
C        IN THE MOBILE BAY PROGRAM.
C
      READ(LR,1000) NPLOT1, NPLOT2
      IF(NPLOT1.EQ.-1) STOP
C
CCC     *NXY, NXZ, NYZ ARE THE NUMBER OF CROSS-SECTIONS IN THE XY-, XZ-,
CCC      AND YZ-PLANES, RESPECTIVELY, TO BE PLOTTED FOR EACH FRAME.
C
      READ(LR,1000) NXY, NXZ, NYZ
C
CCC     *KSECTN, JSECTN, ISECTN ARE THE SPECIFIC VALUES OF THE SPATIAL
CCC      INDICES FOR EACH SECTION IN THE XY-, XZ-, AND YZ-PLANES,
CCC      RESPECTIVELY.
C
      READ(LR,1000) (KSECTN(L), L = 1, NXY)
      READ(LR,1000) (JSECTN(L), L = 1, NXZ)
      READ(LR,1000) (ISECTN(L), L = 1, NYZ)
C
CCC     GENERATE PLOTS FOR ABOVE SEQUENCE OF FRAMES.
C
      DO 350 L = NPLOT1, NPLOT2
C
CCC     READ DATA FOR GIVEN FRAME AND SCALE THEM.
C
      CALL PREP
C
CCC     PLOT VELOCITY VECTORS IN XY-PLANE CROSS-SECTIONS.
C
```

```
      IF(NXY.EQ.0) GO TO 165
      DO  90 M = 1, NXY
      K       = KSECTN(M)
      CALL XYBRDI
      CALL PLOT(  2.25, -1.00,-3)
      CALL DODAD2
      CALL PLOT( -2.25,  1.00,-3)
      CALL VTHICK(1)
      CALL PLOT(  2.50, -4.00,-3)
      VECTOR = USCALE / HREF
      CALL PLOT(  0.00,VECTOR, 2)
      CALL SYMBOL(  0.00,VECTOR,0.07,     2,  0.0,-1)
      CALL SYMBOL(  0.20,VECTOR*0.5-0.07,0.14, LIMPS,  0.0, 5)
      CALL PLOT( -2.50,  4.00,-3)
      CALL PLOT(  0.30, -7.25,-3)
      CALL SYMBOL(  0.70,  0.00,0.14,  LVEL,  0.0,10)
      CALL SYMBOL(  0.00, -0.25,0.14,  LPLO,  0.0,11)
      ELEV    = HREF * (1.0 - Z(K) / ZSCALE)
      CALL NUMBER(  1.54, -0.25,0.14,  ELEV,  0.0, 2)
      CALL SYMBOL(  2.24, -0.25,0.14,  LMMS,  0.0, 6)
      CALL SYMBOL(  0.00, -0.50,0.14,  LTID,  0.0,13)
      CALL NUMBER(  1.82, -0.50,0.14,PERICD,  0.0, 2)
      CALC SYMBOL(  2.52, -0.50,0.14,   LHR,  0.0, 3)
      CALL SYMBOL(  0.00, -0.75,0.14,  LELA,  0.0,14)
      CALL NUMBER(  1.96, -0.75,0.14,  TIME,  0.0, 2)
      CALL SYMBOL(  2.80, -0.75,0.14,   LHR,  0.0, 3)
      CALL PLOT( -0.30,  7.25,-3)
C
CC       FOR THIS PLOT Y IS ALONG THE PAPER ROLL AND X IS ACROSS THE
CC       PAPER ROLL.
C
      DO  80 I = 1, IMAX
      XSTRT   = X(I)
      IF(I.NE.1) GO TO  40
      DO  30 NN = 1, NRIV
      J       = JRIV(NN)
      YSTRT   = Y(J)
      IF(K.GE.KFLOOR(I,J)) GO TO  20
      IF(J.EQ.JWAG) GO TO  15
      CALL SYMBOL( YSTRT, XSTRT,0.07,     6,  0.0,-1)
      GO TO 30
   15 KC      = KMXC - KMAX + K
      UC1     = UC(I,KC)
      XEND    = XSTRT - UC1
      THETA   = 0.0
      IF(UC1.GT.0.0) THETA = 180.0
      CALL PLOT( YSTRT, XSTRT, 3)
```

```
      CALL SYMBOL( YSTRT,  XEND,0.07,      2,THETA,-2)
      GO TO  30
   20 U1      = U(I,J,K) + 0.000001
      V1      = V(I,J,K)
      XEND    = XSTRT - U1
      YEND    = YSTRT + V1
      THETA   = (ATAN2(-U1,V1) - PID2) * DGORD
      CALL PLOT( YSTRT, XSTRT, 3)
      CALL SYMBOL(  YEND,  XEND,0.07,      2,THETA,-2)
   30 CONTINUE
      GO TO  80
   40 JW      = JWEST(I)
      JE      = JEAST(I)
      DO  70 J = JW, JE
      YSTRT   = Y(J)
      IF(K.GE.KFLOOR(I,J)) GO TO  60
      IF(J.EQ.JWAG) GO TO  50
      CALL SYMBOL( YSTRT, XSTRT,0.07,      3,  0.0,-1)
      GO TO  70
   50 KC      = KMXC - KMAX + K
      UC1     = UC(I,KC)
      XEND    = XSTRT - UC1
      THETA   = 0.0
      IF(UC1.GT.0.0) THETA = 180.0
      CALL PLOT( YSTRT, XSTRT, 3)
      CALL SYMBOL( YSTRT,  XEND,0.07,      2,THETA,-2)
      GO TO  70
   60 U1      = U(I,J,K) + 0.000001
      V1      = V(I,J,K)
      XEND    = XSTRT - U1
      YEND    = YSTRT + V1
      THETA   = (ATAN2(-U1,V1) - PID2) * DGORD
      CALL PLOT( YSTRT, XSTRT, 3)
      CALL SYMBOL(  YEND,  XEND,0.07,      2,THETA,-2)
   70 CONTINUE
   80 CONTINUE
      CALL EOPLOT(0,1)
      CALL PLOT(  0.00,  0.00,-3)
   90 CONTINUE
C
CCC      PLOT SALINITIES IN XY-PLANE CROSS-SECTIONS.
C
      DO 140 M = 1, NXY
      K       = KSECTN(M)
      CALL XYBRD1
      CALL PLOT(  2.25, -1.00,-3)
      CALL DOOAD2
```

```
      CALL PLOT( -2.25,  1.00,-3)
      CALL VTHICK(1)
      CALL PLOT(  1.50, -3.00,-3)
      CALL SYMBOL(  0.00,  0.00,0.14,  LWAT,  0.0,11)
      CALL SYMBOL(  0.00, -0.25,0.14,  LFRE,  0.0, 9)
      CALL SYMBOL(  0.00, -0.50,0.14,  LGUL,  0.0,10)
      CALL PLOT( -1.50,  3.00,-3)
      CALL PLOT(  0.30, -7.25,-3)
      CALL SYMBOL(  0.70,  0.00,0.14,  LSAL,  0.0,16)
      CALL SYMBOL(  0.00, -0.25,0.14,  LPLO,  0.0,11)
      ELEV   = HREF * (1.0 - Z(K) / ZSCALE).
      CALL NUMBER(  1.54, -0.25,0.14,  ELEV,  0.0, 2)
      CALL SYMBOL(  2.24, -0.25,0.14,  LMMS,  0.0, 6)
      CALL SYMBOL(  0.00, -0.50,0.14,  LTID,  0.0,13)
      CALL NUMBER(  1.82, -0.50,0.14,PERIOD,  0.0, 2)
      CALL SYMBOL(  2.52, -0.50,0.14,   LHR,  0.0, 3)
      CALL SYMBOL(  0.00, -0.75,0.14,  LELA,  0.0,14)
      CALL NUMBER(  1.96, -0.75,0.14,  TIME,  0.0, 2)
      CALL SYMBOL(  2.80, -0.75,0.14,   LHR,  0.0, 3)
      CALL PLOT( -0.30,  7.25,-3)
      DO 130 I = 1, IMAX
      XSTRT  = X(I) - 0.035
      IF(I.NE.1) GO TO 110
      DO 100 NN = 1, NRIV
      J      = JRIV(NN)
      YSTRT  = Y(J) - 0.035
      IF(K.LT.KFLOOR(I,J).AND.J.EQ.JWAG) GO TO  95
      CALL NUMBER( YSTRT, XSTRT,0.07,S(I,J,K),  0.0,-1)
      GO TO 100
   95 CALL NUMBER( YSTRT, XSTRT,0.07,SC(I,KC),  0.0,-1)
  100 CONTINUE
      GO TO 130
  110 JW     = JWEST(I)
      JE     = JEAST(I)
      DO 120 J = JW, JE
      YSTRT  = Y(J) - 0.035
      IF(K.LT.KFLOOR(I,J).AND.J.EQ.JWAG) GO TO 115
      CALL NUMBER( YSTRT, XSTRT, 0.07,S(I,J,K),  0.0,-1)
      GO TO 120
  115 CALL NUMBER( YSTRT, XSTRT,0.07,SC(I,KC),  0.0,-1)
  120 CONTINUE
  130 CONTINUE
      CALL FOPLOT(0,1)
      CALL PLOT(  0.00,  0.00,-3)
  140 CONTINUE
C
C        PLOT SURFACE HEIGHT PROFILES.
```

