Final Report

TIDAL FREQUENCY ESTIMATION
for
CLOSED BASINS

J.B. Eades, Jr.

AMA Report No. 78-11
Contract NAS5-22927
February 15, 1978

ANALYTICAL MECHANICS ASSOCIATES, INC.
50 Jericho Turnpike
Jericho, New York 11753

ACKNOWLEDGEMENTS

The efforts of Dr. Malcolm Newman, on behalf of this task, must be recognized and commented upon. It was he who did the finite element formulation; and, it was his knowledge and insight into the NASTRAN program, which prompted its use initially. Also, it was Dr. Newman who developed the FEER programs.

Also, in recognition for his counsel and guidance, acknowledgement is given to Dr. M.A. Graber, Code 921, at the NASA/Goddard Space Flight Center, who served as the Technical Monitor for this work.

TABLE OF CONTENTS

page

APPENDIX B (continued)

I. INTRODUCTION

The earth's oceans have, in recent times, became an area for renewed study and a candidate for investigational analysis. This, coupled with the use of satellites as vehicles for the transport of experimental research instrumentation, has prompted much of the activity; and, in fact, has been the impetus for some of the additional research, per se.

One of these "renewed areas" is concerned with the prediction and the estimation of ocean tides. In this regard there are analysts who have concocted "new" mathematical models of the oceans and their driving influences. Still others have an interest in these same, or similar, studies but are seeking new and different techniques to apply in their modelling.

From a literature perusal it would seem that the most frequently used mathematical approach in this work is one based on finite difference techniques. Of course this is only the vehicle for solution; it has very little to do with the mathematical model devised to describe the topography and physical character of the oceans and their disturbances. Many of these models have only "small" differences in their makeup -- that is, they differ primarily in the representations used to describe certain physical characteristics of the real world problem situation. Others may include different "influences" which introduce perturbations to the ocean body; and still others will incorporate measurements, from observations, as a means to "adjust" the predicted output from the ensueing computations. All things considered, the basic thrust for these studies seems to be aimed at more accurate predictions for the global tides -- and, apparently, the present day goal is one of forecasting tides to within a ten centimeter accuracy.

Not all efforts are as ambitious and as far reaching as those noted above. Some investigations are directed to more elementary tasks such as the predictions for lesser sized bodies of water; or, to the development of new techniques which might supplement or even replace some of the more popular procedures.

It was in the spirit of these latter trial efforts that the study reported here was undertaken. The basic aim in this work was to develop a method for determining the fundamental tidal frequencies, for closed basins -- of water -- by means of an eigenvalue analysis. In this regard, then, the mathematical model which was to be employed, was the so-called Laplace Tidal Equations. Obviously these are not "new" equations (LAPLACE, Pierre Simon, Marquis de (1749-1827)); however, the proposed procedure for "solving" these was to represent a different and somewhat unique approach.

It was proposed that these mathematical statements, by Laplace, would be cast in a format employing the finite element method. Once this model of the governing expressions was in hand, it was proposed that solutions for the tidal frequencies be pursued. This obviously would lead to an eigenvalue analysis, but to one which would necessitate the use of some different mathematical machinery as the means to the ends desired. Here, because of formulation, it was proposed that the eigenvalue extraction algorithms residing in the NASTRAN program, would be put to use. The reasoning behind this stemmed from the fact that NASTRAN was designed to handle large scale problems (structural problems) and it possessed the machinery necessary to do the large eigen-analysis expected in this task.

Once the necessary software was in hand, an application of this scheme was to be tested on candidate basins -- natural basins. The obvious candidates for such a task, were the Great Lakes. Thus the "proof of the pudding", so to speak, would lie in the application of this procedure to a determination of the natural (fundamental) tidal frequencies for the selected bodies in the Great Lakes system.

A first candidate for the study was selected to be Lake Erie. Subsequently, the next chosen body of water, for examination, was Lake Superior.

In the following sections of this report the reader will find developments descriptive of the mathematical model used here; also, there will be discussions of the procedures tried and adopted; and, finally, some selected results acquired for the exercising of this method will be noted and described.

The next sections of the report are given to a general discussion on the finite element method, its history and its application in this investigation. Later, following the report of selected results there will be a few statements concerning the outcome of this work, and recommendations for subsequent efforts.

II. FINITE ELEMENT FORMULATION

II.1 Introduction

From a perusal of literature on the finite element method one finds that most previous work in this area has been based on the minimization of some function - a variational statement. Basically, this can be traced to the fact that the finite element method (FEM) was originally developed for use in structural mechanics, a field where variational principles abound.

Contrary to the structures field, fluid mechanics is one area where there is, more often than not, a scarcity of variational principles. Findlayson (see Reference (5)) in preparing a summary for fluids, mentions such areas as perfect fluids, magnetohydrodynamics, non-Newtonian fluids and the low Reynolds number problems as candidates for the variational approach. However, in with these classifications he notes that there is no known variational principles for the Navier-Stokes equations - pointing to the fact that these expressions contain representations for both viscous and inertia forces in their makeup. In addition, it is not apparent just what would be an appropriate function, for minimization, in the hyperbolic-type, vertically averaged equations (see Appendix A) which are often used for the study of tidal ponds and coastal water basins.

In at least one instance, found in the literature, there has been developed a type of variational formulation for hydrodynamic equations. McIver (Reference (11)) has constructed an adjoint variational principle; however, his work does not have a direct (physical) usefulness due to the presence of the added adjoint variables. With the inclusion of these quantities the problem "size" is doubled, in unknowns; accordingly the computational time and problem complexity is increased substantially.

Among the various procedures used to solve problems described by partial differential equations, probably the one most widely used is the method of weighted residuals. In this procedure the unknown solution is simulated by a set of so-called "trial functions". The

constants, in these functions, are adjusted so that the final result provides a "best fit" solution to the problem under investigation. To obtain values for the constants, the trial functions are (first) substituted into the governing equations; however, since these functions do not represent a true solution, residuals are formed. The constants are then adjusted so that the residuals, modified by chosen weighting functions, are "zeroed" in some average sense.

One of the most crucial operations here has to do with selecting the weighting functions. Obviously there are any number of procedures by which these may be chosen; and, obviously, each represents a different method of approach.

Among the more popular procedures are: (1) the Galerkin method; (2) a least squares method; and (3) the method of moments. Of these three the procedure which has, recently, found the most favor among "finite element" investigators is the Galerkin method. One apparent advantage of this procedure is that in it the weighting functions are the trial functions used in the simulation of the problem's solution (see References (3, 5)).

Quite frequently the Galerkin method leads to a simpler and more direct formulation than would otherwise be obtained by constructing the trial functions and subsequently going through the minimization operation.

[As an interesting aside; when the governing equations are self-adjoint, the variational procedure and the Galerkin weighted residual method become identical.]

It goes almost without saying that, because of its simplicity, and the success which the method has enjoyed, the Galerkin procedure was chosen for use in the present investigation. This method, which is used to simulate a solution, coupled with the finite element technique, is well adapted for use here. As has been demonstrated elsewhere, (e.g., References (3 , 5)) this combination of procedures is very well adapted to representing a solution over a region having complex boundaries and continually varying bathymetry. Of course, one consequence of these complications is that no analytic solutions are forthcoming -

hence, numerical solutions will be produced, and these will be acquired through the use of a digital computer.

It is only in the more recent investigations of hydrodynamic problems that the FEM approach has been adopted. Prior to that the usual method employed in acquiring solutions was the finite difference (FD) approach. Necessarily both procedures (the FD and the FEM) are applied to the same governing expressions; the main difference between them being that the FD-method approximates derivatives appearing in the governing equations, while the FEM operates through integrals developed from the same differential equations. A second difference between these methods, and one not necessarily of small consequence, has to do with the geometry used to subdivide the region over which a solutions is sought. In the FD method, as a general procedure, and one almost universally employed, squares (of uniform size) are utilized as subdivisions for the solution region. Contrary to this, the FEM procedure most frequently uses triangles as subdivisions. These are, however, not of a particular size or orientation. There are some specific conditions which must be met (for and by these geometries); however, these are not described at present. Suffice it to say, the triangular subdivisions are more readily and easily adapted to natural boundaries (e.g., coast lines, shores, etc.) and are, therefore, more representative of these bounds. In all cases and methods the subdivisions are finite in number; the triangles, used in the FEM, are of much greater utility in satisfying grid refinements, particularly where steep gradients are present. (That is, in regions where sharp corners, point sources, irregular bottom geometries, etc., are present and need to be modelled or accounted for in the formulations).

With an adoption of the FEM, as an approach to studying tides and circulation models, one finds a most powerful tool for analysis. In this procedure a function - satisfying both the boundary conditions and the governing equations - is approximated by piecewise continuous polynomials. This approach, coupled with the solution elements, is ready made for the inclusion of numerical information, parameters of consequence, analytical relationships, or whatever else (based on experience or experiment) may serve as part of a problem's

formulation. Such models should be truly predictive of a physical case and, of course, with boundary conditions included, as these arise, results should be indicative of observed phenomena.

The approach which has been selected for this study makes use of the finite element method, incorporating linear triangular elements and linear interpolation functions.

The problem statement is deduced from the two-dimensional, vertically integrated equations of motion (see Appendix A); and, the Galerkin method has been used in developing the finite element equations. This implies a use of the method of weighted residuals. The form of the dynamical equations of motion, utilized here, is that most frequently referred to as Laplace's Tidal Equations. These expressions are introduced into the NASTRAN* system, along with an appropriate conservation of mass expression, for the extraction of eigenvalues and a determination of associated eigenvectors. It should be noted that within the NASTRAN system there are several methods available for the eigenvalue extraction; a brief description of these is included, herein, as Appendix B. Not all methods are useful in this instance; also, as an aside, the complex FEER subprogram is not classed as a "standard" method in NASTRAN, at this time.

In view of the impending use of these (somewhat) specialized hydrodynamics equations, described in Appendix A, it is deemed appropriate to comment briefly on them at this point.

As shown in the appendix, the system of equations developed there are a specialization of the expressions for conservation of mass and linear momentum. These are expressions concocted from the Navier-Stokes equations, and the continuity equation, with simplifications, assumptions, and modifications (as noted there). The vertically integrated "shallow water" model, usually employed in tides and circulations work, was developed in about 1960. This system represents an attempt at simplifying the (otherwise) highly complex

*NASTRAN - a computerized system initially designed and developed for the study of problems in structural mechanics.

three-dimensional expressions by an elimination of the vertical (third) dimension coordinate. "Shallow water" here should be viewed as meaning that condition (within a fluid mass) where there is little or no variation in dependent variables with "water depth". Under this umbrella of assumptions the vertical velocity, within the fluid, is neglected, and the momentum expression (in that direction) is replaced by a statement depicting the variation of hydrostatic pressure with depth (the Boussinesq approximation). For most circulation models the internal friction is replaced by surface and bottom frictional actions; and, quite frequently, the convective acceleration terms are ignored. The argument for this latter action is justified on an apparent order of magnitude basis - one which may or may not be adequate. Some of the most recent models have retained these terms suggesting that, in general, the real life situation may not justify the loseing of such terms (in toto).

Most applications of the above mathematical model have occurred in connection with circulation determinations. Unfortunately, very little has been done to establish the necessary and sufficient conditions for a well-posed problem; consequently an occasional inconsistency has been allowed to arise in some problem situations.

This single layer, vertically integrated "shallow water" system does not properly depict conditions when (say) the fluid density has a measureable variation over the water depth; or, when local gradients are present. These latter conditions should more properly be modelled by a multi-layered system, due to (say) density stratification. Some attempts at multi-layered modelling are beginning to appear in the literature; however, the use of such schemes is not wide spread at this time. One example of the need for (say) a two-layer model would be the case where vertical mixing between the epi- and hypo-limnion is reduced because of a (sharp) density gradient. (An example of such is the case of solar heating in a closed basin). Here, the analyst could model the basin, in its depth, as having a top and a bottom fluid layer, physically separated, yet connected through the pressure variation with depth.

It is proper to remark, here, that full three-dimensional models (of the ocean, etc. tides) are very much a "dream" at present. This is not to be construed as a problem

in computational technique - though it would be a large task. Rather, at this time, there does not appear to be a coherent means for handling both surface and internal waves, real time wind and pressure distributions, turbulence exchange and boundary layer phenomenon. Indications, at the moment, tend to favor stochastic processes rather than deterministic modelling, as the way to go, since flow fields and loadings do have a random character.

The topical material, here and above, would seem to deviate from the thesis of the investigation conducted and reported on in this document. As a justification for this deviation, the excuse offered is that the formulation for tides and other hydrodynamic studies departs (sometimes) markedly from the classical Navier-Stokes equations. If one is to appreciate and understand why and how the variations come about, then some delving into the background and evaluation of these events is justified.

Historically, Hansen (6) initially outlined the vertically averaged formulation almost as it is known today. Many investigators have made use of his equations, almost without modification, in their own investigations. Interestingly, Hanson did not include (surface) atmospheric pressure or density variations in his model. He did, however, include viscosity terms - using constant eddy viscosity coefficients - in the (horizontal) momentum equations.

Pritchard (4) has developed a system of vertically averaged equations which are quite like those shown in Appendix A. Both formulations contain the local and convective acceleration terms, both have surface and bottom frictions, and both utilize the Boussinesq approximation to replace pressure gradients with surface (wave form) gradients. Here similarity ceases; Pritchard has included other terms in his formulation; these are expressions for pressure gradients due to fluid density. Neither of these developments, however, incorporate other possible (and likely) content variations (e.g., those attributed to salinity). Generally conditions such as these are defined through additional expressions which must be coupled with and included in the equations system to be solved.

Some others who have contributed to the tides and ocean modelling literature are, as examples: Dronkers (4), who reviewed the harmonic (analysis) methods for tides prediction. (The thrust behind these methods is to use time series in order to derive harmonic functions in known astrodynamic periods. Basically, the input information used here comes from observations data). Leendertse (10) made use of equations very much like Hansens; but without eddy viscosity terms. In this work the importance of central differences - for numerical stability and accuracy - was pointed to. Of course, centered differences, in time, are not to be used for the convective terms if a tridiagonal matrix is to be preserved. Abbott and his co-investigators (1) employed much the same approach as Leendertse; however, they introduced a special, implicit time integration for improved stability and conservation properties. A special feature (included here) allowed for a change in grid size (used for the FD solution). Other procedures, which have been employed in solving these problems include "characteristics methods" and, of course, the relatively new "Finite Element Method". Also, solutions have been developed using semi-analytic approaches; and a variety of schemes have been employed for the time and space integrations needed to achieve desired solution results. (A bibliography worth perusing is presented in Reference (3)). There are numerous persons who have contributed to this topic's literature; investigators such as Hendershott, Plattzman, Laevastu, Simons, Eckert, Defant and Proudman, are among the names which one will find in researching this subject.

The foregoing remarks have been made as a prologue to this report; a description of the investigation is discussed on the following pages. It should not be surmised that the nature of the present study was a revision of tasks undertaken before. Insofar as can be ascertained this work represents a different approach to the problem of determining "tidal frequencies" for enclosed water basins. True, other approaches (see Platzmann) have been reported for this undertaking; however, the procedure utilized here - expecially a use of the NASTRAN system - is indicative of an application of existing software to the solution of a problem type not for which the system was designed. It should be remarked that before this collection of computer programs could be employed, it was necessary to transform the particular governing equations into a format representative of the finite element method.

Having done this, the next move was to make use of selected algorithms for the extraction of eigenvalues and the development of the associated eigenvectors.

In the next section a brief description of, and discussion on, the finite element method (FEM) will be given. There, some of the history, evaluation and characteristics of the method will be noted.

III. THE FINITE ELEMENT METHOD

III.1 Introduction and General Remarks

The Finite Element Method (FEM) is a relatively new procedure which provides a means for approximating solutions to real physical problems. Probably the most significant impact which it has had on numerical methods can be traced to the use of subdivisions, of the solution domain, and to the use of (approximating) polynomial expansions within each subdomain.

For example, consider the situation of the single layered tidal equations being applied to a fluid in a given solution region. First, this spatial domain - an area, in this case - is divided into subregions; and, for each of these subdivisions a function - approximated by a simple polynomial expression in the spatial coordinates - is introduced. In the parlance of the FEM these expressions are known as "trial functions", "interpolating functions", etc. The literature, to date, is not consistent insofar as nomenclature is concerned. Regardless of name, these expansive functions are described in terms of the field variables - at specific loci called "nodes" - where the nodes are (either) specific points on the element's (or subdomain's) boundary; or, possibly, locations within the boundary. Notwithstanding, the nodal values of the field variable(s) plus the interpolating functions, for an element, completely define the "behavior" of the field variable(s) within the particular subdomain. As a consequence, for the finite element representation, nodal values of the field variable(s) become the new unknown(s).

Once these are found, then the interpolating functions are used to "define" the field variables throughout the assembly of elements. An element being identified as a shaped sub-domain.

It should be noted that the choice of interpolating functions is not arbitrary because of compatibility requirements which must be met for the problem at hand. Should these not be satisfied, then convergence to a solution cannot be assured.

Another advantage to the FEM procedure is the inherent ability to formulate solutions, for each of the individual elements, prior to putting them together as a representation of the

total problem. This operation obviously simplifies, and reduces, a complex problem to the study of a series of much simpler problems. In this regard each subdomain (element) has its own approximating polynomial - one which is independent of all other elements. Thus, the entire solution (region) in systematically "assembled" by summing contributions from each and every element.

A more elementary advantage, associated with the FEM, is the several ways the analyst may formulate the required properties for these elements. In fact, there are four basic, but different, approaches which could be used for this purpose.

First, there is the direct approach; this can be traced (back) to the direct stiffness method used in structural analysis. Though quite straight forward this procedure is used only for relatively simple problems. In practice the element properties for this method may be described by means of the more versatile, and more advanced, variational approach.

The variational method, as implied, relies on the calculus of variations; and involves the extremization of some functional. The application of this approach requires a knowledge beyond the introductory level. While the direct approach can be applied to only simple geometries (element shapes); the variational approach is used for both simple and sophisticated shapes. Unfortunately, this latter method is not useful in all instances. As noted in the introduction, the variational method is not useful to many fluid mechanics problems (for example) since an appropriate variational principle has not been developed for the more general case.

The third, and more versatile, approach for deriving element properties is best known as the weighted residual approach. This method commences with the governing equation, and proceeds without any reliance whatsoever on a functional or a variation statement. This procedure extends the FEM (immediately) to those problem situations where no functional is available. (Availability, here, resides in the fact that either a functional does not exist, or that it has not been discovered).

A fourth approach (also) has been applied for real (physical) problems and situations. This method relies on a balance of thermal and/or mechanical energies for the studied system. In this energy balance approach a variational statement is not required; consequently the method considerably broadens the range of possible applications for the FEM.

III. 2 Application Procedure for the FEM

Regardless of which of these approaches one would use, to describe the element properties, the solution to a continuum problem, by the FEM, always follows an orderly step-by-step procedure. These steps are:

(1) A discretization of the continuum - here the "solution region" is divided into elements. A variety of geometric shapes may be used; also, it is possible to utilize differently shaped elements for a given problem. The number, and type, of elements used for a problem is a matter of engineering judgement. Much of the "how" and "why" injected here is a direct consequence of experience - drawn either from personal knowledge, or from the findings of others.

(2) The selection of interpolating functions - for this task the investigators will assign nodal locations and choose function types (to represent field element variations). Here a field variable may be a scalar, a vector, or a higher-ordered tensor. Quite frequently polynomials are chosen for the interpolation function; these are generally the easier type to differentiate and to integrate.

The degree of polynomial (selection) depends on the number of nodes employed; the nature and number of unknowns at each node; continuity requirements imposed at each element and along the boundaries; and, the magnitude of the field variables and their derivatives, which may be unknown at the nodes.

(3) Determination of element properties - after the elements and the interpolation functions have been established, the next task is to determine the matrix equations which will describe the element properties. Here, the analyst will make use of one of the four methods (approaches) described above.

Note: The variational approach, if available, is often the most convenient. However, the approach which is (ultimated) used will, generally, depend on the nature of the problem at hand.

(4) Assembly of the elements - in this operation the properties, and characteristics, of all the elements are systematically joined together. This forms the matrix equations used in obtaining solutions for the entire system (or region). This totality of equations will have a same form as those expressions describing the character of each of the individual elements. Obviously, the collected equations will contain many more terms than the element expressions, since the former relates to all nodes found in the full problem.

One note of consequence: Before these assembled equations can be used to obtain a solution, they must be modified to properly account for boundary conditions.

(5) Solve the system of equations - to this point in the development a system of simultaneous equations has been obtained. These are (next) solved to obtain node values for the unknown field variables.