```
      CALL XYBRD1
      CALL PLOT(  2.25, -1.00,-3)
      CALL DODAD2
      CALL PLOT( -2.25,  1.00,-3)
      CALL VTHICK(1)
      CALL PLOT(  2.00, -3.25,-3)
      VECTOR = HSCALE / HREF
      CALL SYMBOL(  0.00,-VECTOR,0.10,     13, 90.0,-1)
      CALL SYMBOL(  0.00,  0.00,0.10,     13, 90.0,-2)
      CALL SYMBOL(  0.00,VECTOR,0.10,     13, 90.0,-2)
      XSTRT  = 0.10
      YSTRT  = - VECTOR - 0.035
      CALL NUMBER( XSTRT, YSTRT,0.07,-100.0,  0.0, 0)
      YSTRT  = YSTRT + VECTOR
      CALL NUMBER( XSTRT, YSTRT,0.07,   0.0,  0.0, 0)
      YSTRT  = YSTRT + VECTOR
      CALL NUMBER( XSTRT, YSTRT,0.07, 100.0,  0.0, 0)
      CALL SYMBOL(  0.70, -0.70,0.14,  LHEI, 90.0,10)
      CALL PLOT( -2.00,  3.25,-3)
      CALL PLOT(  0.30, -7.25,-3)
      CALL SYMBOL(  0.70,  0.00,0.14,  LSUR,  0.0,15)
      CALL SYMBOL(  0.00, -0.25,0.14,  LTID,  0.0,13)
      CALL NUMBER(  1.82, -0.25,0.14,PERIOD,  0.0, 2)
      CALL SYMBOL(  2.53, -0.25,0.14,   LHR,  0.0, 3)
      CALL SYMBOL(  0.00, -0.50,0.14,  LELA,  0.0,14)
      CALL NUMBER(  1.96, -0.50,0.14,  TIME,  0.0, 2)
      CALL SYMBOL(  2.80, -0.50,0.14,   LHR,  0.0, 3)
      CALL PLOT( -0.30,  7.25,-3)
      DO 160 I = 2, IMAX
      JW     = JWEST(I)
      JE     = JEAST(I)
      X1     = X(I)
      YW     = Y(JW)
      YE     = Y(JE)
      CALL VTHICK(1)
      CALL PLOT(    YE,    X1, 3)
      CALL PLOT(    YW,    X1, 2)
      X2     = X1 + SURF(I,JW)
      CALL PLOT(    YW,    X2, 2)
      CALL VTHICK(-3)
      DO 150 J = JW, JE
      X2     = X1 + SURF(I,J)
      Y2     = Y(J)
      CALL PLOT(    Y2,    X2, 2)
  150 CONTINUE
      CALL VTHICK(1)
```

```
      CALL PLOT(     YE,     X1, 2)
  160 CONTINUE
      CALL EOPLOT(0,1)
  165 CALL PLOT(  0.00, -5.00,-3)
C
CCC       PLOT VELOCITY VECTORS IN XZ-PLANE CROSS-SECTIONS.
C
      IF(NXZ.EQ.0) GO TO 285
      DO 220 M = 1, NXZ
      J        = JSECTN(M)
      CALL PLOT(  2.00,  1.80,-3)
      CALL DODAD1
      CALL PLOT( -2.00, -1.80,-3)
      CALL PLOT(  7.20, -1.50,-3)
      CALL DODAD2
      CALL PLOT( -7.20,  1.50,-3)
      CALL VTHICK(1)
      CALL PLOT(  0.00, -2.70,-3)
      CALL SYMBOL(  2.10,  0.00,0.14,  LVEL,  0.0,16)
      CALL SYMBOL(  0.91, -0.25,0.14,  LNSS,  0.0,12)
      OFFSET = 0.001 * HREF * Y(J) / XSCALE
      CALL NUMBER(  2.59, -0.25,0.14,OFFSET,  0.0, 2)
      CALL SYMBOL(  3.43, -0.25,0.14,LCHANL,  0.0,15)
      CALL SYMBOL(  0.00, -0.50,0.14,  LTID,  0.0,13)
      CALL NUMBER(  1.82, -0.50,0.14,PERIOD,  0.0, 2)
      CALL SYMBOL(  2.52, -0.50,0.14,   LHR,  0.0, 3)
      CALL SYMBOL(  3.29, -0.50,0.14,  LELA,  0.0,14)
      CALL NUMBER(  5.25, -0.50,0.14,  TIME,  0.0, 2)
      CALL SYMBOL(  6.09, -0.50,0.14,   LHR,  0.0, 3)
      CALL PLOT(  0.00,  2.70,-3)
      IF(J.EQ.JWAG) GO TO 190
      CALL XZGRD1
      CALL VTHICK(1)
C
CCC       FOR THIS PLOT X IS ALONG THE PAPER ROLL AND Z IS ACROSS THE
CCC       PAPER ROLL.
C
      DO 180 I = 1, IMAX
      KBOT   = KFLOOR(I,J)
      IF(KBOT.GT.KMAX) GO TO 180
      XSTRT  = - X(I)
      DO 170 K = KBOT, KMAX
      ZSTRT  = Z(K)
      U1     = U(I,J,K) + 0.000001
      W1     = W(I,J,K)
      XEND   = XSTRT + U1
      ZEND   = ZSTRT + W1
```

```
      THETA   = (ATAN2(W1,U1) - PID2) * DGORD
      CALL PLOT( XSTRT, ZSTRT, 3)
      CALL SYMBOL(  XEND,  ZEND,0.07,     2,THETA,-2)
  170 CONTINUE
  180 CONTINUE
      GO TO 210
  190 CALL CHBRD1
      CALL VTHICK(1)
      DO 200 I = 1, IMAX
      KBOT    = KFLORC(I)
      XSTRT   = - X(I)
      DO 200 K = KBOT, KMXC
      ZSTRT   = ZC(K)
      UC1     = UC(I,K) + 0.000001
      WC1     = WC(I,K)
      XEND    = XSTRT + UC1
      ZEND    = ZSTRT + WC1
      THETA   = (ATAN2(WC1,UC1) - PID2) * DGORD
      CALL PLOT( XSTRT, ZSTRT, 3)
      CALL SYMBOL(  XEND,  ZEND,0.07,     2,THETA,-2)
  200 CONTINUE
  210 CALL EOPLOT(0,1)
      CALL PLOT(  0.00,  0.00,-3)
  220 CONTINUE
C
C        PLOT SALINITIES IN XZ-PLANE CROSS-SECTIONS.
C
      DO 280 M = 1, NXZ
      J       = JSECTN(M)
      CALL PLOT(  7.20, -1.50,-3)
      CALL DOUBAD2
      CALL PLOT( -7.20,  1.50,-3)
      CALL VTHICK(1)
      CALL SYMBOL(  0.70,  1.80,0.07, LFRGU,  0.0,34)
      CALL PLOT(  0.00, -2.70,-3)
      CALL SYMBOL(  2.10,  0.00,0.14,  LSAL,  0.0,16)
      CALL SYMBOL(  0.91, -0.25,0.14,  LNSS,  0.0,12)
      OFFSET = 0.001 * HREF * Y(J) / XSCALE
      CALL NUMBER(  2.59, -0.25,0.14,OFFSET,  0.0, 2)
      CALL SYMBOL(  3.43, -0.25,0.14,LCHANL,  0.0,15)
      CALL SYMBOL(  0.00, -0.50,0.14,  LTID,  0.0,13)
      CALL NUMBER(  1.82, -0.50,0.14,PERIOD,  0.0, 2)
      CALL SYMBOL(  2.52, -0.50,0.14,   LHR,  0.0, 3)
      CALL SYMBOL(  3.29, -0.50,0.14,  LELA,  0.0,14)
      CALL NUMBER(  5.25, -0.50,0.14,  TIME,  0.0, 2)
      CALL SYMBOL(  6.09, -0.50,0.14,   LHR,  0.0, 3)
      CALL PLOT(  0.00,  2.70,-3)
```