(6) Make other computations - frequently the investigation will need, require, or desire to have additional information developed from the solution (node) results. (E.g., the velocity field, for a fluid dynamics problem, may be desired. This might be a calculation from a "solved for" pressure distribution, which would have been the field variable).

These listed steps describe the procedure to be followed when solving a problem by the finite element method. What has not been enumerated, as yet, is a categorical listing of problem types solved by this procedure.

III.3 General Applications of the Finite Element Method

Basically, there are three categories of problems solved by the FEM; these are set down according to the nature of the solution.

Category one -- equilibrium problems; problems which have time independent solutions.

Category two -- eigenvalue problems; these lead to steady-state solutions. Most often, here, the analyst seeks to describe natural frequencies and modes of vibration. This category of solutions can be descriptive of both solid and fluid media.

Category three - propagation problems; these are, primarily, problems from continuum mechanics; here a time dimension is added to the solution above.

The fact that the FEM can be used to solve most practical problems does not imply that it is the most practical technique. Actually any of the various techniques, which would apply, has its own advantage(s) and disadvantages. No one technique enjoys the distinction of being "the best" for all problems.

Somewhat in contradiction to the above statement the FEM has received a marked increase in its application to real world problems, since the early 1960's. By 1972 it had become the most frequently used procedure for numerical solutions to continuum problems. In all likelihood future attention, for the method, will be given to modelling nonlinear problems and situations.

III.4 Geometric and Mathematical Considerations

The finite element method (FEM) and the Ritz method are, basically, equivalent procedures. Both schemes make use of "trial functions" as a starting point for finding approximate solutions; both employ linear combinations of these (trial) functions; and, both seek that combination of functions which produce functional stationarity.

The major difference, here, is that the assumed functions, for the FEM, are not descriptive of the entire solution domain. Also, and consequently, these functions do not need to satisfy the problem's boundary conditions (at the element level). However, it is essential and necessary that they do satisfy certain continuity conditions, to be subsequently discussed; necessarily, not all of these conditions need be met at a same time.

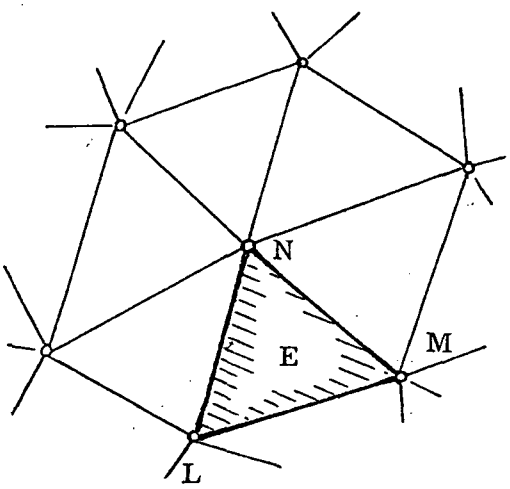
III.4.1 Element Geometry

For two-dimensional problems, as in the case of the tides models studied here, the simplest and, probably most widely used element geometry is the 3-node triangle. This selection is arbitrary; there are no constraints put on the choice of elements; however, the triangle does enjoy significant utility, most likely because of its simplicity in mathematical and geometrical applications. (For a more complete discussion of the choice of elements, the reader is directed to appropriate sections of Reference (8)).

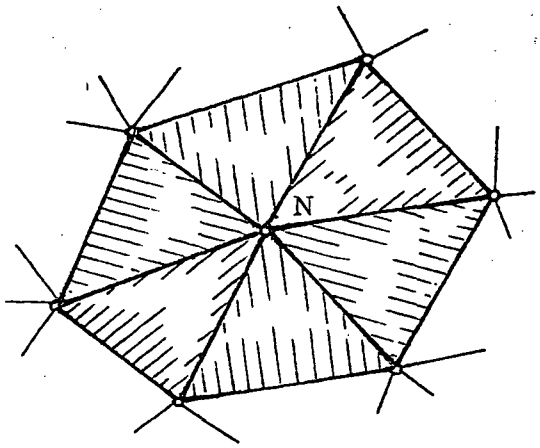
III. 4. 2 Assembly of Element Characteristics

The assembly process, whereby the characteristics of all "elements" are brought together, is an ordered and orderly melding of the element properties based on specific compatibility requirements for the nodes. The nodes represent the junction, or joining, of adjacent elements. At each of these points the unknown nodal variables must be the same for all elements having this locus as a common point. (Actually, this is an important aspect of the FEM - it is the basis for the assembly process). In accomplishing the assembly operation the values of each of the variables, for all elements joining at a node, are added, algebraically; and, this sum is the variable value assigned to that node.

As an aid in distinguishing between the geometric meaning of an element and a set of adjacent elements, joined at a node, the sketch (below) is given. There, the distinction should be apparent; note that triangular elements are used to illustrate both of these situations.



SKETCH A. Illustration of a triangular element (E), having nodes L, M, N



SKETCH B. Illustration showing several elements, joined at Node N.

III. 4. 3 Coordinate Transformations

It was noted above that individual element characteristics are assembled in order to develop those (corresponding) characteristics which describe the entire system, or problem. Sometimes, and especially in the case of complicated geometries, it is convenient (and frequently expedient) to compute the individual element's characteristics using a set of local coordinates. These are called local coordinates since they are typical to the individual element and frequently are different for each of the elements.

Even though these (special) coordinates facilitate calculations for any given element, they are not necessarily compatible with the (local) coordinates for all other elements. Therefore, before the assembly of element characteristics can be carried out, it is imperative that all local coordinates systems be "transformed" into a common system which is compatible for all elements. Such a universal system is referred to in the FEM literatures as "global coordinates". After transforming the local coordinates to global coordinates, the assembly process is carried out without difficulty.

Before entering into a direct application of the finite element technique to the problem at hand, it is felt that some general statements regarding the mathematics of this procedure would be of value. Therefore, the next few paragraphs will address this topic and, hopefully, provide some background, understanding and appreciation for this method.

III.5 A Generalized Mathematics Approach to the FEM

A broad variational approach is that procedure generally used in the derivation of element equations for the finite element method. This is certainly the most convenient procedure to use whenever a classical variational statement exists for a given problem. Unfortunately, for many practical problems, and especially for the more general fluid mechanics situations, classical variational principles are unknown. However, this lack does not negate the use of a finite element approach for this class of problems; rather, it suggests the need for more generalized procedures when deriving the element equations. (See the discussion, presented earlier, describing the four approaches available for such formulations).

In the generalized approach which follows, the finite element equations are derived by a direct and straight forward procedure. That is, the FEM is developed from definitions alone - no recourse to variational principles, etc. is attempted at all.

III.5.1 A Direct Approach to the FEM Solution

One of the basic ideas in the FEM is a separation, or diversion, of the solution domain into a specified number of sub-domains (hereafter referred to as "elements"). These subregions are "joined" to one another only at "nodes" -- i.e., common points, or junctions, in the solution domain -- and at loci on the element boundaries. This concept allows one to reduce the total domain to a "patchwork" of elementary solution regions, each of which is examined as a representative solution element. As a general statement, the domain boundaries are selected to be straight lines - or, planes - consequently, even though the real boundaries may be curved, these are approximated in the solution region by straight lines and/or flat planar segments.

For the fluid mechanics problem to be examined here the nodes are regarded as those loci where the fluid field properties are known. The solution region "elements" and "nodes", however, do not represent a part of the fluid's problem, rather these are only regions, and parts of regions, through which the fluid passes.

The mathematical interpretation given to the FEM requires that the definition of an "element" be generalized; that is, that these sub-domains be considered not as physical regions but more as mathematical entities. In this regard the elements are not viewed as a physical part of the system, but must be thought of as part of the solution domain; one which is partitioned by lines or planes defining boundaries for these elements. In fluids problems the elements are regions over which (say) the pressure field exists and through which the fluid flows. Mathematically the totality of these elements forms a mesh which represents a spatial (solution) domain rather than a material one. Incidentally, it is through this consideration that one is allowed to carry the ideas and concepts gathered from one problem area over into another.

Once the element mesh is described, for a physical problem, then the behavior of the unknown field variable over each of the elements must be approximated by (certain selected) continuous functions. These functions are described in nodal values, representing the variable. Of course, in some cases it is likely that these functions may be described in terms of nodal variable derivatives rather than variable values, per se. These derivatives can be of order greater than the first; however, from a practical standpoint it is not likely that these will be greater than order two. Under any and all conditions, these functions, which are described over the finite element(s), are referred to under various names -- interpolating functions, shape functions, or field variable models. Lastly, it should be mentioned that the full collection of (say) interpolation functions, for the entire solution domain, provides the piecewise approximations to the problem's field variable.

III.5.2 An Example of a Piecewise Approximation

To illustrate a piecewise approximation, for the field variable, consider the function $\Phi(x, y)$ -- a two dimensional field variable.

The aim here is to illustrate how nodal values of Φ can be employed as unique and continuous representations throughout the domain of interest. This will be done by introducing the notation of an integration function.

Let the domain ($\mathcal{D}$) be sub-divided into elementary segments -- each of area Δ (not all the same size or shape) -- with nodes at each of the vertices. It will be assumed that Φ varies linearly over each element; hence these planar areas will (each) contain three nodal points and have three nodal values of the variable Φ. As a consequence for each element (e) the variable will be described by:

$$\Phi^{(e)}(x, y) = \alpha_1^{(e)} + \alpha_2^{(e)} x + \alpha_3^{(e)} y \, . \tag{1}$$

From this expression the constants $\alpha_i^{(e)}$ are to be expressed in terms of the coordinates, locating the nodes, and the nodal values of $\Phi^{(e)}(x, y)$. That is, for each of the nodes (i, j, k) write:

$$\Phi_i^{(e)} = \alpha_1^{(e)} + \alpha_2^{(e)} x_i + \alpha_3^{(e)} y_i \, , \tag{2}$$

(likewise for $(\ )_j$ *and* $(\ )_k$*).*

Next, solving for the α_m coefficients (via, say, matrix inversion); then substituting into the above general expression, for $\Phi^{(e)}(x, y)$, it is found that:

$$\Phi^{(e)}(x, y) = \frac{a_i + b_i x + c_i y}{2\Delta} \Phi_i + \frac{a_j + b_j x + c_j y}{2\Delta} \Phi_j + \frac{a_k + b_k x + c_k y}{2\Delta} \Phi_k \tag{3}$$

wherein

$$a_i \equiv x_j y_k - x_k y_j, \quad b_i \equiv y_j - y_k, \quad \text{and } c_i \equiv x_k - x_j,$$

(with j, k following in cyclic order). Also, Δ is the representation of the element's (planar) area.

Next, define a parameter N, where,

$$N_n \equiv \frac{a_n + b_n x + c_n y}{2\Delta} \, , \quad \text{for } n = i, j, k; \tag{4}$$

then let the field variable $\Phi(x, y)$ be defined by:

$$\Phi^{(e)}(x, y) \equiv \left\{ \begin{array}{c} \Phi_i \\ \Phi_j \\ \Phi_k \end{array} \right\}, \tag{5}$$

consequently, for the element, the $N^{(e)}$ is represented by a row vector,

$$\lfloor N^{(e)} \rfloor \equiv \left\lfloor N_i^{(e)},\ N_j^{(e)},\ N_k^{(e)} \right\rfloor \tag{6}$$

(these quantities are, generally, referred to as the "shape functions" -- they play a most important role in the FEM -- they are the, presently defined, linear interpolating functions for the 3-node triangular element employed here!).

Therefore, over the solution domain ($\mathcal{R}$), which is composed of P elements, the full and complete representation of the field variable, over $\mathcal{R}$, is to be given by:

$$\Phi(x, y) \equiv \sum_{e=1}^{P} \Phi^{(e)}(x, y) = \sum_{1}^{P} \left\lfloor N^{(e)} \right\rfloor \left\{ \Phi^{(e)} \right\}. \tag{7}$$

In this evaluation the nodal values of Φ are known, thus the full solution for $\Phi(x, y)$ is obtained from the complete solution as represented by the surface of interconnected triangular elements.

Note that for this procedure there will not be discontinuities in Φ since values of this field variable -- at any two adjacent nodes defining an element boundary -- will lead immediately to a linear variation along that boundary.

Even though the method outlined above has been for a particular interpolation scheme, namely a linear system; and has been presented in reference to a specific element type (triangular in shape); the expressions shown here are generally valid.

That is, the method shown is valid for the more complicated element shapes, and for interpolation functions which are (also) more complex in format.

The ideas described here are generally definitive of the finite element method; however, the procedure does not allude to the "why" of the procedure. In the next few paragraphs a more general mathematical basis for this scheme will be outlined.

III.5.3 The Galerkin Method

Weighted residuals is one of the methods used to obtain approximate solutions for "systems" described by linear and non-linear partial differential equations. (The procedure is not constrained to the FEM; it is, however, another means for formulating the finite element equations).

Basically, in this operation, there are two steps involved. First, an assumption is made regarding the behavior of the dependent field variable(s). This is done in order that it might approximately satisfy the solution sought after.

The assumed solution is substituted into the partial differential equations; but, being only an approximation, residuals are developed. These errors are, however, required to vanish -- in some average sense -- over the entire solution domain.

During the second step, the equations from the first step are solved. These are specialized, from the general functional form, to a particular function which becomes the approximate solution sought.

As an example we will demonstrate an approximate functional representation for a field variable, Φ, which is governed by the relationship;

$$\mathcal{L}(\Phi) = f, \tag{8}$$

over some specific domain, $\mathcal{D}$. This solution region is assumed to be bounded by a surface, Σ. The function, f, above, is a "known" relation in the independent variables; also, here, we assume the existance of appropriate boundary conditions.

As a first operational step, let the approximate solution $(\hat{\Phi})$ be represented by the following relationship:

$$\Phi \cong \hat{\Phi} \equiv \sum_{1}^{m} N_i C_i \,. \tag{9}$$

(here, the N_i are assumed (shape) functions while the C_i are either unknown parameters, or unknown functions, of one of the independent variables. The "m" functions, N_i, are generally chosen so that they will satisfy the boundary conditions in global coordinates).

Next, using $\hat{\Phi}$ in the governing expressions, we then find that:

$$\mathcal{L}(\hat{\Phi}) - f = \mathcal{R}, \tag{10}$$

where, $\mathcal{R}$, is the residual (error) resulting from this approximation.

Now, according to the method of weighted residuals, the C_i are selected so that $\mathcal{R}$ is "small" over the entire domain $(\mathcal{D})$.

As a consequence of these procedures, there will (now) be "m" linearly independent weighting functions (W_i) which are to satisfy the integral expressions:

$$\int_{\mathcal{R}} [\mathcal{L}(\hat{\Phi}) - f] W_i \, d = \int_{\mathcal{D}} \mathcal{R} \, W_i \, d\mathcal{D} = 0, \qquad \text{for } i = (1, \ldots, m). \tag{11}$$

The result from this calculation implies that $\mathcal{R} = 0$, in some average sense.

The next step, in this procedure, is to solve the weighted equations (above) for the quantities, C_i. After this is done, we are able to provide an approximate representation for the field variable, Φ, according to Equation (9). (One note of caution should be injected here: there is a possibility that difficulties, in representing the solution, could arise since the subjects of convergence and error bounds -- for this approach -- are not too well defined at the present time. It is found that studies on these topics are indeed scarce).

Throughout the literature one will find that there are a variety of weighted residual techniques available for use in the solving of practical problems. However, the one most often employed in the derivation of finite element equations is the error distribution principle; it is more generally known as the Galerkin Method.

In the parlance of present notation, for this method, the weights are chosen to be the same as the approximating functions used to represent the field variable (Φ). Thus, we (now) have $W_i \equiv N_i$; and, according to the Galerkin Method:

$$\int_{\mathcal{D}} [\mathcal{L}(\hat{\Phi}) - f]\, N_i \, d\mathcal{D} = 0, \quad \text{for } i = (1, \ldots, m). \tag{12}$$

Recall that in this expression the field variable, Φ, is approximated as shown in Equation (9).

Here, since Equation (8) is presumed to hold for any point in region $\mathcal{D}$, it also holds for a collection of points defining any arbitrary sub-domain (or element) in the entire solution domain, $\mathcal{D}$.

Recognizing the N_i as interpolating functions ($\equiv N_i^{(e)}$) over the element; and noting the C_i to be undetermined parameters (which will subsequently be the nodal values of the field variable or its derivative(s)), then, by the Galerkin method, the equation(s) governing the behavior of an element can be expressed by:

$$\int_{\mathcal{D}^{(e)}} [\mathcal{L}(\hat{\Phi}^{(e)}) - f^{(e)}]\, N_i^{(e)} \, d\mathcal{D}^{(e)}, \quad \text{for } i = 1, \ldots, k \ . \tag{13a}$$

Here the superscript (e) suggests a restriction of the operation to one element; and, also

$$\hat{\Phi}^{(e)} \equiv \left\lfloor N^{(e)} \right\rfloor \{\Phi\}^{(e)},$$

$f^{(e)} \equiv$ forcing function (over the element),

with

$k \equiv$ the number of unknown parameters assigned to the element.

Since it is required that the residuals vanish (see Equation (11)) then Equation (13) can be rewritten as:

$$\int_{\mathcal{D}^{(e)}} \begin{Bmatrix} N_1 \\ N_2 \\ \vdots \\ N_n \end{Bmatrix} \left[\mathcal{L}\left(\lfloor N^{(e)} \rfloor \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_n \end{Bmatrix} \right) - f^{(e)} \right] d\mathcal{D}^{(e)} = \{0\} \qquad (13b)$$

which is used to yield the finite element equations and the properties of each (or a representative) element.

Before assembling these equations, for the system, from the element equations, it is required that the N_i quantities guarantee inter-element continuity; this is needed in the assembly process.

Note that the higher the order of continuity, which is required for the interpolations, the narrower the choice of functions which is available, for use. Many interpolation functions provide continuity in value; fewer provide continuity in slope (the first derivative); and, only a very few can ensure continuity of curvature. One means of escaping this problem is to change the form of Equation (13a). When one applies the idea of integration by parts, to the integral, an expression in lower ordered derivatives is obtained. This allows the use of interpolation functions requiring lower ordered continuity. There is another (and added) advantage, here, also; when integration by parts is possible, there is (then) available a convenient means to introduce natural boundary conditions (to be satisfied at points on the boundary).

Having outlined the method which will be used here, we shall proceed (next) to a discussion and description of expressions and operations described in terms of "natural coordinates".

III.5.4 Natural Coordinates

The development here, given in terms of natural coordinates, is related to triangular shaped elements outlined on planar surfaces. The goal is to choose a set of

linear coordinates (say ξ_i), particular to the element, which can define the locus of a given point ((x_p, y_p) say) within the element and/or on its boundary (see the sketch below).

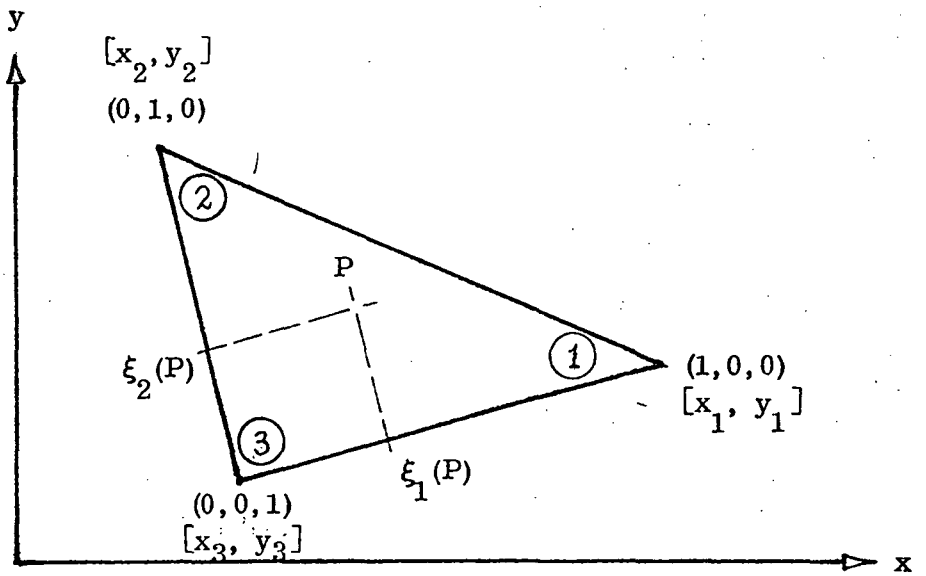
The cartesian coordinates* of a point, in the element, are linearly related to the local coordinates by the following expressions:

$$x \equiv \xi_1 x_1 + \xi_2 x_2 + \xi_3 x_3 = \lfloor \xi_i \rfloor \{x_i\} \tag{14a}$$

and

$$y \equiv \xi_1 y_1 + \xi_2 y_2 + \xi_3 y_3 = \lfloor \xi_i \rfloor \{y_i\}, \tag{14b}$$

wherein the notation $\lfloor \sim \rfloor$ infers a row matrix, $\{\sim\}$ infers a column matrix, and the sequence (1, 2, 3) must be interperted as the cyclic indication of nodes, described in a counter-clockwise manner about the element's perimeter.



A sketch describing the relationship between local and global coordinates for a linear triangle element (of area). The numbers, in parens, denote values for the local coordinates at each of the nodes (i) . Point "P" is an interior locus within the "element".

*Cartesian coordinates are the so-called "global coordinates" in this example. As such they are usually suitable for most physical situations.

In addition to these transformation expressions there is a constraint relation imposed on the weighting functions. This is a requirement that the sum of the parameters must be unity; thus,

$$\Sigma\ \xi_i = \xi_1 + \xi_2 + \xi_3 \equiv 1 \tag{15}$$

(for each element locus).

From this last relationship it is immediately apparent that only two of the natural coordinates can be independent. Of course, in the global system there will be only two independent coordinates since the area elements are planar configurations.

Collecting the above relations into a single matrix format, we write

$$\begin{bmatrix} 1 \\ x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix} , \tag{16a}$$

or

$$\{\sigma\} = [\Lambda]\{\xi\} . \tag{16b}$$

Obviously, in order to describe the vector elements $\{\xi\}$ the expression above must be inverted; i.e.:

$$\{\xi\} = [\Lambda]^{-1}\{\sigma\}, \tag{17}$$

assuming the matrix inverse does exist. When the inverse is obtained, it is described to be:

$$[\Lambda]^{-1} \equiv \frac{[\Lambda]^*}{\text{Det}\,[\Lambda]} ,$$

where $[\Lambda]^*$ is the adjoint of $[\Lambda]$, and Det $[\Lambda]$ denotes the matrix determinant. In a straight forward manner these quantities are easily obtained; and, interestingly the determinant of $[\Lambda]$ is found to be equal to twice the area of the (triangular) element; that is, Det $[\Lambda] = 2\Delta$.

Rewriting Equation (4), as follows:

$$\{\xi\} \equiv [A]\{\sigma\},$$

then an appropriate relationship for the local coordinates, in terms of the global ones, is:

$$\begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix} = \frac{1}{2\Delta} \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \end{bmatrix} . \tag{18}$$

When this last expression is compared to (say) Equation (17), in expanded form, it is found that:

$$a_i = x_j y_k - x_k y_j ,$$

$$b_i = y_j - y_k ,$$

and

$$c_i = x_k - x_j ,$$

(for $i, j, k = 1, 2, 3$ in cyclic order). These indices refer to nodes for the element area.

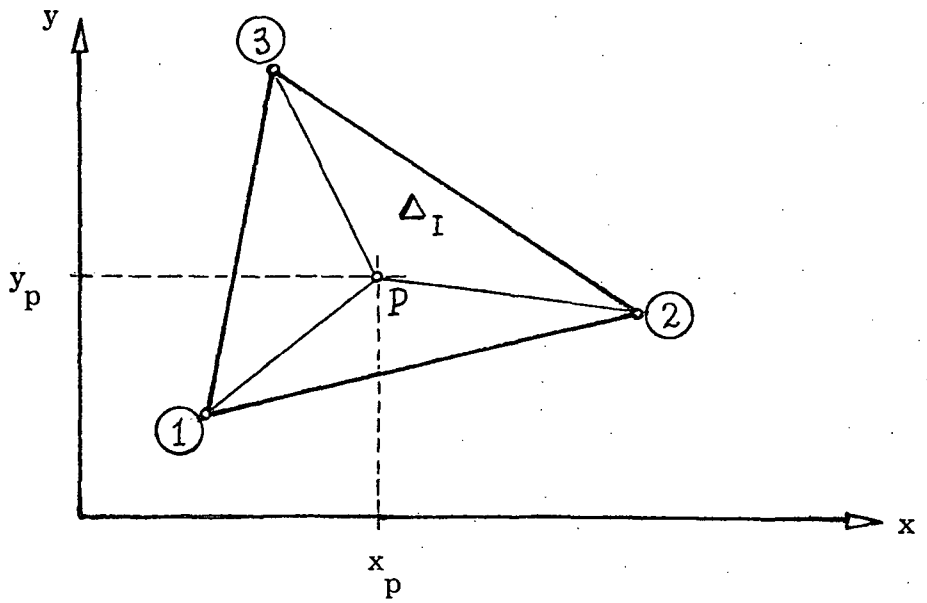
III.5.5 Interpretation of the Local Coordinates

An interpretation of these local coordinates (ξ_i) is rather easily acquired,

geometrically. Considering an interior point (P), in the element domain, then it follows that at $(x, y)_P$:

$$\xi_1 = \frac{1}{2\Delta} (a_1 + b_1 x_p + c_1 y_p) \tag{19a}$$

(see the sketch below).



Sketch, for the element showing an interior point, P, nodes (1), (2), (3) and a sub-element (triangle) area, Δ_I.

When the expression for ξ_1 is written in global (cartesian) coordinates, it is found to be:

$$\xi_1 = \frac{1}{2\Delta} [(x_2 y_3 - x_3 y_2) + (y_2 - y_3) x_p + (x_3 - x_2) y_p] \tag{19b}$$

with

$$2\Delta = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}, \text{ (twice the element's area).} \tag{20a}$$

(Note also that $2\Delta = b_1 c_2 - b_2 c_1$).

Applying this description, for elemental areas, to the triangle formed by "nodes" (2) , (3) , and P, it is apparent that (now)

$$2\Delta_I = \begin{vmatrix} 1 & x_p & y_p \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} , \tag{20b}$$

where Δ_I is the sub-area shown on the sketch above.

A comparison of expressions, and the expansion for ξ_1 (above), shows immediately that

$$\xi_1 = \frac{\Delta_I}{\Delta} ; \tag{21}$$

(a similar treatment would indicate that results for the coordinates ξ_2, ξ_3 would be expressed in area ratios, also. The "appropriate areas" are described using nodes (1) , P, (2) and (1) , (2) , P, respectively).

To better understand the meaning of these coordinates consider the locus P on, first, the line between nodes (2) and (3) ; and, second, let P be coincident with node (1) . In the first case, $\xi_1 = 0$, since Δ_I vanishes; and, in the second case $\xi_1 = 1$, since Δ_I and Δ become identical. It follows, hueristically, that lines of constant ξ_1 are parallel to the boundary line from (2) to (3) , with values varying from zero to unity. (Similar results are obtained for ξ_2 and ξ_3). Thus, any point P for the element is described in terms of the ξ_i -- and these (natural coordiantes) are defined, geometrically, according to intersections of the lines for the ξ_i, as appropriate.

III.5.6 Solution Integrals in Local Coordinates

It should be recognized that the primary objective here is to acquire a problem solution rather than making an attempt at optimizing a method of solution. As a consequence of this fact, we have chosen to work with one of the simplest geometric forms as a representative of the sub-domain (element) shape. Also, we have selected a simple expression for the interpolating (on trial) function. As a result of these choices "the problem" will be expressed in terms of local coordinates and a "field variable". The solution strategy, now, is to establish functional relationships for the typical element, then to sum these (at each nodal locus) as a means to describe the contribution(s) from each and every element (appropriate to the node). The consequence of these operations is a development of equations which will describe the full system.

It will be demonstrated, subsequently, that there are three basic integral forms to be dealt with in the present analysis. Each of these is written in the transformed (local coordinate) space; and, since each integral is to be evaluated over each sub-domain, then the variables involved will be expressed in local coordinates.

These operations, noted above, lead to the following integral types -- expressed in local coordinates -- and have solution forms as noted below:

(a) the integrals, linear in local coordinates, evaluated over an element, lead to results of the form:

$$\int_{\Delta} \{\xi\}\, d\mathcal{D} = \frac{\Delta}{3} \{1\ 1\ 1\}^{T}; \tag{22a}$$

(b) those integrals which are quadratic in the local coordinates have solutions of the form:

$$\int_{\Delta} \{\xi\}\, d\mathcal{D} = \frac{\Delta}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}; \tag{22b}$$

and (c) those which are cubics, lead to:

$$\int_{\Delta} \{\xi\}^T \{\xi\} \{\xi\} \, d\mathcal{D} = \frac{\Delta}{60} * \left[\begin{array}{ccc|ccc|ccc} 6 & 2 & 2 & 2 & 2 & 1 & 2 & 1 & 2 \\ 2 & 2 & 1 & 2 & 6 & 2 & 1 & 2 & 1 \\ 2 & 1 & 2 & 1 & 2 & 2 & 2 & 2 & 6 \end{array}\right] . \qquad (22c)$$

It is of more than passing interest to recognize that there is a generalized integral relationship, which has been developed, accounting for these expressions, and for others. Evaluating a general integral, but employing a specialization for the integrands, leads to results which have been tabulated (see Ref. (1)). This generalized integral for the linear interpolation function, across a triangular element, can be expressed as follows:

$$\int_{\Delta} \xi_1^{\alpha} \xi_2^{\beta} \xi_3^{\gamma} \, d\mathcal{D} = \frac{\alpha! \, \beta! \, \gamma!}{(\alpha+\beta+\gamma+2)!} \, 2\Delta \qquad (23)$$

where Δ represents the area of the element considered. Here the (α, β, γ) are exponents -- as noted in the integral -- and, according to Ref. (8)) , the integral's evaluation, which depends on the integer values assigned to the exponents, leads to the following results:

$$I_{\alpha\beta\gamma} \equiv \frac{1}{\Delta} \int \xi_1^{\alpha} \xi_2^{\beta} \xi_3^{\gamma} \, d\mathcal{D} \equiv \frac{K_1}{K_2} = \frac{2(\alpha! \, \beta! \, \gamma!)}{(\alpha+\beta+\gamma+2)!} \; ;$$

with the evaluations tabulated as follows:

(see next page)

$\alpha+\beta+\gamma$	α	β	γ	K_1	K_2
0	0	0	0	1	1
1	1	0	0	1	3
2	2	0	0	2	12
2	2	1	0	1	12
3	3	0	0	6	60
3	2	1	0	2	60
3	1	1	1	1	60
4	4	0	0	12	180
4	3	1	0	3	180
4	2	2	0	2	180
4	2	1	1	1	180
5	5	0	0	60	1260
5	4	1	0	12	1260
5	3	2	0	6	1260
5	3	1	1	3	1260
5	2	2	1	2	1260
etc.					

This tabulation is included to aid in establishing the integral forms which will be encountered subsequently. Needless to say, this description is valid only for planar area (elements) having the triangular shape, and being spanned by the linear interpolation functions.

The interested reader can expand these tables by incorporating higher ordered functions and more complex element areas.

III. 6 An Application of the FEM

In the foregoing sections of this chapter, an explanation and a development of the finite element method was given. There the procedure was carried out in terms of a general problem statement; in this section, however, the method is applied to a specific problem. This problem, obviously, will make use of equations developed in Appendix A; and, in particular, use is made of a reduced set of the equations of motion (see Equations (16) in the Appendix).

For purposes of reference, the particular expressions are noted below:

(a) the continuity equation (conservation of mass), but expressed in terms of quantity of flow),

$$\frac{\partial \zeta}{\partial t} + \frac{\partial}{\partial x}(h\bar{u}) + \frac{\partial}{\partial y}(h\bar{v}) = 0; \tag{24a}$$

and (b) the conservation of momentum equations (more appropriately referred to as dynamic expressions for the flow field)

$$\frac{\partial \bar{u}}{\partial t} - f\bar{v} + g\frac{\partial \zeta}{\partial x} = F(x, y, t) \tag{24b}$$

and

$$\frac{\partial \bar{v}}{\partial t} + f\bar{u} + g\frac{\partial \zeta}{\partial y} = G(x, y, t). \tag{24c}$$

In these equations the quantities appearing are:

$(\bar{u}, \bar{v})$ ≡ the (averaged) velocity components, in directions (x, y), respectively.

(x, y) ≡ coordinates (described on the mean surface), referred to as global coordinates.

f ≡ the Coriolis parameter ($\equiv 2\Omega \sin\phi$, with ϕ the latitude, and Ω the earth's rotational rate).

g ≡ gravitational attraction (assumed constant in this stucy).

h ≡ local depth of water, measured from the mean surface.

ζ ≡ local water height, measured from the mean surface.

F, G ≡ a representation for general forces which may be acting on the fluid (field).

A more convenient, and particularly useful, form of these expressions is described below. Actually, these equations are the ones employed for the analysis.

First, as a matter of convenience, the flow rates q_x, q_y) are defined; these are given in terms of local depth and averaged velocity components; i.e.,

$$q_x \equiv h\bar{u} \quad \text{and} \quad q_y \equiv h\bar{v}. \tag{25}$$

Next, the homogeneous forms of the expressions above are written (these are in a form necessary to the eigenvalue analysis); that is:

$$\frac{\partial \zeta}{\partial t} + \frac{\partial}{\partial x}(q_x) + \frac{\partial}{\partial y}(q_y) = 0 \tag{26a}$$

$$\frac{\partial q_x}{\partial t} - f q_y + g h \frac{\partial \zeta}{\partial x} = 0 \tag{26b}$$

$$\frac{\partial q_y}{\partial t} + f q_x + g h \frac{\partial \zeta}{\partial y} = 0. \tag{26c}$$

These equations are to be manipulated; they will be recast into a finite element format. Subsequently these FEM equations will be studied and used in the eigenvalue extraction procedures.

III.6.1 Tidal Oscillations in Shallow Water Basins (Formulation)

For the determination of eigenvalues, those values describing free oscillations

in shallow water tidal basins, equations (26) are to be rewritten employing the finite element approach. The resulting expressions, being a set of homogeneous differential equations, are considered as descriptive of a "typical" element. Consequently, they must be summed, in an appropriate manner, as required by the FEM, before the tidal frequencies can be obtained by use of the (more or less) standard procedures for eigenvalue extraction.

When applying the finite element techniques we must first describe appropriate natural coordinates (see Sections III.5.4); denote the nodal (field) variables; and write equations (26) using a proper notation for the eigenanalysis we wish to perform.

Defining, as natural coordinates, the vector $\{\xi\}^{(e)}$, typical to each (individual) field element (per Section III.5.3); and, choosing as nodal variables the quantities

$$\{U\},\ \{V\},\ \{\zeta\},$$

representing the averaged velocities and the free surface heights, for each of the triangular shaped sub-domains, then we define:

$$q_x \equiv \lfloor \xi_1\ \xi_2\ \xi_3 \rfloor \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \end{Bmatrix}, \tag{27a}$$

$$q_y \equiv \lfloor \xi_1\ \xi_2\ \xi_3 \rfloor \begin{Bmatrix} V_1 \\ V_2 \\ V_3 \end{Bmatrix}, \tag{27b}$$

$$\zeta \equiv \lfloor \xi_1\ \xi_2\ \xi_3 \rfloor \begin{Bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \end{Bmatrix}. \tag{27c}$$

For use in subsequent computations, we define the fluid depth (vector) as:

$$h \equiv \left[\xi_1 \; \xi_2 \; \xi_3\right] \begin{Bmatrix} h_1 \\ h_2 \\ h_3 \end{Bmatrix} \quad . \tag{27d}$$

(Here, the subscripts (1, 2, 3) denote vertices of the triangular area taken in a counterclockwise manner).

This choice of variables, constituting a linear functional variation within the element, renders the element compatible and complete, as described in earlier discussions. Thus, since we are dealing only with first order derivatives, in the governing differential equations, then only continuity of a nodal variable value across the element boundaries, is required.

Following the ideas noted in Section III.5.2, it is apparent that for the Galerkin Method, one of the requirements to be satisfied is:

$$\int \left[\frac{\partial \zeta}{\partial t} + \frac{\partial}{\partial x}(q_x) + \frac{\partial}{\partial y}(q_y)\right] g h \, \delta \zeta \, d\mathcal{D} = 0, \tag{28a}$$

$$\int \left[\frac{\partial}{\partial t}(q_x) - f q_y + g h \frac{\partial \zeta}{\partial x}\right] \delta q_x \, d\mathcal{D} = 0, \tag{28b}$$

and

$$\int \left[\frac{\partial}{\partial t}(q_y) + f q_x + g h \frac{\partial \zeta}{\partial y}\right] \delta q_y \, d\mathcal{D} = 0. \tag{28c}$$

Expressing these integrals, according to the FEM procedure, we will acquire the equations representing this problem.

Casting (say) the first of the above expressions into a finite element form, it is necessary to assume that Equations (27) hold, and that the derivatives (above) are expressed as follows:

$$\frac{\partial}{\partial x}(q_x) = \frac{\partial}{\partial \xi_i}(q_x)\frac{\partial \xi_i}{\partial x} \qquad (i = 1, 2, 3)$$

but with $q_x \equiv \lfloor \xi \rfloor \{U\}$, then

$$\frac{\partial}{\partial x}(q_x) = U_i \frac{b_i}{2\Delta} \qquad (i = 1, 2, 3)$$

or

$$\frac{\partial}{\partial x}(q_x) = \frac{1}{2\Delta} \lfloor b_1 \; b_2 \; b_3 \rfloor \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \end{Bmatrix} . \qquad (29a)$$

In a similar sequence of operations it is easy to show that:

$$\frac{\partial}{\partial y}(q_x) = \frac{1}{2\Delta} \lfloor c_1 \; c_2 \; c_3 \rfloor \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \end{Bmatrix} . \qquad (29b)$$

Next, introducing Equations (27) and (29), etc., into Equation (28a) yields, for a general element:

$$g \int_{\beta^{(e)}} \lfloor \delta\zeta_1 \; \delta\zeta_2 \; \delta\zeta_3 \rfloor^{(e)} \begin{Bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{Bmatrix} \lfloor \xi_1 \; \xi_2 \; \xi_3 \rfloor \begin{Bmatrix} h_1 \\ h_2 \\ h_3 \end{Bmatrix}^{(e)} *$$

$$\left(\lfloor \xi_1 \; \xi_2 \; \xi_3 \rfloor \begin{Bmatrix} \dot{\zeta}_1 \\ \dot{\zeta}_2 \\ \dot{\zeta}_3 \end{Bmatrix}^{(e)} + \frac{1}{2\Delta} \lfloor b_1 \; b_2 \; b_3 \rfloor^{(e)} \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \end{Bmatrix}^{(e)} + \right.$$

(equation continued on next page)

$$+ \frac{1}{2\Delta} \left[c_1 \; c_2 \; c_3 \right]^{(e)} \begin{Bmatrix} V_1 \\ V_2 \\ V_3 \end{Bmatrix}^{(e)} \Bigg) \; d\,\theta^{(e)} = 0. \tag{30a}$$

After integration, the resulting expression is represented as:

$$\left[M_h \right]^{(e)} \left\{ \dot{\zeta} \right\}^{(e)} + \left[B_x \right] \left\{ U \right\}^{(e)} + \left[B_y \right]^{(e)} \left\{ V \right\}^{(e)} = 0 \tag{30b}$$

wherein

$$\left\{ \zeta \right\}^{(e)} = \begin{Bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \end{Bmatrix}^{(e)} ,$$

$$\left\{ U \right\}^{(e)} = \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \end{Bmatrix}^{(e)} ,$$

$$\left\{ V \right\}^{(e)} = \begin{Bmatrix} V_1 \\ V_2 \\ V_3 \end{Bmatrix}^{(e)} .$$

Also, in expanded form:

$$\left[M_h \right]^{(e)} \equiv \frac{g\,\Delta^{(e)}}{60} * \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}^{(e)} \tag{31}$$

wherein

$$a_{11}^{(e)} = (6h_1 + 2h_2 + h_3)^{(e)},$$

$$a_{22}^{(e)} = (2h_1 + 6h_2 + 2h_3)^{(e)},$$

$$a_{33}^{(e)} = (2h_1 + 2h_2 + 6h_3)^{(e)},$$

$$a_{12}^{(e)} = (2h_1 + 2h_2 + h_3)^{(e)} = a_{21}^{(e)},$$

and

$$a_{23}^{(e)} = (h_1 + 2h_2 + 2h_3)^{(e)} = a_{32}^{(e)}.$$

In addition, it is found that

$$\left[B_x\right]^{(e)} \equiv \frac{g}{24} * \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}^{(e)}, \tag{32}$$

wherein the elements of the matrix (b_{rs}) are given by:

$$(b_{rs})^{(e)} = b_s \left[(1 + \delta_{r1}) h_1 + (1 + \delta_{r2}) h_2 + (1 + \delta_{r3}) h_3\right]^{(e)}$$

$$(r = 1, 2, 3)$$

with δ_{ri} representing elements of the Idem matrix, and the b_s being coefficients describing the natural coordinates of the element.