```
      IF(J.EQ.JWAG) GO TO 250
      CALL XZBRD1
      CALL VTHICK(1)
      DO 240 I = 1, IMAX
      KBOT   = KFLOOR(I,J)
      IF(KBOT.GT.KMAX) GO TO 240
      XSTRT  = - X(I) - 0.035
      DO 230 K = KBOT, KMAX
      ZSTRT  = Z(K) - 0.035
      CALL NUMBER( XSTRT, ZSTRT,0.07,S(I,J,K),  0.0,-1)
  230 CONTINUE
  240 CONTINUE
      GO TO 270
  250 CALL CHORD1
      CALL VTHICK(1)
      DO 260 I = 1, IMAX
      KBOT   = KFLORC(I)
      XSTRT  = - X(I) - 0.035
      DO 260 K = KBOT, KMXC
      ZSTRT  = ZC(K) - 0.035
      CALL NUMBER( XSTRT, ZSTRT,0.07,SC(I,K),  0.0,-1)
  260 CONTINUE
  270 CALL EOPLOT(0,1)
      CALL PLOT(  0.00,  0.00,-3)
  280 CONTINUE
C
C        PLOT VELOCITY VECTORS IN YZ-PLANE CROSS-SECTIONS.
C
  285 IF(NYZ.EQ.0) GO TO 345
      DO 310 M = 1, NYZ
      I      = ISECTN(M)
      CALL PLOT(  0.00,  1.80,-3)
      CALL DODAD1
      CALL PLOT(  0.00, -1.80,-3)
      CALL PLOT(  4.00,  1.00,-3)
      CALL DODAD2
      CALL PLOT( -4.00, -1.00,-3)
      CALL VTHICK(1)
      CALL PLOT( -2.20, -2.60,-3)
      CALL SYMBOL(  2.10,  0.00,0.14,  LVEL,  0.0,16)
      CALL SYMBOL(  0.91, -0.25,0.14,  LEWS,  0.0,12)
      OFFSET = - 0.001 * HREF * X(I) / XSCALE
      CALL NUMBER(  2.59, -0.25,0.14,OFFSET,  0.0, 2)
      CALL SYMBOL(  3.43, -0.25,0.14,LCASWY,  0.0,16)
      CALL SYMBOL(  0.00, -0.50,0.14,  LTID,  0.0,13)
      CALL NUMBER(  1.82, -0.50,0.14,PERIOD,  0.0, 2)
      CALL SYMBOL(  2.52, -0.50,0.14,   LHP,  0.0, 3)
```

```
      CALL SYMBOL(  3.29, -0.50,0.14,  LELA,  0.0,14)
      CALL NUMBER(  5.25, -0.50,0.14,  TIME,  0.0, 2)
      CALL SYMBOL(  6.09, -0.50,0.14,   LHR,  0.0, 3)
      CALL PLOT(  2.20,  2.60,-3)
      JW      = JWEST(I)
      JE      = JEAST(I)
      CALL YZBRD1
      CALL VTHICK(1)
C
CC       FOR THIS PLOT Y IS ALONG THE PAPER ROLL AND Z IS ACROSS THE
CC       PAPER ROLL.
C
      DO 300 J = JW, JE
      KBOT    = KFLOOR(I,J)
      IF(KBOT.GT.KMAX) GO TO 300
      YSTRT   = Y(J)
      DO 290 K = KBOT, KMAX
      ZSTRT   = Z(K)
      V1      = V(I,J,K) + 0.000001
      W1      = W(I,J,K)
      YEND    = YSTRT + V1
      ZEND    = ZSTRT + W1
      THETA   = (ATAN2(W1,V1) - PID2) * DGORD
      CALL PLOT( YSTRT, ZSTRT, 3)
      CALL SYMBOL(  YEND,  ZEND,0.07,     2,THETA,-2)
  290 CONTINUE
  300 CONTINUE
      CALL EOPLOT(0,1)
      CALL PLOT(  0.00,  0.00,-3)
  310 CONTINUE
C
CC       PLOT SALINITIES IN YZ-PLANE CROSS-SECTIONS.
C
      DO 340 M = 1, NYZ
      I       = ISECTN(M)
      CALL VTHICK(1)
      CALL SYMBOL(  0.00,  1.80,0.07, LFRGU,  0.0,34)
      CALL PLOT(  4.00,  1.00,-3)
      CALL DODAD2
      CALL PLOT( -4.00, -1.00,-3)
      CALL PLOT( -2.20, -2.60,-3)
      CALL SYMBOL(  2.10,  0.00,0.14,  LSAL,  0.0,16)
      CALL SYMBOL(  0.91, -0.25,0.14,  LEWS,  0.0,12)
      OFFSET = - 0.001 * HREF * X(I) / XSCALE
      CALL NUMBER(  2.59, -0.25,0.14,OFFSET,  0.0, 2)
      CALL SYMBOL(  3.43, -0.25,0.14,LCASWY,  0.0,16)
      CALL SYMBOL(  0.00, -0.50,0.14,  LTID,  0.0,13)
```

```
      CALL NUMBER(   1.82, -0.50,0.14,PERIOD,   0.0, 2)
      CALL SYMBOL(   2.52, -0.50,0.14,   LHR,   0.0, 3)
      CALL SYMBOL(   3.29, -0.50,0.14,  LELA,   0.0,14)
      CALL NUMBER(   5.29, -0.50,0.14,  TIME,   0.0, 2)
      CALL SYMBOL(   6.09, -0.50,0.14,   LHR,   0.0, 3)
      CALL PLOT(  2.20,  2.60,-3)
      JW      = JWEST(I)
      JE      = JEAST(I)
      CALL YZBRD1
      CALL VTHICK(1)
      DO 330 J = JW, JE
      KBOT    = KFLOOR(I,J)
      IF(KBOT.GT.KMAX) GO TO 330
      YSTRT   = Y(J) - 0.035
      DO 320 K = KBOT, KMAX
      ZSTRT   = Z(K) - 0.035
      CALL NUMBER( YSTRT, ZSTRT,0.07,S(I,J,K),  0.0,-1)
  320 CONTINUE
  330 CONTINUE
      CALL EOPLOT(0,1)
      CALL PLOT(  0.00,  0.00,-3)
  340 CONTINUE
  345 CALL PLOT(  0.00,  5.00,-3)
  350 CONTINUE
C
C        PUT ORIGIN IN BOTTOM LEFT HAND CORNER OF VARIAN FRAME.
C
      CALL PLOT( -5.00,-12.00,-3)
      GO TO  10
 1000 FORMAT(8I10)
      END
```

```
      SUBROUTINE DODAD1
      DIMENSION                 L1MMPS(2),          L1MPS(2)
      COMMON/SCALE/             HSCALE,             SSCALE,
     1   USCALE,                WSCALE,             XSCALE,
     2   ZSCALE
      COMMON/UNITS/             BETA,               BETAD2,
     1   FETCH,                 GRAV,               HREF,
     2   OMEGA,                 PI,                 TREF,
     3   VREF,                  YMAX
      DATA L1MMPS/'1.MM','/S  '/
      DATA L1MPS /'1.M/','S   '/
C
C        GENERATE VELOCITY SCALES FOR VERTICAL CROSS-SECTIONS.
C
      CALL VTHICK(1)
      VECTOR = USCALE / HREF
      CALL SYMBOL(VECTOR,  0.00,0.07,      2,-90.0,-2)
      CALL SYMBOL(VECTOR+0.07,-0.035,0.07, L1MPS,  0.0, 5)
      CALL PLOT(  0.00,  0.00, 3)
      VECTOR = WSCALE / (HREF * 1000.0)
      CALL SYMBOL(  0.00,VECTOR,0.07,      2,  0.0,-2)
      CALL SYMBOL( -0.49,VECTOR*0.5-0.035,0.07,L1MMPS,  0.0, 6)
      RETURN
      END
```

```
      SUBROUTINE DODAD2
      DIMENSION                 LST(2)
      COMMON/PLOTT/             DXD2,                   L,
     1    NPLOT1,                NPLOT2,                 PERIOD,
     2    ZMAX,                  ZOHSCL
      COMMON/PULL/              SY(17),                 SYY(17),
     1    TIME,                  X(18),                  Y(17),
     2    YK2,                   Z(8)
      COMMON/TIDE/              PHSEMP,                 PHSEPH,
     1    TAMPMP,                TAMPPH,                 TAVGMP,
     2    TAVGPH,                THGTMP,                 THGTPH
      COMMON/TUNE/              ACCEL,                  ARTVSC,
     1    AVCOMP
      COMMON/UNITS/             BETA,                   BETAD2,
     1    FETCH,                 GRAV,                   HREF,
     2    OMEGA,                 PI,                     TREF,
     3    VREF,                  YMAX
      DATA LGU    /'GULF'/
      DATA LHI    /'HI   '/
      DATA LLO    /'LO.  '/
      DATA LMN    /'MN   '/
      DATA LST    /'STAG','E    '/
      DATA LTI    /'TIDE'/

C
C     GENERATE GULF TIDE GAGE.
C
      CALL VTHICK(1)
      CALL SYMBOL(  0.00,-TAMPMP,0.10,     13, 90.0,-1)
      CALL SYMBOL(  0.00,  0.00,0.10,      13, 90.0,-2)
      CALL SYMBOL(  0.00,TAMPMP,0.10,      13, 90.0,-2)
      XSTRT  = -0.24
      YSTRT  = - TAMPMP - 0.035
      CALL SYMBOL( XSTRT, YSTRT,0.07,    LLO,  0.0, 2)
      YSTRT  = YSTRT + TAMPMP
      CALL SYMBOL( XSTRT, YSTRT,0.07,    LMN,  0.0, 2)
      YSTRT  = YSTRT + TAMPMP
      CALL SYMBOL( XSTRT, YSTRT,0.07,    LHI,  0.0, 2)
      TIDSCL = THGTMP - TAVGMP
      CALL SYMBOL(  0.10,TIDSCL,0.07,      2, 90.0,-1)
      DTHTMP = - SIN(2. * PI * (TIME - PHSEMP) / PERIOD)
      THETA  = 0.0
      IF(DTHTMP.LT.0.0) THETA = 180.0
      CALL SYMBOL(  0.18,TIDSCL,0.07,      2,THETA,-1)
      CALL SYMBOL( -0.14, -0.40,0.07,    LGU,  0.0, 4)
      CALL SYMBOL( -0.14,-0.505,0.07,    LTI,  0.0, 4)
      CALL SYMBOL(-0.175, -0.61,0.07,    LST,  0.0, 5)
      RETURN
```

END