In a like manner it is evident that the matrix

$$\left[B_y\right]^{(e)} \equiv \frac{g}{24} \left[c_{rs}\right]^{(e)} \qquad (r, s = 1, 2, 3) \tag{33}$$

with

$$(c_{rs})^{(e)} = c_s \left[(1 + \delta_{r1}) h_1 + (1 + \delta_{r2}) h_2 + (1 + \delta_{r3}) h_3\right],$$

wherein the c_s are coefficients used to describe the natural coordinates of the element.

Next, make similar substitutions into Equation (28b); then after integration the resulting expression can be expressed by:

$$\left[M\right]^{(e)}\left\{\dot{U}\right\}^{(e)}+\left[B_x\right]^{(e)}\left\{\zeta\right\}^{(e)}-f\left[M\right]^{(e)}\left\{V\right\}^{(e)}=0 \tag{34}$$

with

$$\left[M\right]^{(e)}\equiv\frac{\Delta^{(e)}}{12}\begin{bmatrix}2&1&1\\1&2&1\\1&1&2\end{bmatrix}. \tag{35}$$

Likewise, the corresponding finite element expression corresponding to Equation (28c) is:

$$\left[M\right]^{(e)}\left\{\dot{V}\right\}^{(e)}+\left[B_y\right]^{(e)}\left\{\zeta\right\}^{(e)}+f\left[M\right]^{(e)}\left\{U\right\}^{(e)}=0. \tag{36}$$

Collecting expressions, as these refer to each of the elements, we have the set:

$$\left[M_h\right]^{(e)}\left\{\dot{\zeta}\right\}^{(e)}+\left[B_x\right]^{(e)}\left\{U\right\}^{(e)}+\left[B_y\right]^{(e)}\left\{V\right\}^{(e)}=0, \tag{37a}$$

$$\left[M\right]^{(e)}\left\{\dot{U}\right\}^{(e)}+\left[B_x\right]^{(e)}\left\{\zeta\right\}^{(e)}-f\left[M\right]^{(e)}\left\{V\right\}^{(e)}=0\,, \tag{37b}$$

and

$$\left[M\right]^{(e)}\left\{\dot{V}\right\}^{(e)}+\left[B_y\right]^{(e)}\left\{\zeta\right\}^{(e)}+f\left[M\right]^{(e)}\left\{U\right\}^{(e)}=0\,. \tag{37c}$$

Even though Equations (37) are descriptive of "the problem" for each and every finite element, there is a more useful arrangement of these expressions. This alternate formulation, for a unit element, is developed as follows:

First, defining a nodal variable, say ϕ_1, to be:

$$\left\{\phi_1\right\}\equiv\begin{Bmatrix}\zeta_1\\U_1\\V_1\end{Bmatrix}, \tag{38}$$

then it is convenient to recast Equations (37) into the compact form:

$$\left[A\right]^{(e)} \begin{Bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \end{Bmatrix}^{(e)} + \left[B\right]^{(e)} \begin{Bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{Bmatrix}^{(e)} = 0, \tag{39}$$

with

$$\left[A\right]^{(e)} \equiv \left[A_{rs}\right]^{(e)} \qquad (r, s = 1, 2, 3)$$

and

$$\left[B\right]^{(e)} \equiv \left[B_{rs}\right]^{(e)} \qquad (r, s = 1, 2, 3)$$

where the matrix [A] is real, symmetric and positive definite, while [B] is real, skew symmetric in the Coriolis terms, but symmetric in all other terms.

To illustrate the formation of these two matrices, for a representative element, the matrix compositions are shown below:

$$\left[A\right]^{(e)} \equiv \begin{bmatrix}
a_{11} & & & a_{12} & & & a_{13} & & \\
 & \Delta/_6 & & & \Delta/_{12} & & & \Delta/_{12} & \\
 & & \Delta/_6 & & & \Delta/_{12} & & & \Delta/_{12} \\
a_{21} & & & a_{22} & & & a_{23} & & \\
 & \Delta/_{12} & & & \Delta/_6 & & & \Delta/_{12} & \\
 & & \Delta/_{12} & & & \Delta/_6 & & & \Delta/_{12} \\
a_{31} & & & a_{32} & & & a_{33} & & \\
 & \Delta/_{12} & & & \Delta/_{12} & & & \Delta/_6 & \\
 & & \Delta/_{12} & & & \Delta/_{12} & & & \Delta/_6
\end{bmatrix}$$

As multipliers for the vector of nodal variables

$$\begin{Bmatrix} \dot{\zeta}_1 \\ \dot{U}_1 \\ \dot{V}_1 \\ \dot{\zeta}_2 \\ \dot{U}_2 \\ \dot{V}_2 \\ \dot{\zeta}_3 \\ \dot{U}_3 \\ \dot{V}_3 \end{Bmatrix}$$

(40a)

and

$$
\left[B\right]^{(e)} \equiv
\begin{array}{|c|c|c|c|c|c|c|c|c|}
\hline
 & b_{11} & c_{11} & & b_{12} & c_{12} & & b_{13} & c_{13} \\
\hline
b_{11} & & -\frac{f\Delta}{6} & b_{12} & & -\frac{f\Delta}{12} & b_{13} & & -\frac{f\Delta}{12} \\
\hline
c_{11} & \frac{f\Delta}{6} & & c_{12} & \frac{f\Delta}{12} & & c_{13} & \frac{f\Delta}{12} & \\
\hline
 & b_{21} & c_{21} & & b_{22} & c_{22} & & b_{32} & c_{32} \\
\hline
b_{21} & & -\frac{f\Delta}{12} & b_{22} & & -\frac{f\Delta}{6} & b_{32} & & -\frac{f\Delta}{12} \\
\hline
c_{21} & \frac{f\Delta}{12} & & c_{22} & \frac{f\Delta}{6} & & c_{32} & \frac{f\Delta}{12} & \\
\hline
 & b_{31} & c_{31} & & b_{32} & c_{32} & & b_{33} & c_{33} \\
\hline
b_{31} & & -\frac{f\Delta}{12} & b_{32} & & -\frac{f\Delta}{12} & b_{33} & & -\frac{f\Delta}{6} \\
\hline
c_{31} & \frac{f\Delta}{12} & & c_{32} & \frac{f\Delta}{12} & & c_{33} & \frac{f\Delta}{6} & \\
\hline
\end{array}^{(e)}
\quad \text{As multipliers for the vector of nodal values} \quad
\left\{
\begin{array}{c}
\zeta_1 \\ U_1 \\ V_1 \\ \zeta_2 \\ U_2 \\ V_2 \\ \zeta_3 \\ U_3 \\ V_3
\end{array}
\right\}
\tag{40b}
$$

Equations (40) represent the form of the finite element expressions corresponding to Equations (28). These matrix expressions are valid for the finite elements (individually) composing the solution domain. In order to "solve" the problem it is (next) necessary to collect the totality of these finite element expressions, summing them so that the influence of all elements is represented, and so the interaction of these elements is accounted for. Thus, the influence from adjacent elements is accounted for by algebraically summing the matrix components where the common nodal points are apparent in the system.

III.6.2 The Eigenvalue Problem

Equations (39), after being summed for the individual elements comprising the solution domain ($\mathcal{B}$), represents the problem statement sought after in this formulation.

These equations are, of course, written in global corrdinates, and can be expressed (compactly) as:

$$[A]\{\dot{\Phi}\} + [B]\{\Phi\} = 0 \tag{41}$$

wherein $[A]$ and $[B]$ are the assembly of all matrices $[A]^{(e)}$ and $[B]^{(e)}$, shown by Equation (40). Also, here, $\{\Phi\}$ represents the ordered vector of all nodal variables descriptive of "the problem".

Equation (41) can be used for an eigenvalue analysis quite easily. For instance, let it be assumed that the "nodal variable", $\Phi(t)$, is expressed, generally, as the assembled vector,

$$\{\Phi(t)\} \equiv \{\tilde{\Phi}\} \exp(-\lambda(t), \tag{42}$$

then, after substitution into Equation (41), the result is

$$[B]\{\tilde{\Phi}\} = \lambda [A]\{\tilde{\Phi}\}, \tag{43}$$

which is a complex eigenvalue problem involving real matrices, $[A]$ and $[B]$.

In order to solve this relationship, for appropriate eigenvalues, it is necessary to employ some one (or more) of the procedures applicable to such a problem statement. In the next section one such procedure is outlined, in theory; the application of this method, and others, as these are present to the NASTRAN system is outlined in Appendix B.

The one method described, next, is the so-called Inverse Power Method - with shifts. This scheme is one of the several available to NASTRAN users.

III.6.3 An Eigenvalue Extraction Method

One of the eigenvalue extraction procedures available in the NASTRAN system,

and the one proposed, initially, for this problem is that method referred to as: The Inverse Power Method with Shifts. The few statements noted below provide an elementary outline of this procedure.

In this scheme the basic idea is to shift a "previous" eigenvalue problem, to some (new) central point of interest in the eigenspectrum. In this regard, write (analagous to Equation (43))

$$[B - \lambda_o \Delta]\{\tilde{\Phi}\} = (\lambda - \lambda_o)[A]\{\tilde{\Phi}\}, \tag{44}$$

where λ_o is the "previous" eigenvalue.

Now, writing this problem in its inverse form, we have:

$$\frac{1}{\lambda - \lambda_o}\{\tilde{\Phi}\} = [B - \lambda_o A]^{-1}[A]\{\tilde{\Phi}\}; \tag{45a}$$

or, defining a new matrix $[C]$, where

$$[C] \equiv [B - \lambda_o A]^{-1}[A],$$

then the expression above is replaced by:

$$[C]\{\tilde{\Phi}\} = \frac{1}{\lambda - \lambda_o}\{\tilde{\Phi}\}. \tag{45b}$$

Next, choose a random "starting" vector, say $\{V_1\}$, and perform the iterations indicated below:

$$\{V_2\} = [C]\{V_1\}$$

$$\{\bar{V}_3\} = [C]\{V_2\}$$

$$\vdots$$

$$\{V_k\} = [C]\{V_{k-1}\}$$

where the vectors are normalized after each iteration.

Through this procedure the iterations converge to the eigenvector $\{X\}_s$, say, which is closest to the shift point. The corresponding ratio between successive vectors will be the eigenvalue $(\lambda_s - \lambda_o)$; that is,

$$\{\bar{V}_{k-1}\} \rightarrow (\lambda_s - \lambda_o)\{V_k\}. \tag{46}$$

This statement indicates the converged eigensolution for this problem's representation.

III.7 Boundary Conditions

Boundary conditions, for the current study of water basins, can be generally classed as "open" or "closed". The open condition infers one where the boundary is of a water-water type. In this case the fluid domain under investigation terminates at an adjacent water body.

For closed boundaries the condition is simplest explained as a water-land type; typically this would infer a shoreline.

For these two bounding conditions the state will be obviously different. On the open boundaries it is most feasible to specify water heights and/or velocities; closed boundaries, on the other hand, are best described by assigning a zero velocity normal to the physical boundary (e.g., the shore).

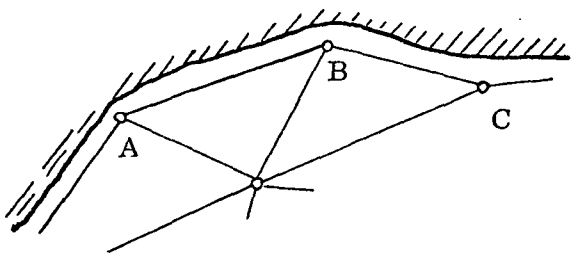
A comment, of some interest at this point, regarding differences in boundary conditions, as these apply to finite difference and to finite element methods, should be made.

The finite difference scheme, since it is associated with "squared" sub-domains, is not difficult to treat. Here adjacent sides (of each sub-region) are orthogonal, hence the velocity components are either nulled (for closed boundaries) or left "free" (if the boundary is adjacent to another sub-domain). Open boundaries, in this scheme, would be specified the same as noted above.

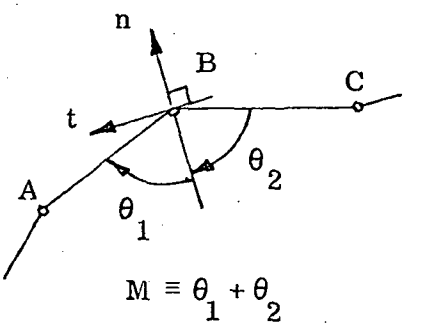
The finite element boundaries, being irregular (e.g., in this study the sub-domains are triangular), makes it difficult to describe a "normal direction" at element nodes. Generally the nodes are formed by adjacent line segments which intersect at "odd" angles. Hence it is not possible to arbitrarily assign a properly directed normal at a node. Certainely it would be incorrect to assign a direction which is orthogonal to either of the intersecting line segments. Hueristically, one would be inclined to assign a direction based on some "weighted average" of the normals at the adjacent linear bounds. Giving consideration to the physics of the present situation it appears that an appropriate weighting should be associated with the mass flow rates over the boundaries. As a matter of fact this concept was introduced by previous investigators (see Reference (15)).

A scheme, based on element geometry and flow conditions, is described (below); this shows one means for determining a weighted, averaged normal direction. This scheme is used in this study to describe the nulled velocity component at a "closed" boundary node. (Incidentally, the method is programmed into the "preprocessor" which develops and describes input data for current problem cases).

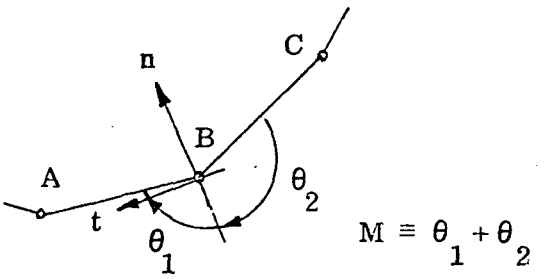
The sketches, shown below, indicate: (a) the probable assignment of "elements" adjacent to a closed boundary; and (b) certain geometries associated with the assignment of nodal normals.



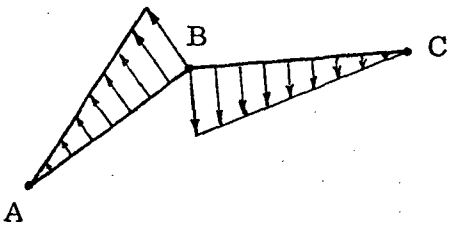
(a) SKETCH showing a closed boundary and probably finite element sub-domains.



(b) SKETCH showing conditions needed to determine a nodal normal at a concave node.



(b2) SKETCH depicting conditions for normal determination at a convex nodal point.



(b3) SKETCH graphically depicting summed flow rates over adjacent element boundaries.

In sketch (b1) the normal, at node B, is assigned by balancing the flow rate, over line segment BC, against the flow over segment AB. The presumption being that over one boundary the flow is <u>into</u> the interior region, while over the other (line) the flow is directed outward. The condition which must be satisfied (by the proper normal n) is simply stated as

$$L_1 \cos \theta_1 = L_2 \cos \theta_2 , \tag{47}$$

where L_1 represents the line (length) AB, and L_2 represents BC. Obviously, the interior angle (M) is defined as:

$$M = \theta_1 + \theta_2 . \tag{48}$$

Incidentally, these same conditions must be satisfied by convex boundary cornors (see Sketch (b2)). (Note: Sketch (b3) graphically illustrates this boundary condition being satisfied -- here the triangular areas depict the balanced noted above).

Obviously there are limits to be imposed on this method for ascertaining a normal direction. Generally speaking these limits are obtained from experience, not from analysis.

Since, in the general case, we attempt to locate a nodal normal direction; and, we null the velocity component in that direction; then we must retain the velocity component which is orthogonal to the local normal (n). Retaining this one component describes the local velocity condition at the node.

Some analysts have chosen to null the full velocity at nodes which occur at a "closed" bounding node. Experience, however, has indicated this is not a best representation; hence the conditions described just above. Also, experience has indicated that when the included interior angle is less than $\pi/2$ it is acceptable (and frequently advisable) to null the nodal velocity. The converse situation applies for convex cornors (see Sketch (b2)).

III.8 The Pre-Processor

Probably the most time consuming and error prone operation associated with the present analysis (and other large scale problems of this class and type) is the preparation of input data for the NASTRAN system. Even for simple problems, such as the test case, described elsewhere, the amount of labor involved in preparing the needed input information is quite conducive to errors and mistakes. In order to reduce this deficiency, and to make the associated work tasks as small as possible, a pre-processor algorithm was built, programmed and used extensively.

The basic purpose of the pre-processor is to take the barest minimum of input information, translate this into useful quantities needed for building and assembling the matrices (required for NASTRAN operations) and to perform those other tasks which can be automated for the problem.

In constructing the matrix components, for each "element" the input requirements can be gleaned from a study of equations (40) and the associated parameters described in Equations (31), (32), (33) and (35). Of course a proper interpertation of the individual components appearing there is assumed. In addition, the boundary conditions will be accounted for by noting the type of boundary associated with each node, and solving for the "closed" boundary conditions requirement (see Section III.7), as this occurs. Lastly, when the individual matrices (see Equations (40)) have been developed, for each node, the final step (for the pre-processor) is to assemble the appropriate elements according to the commonality of nodes in the solution domain. While this may, at first, seem to be a rather cumbersome and unyieldly task, it is not too difficult to perform if one makes haste slowly and goes about this chore in a systematic manner.

Since, in the present problem, we are dealing with water basins which are comparatively small (on a global scale) it is permissible to treat the Coriolis parameter ($f \equiv 2\Omega \sin\phi$) as a constant. Likewise the gravitational acceleration is treated as a fixed parameter; all other physical quantities, however, are nominally treated as variables.

The general information required at each and every nodal point is the following: nodal coordinates (x, y) -- expressed in the so-called "global coordinate system"; the local water depth; and an indication of the type of boundary associated with the nodes. As noted earlier, we are (here) concerned with only two types of boundaries (open and/or closed); all interior points are "general node" types. The general nodes play no special roles in the assembly process.

In order to assemble the matrices, properly, and to build the input deck according to NASTRAN requirements, it is necessary that each "element" be identified, and that the nodes associated with each node be properly "numbered". (Recall that nodes are "numbered" in a counter-clockwise manner about the elements -- one cannot be arbitrary in the sequencing and designation of nodes).

After all of this input information has been fed into the pre-processor, that subprogram generates the needed quantities for each element; assembles the input matrices -- by summing contributions from all elements joining at each node; adjusts those parameters which are indicative of boundary conditions; and, generally, readies these data for processing by the NASTRAN system.

Incidentally, due to certain requirements, for NASTRAN, it was deemed advisable to separate the matrix [A], after assembly, so that a part of this (symmetric) array could be input as another matrix. (These are treated, internally, in an additive fashion; hence the matrix assembly is not altered).

As noted above, the output from the pre-processor is simply the input data required for the NASTRAN system; and, in particular, the form of these data is consistent with the requirements for the eigenvalue routines to be employed.

As an added step in the pre-processing operations, this subprogram was used to construct the entire NASTRAN input deck -- including NASTRAN routine calls, comment cards, etc. -- this greatly alleviates the investigator's work load and reduces one more source of error in the operations.

Before passing to the next section, it should be remarked that the pre-processor developed here is quite similar to the many others used in NASTRAN operations. However, since the current operation represented a very new and unusual utilization of NASTRAN, the investigators were unable to make use of other (available) pre-processors. Hence, the one developed and employed for this study represents an added task in this operation.

III.9. The Test Problem

As a means for checking the formulation of this problem, and to ascertain whether or not the system was operating in an error free fashion, a sample problem was designed. This simple case was chosen prior to the completion of the pre-processor (sub-program) hence all input information was produced by hand. This necessitated numerous parameter calculations; the construction of matrices for each element, and, ultimately, the assembly of the element matrices for the entire solution domain.

The problem selected was one which had a known analytic solution (obviously). However, in order to verify the program's operation, and to learn how to manipulate the NASTRAN system, this case (selected) was exercised on the Goddard Space Center's computational system(s).

As a general description of this test case, the following paragraph should suffice.

The tidal basin selected was square in shape; it had a fixed depth (overall); the boundaries were assumed to be "closed"; gravity and the Coriolis parameters were assigned fixed values each. The one primary variable in this example was the selection of nodes and the subdivision of the solution domain. Not having a criterion for the selection of elements and element size, the procedure adopted was one of incrementing the element count and checking solutions for convergence to acceptable values. Because of geometric symmetry it was not deemed necessary to vary element geometry or size within the basin's bounding perimeter. Hence the elements were systematically oriented and uniform in shape and size, for each sample case studied.

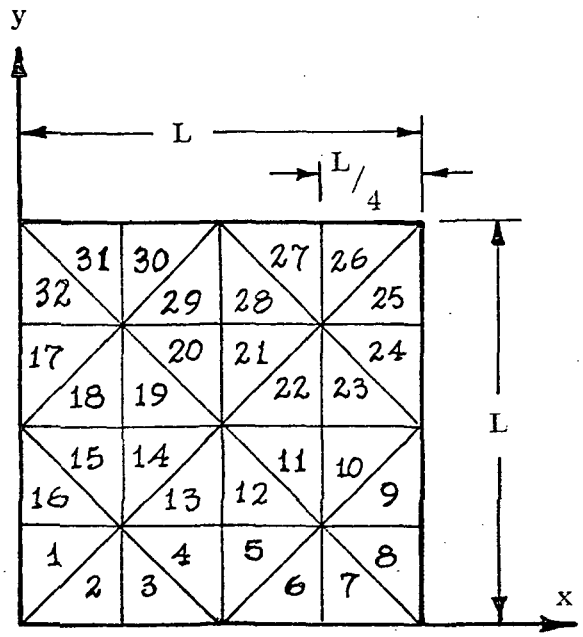
As an indicatian of physical dimensions and parameter sizing for the example cases, the information noted below is given. This is representative of the sample solutions, and is most probably the "best" results acquired during this phase of the investigation.

III.9.1 Problem Statement

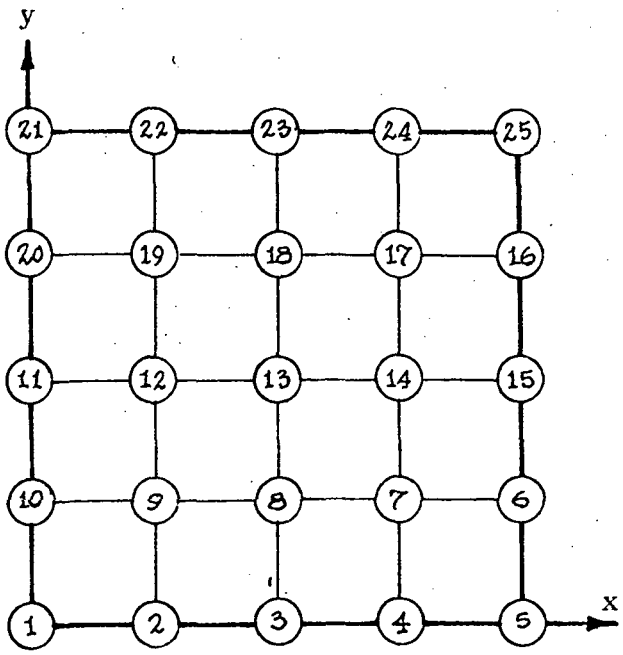
The square basin, for this problem, was assumed to measure 6000 feet on each side. The water depth was so selected that the product "gh" would be set at 9600 ft^2/sec^2; and, the Coriolis parameter (f) was given a value of 6.0×10^{-5} rad/sec -- corresponding to a geographic latitude of approximately 27 degrees.

The geometry of this problem is sufficiently regular to allow the entire formulation to be cast in (so-called) global coordinates. Each sub-domain (element) had a same (surface) area; consequently the number of computations needed to describe an element's input was minimal.

The solution domain, here, was divided into 32 elements; with 25 nodes being assigned. The (general) geometry for this basin, separated into elements, is shown in the sketch below.



SKETCH showing basin and designation of elements



SKETCH showing basin and designation of nodes.

Note that in the problem shown here each element has an area (Δ) which can be expressed by:

$$\Delta = L^2/32,$$

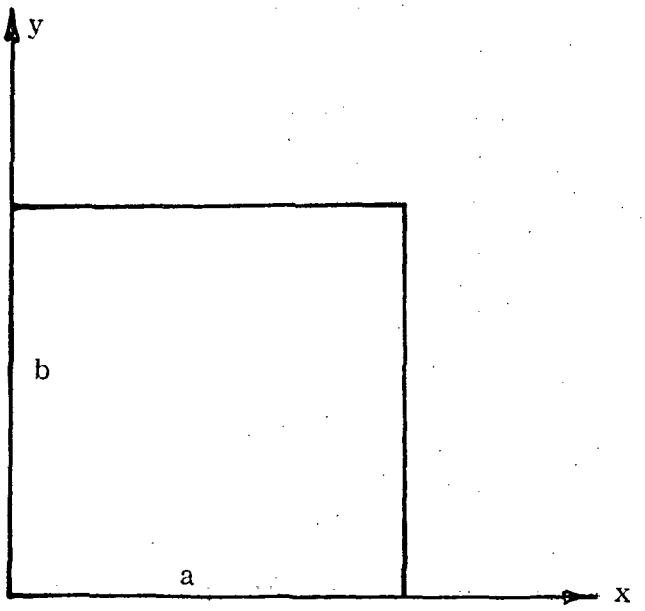
and with $L \equiv 6 \times 10^2$ ft., this yields

$$\Delta = \frac{1}{2} \left(\frac{3}{2} \times 10^3\right)^2 \text{ ft}^2.$$

III.9.2 Analytical Results

As noted earlier the example selected, for verification of the method and the solution proposed in this investigation, was a rectangular (square) basin of fixed depth. The analytical solution for this case is found in Lamb (Ref. (9)), pages 282-284.

The sketch (below) depicts the notation employed here; other terms are described in the text to follow.



SKETCH showing a rectangular basin with sides of lengths "a" and "b".

Modal frequencies for this rectangular basin, normalized by the parameter "gh" -- where these quantities are gravity and water depth, respectively -- are given by

$$\frac{\beta^2}{gh} = \pi^2 \left(\frac{m^2}{a^2} + \frac{n^2}{b^2} \right) \qquad \text{for} \quad \begin{cases} m = 1, 2, \ldots \\ n = 1, 2, \ldots \end{cases} \tag{49}$$

with m, n depicting the modes, per se. The tidal heights, for these modal oscillations are expressed in Lamb, by

$$\zeta = A_{mn} \cos \frac{m \pi x}{a} \cos \frac{n \pi y}{b} \ . \tag{50}$$

For the present case, the rectangular basin, the lowest modal frequency would be expressed as (say)

$$\beta_{10}^2 = gh \, \pi^2 \left(\frac{1}{a^2} \right) \qquad \text{for} \quad m = 1, \quad n = 0. \tag{51}$$

with the associated period, for the oscillation, being given by:

$$T_{10} = \frac{2\pi}{\beta} \tag{52}$$

III.9.3 NASTRAN Results

The output, from NASTRAN, does not give the frequencies directly (see Appendix B). Rather, values described from the NASTRAN eigenvalue routines, are related to the (real) frequencies (e.g., quantities like those computed above) as:

$$\lambda^2 = \pm \, i \, \beta. \tag{53}$$

Since the NASTRAN output presents only the positive imaginary parameters, from the pair(s) above, i.e.,

$$\lambda = + \sqrt{\beta/2} \ (\pm 1 + i), \tag{54}$$

then the printed quantities, appearing from NASTRAN runs, would be arranged with paired parameter terms like

$$\lambda^{+}_{10}, \lambda^{-}_{10} = \sqrt{\beta/2}\,[(+1 + i, -1 + i)] \tag{55}$$

As a consequence of this output format it is evident that there is an automatic check on the roots being extracted by the various eigenvalue extraction methods available (and used) in the NASTRAN system.

III.9.4 Comparison of Results

In the foregoing paragraphs the test model, for this comparison case study, was depicted as a 25 mode, 32 element (solution domain) arranged over the square basin. Suffice it to say that this arrangement was a final selection of the several geometries considered. The basis for presuming this to be an adequate description for the solution domain was the fact that the fundamental modal frequencies appeared to be converged (with this choice of elements and geometry). This point will be illustrated, below, in the tabulation of results, where results for a 16 node model, and the 25 node model, are both noted.

Before presenting these results, and comparing the NASTRAN values with the analytic solution, a comment should be made concerning methods and methodoly used here. First, the results tabulated below are the consequence of obtaining eigensolutions via the DETERMINANT METHOD, in NASTRAN. This scheme (see Appendix B) is obviously a more exact extraction procedure than any of the others available. (At the time when these test case results were acquired, via NASTRAN operations, there was no complex FEER routine in existance. This particular method was being developed, but was not available for use until late in the time frame spanning this investigation. The FEER method, however, was utlized in computing eigenvalues for the real tidal basins near to the termination time of this work).

The procedure found to be most expedient in the extraction of eigenvalues was as follows:

A first estimate of where the eigensolutions resided (in a complex field representing the frequencies, λ_{ij}) was made using the HESSENBERG Routine in NASTRAN (see Appendix B). Once a range for the eigenvalues was established, then a more (or most) refined value could be obtained through the application of the DETERMINANT Method. The philosophy behind this (seemingly cumbersome) procedure is explained in terms of "accuracy" and "error bounds" for the routine. That is, the DETERMINANT method is inherently the most accurate of the several routines, in NASTRAN, by virture of its computational algorithm. The HESSENGERG routine, being a much faster procedure, is, nonetheless, less accurate because of "built in" error bounds residing in the computational algorithm. As a consequence of this, and due to the fact that a fairly well defined "search region" had to be prescribed for the DETERMINANT method, the approach recommended (and used) for eigensolutions consisted of a combination of these procedures. First, eigenvalues were acquired using the HESSENBERG algorithm. Next, these solutions were refined by means of the DETERMINANT method procedure. (Incidentally, a visual check on the accuracy of these solutions is made by studying the root structure of the eigensolutions. Since the NASTRAN procedures print, as output, the eigenvalues as λ (paired) parameters - see the expressions above - then a "converged" root structure is one which lies on a 45^{o} line through the complex plane representing these solutions. Indeed, this fact was employed, throughout this investigation, to ascertain the degree of refinement (or "exactness", if such a term can be loosely implied) in the eigensolutions.

In summary, then, the HESSENBERG algorithm was used (first) to "locate" roots in the (frequency) solutions region. With a "confined" region known, the DETERMINANT method was employed to refine the root structure (as acquired by NASTRAN).

One additional comment should be made, here, concerning the use of the various routines available in the NASTRAN system. Initially, it was planned that the "INVERSE METHOD - with shifts" would be the method to use for extracting eigenvalues. Ultimately

this particular procedure was abandoned, for this investigation, because of its "eratic" behavior during useage. The cause, or causes, of its anomalistic behavior (in eigenvalue determination) could not be ascertained -- nor was much effort expended in the search for such cause(s) -- hence this procedure was abandoned as a candidate method. This, as a consequence, played an obvious role in the eigenvalue extraction procedure adopted for the current investigation.

The tabulation of eigenvalues, etc. listed below represents results taken from NASTRAN runs (made) using the DETERMINANT method algorithm. (It is not deemed necessary, or of use, to list the corresponding results acquired from the HESSENBERG operations). In this tabulation one will find a listing for results obtained using a 16 node model and results acquiared using a 25 node model. The two models are included to illustrate the "convergence" of the solution for the more refined (25 node) case. Also, results for this model are shown with and without Coriolis effects included. (The analytical results, from Lamb, do not include the Coriolis influence). Remember that all eigenvalues (here) were obtained using the DETERMINANT extraction routine.

TABLE I

EIGENVALUES, AND RELATED DATA FOR THE SQUARE BASIN MODEL

Parameter		16 node Model (with Coriolis)	25 node Model (with Coriolis)	25 node Model (without Coriolis)
Eigenvalues, as obtained from NASTRAN	λ_1	$1.3764 * 10^{-1}$	$1.59976 * 10^{-1}$	$1.60013 * 10^{-1}$
	λ_2	$1.5701 * 10^{-1}$	$1.60051 * 10^{-1}$	$1.60013 * 10^{-1}$
	λ_3	$1.5818 * 10^{-1}$	$1.73434 * 10^{-1}$	$1.73449 * 10^{-1}$
	λ_4		$1.73464 * 10^{-1}$	$1.73449 * 10^{-1}$
	λ_5		$1.89901 * 10^{-1}$	$1.8901 * 10^{-1}$
Eigenfrequencies $\beta_i \equiv 2\lambda_i^2$ (rad/sec)	β_1	$3.78895 * 10^{-2}$	$5.11845 * 10^{-2}$	$5.12086 * 10^{-2}$
	β_2	$4.9304 * 10^{-2}$	5.12326×10^{-2}	$5.12086 * 10^{-2}$
	β_3	$5.0042 * 10^{-2}$	$6.01589 * 10^{-2}$	$6.01693 * 10^{-2}$
	β_4		$6.01798 * 10^{-2}$	$6.01693 * 10^{-2}$
	β_5		$7.21248 * 10^{-2}$	$7.212501 * 10^{-2}$
Period of Oscillation $T_i = 2\pi/\beta_i$ (in seconds)	T_1	331.66	122.755	122.698
	T_2	127.44	122.640	122.698
	T_3	125.56	104.443	104.425
	T_4		104.407	104.425
	T_5		87.115	87.115

By comparing the column results shown in Table I, it is quite evident that the more refined model (25 nodes) predicts results which are quite different from those acquired using the more coarse spacing. It remains, yet, to test these results against the analytical predictions. Returning to Equations (49) and (52) it is seen that the

frequency and period of tidal oscillations, in the test basins, are acquired from

$$\beta_i = \frac{\pi}{a}\sqrt{gh\,(m^2 + n^2)} \qquad \text{for } m,\ n = 0,\ 1,\ \ldots$$

and

$$T_i = \frac{2\pi}{\beta_i}$$

wherein, for the present situation $gh = 9600\ ft^2/sec^2$, and $a = 6000$ ft.; the m, n numbers represent the modal characteristics for this basin.

Since the fundamental mode of oscillation is the result most sought for and expected here, then the modal characteristics are either ($m = 0$, $n = 1$) or ($m = 1$, $n = 0$). After these mode shapes, the next shape would be described by the characteristic pair ($m = n = 1$); and so on. Tabulated below are values describing the first three (theoretical) modes.

TABLE II

ANALYTICAL RESULTS FOR A SQUARE TIDAL BASIN

Parameter	Modal Frequency (rad/sec)	Modal Period (sec)
Fundamental, β_1	$5.130199 * 10^{-2}$	122.474
β_2	$5.130199 * 10^{-2}$	122.474
β_3	$7.25518 * 10^{-2}$	86.603

Comparing results in Table I and Table II it is seen that the 25 node model, without Coriolis effect, yields results -- for the fundamental mode -- which are in excellent agreement with the theoretical prediction for this model basin.

Fortified with these results, the investigation was next turned to the task of determining the fundamental mode shape, frequency and period for selected natural basins. The two basins chosen for this study were Lake Erie and Lake Superior.

Before moving to the descriptions and discussions of the natural basins and their free tidal modes of motion, some additional remarks regarding the test problem are in order.

So far, for the test case we have looked at the basic modal frequencies, and the attendant periods of motion only. It would be useful to have some idea of how the tides might appear for these free modes.

In keeping with the results reported in Table I, Table III lists the normalized tidal heights (data acquired from the eigenvectors produced with the eigenvalue analysis, in NASTRAN) and the relative phase (taken with respect to node 1) for each of the nodes. Note that in Table III the relative heights and phases are listed for each node, and for the three periods (from NASTRAN results); $T \cong 122.7$ sec., $T = 104.4$ sec and $T \cong 87.1$ sec. These data are included here to point to the surface behavior, in free oscillations, for the periods noted. As an added aid in clarifying these results, there are two sketches (A and B), found, following the table, which show a representative plot of the data in the columns headed by $T = 122.7$ sec. and 87.1 sec., respectively. On the plots the positive relative heights are shown above the base rectangle and the negative values are plotted opposite in direction.

The shape of the relative amplitude-phase surface is readily seen in the two sketches. The differences in these are obvious and need no comment. (Recall, however, that the information depicted here is relative valued - the data were taken from eigenvector tabulations, for the modes indicated; there is no indication of true tidal heights implied here).

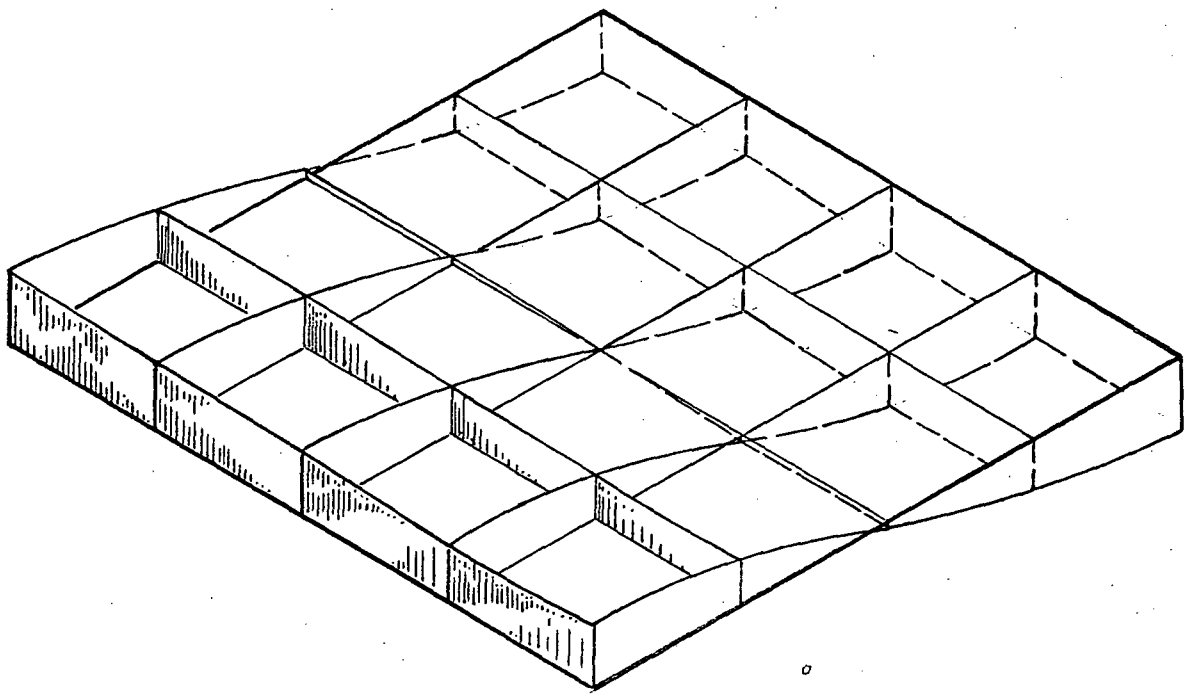
TABLE III

RELATIVE TIDAL HEIGHTS*, WITH PHASE, FROM NASTRAN RESULTS

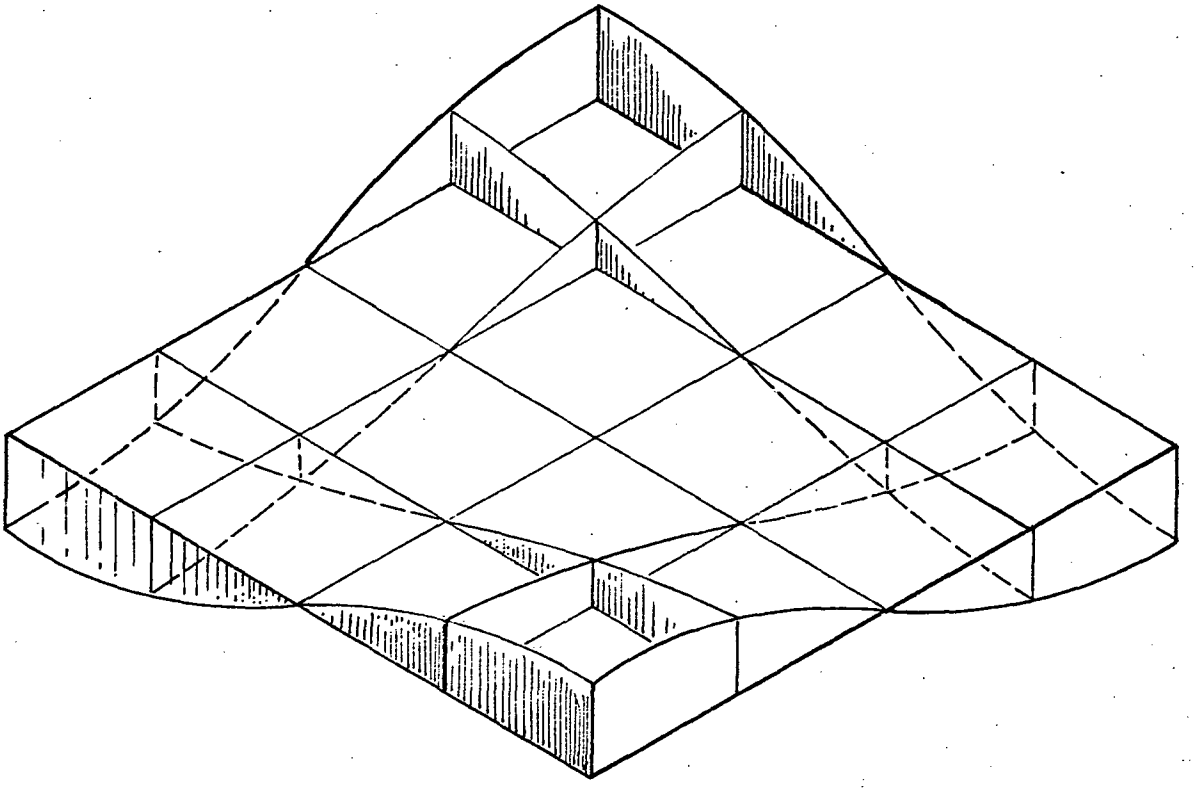
	Relative Amplitude and Phase for modes with Periodicity		
Node No.**	T=122.7 sec.	T=104.4 sec.	T=87.1 sec.
1.	1.111	-0.62	-1.443
2.	1.03	-0.70	-1.127
3.	1.02	-0.012	0
4.	0.909	0.74	1.127
5.	0.93	0.60	1.443
6.	0.60	-1.03	1.127
7.	0.657	-0.42	0.722
8.	0.686	-0.014	0
9.	0.786	0.44	-0.722
10.	0.772	1.0	-1.127
11.	0.09	-0.607	0
12.	0.06	-0.72	0
13.	0	0	0
14.	-0.06	0.72	0
15.	-0.09	0.607	0
16.	-0.772	-1.0	-1.127
17.	-0.786	-0.44	-0.722
18.	-0.686	0.014	0
19.	-0.657	0.42	0.722
20.	-0.60	1.03	1.127
21.	-0.93	-0.60	1.443
22.	-0.909	-0.74	1.127
23.	-1.02	0.012	0
24.	-1.03	0.70	-1.127
25.	-1.111	0.62	-1.443

*Tide heights are normalized, in the Eigenvector descriptions; the sign on each normalized height denotes the relative phase (relative to node 1). These data are for free oscillations without Coriolis effect.