```
      SUBROUTINE PREP
      LOGICAL                   ISTEP,                  KEYOUT
      DOUBLE PRECISION          SURF(18,17),            SURFC(18)
      COMMON/ACCT/              QDOT(7),                QNET(7),
     1    QSDOT(7),              QSNET(7)
      COMMON/BARS/              PTCTIN,                 SBAR,
     1    UBAR,                  VBAR
      COMMON/BNKCRD/            EAST(22),               IIMAX,
     1    JE,                    JEAST(22),              JW,
     2    JWEST(22),             SOUTH(22),              WEST(22)
      COMMON/CHNNEL/            CDEPTH,                 CWIDTH,
     1    JWAG,                  KMXC,                   KMXCM1,
     2    KMXCM2
      COMMON/FLOOR/             KFLOOR(18,17),          ZB(18,17)
      COMMON/FLOORC/            KFLORC(18),             ZBC(18)
      COMMON/FLOW1/             P(18,17,08),            S(18,17,08),
     1    U(18,17,08),           V(18,17,08),            W(18,17,08)
      COMMON/FLOW2/             SURF,                   DHDT(18,17),
     1    DUDT(18,17,08),        DVDT(18,17,08)
      COMMON/FLOWC1/            PC(18,20),              SC(18,20),
     1    UC(18,20),             WC(18,20)
      COMMON/FLOWC2/            SURFC,                  DHCDT(18),
     1    DUCDT(18,20)
      COMMON/FORCES/            F,                      FWINDX(18,17),
     1    FWINDY(18,17),         TOPLYR
      COMMON/FORTNO/            LR,                     LT,
     1    LW
      COMMON/GRID/              DT,                     DTD2,
     1    DX,                    DXINV,                  DXT2IN,
     2    DXINSQ,                DY,                     DYINV,
     3    DYT2IN,                DYINSQ,                 DZ,
     4    DZD2,                  DZINV,                  DZT2IN,
     5    DZINSQ
      COMMON/GULF/              CUTPT,                  KCUT,
     1    PERIOI
      COMMON/INDEX/             I,                      J,
     1    K,                     N
      COMMON/LIMITS/            IMAX,                   IMAXM1,
     1    JMAX,                  JMAXM1,                 KMAX,
     2    KMAXM1,                KMAXM2,                 NMAX,
     3    KEYOUT
      COMMON/PASS/              IPASS,                  PDEPTH,
     1    PMOMAF,                PSALT,                  PWIDTH
      COMMON/PLOTT/             DXD2,                   L,
     1    NPLOT1,                NPLOT2,                 PERIOD,
     2    ZMAX,                  ZOHSCL
      COMMON/PULL/              SY(17),                 SYY(17),
```

```
     1    TIME,                X(18),               Y(17),
     2    YK2,                 Z(8)
      COMMON/PULLC/            ZC(20)
      COMMON/RIVERS/           JRIV(4),             NRIV,
     1    PHASE1,              PHASE2,              RDEPTH(4),
     2    RMOMAF(4),           RWIDTH(4),           UAVG,
     3    UVAR1,               UVAR2
      COMMON/RYTE/             IO,                  LABELI,
     1    LABELO
      COMMON/SCALE/            HSCALE,              SSCALE,
     1    USCALE,              WSCALE,              XSCALE,
     2    ZSCALE
      COMMON/STEP/             MN,                  MO,
     1    NBAR,                NWRITE,              ISTEP
      COMMON/TIDE/             PHSEMP,              PHSEPH,
     1    TAMPMP,              TAMPPH,              TAVGMP,
     2    TAVGPH,              THGTMP,              THGTPH
      COMMON/TUNE/             ACCEL,               ARTVSC,
     1    AVCOMP
      COMMON/TURB/             CDIF,                CRICH,
     1    CVIS,                DIFFUS,              VISC
      COMMON/UNITS/            BETA,                BETAD2,
     1    FETCH,               GRAV,                HREF,
     2    OMEGA,               PI,                  TREF,
     3    VREF,                YMAX
C
C
C         LOCATE DESIRED FRAME ON TAPE.
C
      IF(N.NE.0) GO TO 10
      N      = 1
      IMAX   = 18
      JMAX   = 17
      NBAR   = 60
      NWRITE = 375
      NMAX   = 6000
      GO TO 10
      IO     = LT
      CALL REED
   10 N      = N + 1
      IF(N.NE.NBAR*(N/NBAR)) GO TO  20
      IO     = LT
      CALL REED2
   20 IF(N.NE.NWRITE*(N/NWRITE)) GO TO  30
      IO     = LT
      CALL REED
      IF(N.GE.NPLOT1*NWRITE) GO TO  50
   30 IF(N.GE.NMAX) GO TO  40
```

```
      GO TO  10
   40 WRITE(LW,1000)
      STOP
C
C        GENERATE SCALE FACTORS AND CONVERT FROM NONDIMENSIONAL VARI-
C        BLES TO VARIABLES SCALED FOR PLOTTING.
C
   50 THGTMP = TAVGMP + TAMPMP * COS((TIME - PHSEMP) * PERIDI)
      TIME   = TIME * TREF / 3600.0
      PHSEMP = PHSEMP * TREF / 3600.0
      PERIOD = 2.0 * PI * TREF / (3600.0 * PERIDI)
      IMAXM1 = IMAX - 1
      JMAXM1 = JMAX - 1
      KMAXM1 = KMAX - 1
      XSCALE = 6.50 / X(IMAX)
      ZSCALE = 1.5
      I      = 0
   60 I      = I + 1
      X(I)   = - X(I) * XSCALE
      IF(I.LT.IMAX) GO TO  60
      J      = 0
   70 J      = J + 1
      Y(J)   = Y(J) * XSCALE
      IF(J.LT.JMAX) GO TO  70
      K      = 0
   80 K      = K + 1
      Z(K)   = Z(K) * ZSCALE
      IF(K.LT.KMAX) GO TO  80
      ZMAX   = Z(KMAX)
      K      = 0
   85 K      = K + 1
      ZC(K)  = ZC(K) * ZSCALE
      IF(K.LT.KMXC) GO TO 85
      II     = 0
   86 II     = II + 1
      SOUTH(II) = - SOUTH(II) * XSCALE
      EAST(II)  = EAST(II) * XSCALE
      WEST(II)  = WEST(II) * XSCALE
      IF(II.LT.IIMAX) GO TO 86
      USCALE = 2.5
      WSCALE = 500.0
      HSCALE = 2.5
      SSCALE = 10.0
      ZOHSCL = ZSCALE / HSCALE
      DO 130 I = 1, IMAX
      DO 110 J = 1, JMAX
      IF(KFLOOR(I,J).LT.KMAX) GO TO  90
```

```
      SURF(I,J) = 0.0
      ZB(I,J)   = ZB(I,J) * ZSCALE + 0.225
      GO TO 110
   90 SURF(I,J) = SURF(I,J) * HSCALE
      ZB(I,J)   = ZB(I,J) * ZSCALE
      KBOT      = KFLOOR(I,J)
      DO 100 K = KBOT, KMAX
      U(I,J,K) = U(I,J,K) * USCALE
      V(I,J,K) = V(I,J,K) * USCALE
      W(I,J,K) = W(I,J,K) * WSCALE
  100 S(I,J,K) = S(I,J,K) * SSCALE
  110 CONTINUE
      ZBC(I)    = ZBC(I) * ZSCALE
      SURFC(I) = SURFC(I) * HSCALE
      KBOT      = KFLORC(I)
      DO 120 K = KBOT, KMXC
      UC(I,K) = UC(I,K) * USCALE
      WC(I,K) = WC(I,K) * WSCALE
  120 SC(I,K) = SC(I,K) * SSCALE
  130 CONTINUE
      DX      = DX * XSCALE
      DXD2    = 0.5 * DX
      PWIDTH = PWIDTH * XSCALE
      THGTMP = THGTMP * HSCALE
      TAMPMP = TAMPMP * HSCALE
      TAVGMP = TAVGMP * HSCALE
C
C        PRINT FRAME IDENTIFICATION.  IF THIS IS THE FIRST FRAME
C        PLOTTED, PRINT SCALED GEOMETRY.
C
      WRITE(LW,1001) L, LABELI
      IF(L.NE.NPLOT1) GO TO 140
      WRITE(LW,1002) HSCALE, SSCALE, USCALE, WSCALE, XSCALE, ZSCALE
      WRITE(LW,1003)
      WRITE(LW,1004) (I, X(I), I = 1, IMAX)
      WRITE(LW,1005)
      WRITE(LW,1004) (J, Y(J), J = 1, JMAX)
      WRITE(LW,1006)
      WRITE(LW,1004) (K, Z(K), K = 1, KMAX)
      WRITE(LW,1007)
      WRITE(LW,1004) (K, ZC(K), K = 1, KMXC)
  140 WRITE(LW,1008)
      RETURN
C
 1000 FORMAT(///,5X,'RAN OUT OF INPUT DATA.')
 1001 FORMAT(///,5X,'L = ',I3,'    LABELI = ',I3)
 1002 FORMAT(5X,'HSCALE = ',F8.3,/,5X,'SSCALE = ',F8.3,/,5X,'USCALE = ',
```