**See sketches in Section 9.1 (here) for node loci on the square basin.



SKETCH A. Graphical representation of the normalized, free oscillation tide heights (and phase) for the mode whose period is T = 122.7 sec.



SKETCH B. Graphical representation of the normalized, free oscillation tide heights (and phase) for the mode with period T = 87.1 sec.

IV. THE CLOSED BASINS STUDIED

IV.1 Basin Models

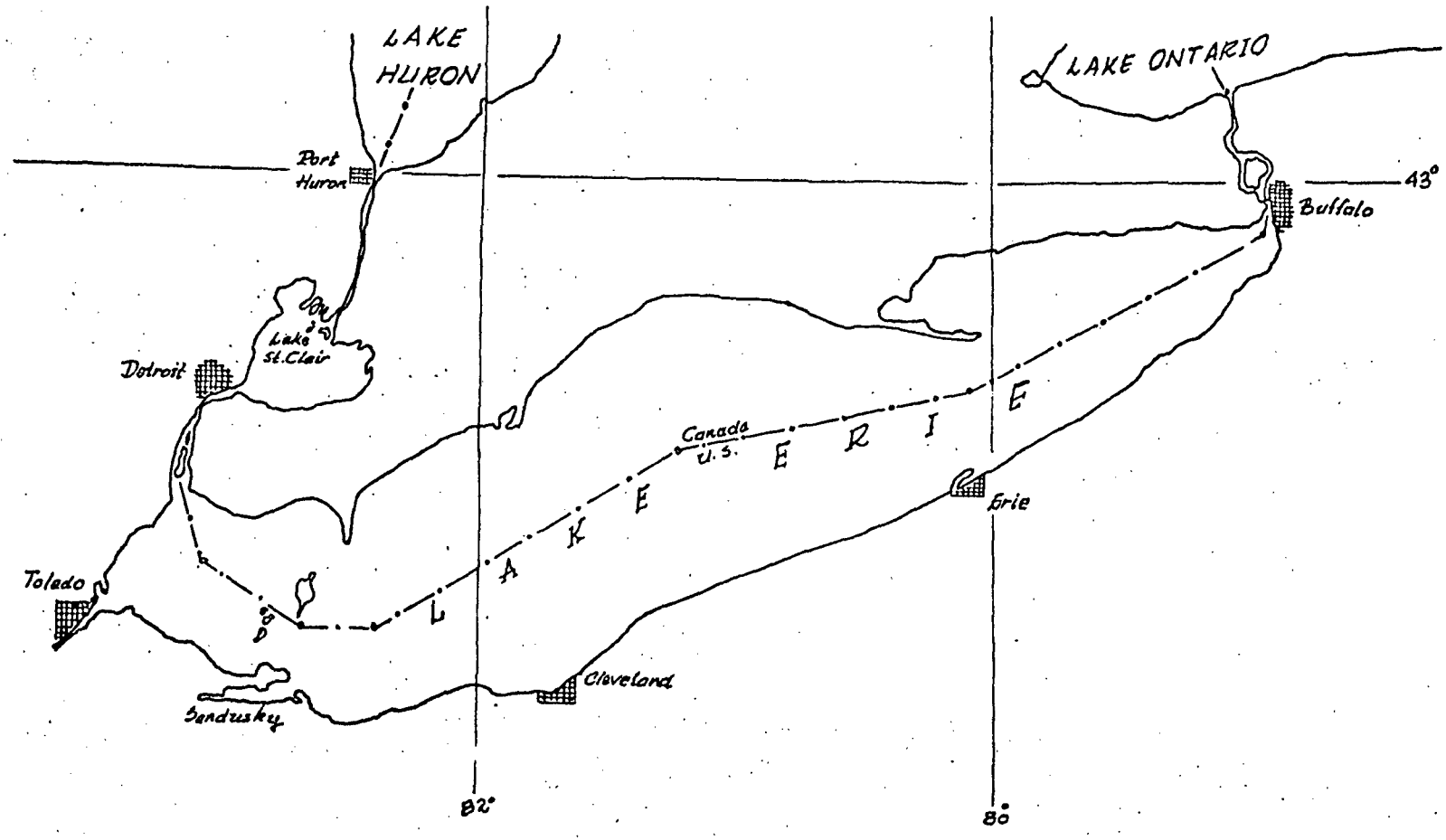
The models of closed tidal basins, numerically studied in this investigation, were three in number - the test basin, Lake Erie and Lake Superior. The test basin - a square tank-like geometry, having a fixed depth - has been described and discussed earlier. Lakes Erie and Superior are, of course, natural basins situated along the central northeastern boundary between the United States and Canada. These bodies of water are, for all intents and purposes, "closed" basins; in addition both have irregular boundaries and are not uniform in depth.

The next few paragraphs will be devoted to a description of these water bodies, with an aim of pointing out those features most pertinent to the present investigation.

(a) Lake Erie. The finite element model, which was constructed to represent Lake Erie in this study, came from information found on the National Ocean Survey Chart No. 3. This chart is a polyconic projection of the lake, scaled at 1:400,000. Geographically, Lake Erie is situated between the 42nd and 43rd (north) parallels, between the latitudes of 79^{o} - 83^{o} west. A sketch of the lake and some of its geographic landmarks is seen below.

Basically, Lake Erie has a rather smooth bottom contour. It does not have abrupt depth changes, except in the near shore and (some) few bay areas; and, comparatively, it is fairly shallow*. The maximum depth of the lake is near thirty fathoms (at the eastern limb of the lake); and, it slopes (upward) to approximately seven fathoms at the western end. (The "western end" here infers the western portion of the main body, not the extreme western portion - from (say) Toledo to Pelee Island, above Sandusky). Over the central portions of this lake, the depth is fairly uniform - varying from 10 to 13 fathoms in its off-shore regions.

*The mean depth of Lake Erie is 19 meters (Ref. (13)).



SKETCH 1. Illustration of Lake Erie and the adjacent geography.

(b) Lake Superior. Lake Superior was modelled using information obtained from the National Ocean Survey Chart No. 14961 (formerly Chart L.S. 9M). This is a Mercator Projection, scaled at 1:600,000 at latitude $47^{o}30'$N.

Lake Superior is a larger body of water than Lake Erie; it is situated between the 46th and 48th parallels, and extends between the longitudes of 84^{o} - 92^{o} west. (See a sketch of this lake, found below, for these and other details).

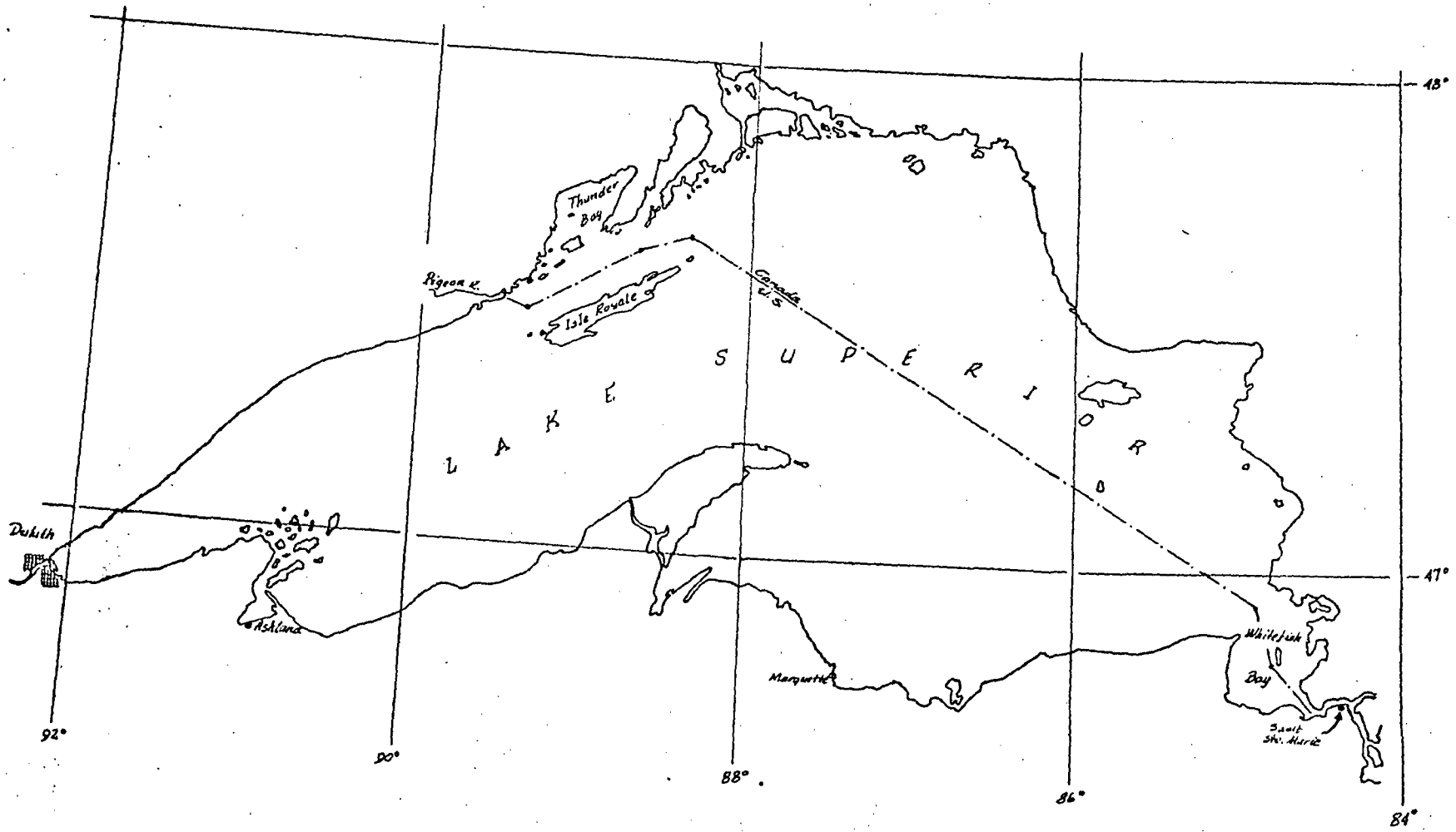
Lake Superior has a more variable bottom topography than does Lake Erie. Also, this basin is the deeper* of the two -- having depths in excess of one hundred fathoms over a sizeable portion of its area. The general contrasting differences in these two bodies of water, aside from water depths, is found in the boundary features and the average width to length variations. While Lake Erie is rather slender and nearly uniform in width, Lake Superior is more oval in character and has a pronounced curvature to its lengthwise median line. For modelling purposes the boundaries and exclusions, in surface topography, for Lake Superior, are more extensive than those for Lake Erie. Additionally, these variations suggest a need for the finite element method's ability to accommodate irregularities in element area sizes and arrangements, for modelling purposes.

IV.2 The FEM Models

The finite element (method) models, constructed to represent these two natural water basins, are seen on Figures 1 and 2, below. There, Figure 1 shows the arrangement of surface "elements" used to represent Lake Erie, while Figure 2 indicates the arrangement for Lake Superior. An examination of these figures and the two sketches (of the lakes) will provide the insight needed to ascertain which features were included and which were excluded. A few comments, however, on each of these models are appropriate here.

Comparing Figure 1 with Sketch 1, one sees that the extreme western end of Lake Erie is excluded from the FEM model. The reasoning behind this stems from the fact that the "bay" adjacent to Toledo - stretching to Pelee Island - is rather shallow,

*Lake Superior has a mean depth of 147 meters (Ref. (13)).



SKETCH 2. Illustration of Lake Superior and its local geography.

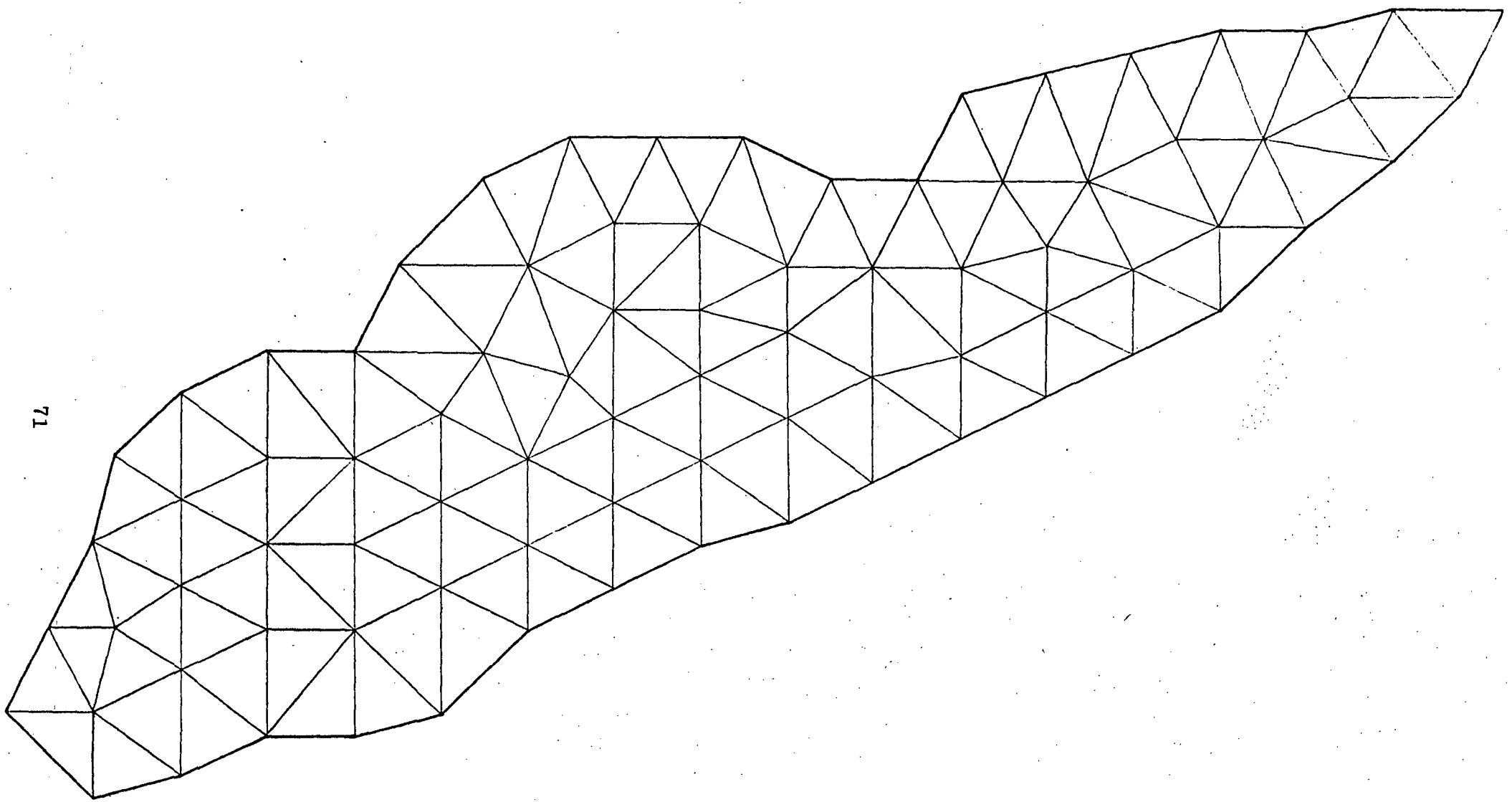


FIG. 1. A reproduction of the FEM subdivisions used to model Lake Erie.

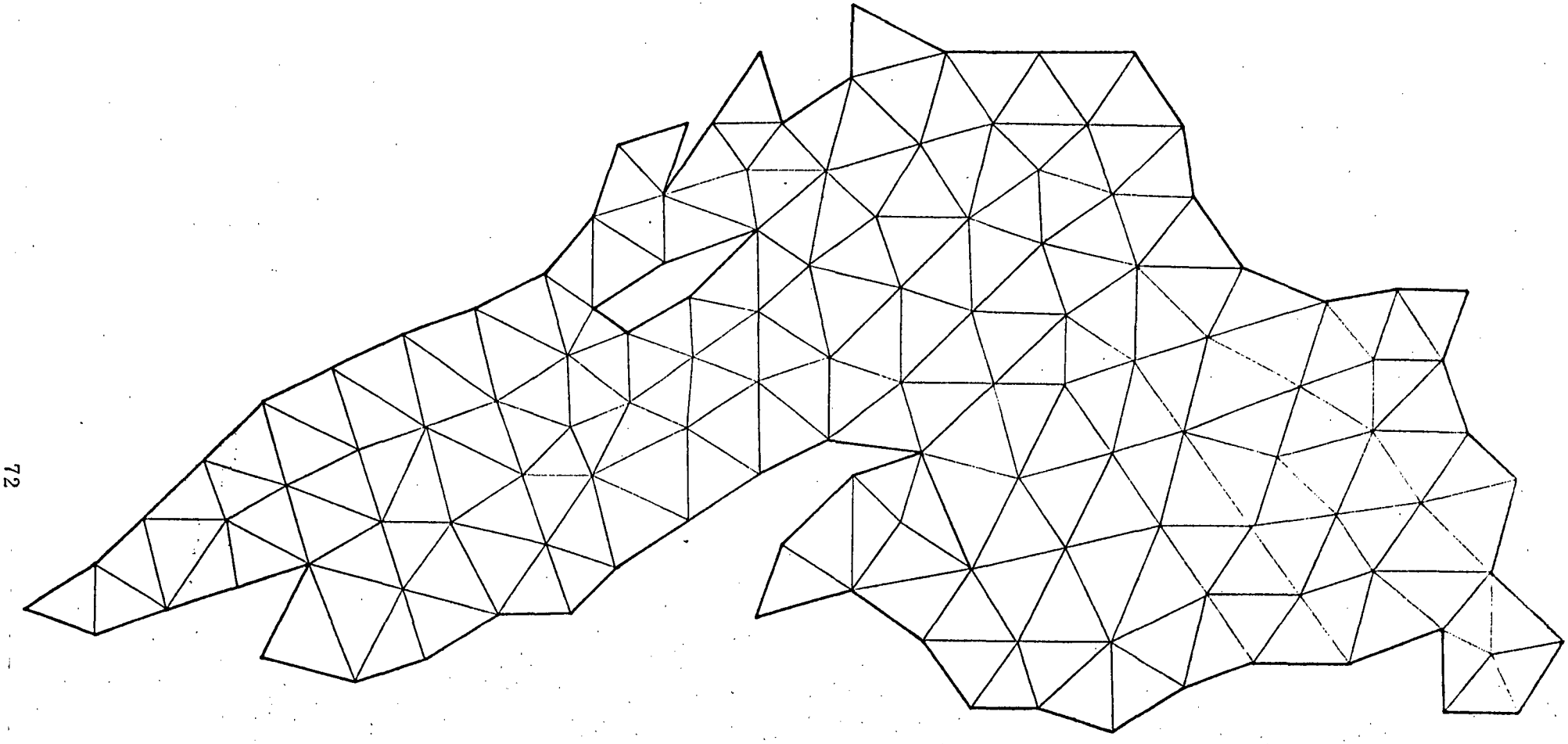


FIG. 2. A reproduction of the FEM subdivisions used to model Lake Superior.

and is separated from the main water body by islands above Sandusky (Ohio). It was felt that these conditions were sufficient to imply a small to negligible response from the (Toledo) bay region; hence that region would not significantly influence the problem's output. Also, the land arm, opposite to Erie, Pennsylvania has been ignored in the FEM model. The belief that this (water) region, behind the arm, was sufficiently open to the main body of water prompted the action taken; also, the fact that this bay region was not deep, and the bottom topography was regular, likewise prompted the investigators to ignore this slender land mass.