```
     1 F8.3,/,5X,'WSCALE = ',F8.3,/,5X,'XSCALE = ',F8.3,/,5X,
     2 'ZSCALE = ',F8.3)
1003 FORMAT(/,5X,' I    X(SCALED)',/)
1004 FORMAT(/,5X,I2,F9.2)
1005 FORMAT(/,5X,' J    Y(SCALED)',/)
1006 FORMAT(/,5X,' K    Z(SCALED)',/)
1007 FORMAT(/,5X,' K    ZC(SCALED)',/)
1008 FORMAT(/,5X,'PREPARATORY CALCULATIONS COMPLETED.')
     END
```

```
      SUBROUTINE REED
C
C     SUBROUTINE FOR I/O TO SEQUENTIAL STORAGE DEVICE.
C
      LOGICAL                 ISTEP,                  KEYOUT
      DOUBLE PRECISION        SURF(18,17),            SURFC(18)
      COMMON/ACCT/            QDOT(7),                QNET(7),
     1    QSDOT(7),          QSNET(7)
      COMMON/BARS/            PTCTIN,                 SBAR,
     1    UBAR,              VBAR
      COMMON/BNKCRD/          EAST(22),               IIMAX,
     1    JE,                JEAST(22),              JW,
     2    JWEST(22),         SOUTH(22),              WEST(22)
      COMMON/CHNNEL/          CDEPTH,                 CWIDTH,
     1    JWAG,              KMXC,                   KMXCM1,
     2    KMXCM2
      COMMON/FLOOR/           KFLOOR(18,17),          ZB(18,17)
      COMMON/FLOORC/          KFLORC(18),             ZBC(18)
      COMMON/FLOW1/           P(18,17,08),            S(18,17,08),
     1    U(18,17,08),       V(18,17,08),            W(18,17,08)
      COMMON/FLOW2/           SURF,                   DHDT(18,17),
     1    DUDT(18,17,08),    DVDT(18,17,08)
      COMMON/FLOWC1/          PC(18,20),              SC(18,20),
     1    UC(18,20),         WC(18,20)
      COMMON/FLOWC2/          SURFC,                  DHCDT(18),
     1    DUCDT(18,20)
      COMMON/FORCES/          F,                      FWINDX(18,17),
     1    FWINDY(18,17),     TOPLYR
      COMMON/GRID/            DT,                     DTD2,
     1    DX,                DXINV,                  DXT2IN,
     2    DXINSQ,            DY,                     DYINV,
     3    DYT2IN,            DYINSQ,                 DZ,
     4    DZD2,              DZINV,                  DZT2IN,
     5    DZINSQ
      COMMON/GULF/            CUTPT,                  KCUT,
     1    PERIDI
      COMMON/LIMITS/          IMAX,                   IMAXM1,
     1    JMAX,              JMAXM1,                 KMAX,
     2    KMAXM1,            KMAXM2,                 NMAX,
     3    KEYOUT
      COMMON/PASS/            IPASS,                  PDEPTH,
     1    PMOMAF,            PSALT,                  PWIDTH
      COMMON/PULL/            SY(17),                 SYY(17),
     1    TIME,              X(18),                  Y(17),
     2    YK2,               Z(8)
      COMMON/PULLC/           ZC(20)
      COMMON/RIVERS/          JRIV(4),                NRIV,
```

```
1    PHASE1,                PHASE2,                RDEPTH(4),
2    RMOMAF(4),             RWIDTH(4),             UAVG,
3    UVAR1,                 UVAR2
 COMMON/RYTE/               IO,                    LABELI,
1    LABELO
 COMMON/STEP/               MN,                    MO,
1    NBAR,                  NWRITE,                ISTEP
 COMMON/TIDE/               PHSEMP,                PHSEPH,
1    TAMPMP,                TAMPPH,                TAVGMP,
2    TAVGPH,                THGTMP,                THGTPH
 COMMON/TUNE/               ACCEL,                 ARTVSC,
1    AVCOMP
 COMMON/TURB/               CDIF,                  CRICH,
1    CVIS,                  DIFFUS,                VISC
 COMMON/UNITS/              BETA,                  BETAD2,
1    FETCH,                 GRAV,                  HREF,
2    OMEGA,                 PI,                    TREF,
3    VREF,                  YMAX
 READ(IO) ACCEL,  ARTVSC, BETA,   CDEPTH, CDIF,   CUTPT,  CVIS,
1 CWIDTH, DT,     DX,     DY,     DZ,     FETCH,  HREF,   IIMAX,
2 IMAX,   IPASS,  JMAX,   JWAG,   KCUT,   KMAX,   KMXC,   LABELI,
3 DUM,    NBAR,   NRIV,   NMAX,   NWRITE, PDEPTH, PERICI, PMOMAF,
4 PTCTIN, PWIDTH, TREF,   VREF,   YMAX
 READ(IO) (EAST(I), SOUTH(I), WEST(I), I = 1, IIMAX)
 READ(IO) (JEAST(I), JWEST(I), I = 1, IMAX)
 READ(IO) ((KFLOOR(I,J), ZB(I,J), I = 1, IMAX), J = 1, JMAX)
 READ(IO) (KFLORC(I), ZBC(I), I = 1, IMAX)
 READ(IO) (((P(I,J,K),     I = 1, IMAX), J = 1, JMAX), K = 1, KMAX)
 READ(IO) (((S(I,J,K),     I = 1, IMAX), J = 1, JMAX), K = 1, KMAX)
 READ(IO) (((U(I,J,K),     I = 1, IMAX), J = 1, JMAX), K = 1, KMAX)
 READ(IO) (((V(I,J,K),     I = 1, IMAX), J = 1, JMAX), K = 1, KMAX)
 READ(IO) (((W(I,J,K),     I = 1, IMAX), J = 1, JMAX), K = 1, KMAX)
 READ(IO) ((DHDT(I,J),     I = 1, IMAX), J = 1, JMAX)
 READ(IO) (((DUDT(I,J,K),  I = 1, IMAX), J = 1, JMAX), K = 1, KMAX)
 READ(IO) (((DVDT(I,J,K),  I = 1, IMAX), J = 1, JMAX), K = 1, KMAX)
 READ(IO) ((PC(I,K), SC(I,K), UC(I,K), WC(I,K), I = 1, IMAX),
1 K = 1, KMXC)
 READ(IO) (DHCDT(I), (DUCDT(I,K), K = 1, KMXC), SURFC(I), I = 1,
1 IMAX)
 READ(IO) ((FWINDX(I,J), FWINDY(I,J), I = 1, IMAX), J = 1, JMAX)
 READ(IO) (SY(J), SYY(J), Y(J), J = 1, JMAX), (X(I), I = 1, IMAX),
1 (Z(K), K = 1, KMAX), (ZC(K), K = 1, KMXC)
 READ(IO) (JRIV(I), RDEPTH(I), RMOMAF(I), RWIDTH(I), I = 1, NRIV),
1 PHASE1, PHASE2, UAVG, UVAR1, UVAR2
 READ(IO) PHSEMP, PHSEPH, TAMPMP, TAMPPH, TAVGMP, TAVGPH
 ENTRY REED2
 READ(IO) (QDOT(L), QNET(L), QSDOT(L), QSNET(L), L = 1, 7), TIME
```

```
      READ(IO) SBAR, UBAR, VBAR
      READ(IO) ((SURF(I,J),     I = 1, IMAX), J = 1, JMAX)
      RETURN
      END
```

```
      SUBROUTINE XYBRD1
      LOGICAL                 FLAG
      LOGICAL                 ISTEP,              KEYOUT
      DIMENSION               LN10KM(2),          L30KM(2),
     1    L50KM(2)
      DOUBLE PRECISION        SURF(18,17),        SURFC(18)
      COMMON/ACCT/            QDOT(7),            QNET(7),
     1    QSDOT(7),           QSNET(7)
      COMMON/BARS/            PTCTIN,             SBAR,
     1    UBAR,               VBAR
      COMMON/BNKCRD/          EAST(22),           IIMAX,
     1    JE,                 JEAST(22),          JW,
     2    JWEST(22),          SOUTH(22),          WEST(22)
      COMMON/CHNNEL/          CDEPTH,             CWIDTH,
     1    JWAG,               KMXC,               KMXCM1,
     2    KMXCM2
      COMMON/FLOOR/           KFLCOR(18,17),      ZB(18,17)
      COMMON/FLOORC/          KFLORC(18),         ZBC(18)
      COMMON/FLOW1/           P(18,17,08),        S(18,17,08),
     1    U(18,17,08),        V(18,17,08),        W(18,17,08)
      COMMON/FLOW2/           SURF,               DHDT(18,17),
     1    DUDT(18,17,08),     DVDT(18,17,08)
      COMMON/FLOWC1/          PC(18,20),          SC(18,20),
     1    UC(18,20),          WC(18,20)
      COMMON/FLOWC2/          SURFC,              DHCDT(18),
     1    DUCDT(18,20)
      COMMON/FORCES/          F,                  FWINDX(18,17),
     1    FWINDY(18,17),      TOPLYR
      COMMON/GRID/            DT,                 DTD2,
     1    DX,                 DXINV,              DXT2IN,
     2    DXINSQ,             DY,                 DYINV,
     3    DYT2IN,             DYINSQ,             DZ,
     4    DZD2,               DZINV,              DZT2IN,
     5    DZINSQ
      COMMON/GULF/            CUTPT,              KCUT,
     1    PERIDI
      COMMON/INDEX/           I,                  J,
     1    K,                  N
      COMMON/LIMITS/          IMAX,               IMAXM1,
     1    JMAX,               JMAXM1,             KMAX,
     2    KMAXM1,             KMAXM2,             NMAX,
     3    KEYOUT
      COMMON/PASS/            IPASS,              PDEPTH,
     1    PMOMAF,             PSALT,              PWIDTH
      COMMON/PLOTT/           DXD2,               L,
     1    NPLOT1,             NPLOT2,             PERIOD,
     2    ZMAX,               ZOHSCL
```

```
      COMMON/PULL/              SY(17),             SYY(17),
     1    TIME,                X(18),              Y(17),
     2    YK2,                 Z(8)
      COMMON/PULLC/             ZC(20)
      COMMON/RIVERS/            JRIV(4),            NRIV,
     1    PHASE1,              PHASE2,             RDEPTH(4),
     2    RMOMAF(4),           RWIDTH(4),          UAVG,
     3    UVAR1,               UVAR2
      COMMON/RYTE/              IO,                 LABELI,
     1    LABELO
      COMMON/STEP/              MN,                 MO,
     1    NBAR,                NWRITE,             ISTEP
      COMMON/SCALE/             HSCALE,             SSCALE,
     1    USCALE,              WSCALE,             XSCALE,
     2    ZSCALE
      COMMON/TURB/              CDIF,               CRICH,
     1    CVIS,                DIFFUS,             VISC
      COMMON/UNITS/             BETA,               BETAD2,
     1    FETCH,               GRAV,               HREF,
     2    OMEGA,               PI,                 TREF,
     3    VREF,                YMAX
      DATA LN10KM/'-10.','KM  '/
      DATA L0KM  /'0.KM'/
      DATA L50KM /'50.K','M   '/
      DATA L30KM /'30.K','M   '/
C
CC        PLOT BAY OUTLINE.
C
      CALL VTHICK(2)
      JW       = JWEST(1)
      JE       = JEAST(1)
      YWEST    = 0.5 * (Y(JW-1) + Y(JW))
      YEAST    = 0.5 * (Y(JE) + Y(JE+1))
      CALL PLOT( YWEST,  0.00, 3)
      NN       = 0
   10 NN       = NN + 1
      J        = JRIV(NN)
      Y1       = Y(J)
      Y1M      = Y1 - 0.05
      Y1P      = Y1 + 0.05
      CALL PLOT(   Y1M,  0.00, 2)
      CALL PLOT(   Y1M,  0.25, 2)
      CALL PLOT(   Y1P,  0.25, 3)
      CALL PLOT(   Y1P,  0.00, 2)
      IF(NN.LT.NRIV) GO TO 10
      CALL PLOT( YEAST,  0.00, 2)
      CALL PLOT( YWEST,  0.00, 3)
```

```
      XPASS  = X(IPASS)
      PWTHD2 = 0.5 * PWIDTH
      XBANKN = XPASS + PWTHD2
      XBANKS = XPASS - PWTHD2
      YWSTIP = YWEST
      XAVGIP = X(1) - DXD2
      IPSSM1 = IPASS - 1
      DO  30 I = 1, IMAXM1
      IP1    = I + 1
      XAVG   = XAVGIP
      XAVGIP = XAVG - DX
      IF(IP1.EQ.IMAX) XAVGIP = XAVG - DXD2
      JWIP   = JWEST(IP1)
      JWIPM1 = JWIP - 1
      YWEST  = YWSTIP
      YWSTIP = 0.5 * (Y(JWIPM1) + Y(JWIP))
      CALL PLOT(YWSTIP,  XAVG, 2)
      IF(I.NE.IPSSM1) GO TO  20
      CALL PLOT(YWSTIP,XBANKN, 2)
      CALL PLOT(YWSTIP-0.25,XBANKN, 2)
      CALL PLOT(YWSTIP-0.25,XBANKS, 3)
      CALL PLOT(YWSTIP,XBANKS, 2)
   20 CALL PLOT(YWSTIP,XAVGIP, 2)
   30 CONTINUE
      CALL PLOT( YEAST,  0.00, 3)
      YESTIP = YEAST
      XAVGIP = X(1) - DXD2
      DO  40 I = 1, IMAXM1
      IP1    = I + 1
      XAVG   = XAVGIP
      XAVGIP = XAVG - DX
      IF(IP1.EQ.IMAX) XAVGIP = XAVG - DXD2
      JEIP   = JEAST(IP1)
      JEIPP1 = JEIP + 1
      YEAST  = YESTIP
      YESTIP = 0.5 * (Y(JEIP) + Y(JEIPP1))
      CALL PLOT(YESTIP,  XAVG, 2)
      CALL PLOT(YESTIP,XAVGIP, 2)
   40 CONTINUE
C
C        GENERATE LENGTH SCALES.
C
      CALL VTHICK(1)
      CALL PLOT( -1.50,  0.00,-3)
      XREAL  = 10000.0
   50 XREAL  = XREAL - 10000.0
      X1     = (XREAL / HREF) * XSCALE
```

```
      CALL SYMBOL(  0.00,    X1,0.07,    13, 90.0,-1)
      IF(XREAL.GT.-50000.0) GO TO  50
      X1      = X1 - 0.035
      CALL SYMBOL( -0.44,    X1,0.07, L50KM,  0.0, 5)
      CALL SYMBOL( -0.37,-0.035,0.07,  L0KM,  0.0, 4)
      CALL PLOT(  1.50,  0.00,-3)
      CALL PLOT(  0.00, -6.60,-3)
      YREAL   = -20000.0
   60 YREAL   = YREAL + 10000.0
      Y1      = (YREAL / HREF) * XSCALE
      CALL SYMBOL(    Y1,  0.00,0.07,    13,  0.0,-1)
      IF(YREAL.LT.30000.0) GO TO  60
      Y1      = Y1 - 0.175
      CALL SYMBOL(    Y1, -0.20,0.07, L30KM,  0.0, 5)
      Y1      = (-10000.0 / HREF) * XSCALE - 0.21
      CALL SYMBOL(    Y1, -0.20,0.07,LN10KM,  0.0, 6)
      CALL PLOT(  0.00,  6.60,-3)
C
C        GENERATE 8.5 BY 11-INCH PAGE BORDER.
C
      CALL VTHICK(1)
      CALL PLOT( -3.50,  1.50, 3)
      CALL PLOT( -3.50, -9.50, 2)
      CALL PLOT(  5.00, -9.50, 2)
      CALL PLOT(  5.00,  1.50, 2)
      CALL PLOT( -3.50,  1.50, 2)
      RETURN
      END
```