The FEM model shown on Figure 1 represents that one which was more extensively exercised in this investigation. It, incidentally, was not the only model of Lake Erie (constructed); there were two additional models described and examined during the course of this investigation. (More will be said about these several models, subsequently).

Next, looking at Figure 2 and Sketch 2 one sees that, here, more of the lake's structure is included in the FEM model. In part this occurred because of the arguments noted above; and, in part, these are included because of land mass size considerations.

For orientation purposes the FEM model is briefly described as follows: Moving from Duluth, along the southern shoreline of the lake, toward Sault Ste. Marie, it is seen that (first) the land -- jutting into the lake -- above Ashland has been included in the FEM model. Next, the tongue of land west of Marquette has also been accounted for; and, finally, Whitehead Bay -- adjacent to Sault Ste. Maries -- has been modelled into the solution region.

Moving around the northern shore, from Duluth to Sault Ste. Marie, it is evident that several of the natural topological features here are included in the FEM geometry. First, Thunder Bay, and the adjacent irregular shorelines, are modelled. Likewise, an exclusion of area -- from the FEM model -- representing Isle Royale, has been accommodated. Following the eastern shoreline, down toward Sault Ste. Marine, we note that some of the larger irregularities are included, but that the small island (in the Lake, proper) has been ignored. The investigators concluded that the analysis, here, was not

sensitive to small land masses and that these could be ignored without inducing serious error. By and large, the water around this island is deep; also the bottom contours are fairly regular, by comparison, and the surface area, per se, is small in terms of total water surface for the problem.

In summary, the two FEM models -- shown here -- for these two of the Great Lakes (Erie and Superior) may be classed as follows:

Closed Basins (Body Name)	Element Shape	Interpolation Fn. Used	No. of Nodes	No. of Elements
Lake Erie	Triangular	Linear	81	122
Lake Superior	Triangular	Linear	124	184

As indicated earlier these were not the only FEM models constructed and examined during the course of this study. Aside from the square basin test (case) there were two additional models of Lake Erie, concocted; however, neither of these served as a primary model in the investigation. Rather, the models noted (just above) were those most extensively used throughout the eigenvalue analysis.

IV.3 Other Models

Before leaving this section it may be well to describe the "other models" -- of Lake Erie -- as a means of acquainting the reader with them, and to indicate some few of the difficulties encountered in this work.

Generally speaking there were three separate model "sizes" developed and examined during the course of this study. One of these was a "larger" model of Lake Erie, while the other was a "smaller" model. Larger and smaller, here, must be reckoned in terms of FEM model size (numbers of nodes and elements) in contrast to physical size (obviously the lake's physical size remains an invariant in the analysis).

In the early production stages of the investigation a rather detailed FEM "map" of Lake Erie was developed. This model, in its finished state, had three hundred and

thirty one nodes; this translated into five hundred and eighty elements, serving as the solution subdomain structure for the entire study region. The reasoning behind such a "large" model was that with this much detail one should surely be able to remove any convergence and continuance problems which might (possibly) be present in the analysis. Also, this much large scale input would surely check the main operating program's supporting software, and would provide confidence in the investigator's ability to handle macro-sized inputs. While these aims were (generally) satisfied it became apparent (not too quickly, unfortunately) that the size of the problem posed by this model led to operational difficulties within the computer system. Some of these problems were eventually solved, others were never completely resolved to any significant degree of satisfaction. (More will be noted regarding this, subsequently).

The second model of Lake Erie, constructed primarily for exploratory studies, and to achieve more rapid machine response, consisted of a FEM model consisting of 43 nodes and (thus) 58 elements. While this collection of sub-domains was not deemed adequate for purposes of the investigational analysis it was, nonetheless, sufficient for the needs of diagnostics and machine check-out.

This second model was developed when it became evident that a less sophisticated computer operations requirement was the only answer for conducting diagnostics and program checkout. In general, this model and the large-scaled one, covered a same area in Lake Erie, but the CPU and I/O machine requirements were drastically reduced for the lesser sized unit. Ultimately this mini-model became a "work-horse" for a variety of study situations which were undertaken in attempts to resolve other difficulties (both real and imaginary) which cropped up.

As an indication of the advantages gained by using the small-scale model, it is worth noting the size of the "input" deck needed for the macro-model, in comparison to that for the mini-model. Even though the input deck generations were automated (via the pre-processor described elsewhere) the mere physical size of this larger system was a deterant to the computational procedure. For comparison purposes, the "input" for the 331 node model was just in excess of 24 thousand cards. On the other hand, the input

deck for the mini-problem was approximately 6000 cards. Also, the macro-problem was one which required the manipulating (including the inversion) of matrices having some nine-hundred plus degrees of freedom. Such a problem (size) required that the computer (360/95 system) be dedicated, almost exclusively, to this problem's solution when it was in the machine and operating. Obviously when problems of this size requirement (per case run), coupled with long running times (up to an hour's operation), are to be handled, the available computer operations time is severely restricted. Generally, such an operational constraint is more factually translated into "week-end runs" -- only -- and then when the machine can be made available.

In addition to these restrictions, it is equally evident that other limiting factors must be considered. For example, the "checking" of the input, to ascertain that what is "fed" to the computer is truly the information desired, can be and is a very time consuming and tedious process. The sheer monotony of scanning twenty-four thousand column entries, of data, is conducive to the committment of errors, also. Thus, the "safeguards" in this operation are also "error sources".

The mere fact that the input is as extensive as it is likewise causes problems. The handling of these data certainly cannot rule out the loss or misplacement of cards; yet such a circumstance, when it occurs, immediately stalls the computational procedure. Experience is a good teacher -- and the investigators, in this work, were taught a number of very good lessons. Unfortunately the bulk of these were learned at the expense of time -- operational time and delays.

IV.4 Problem Dimensions

It has been mentioned, earlier, that the Laplace Tidal Equations, which govern this study's analysis, are written here in the lb-sec-ft system of units. In the initial operations, when the test basin problem was being utilized, these were the actual dimensions used to evaluate terms in the input stream and elsewhere. However, when it came time to construct the input for the actual (lake) basins it was apparent that these same quantities could not be employed because of the resulting numerics evolving from

arithmetic operations used to describe the input parameters. The obvious solution to this dilemna was to "scale" the input numbers and reduce the size of the problem's parameters*.

Consequently, the following scheme was adopted for dimensioning the input; these units were used by the pre-processor in developing an "input deck" for each lake's problems.

Quantity	Input Dimension	Term Definition
(x, y)	10^4 Yards ("Big Yards")	surface coordinates; i.e., point loci, etc. for models
(h)	fathoms	bottom depth; measured at node loci
(t)	seconds	time

With this system of units used for dimensioning the physical parameters, the pre-processor was written so that it computed a proper set of values for the variables, etc. appearing in the governing equations. This meant that the appropriate scaling laws were written into the pre-processor's algorithms, and that the generated output decks was developed with the required (scaled) dimensions**.

Incidentally, in these problems the gravitational constant and the Coriolis parameter were (each) assigned a fixed value for the models constructed. Needless to say, the value of "g" was universally assigned; however, the Coriolis parameter (f) was given a numerical value appropriate to the latitude range for the model considered.

---

*For the sake of accuracy and input numerics it was necessary to go to the "extended size" input block in NASTRAN. The input field, here, is approximately halved -- the input parameters are 16 characters in length compared to the usual 8 character length for normal matrix input (see Appendix C for representative examples).

**At a later time point in the investigation the scaling laws were altered. This was done to verify that some of the problems, which were noted in the program's operation, were not a consequence of the scaling. After producing several check runs, with the new scaling laws, it was evident that this was not the cause.

IV.5 Selected Results

During the course of this study the investigators were alternately running, or attempting to run, sample cases of the closed basins. This was done to find the eigenvalues and, hence, the frequencies for these natural basin's free oscillations. As noted earlier, the models studied (here) were varied (cronologically) from the macro-sized model of Lake Erie (331 nodes), to the mini-sized model (43 nodes), to the operational model (86 nodes). Finally, in the late stages of the study, an operational Lake Superior model was constructed (this has been described in one of the foregoing sections, here).

From a computer operational consideration the production sized Erie model was limited to eighty-two nodes. This came about because of the scheme used to ascertain estimates of the eigenvalues. Here the limit on model size was dictated by the allowable input to the Hessenberg algorithm (see Appendix B, and Conclusions). This statement does not imply that the other models were not used. Actually, the mini-model was extensively exercised for diagnostic purposes and for program check-out. Also, it saw service during the study phase when the investigators were probing for possible remedies to the relatively poor root structure. Recall that these roots were acquired when the Coriolis parameter was included in the governing equations. The macro-model was used, with NASTRAN, in the earlier stages of the study; when it was expected that the eigen-analysis would proceed without difficulty. There were, however, problems encountered; basically these were operational in nature; and, ultimately, it was apparent that the system either became computer bound or that the (selected) eigenvalue routine was behaving erratically. It was at this point in the investigation that the Inverse Method, for eigenvalue extractions, was abandoned.

Near to the end of operations, after the complex FEER sub-program was operating, the macro-problem was reexamined and results were obtained. Unfortunately, the root structure (here) was likewise of not-acceptable-quality; consequently those roots (too) are not reported herein.

It has never been clearly ascertained where the difficulties lay with the use of these various eigenvalue routines, in NASTRAN, and in the study of Laplace equations containing a Coriolis influence. A number of remedies and "fixes" were tried, in the hope that some inkling of the cause could be uncovered. Unfortunately, there were no clear indicators, and time and resources came to an end before an understanding could be gained. The two remaining candidates trials, for alleviating these difficulties, but which were not tried, are likely to be the most expensive of all. However, these were not attempted. They would require an extensive modification to the program and would necessitate the development of a wholly new input string. The essence of these changes can be simply given as follows: It is expected that an explicit statement for irrotationality needs to be included in with the governing equations; and, that some relaxation of the globally defined nodal net should be included. Hopefully, with these alterations, the eigenvalues could be brought into the expected range, and the problem could be solved as initially intended.

The comments, above, provide some broad coverage of the problems encountered during the course of this investigation. Passing now to the positive results, a few paragraphs outlining these -- with comments -- will be written next.

Neglecting the Coriolis parameter, it was possible to determine eigenvalues and to convert these results into meaningful information regarding the free oscillation modes. Remembering that the Great Lakes models are those described in Section IV.2, above, and that comparisons are made with information presented in Reference (13), then the following summary is given:

(a) The eigenvalues, obtained here, for the Lake Erie model, give the first three free modal oscillations frequencies as:

$$\beta_1 = 1.20215 \times 10^{-4} \text{ rad/sec}$$

$$\beta_2 = 1.7617 \times 10^{-4} \text{ rad/sec}$$

and

$$\beta_3 = 2.46702 \times 10^{-4} \text{ rad/sec}$$

These frequencies denote free oscillation periods of:

$$T_1 = 14.518 \text{ hrs.}$$

$$T_2 = 9.907 \text{ hrs.}$$

and

$$T_3 = 7.07 \text{ hrs., respectively.}$$

Turning next to results from other studies, it is found that the observed periods for the first three longitudinal modes of Lake Erie are:*

$$T_1 = 14.4 \text{ hrs.}$$

$$T_2 = 9.1 \text{ hrs.}$$

and

$$T_3 = 5.9 \text{ hrs.}$$

In that same reference Platzman calculated a period for the fundamental frequency as:

$$T = 14.86 \text{ hrs.}$$

for Lake Erie. This can be compared to the period, calculated from a spectral analysis, by Platzman and Rao**, which is reported to be:

$$T = 14.38 \text{ hrs.}$$

The analyses conducted by Platzman, Rao and others, imply that the channel approximation is not too much in error for Lake Erie (recall that this body of water is "long and slender").

(b) The analysis for Lake Superior is not so well documented (in comparison to (say) Lake Erie). It is evident, from the geometry of this basin, that it is <u>not</u> so amenable to the channel approximation. Nonetheless, Rockwell (Theoretical free

*Platzman, G.W. & D.B. Rao, Spectra of Lake Erie water levels, J. Geophys. Res.69,1964.

**See reference by same authors. In the past, spectral analysis has been the mainstay for most investigators.

oscillations of the Great Lakes, in Proc. Ninth Conference on Great Lakes Research, Pub. 15, Great Lakes Research Division, U. of Michigan, 1966) used this approximation to obtain a "fundamental period for Lake Superior"; he reports this to be T = 7.49 hours. Subsequently, Platzman (using his resonance iteration method) obtained a period of 7.84 hours. Following this work, Mortimer and Fee reported a fundamental mode period somewhere between 7.79 and 7.89 hours; in their analysis they used a spectral analysis also. (Mortimer, C.H. and E.J. Fee, Free-Surface Oscillations and Tides of Lakes Michigan and Superior. Special Report Center for Great Lakes studies, Univ. of Wisc., 1972).

In the present study, the eigenvalue analysis yielded periods for the first three free oscillations of:

$$T_1 = 14.637 \text{ hrs.}$$

$$T_2 = 8.438 \text{ hrs.}$$

and

$$T_3 = 6.98 \text{ hrs.}$$

The second period (here) is apparently that which corresponds to the values obtained by these other investigators. In retrospect, this value is high by some seven percent as compared to Platzman's resonance iteration method. Evidentally, the period (T_1) does not have a counterpart from either spectral analyses or from other investigations. From a plot of the eigenvector results, the tidal amplitudes from this period appear to be reasonable; and, the phase of the motion is well within the realm of expectations. Why this mode would appear here, and not be accountable from other studies, is not known.

This extra root type should not be passed over lightly. The eigen-analysis conducted here was not always reliable; but the technique which was employed for those determinations without Coriolis effect could not be faulted. As an illustration of the method's sensitivity the following situation is mentioned.

During the study of Lake Erie there was one particular case where the eigenvalues (extracted) contained a root which could only be interpreted as a divergence. Physically this appeared to be a case where the lake could be "emptied". Obviously such a root type must be in error; and, after a careful examination of the input, it was found that one boundary condition card (specifying a zero velocity at a water-land type node) had been omitted (or deleted) from the deck. By leaving out this card the lake had an "opening" where it could be dumped -- thus the error in output; and, correspondingly, the sensitivity of this analysis to such conditions and/or omissions.

As an indication of how the free surface responded to the oscillatory modes, pseudo-plots of the tidal amplitudes (plus phase) have been constructed for the calculated mode types noted above. On the six figures to follow the relative amplitudes (plus phase) for the tide heights at all nodes (of the models) are shown. These figures are duplicates of the FEM nets shown on Figures 1 and 2, except that (here) the internal connectors between nodes are eliminated. The noted (interior and exterior) nodes are indicated by "dots", and the appropriate relative amplitudes and phase are indicated (numerically) adjacent to each nodal location. These graphical depictions are included here for reference purposes (only). The relative phase, at each locus, is indicated by the sign assigned to each numerical indicator.

The first three figures are for Lake Erie (at the modal periods noted earlier); and, the last figures describe the above conditions for the three modal periods noted for Lake Superior.

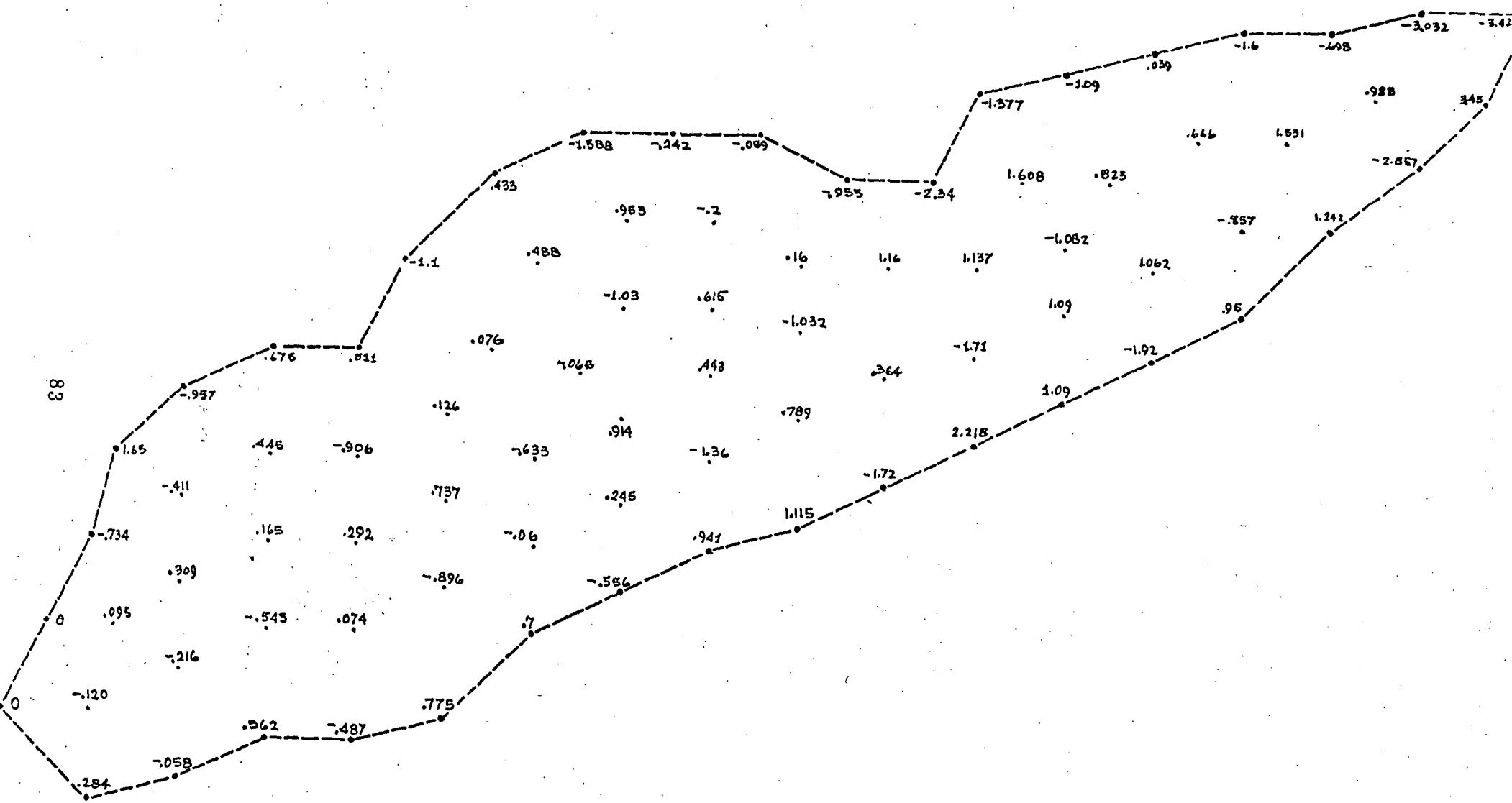


FIG. 3. A description of the eigenvector relative amplitudes, with phase, for Lake Erie, as described by the present analysis at a modal period of 14.518 hours.