```
      SUBROUTINE XZBRD1
      LOGICAL                 ISTEP,          KEYOUT
      DIMENSION               L50KM(2)
      DOUBLE PRECISION        SURF(18,17),    SURFC(18)
      COMMON/ACCT/            QDCT(7),        QNET(7),
     1    QSDOT(7),           QSNET(7)
      COMMON/BARS/            PTCTIN,         SBAR,
     1    UBAR,               VBAR
      COMMON/BNKCRD/          EAST(22),       IIMAX,
     1    JE,                 JEAST(22),      JW,
     2    JWEST(22),          SOUTH(22),      WEST(22)
      COMMON/CHNNEL/          CDEPTH,         CWIDTH,
     1    JWAG,               KMXC,           KMXCM1,
     2    KMXCM2
      COMMON/FLOOR/           KFLOOR(18,17),  ZB(18,17)
      COMMON/FLOORC/          KFLORC(18),     ZBC(18)
      COMMON/FLOW1/           P(18,17,08),    S(18,17,08),
     1    U(18,17,08),        V(18,17,08),    W(18,17,08)
      COMMON/FLOW2/           SURF,           DHDT(18,17),
     1    DUDT(18,17,08),     DVDT(18,17,08)
      COMMON/FLOWC1/          PC(18,20),      SC(18,20),
     1    UC(18,20),          WC(18,20)
      COMMON/FLOWC2/          SURFC,          DHCDT(18),
     1    DUCDT(18,20)
      COMMON/FORCES/          F,              FWINDX(18,17),
     1    FWINDY(18,17),      TOPLYR
      COMMON/GRID/            DT,             DTD2,
     1    DX,                 DXINV,          DXT2IN,
     2    DXINSQ,             DY,             DYINV,
     3    DYT2IN,             DYINSQ,         DZ,
     4    DZD2,               DZINV,          DZT2IN,
     5    DZINSQ
      COMMON/GULF/            CUTPT,          KCUT,
     1    PERIDI
      COMMON/INDEX/           I,              J,
     1    K,                  N
      COMMON/LIMITS/          IMAX,           IMAXM1,
     1    JMAX,               JMAXM1,         KMAX,
     2    KMAXM1,             KMAXM2,         NMAX,
     3    KEYOUT
      COMMON/PASS/            IPASS,          PDEPTH,
     1    PMOMAF,             PSALT,          PWIDTH
      COMMON/PLOTT/           DXD2,           L,
     1    NPLOT1,             NPLOT2,         PERIOD,
     2    ZMAX,               ZOHSCL
      COMMON/PULL/            SY(17),         SYY(17),
     1    TIME,               X(18),          Y(17),
```

```
     2    YK2,                    Z(8)
          COMMON/PULLC/           ZC(20)
          COMMON/RIVERS/          JRIV(4),              NRIV,
     1    PHASE1,                 PHASE2,               RDEPTH(4),
     2    RMOMAF(4),              RWIDTH(4),            UAVG,
     3    UVAR1,                  UVAR2
          COMMON/RYTE/            IO,                   LABELI,
     1    LABELO
          COMMON/SCALE/           HSCALE,               SSCALE,
     1    USCALE,                 WSCALE,               XSCALE,
     2    ZSCALE
          COMMON/STEP/            MN,                   MC,
     1    NBAR,                   NWRITE,               ISTEP
          COMMON/TIDE/            PHSEMP,               PHSEPH,
     1    TAMPMP,                 TAMPPH,               TAVGMP,
     2    TAVGPH,                 THGTMP,               THGTPH
          COMMON/TUNE/            ACCEL,                ARTVSC,
     1    AVCOMP
          COMMON/TURB/            CDIF,                 CRICH,
     1    CVIS,                   DIFFUS,               VISC
          COMMON/UNITS/           BETA,                 BETAD2,
     1    FETCH,                  GRAV,                 HREF,
     2    OMEGA,                  PI,                   TREF,
     3    VREF,                   YMAX
          DATA LN5M   /'-5.M'/
          DATA L0KM   /'0.KM'/
          DATA L0M    /'0.M '/
          DATA L5M    /'5.M '/
          DATA L50KM /'50.K','M   '/
  10      CALL VTHICK(2)
          XNORTH = - X(1) - DXD2
          ZBOTIP = ZB(1,J)
          CALL PLOT(XNORTH,ZMAX+0.3, 3)
          IF(ZBOTIP.LT.ZMAX) CALL VTHICK(-4)
          M      = 2
          IF(J.EQ.JWAG) M = 3
          CALL PLOT(XNORTH,ZBOTIP, M)
          CALL VTHICK(2)
          DO  20 I = 1, IMAXM1
          IP1    = I + 1
          ZBOT   = ZBOTIP
          ZBOTIP = ZB(IP1,J)
          XAVG   = - X(I) + DXD2
          CALL PLOT(  XAVG,  ZBOT, 2)
          CALL PLOT(  XAVG,ZBOTIP, 2)
  20      CONTINUE
          XSOUTH = - X(IMAX) + DXD2
```

```
      CALL PLOT(XSOUTH,ZBOTIP, 2)
      IF(ZBOTIP.GT.ZMAX) GO TO  30
      IF(J.EQ.JWAG) GO TO  30
      CALL VTHICK(-4)
      CALL PLOT(XSOUTH,ZMAX+0.3, 3)
      CALL PLOT(XSOUTH,ZBOTIP, 2)
   30 CALL VTHICK(-3)
      CALL PLOT(  0.00,  ZMAX,-3)
      SURFIP = SURF(1,J) * ZOHSCL
      KBOTIP = KFLOOR(1,J)
      CALL PLOT(XNORTH,SURFIP, 3)
      DO  70 I = 1, IMAXM1
      IP1    = I + 1
      KBOT   = KBOTIP
      KBOTIP = KFLOOR(IP1,J)
      SURF1  = SURFIP
      SURFIP = SURF(IP1,J) * ZOHSCL
      XAVG   = - X(I) + DXD2
      IF(KBOT.GT.KMAX.AND.KBOTIP.GT.KMAX) GO TO  60
      IF(KBOT.GT.KMAX.OR.KBOTIP.GT.KMAX) GO TO  40
      SRFAVG = 0.5 * (SURF1 + SURFIP)
      CALL PLOT(  XAVG,SRFAVG, 2)
      GO TO  60
   40 IF(KBOT.LT.KMAX) GO TO  50
      CALL PLOT(  XAVG,SURFIP, 3)
      GO TO  60
   50 CALL PLOT(  XAVG, SURF1, 2)
   60 CONTINUE
      IF(IP1.LT.IMAX) GO TO 70
      SRFAVG = 0.5 * (SURFIP + THGTMP * ZOHSCL)
      CALL PLOT(XSOUTH,SRFAVG, 2)
   70 CONTINUE
      CALL PLOT(  0.00, -ZMAX,-3)
C
C        GENERATE LENGTH SCALES.
C
      CALL VTHICK(1)
      ZSTRT  = 0.00
      IF(J.EQ.JWAG) ZSTRT = - 2.30
      CALL PLOT(  0.00, ZSTRT,-3)
      XREAL  = -10000.0
   80 XREAL  = XREAL + 10000.0
      X1     = (XREAL / HREF) * XSCALE
      CALL SYMBOL(    X1, -0.00,0.07,    13,  0.0,-1)
      IF(XREAL.LT.50000.0) GO TO  80
      X1     = X1 - 0.175
      CALL SYMBOL(    X1, -0.20,0.07, L50KM,  0.0, 5)
```

```
      CALL SYMBOL( -0.14, -0.20,0.07,  L0KM,  0.0, 4)
      CALL PLOT(  0.00,-ZSTRT,-3)
      ZREAL  = - 1.0
      IF(J.EQ.JWAG) ZREAL = - 6.0
   90 ZREAL  = ZREAL + 1.0
      Z1     = (ZREAL / HREF) * ZSCALE
      CALL SYMBOL( -0.80,    Z1,0.07,    13, 90.0,-1)
      IF(ZREAL.LT.5.0) GO TO  90
      Z1     = Z1 - 0.035
      CALL SYMBOL( -1.10,    Z1,0.07,   L5M,  0.0, 3)
      CALL SYMBOL( -1.10,-0.035,0.07,   L0M,  0.0, 3)
      IF(J.NE.JWAG) GO TO 95
      Z1     = - (Z1 + 0.07)
      CALL SYMBOL( -1.17,    Z1,0.07,  LN5M,  0.0, 4)
   95 CONTINUE
C
C        GENERATE 8.5 BY 11-INCH PAGE BORDER.
C
      CALL PLOT( -2.50,  3.50, 3)
      CALL PLOT( -2.50, -5.00, 2)
      CALL PLOT(  8.50, -5.00, 2)
      CALL PLOT(  8.50,  3.50, 2)
      CALL PLOT( -2.50,  3.50, 2)
      RETURN
C
C        SPECIAL ENTRY FOR PLOTTING SHIP CHANNEL OUTLINE.
C
      ENTRY CHBRD1
      XNORTH = - X(1) - DXD2
      ZBOTIP = ZBC(1)
      CALL PLOT(XNORTH,ZMAX+0.3, 3)
      CALL VTHICK(-4)
      CALL PLOT(XNORTH,ZBOTIP, 2)
      CALL VTHICK(2)
      DO 100 I = 1, IMAXM1
      IP1    = I + 1
      ZBOT   = ZBOTIP
      ZBOTIP = ZBC(IP1)
      XAVG   = - X(I) + DXD2
      CALL PLOT(  XAVG,  ZBOT, 2)
      CALL PLOT(  XAVG,ZBOTIP, 2)
  100 CONTINUE
      XSOUTH = - X(IMAX) + DXD2
      CALL PLOT(XSOUTH,ZBOTIP, 2)
      CALL VTHICK(-4)
      CALL PLOT(XSOUTH,ZMAX+0.3, 3)
      CALL PLOT(XSOUTH,ZBOTIP ,2)
```

```
GO TO 10
END
```

```
      SUBROUTINE YZBRD1
      LOGICAL          ISTEP,          KEYOUT
      DIMENSION        LN10KM(2),      L30KM(2)
      DOUBLE PRECISION SURF(18,17),    SURFC(18)
      COMMON/ACCT/     QDOT(7),        QNET(7),
     1    QSDOT(7),
      COMMON/BARS/     PTCTIN,         SBAR,
     1    UBAR,        VBAR
      COMMON/BNKCRD/   EAST(22),       IIMAX,
     1    JE,          JEAST(22),      JW,
     2    JWEST(22),   SOUTH(22),      WEST(22)
      COMMON/CHNNEL/   CDEPTH,         CWIDTH,
     1    JWAG,        KMXC,           KMXCM1,
     2    KMXCM2
      COMMON/FLOOR/    KFLOCR(18,17),  ZB(18,17)
      COMMON/FLOORC/   KFLORC(18),     ZBC(18)
      COMMON/FLOW1/    P(18,17,08),    S(18,17,08),
     1    U(18,17,08), V(18,17,08),    W(18,17,08)
      COMMON/FLOW2/    SURF,           DHDT(18,17),
     1    DUDT(18,17,08), DVDT(18,17,08)
      COMMON/FLOWC1/   PC(18,20),      SC(18,20),
     1    UC(18,20),   WC(18,20)
      COMMON/FLOWC2/   SURFC,          DHCDT(18),
     1    DUCDT(18,20)
      COMMON/FORCES/   F,              FWINDX(18,17),
     1    FWINDY(18,17), TOPLYR
      COMMON/GRID/     DT,             DTD2,
     1    DX,          DXINV,          DXT2IN,
     2    DXINSQ,      DY,             DYINV,
     3    DYT2IN,      DYINSQ,         DZ,
     4    DZD2,        DZINV,          DZT2IN,
     5    DZINSQ
      COMMON/GULF/     CUTPT,          KCUT,
     1    PERIOD
      COMMON/INDEX/    I,              J,
     1    K,           N
      COMMON/LIMITS/   IMAX,           IMAXM1,
     1    JMAX,        JMAXM1,         KMAX,
     2    KMAXM1,      KMAXM2,         NMAX,
     3    KEYOUT
      COMMON/PASS/     IPASS,          PDEPTH,
     1    PMOMAF,      PSALT,          PWIDTH
      COMMON/PLOTT/    DXD2,           L,
     1    NPLOT1,      NPLOT2,         PERIOD,
     2    ZMAX,        ZOHSCL
      COMMON/PULL/     SY(17),         SYY(17),
     1    TIME,        X(18),          Y(17),
```

```
2   YK2,                    Z(8)
 COMMON/PULLC/              ZC(20)
 COMMON/RIVERS/             JRIV(4),            NRIV,
1    PHASE1,                PHASE2,             RDEPTH(4),
2    RMUMAF(4),             RWIDTH(4),          UAVG,
3    UVAR1,                 UVAR2
 COMMON/RYTE/               IO,                 LABELI,
1    LABELO
 COMMON/SCALE/              HSCALE,             SSCALE,
1    USCALE,                WSCALE,             XSCALE,
2    ZSCALE
 COMMON/STEP/               MN,                 MO,
1    NBAR,                  NWRITE,             ISTEP
 COMMON/TIDE/               PHSEMP,             PHSEPH,
1    TAMPMP,                TAMPPH,             TAVGMP,
2    TAVGPH,                THGTMP,             THGTPH
 COMMON/TUNE/               ACCEL,              ARTVSC,
1    AVCOMP
 COMMON/TURB/               CDIF,               CRICH,
1    CVIS,                  DIFFUS,             VISC
 COMMON/UNITS/              BETA,               BETAD2,
1    FETCH,                 GRAV,               HREF,
2    OMEGA,                 PI,                 TREF,
3    VREF,                  YMAX
 DATA LN1CKM/'-10.','KM   '/
 DATA L0M    /'0.M '/
 DATA L5M    /'5.M '/
 DATA L30KM /'30.K','M    '/
C
C       GENERATE YZ-CROSS SECTION AT X(I).
C
 CALL VTHICK(2)
 JWM1    = JW - 1
 JEM1    = JE - 1
 JEP1    = JE + 1
 YWEST   = 0.5 * (Y(JWM1) + Y(JW))
 ZBOTJP = ZB(I,JW)
 CALL PLOT( YWEST,ZMAX+0.3, 3)
 IF(I.EQ.IPASS) CALL VTHICK(-4)
 CALL PLOT( YWEST,ZBOTJP, 2)
 CALL VTHICK(2)
 YJP     = YWEST
 DO  10 J = JW, JE
 JP1     = J + 1
 ZBOT    = ZBOTJP
 ZBOTJP = ZB(I,JP1)
 Y1      = YJP
```

```
      YJP     = Y(JP1)
      YAVG    = 0.5 * (Y1 + YJP)
      CALL PLOT(  YAVG,  ZBOT, 2)
      CALL PLOT(  YAVG,ZBOTJP, 2)
   10 CONTINUE
      YEAST   = YAVG
      CALL VTHICK(-3)
      CALL PLOT(  0.00,  ZMAX,-3)
      SURFJP = SURF(I,JW) * ZOHSCL
      CALL PLOT( YWEST,SURFJP, 3)
      KBOTJP = KFLOOR(I,JW)
      YJP     = YWEST
      DO  40 J = JW, JEM1
      JP1     = J + 1
      KBOT    = KBOTJP
      KBOTJP = KFLOOR(I,JP1)
      SURF1   = SURFJP
      SURFJP = SURF(I,JP1) * ZOHSCL
      Y1      = YJP
      YJP     = Y(JP1)
      YAVG    = 0.5 * (Y1 + YJP)
      IF(KBOT.GT.KMAX.AND.KBOTJP.GT.KMAX) GO TO  40
      IF(KBOT.GT.KMAX.OR.KBOTJP.GT.KMAX) GO TO  20
      SRFAVG = 0.5 * (SURF1 + SURFJP)
      CALL PLOT(  YAVG,SRFAVG, 2)
      GO TO  40
   20 IF(KBOT.LT.KMAX) GO TO  30
      CALL PLOT(  YAVG,SURFJP, 3)
      GO TO  40
   30 CALL PLOT(  YAVG, SURF1, 2)
   40 CONTINUE
      CALL PLOT( YEAST,SURFJP, 2)
      CALL PLOT(  0.00, -ZMAX,-3)
C
CC       GENERATE LENGTH SCALES.
C
      CALL VTHICK(1)
      YREAL   = -20000.0
   50 YREAL   = YREAL + 10000.0
      Y1      = (YREAL / HREF) * XSCALE
      CALL SYMBOL(    Y1, -0.00,0.07,    13,  0.0,-1)
      IF(YREAL.LT.30000.0) GO TO 50
      Y1      = Y1 - 0.175
      CALL SYMBOL(    Y1, -0.20,0.07, L30KM,  0.0, 5)
      Y1      = (-10000.0 / HREF) * XSCALE - 0.21
      CALL SYMBOL(    Y1, -0.20,0.07,LN10KM,  0.0, 6)
      ZREAL   = -1.0
```

```
   60 ZREAL   = ZREAL + 1.0
      Z1      = (ZREAL / HREF) * ZSCALE
      CALL SYMBOL( -1.70,     Z1,0.07,     13, 90.0,-1)
      IF(ZREAL.LT.5.0) GO TO  60
      Z1      = Z1 - 0.035
      CALL SYMBOL( -2.00,     Z1,0.07,   LSM,  0.0, 3)
      CALL SYMBOL( -2.00,-0.035,0.07,   LOM,  0.0, 3)
C
C        GENERATE 8.5 BY 11-INCH PAGE BORDER.
C
      CALL PLOT( -4.40,  3.50, 3)
      CALL PLOT( -4.40, -5.00, 2)
      CALL PLOT(  6.60, -5.00, 2)
      CALL PLOT(  6.60,  3.50, 2)
      CALL PLOT( -4.40,  3.50, 2)
      RETURN
      END
```

```
      BLOCK DATA
C
CC        VARIOUS CONSTANTS ARE DEFINED.
C
      LOGICAL                   ISTEP
      COMMON/FORTNO/            LR,                 LT,
     1   LW
      COMMON/INDEX/             I,                  J,
     1   K,                     N
      COMMON/STEP/              MN,                 MO,
     1   NBAR,                  NWRITE,             ISTEP
      COMMON/UNITS/             BETA,               BETAD2,
     1   FETCH,                 GRAV,               HREF,
     2   OMEGA,                 PI,                 TREF,
     3   VREF,                  YMAX
C
CC        LW, LR, LT ARE THE LOGICAL I/O UNIT NUMBERS FOR WRITING ON LINE
CC        PRINTER, READING FROM CARD READER, USING TAPE, RESPECTIVELY.
C
      DATA LW,LR,LT/6,5,3/
      DATA MN,MO,N/2,1,0/
      DATA GRAV,OMEGA,PI/9.790,7.29E-5,3.14159/
      END
```

```
DATA
      13          16
       1           0          0
       4
       0
       0
      -1
```