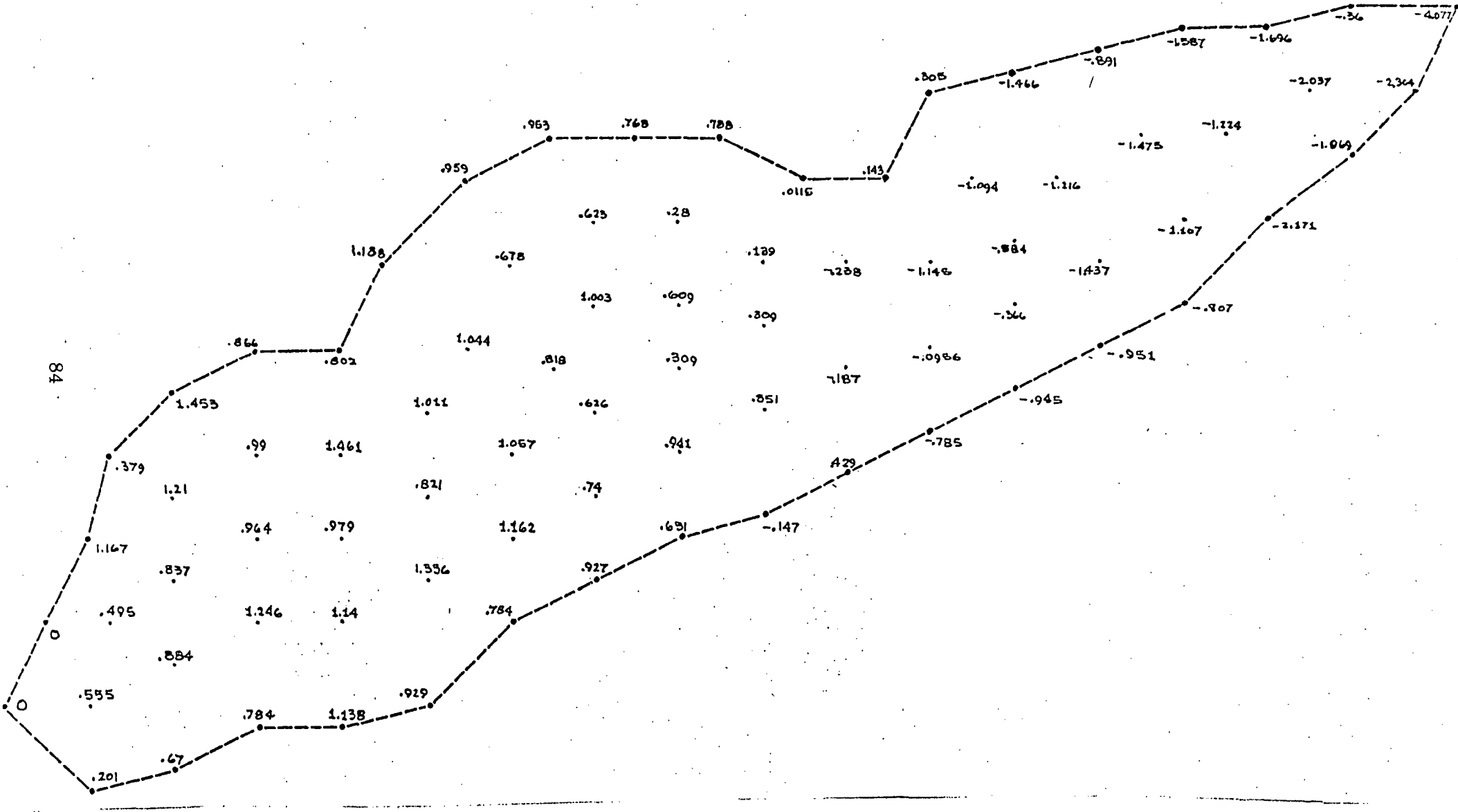


FIG. 4. A description of the eigenvector relative amplitudes, with phase, for Lake Erie, as described by the present analysis at a modal period of 9.907 hours.

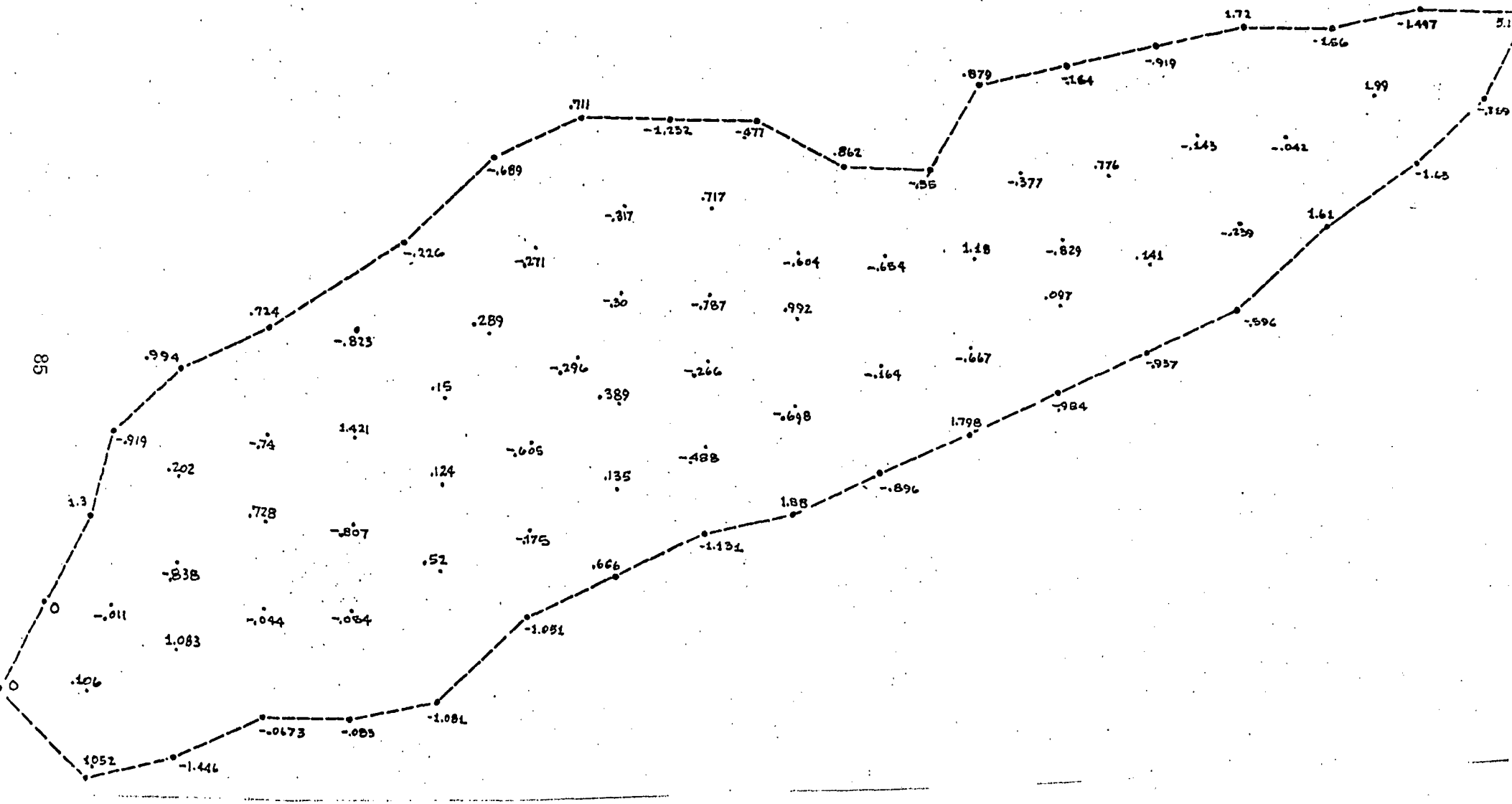


FIG. 5. A description of the eigenvector relative amplitudes, with phase, for Lake Erie, as described by the present analysis at a modal period of 7.07 hours.

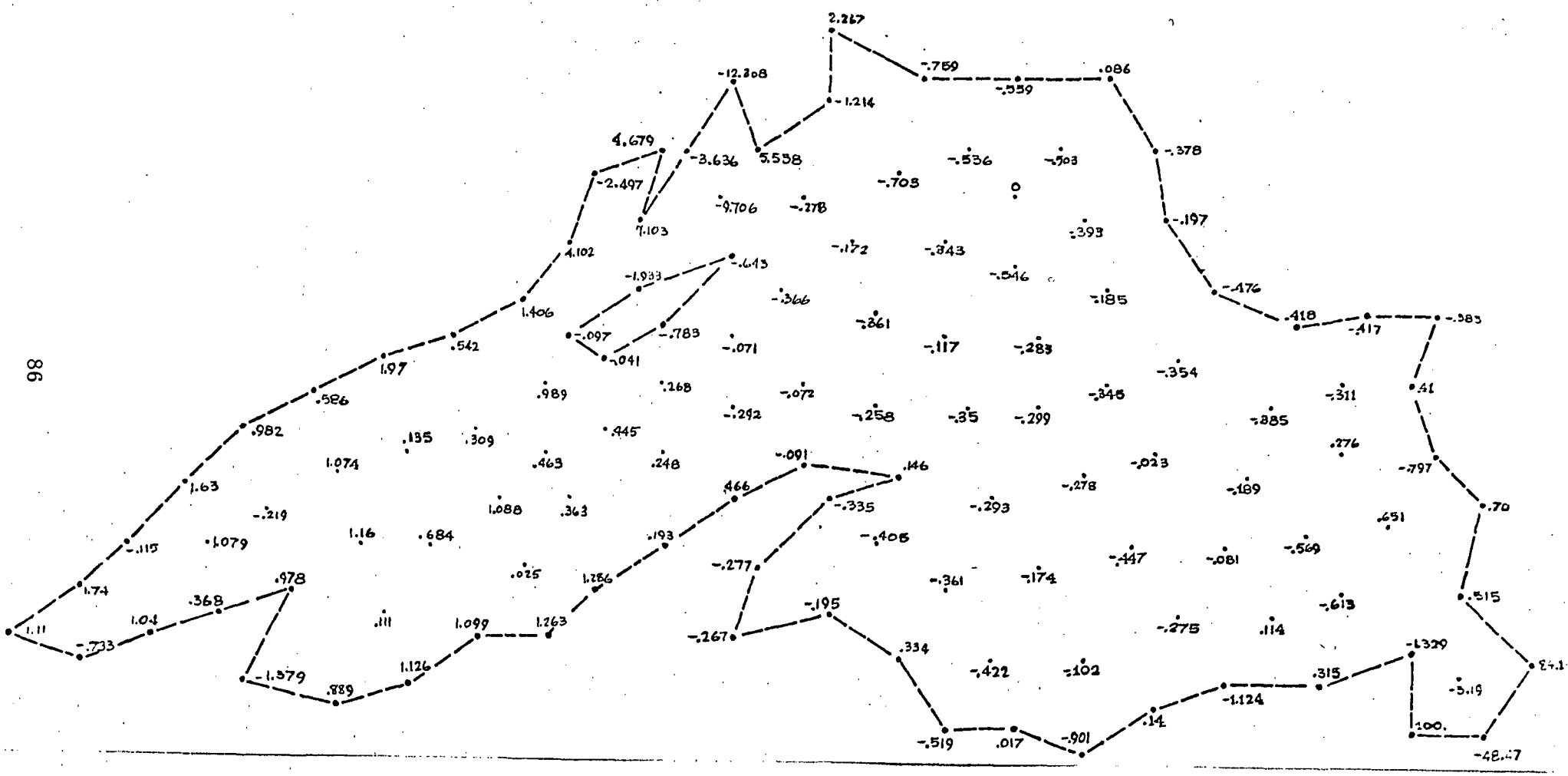


FIG. 6. A description of the eigenvector relative amplitudes, with phase, for Lake Superior, as described by the present analysis at a modal period of 14.637 hours.

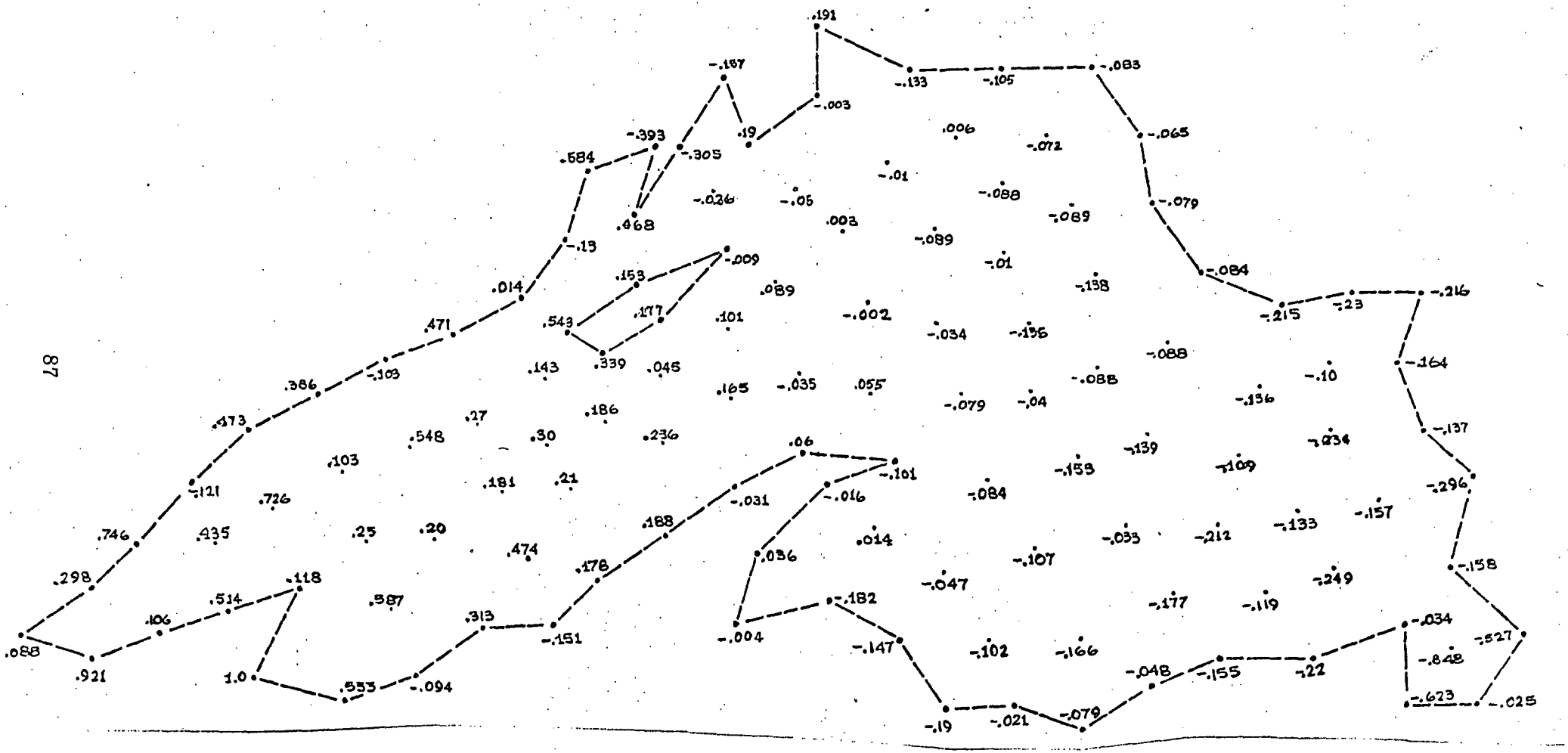


FIG. 7. A description of the eigenvector relative amplitudes, with phase, for Lake Superior, as described by the present analysis at a modal period of 8.438 hours.

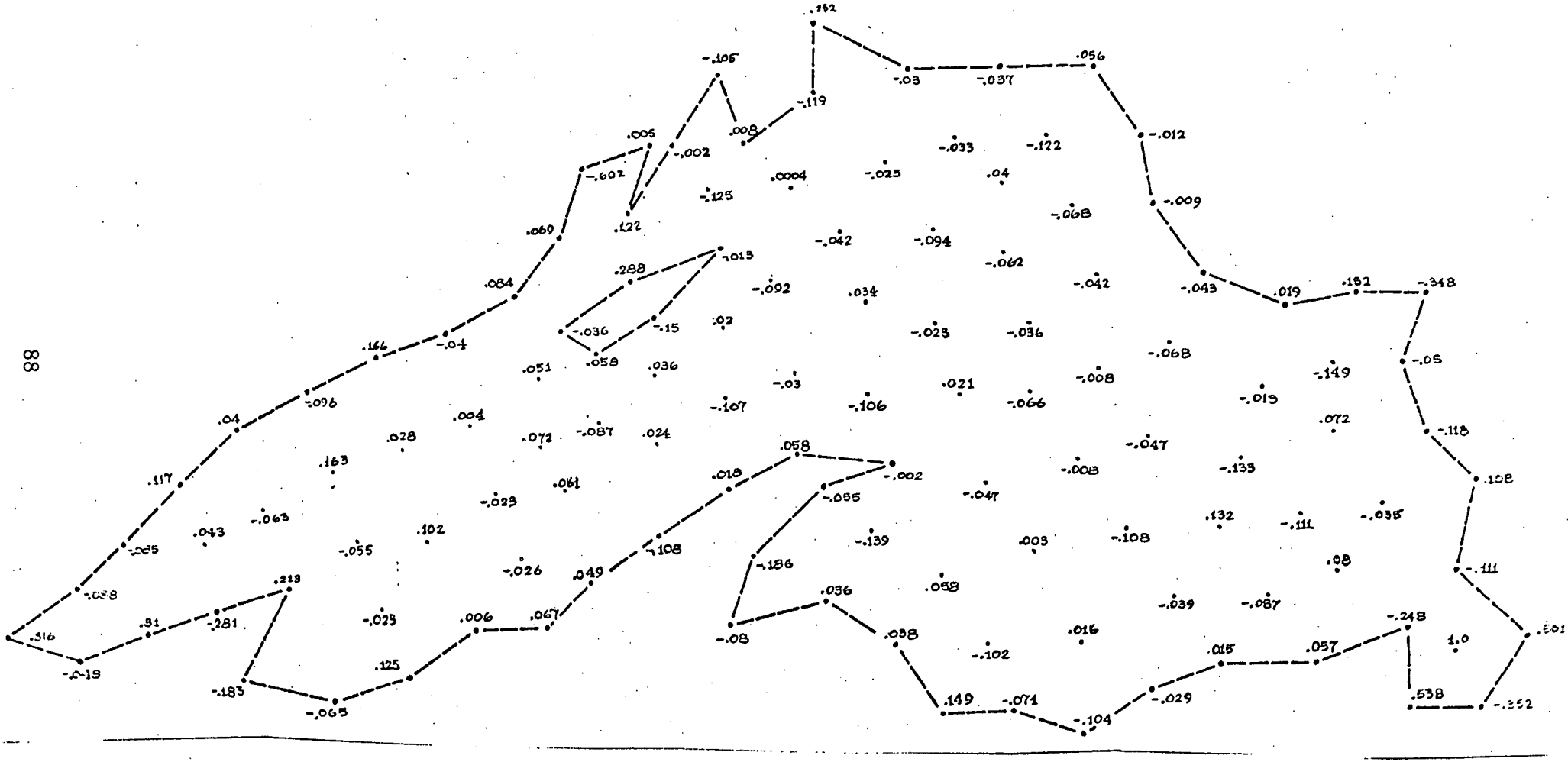


FIG. 8. A description of the eigenvector relative amplitude, with phase, for Lake Superior, as described by the present analysis at a modal period of 6.98 hours.

V. SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

The p rocedure and methods employed in this study indicate that this scheme can be used to obtain fundamental frequencies for free oscillations in closed tidal basin, but with some restrictions.

The use of the Laplace tidal equations for this purpose is, certainly, not new. However, the approach adopted here appears to represent a certain newness in at least two aspects. One of these is the formulation of the problem by the finite element method; while the second is the utilization of NASTRAN for solving an associated eigenvalue problem; these two items appear to be unique with this study. Even though the success anticipated and hoped for at the initiation of the investigation was not fully realized, a great deal was learned about the problem, in general, and about difficulties with large scaled numerical (matrix) operations, in particular.

In the early planning stages of the investigation thoughts were given to the idea of somehow adapting this technique to a similar study of the ocean tides. Now that a familarity with the "full operation" has been gained, it is quite apparent that the present approach cannot be suggested for a global tides analysis. The reasons for this are quite simply given -- this tides problem would quickly exceed machine capabilities, and the time requirement would, in all likelihood, be prohibitive. Until late in the present study, when the complex FEER algorithm (for eigenvalue extraction) was available, the time needed to execute a "run" for one of the Great Lakes models was of the order of an hour, or more. Recognizing that the lakes are relatively small bodies of water, by comparison, the time requirement for ocean studies, with a conservative extropolation, is too extravagant for consideration.

The use of NASTRAN, for the present eigenvalue determination, represented a departure from normal applications of this program. Actually, only the eigenvalue subprograms were employed in this study; however, any success which was to be had here required a comprehensive understanding of NASTRAN and its operations. Fortunately one of the investigators had that knowledge; and, as a matter of fact, it was through

the use of his developed complex FEER algorithm which ultimately provided the verification and confidence in those successes claimed here.

As noted earlier in the body of this report, the initial aim was to use the Inverse Method, with shifts, for the eigenvalue extraction process. Unfortunately, this approach had to be abandoned -- inconsistencies in results, and a growing concern regarding the method's operational integrity, led to a decision to change procedures. This, of course, produced a new dilemma in that the other candidate eigenvalue procedures, in NASTRAN, led to new difficulties; these, of course, were to have been circumvented by using the Inverse Method, per se.

Briefly, the difficulties with the other two routines can be summarized as follows: The upper Hessenberg routine, while it appeared to be working correctly, and required a lesser time for the computation of eigenvalues, was space limited in its application. That is, the size of the problems which could be handled, in NASTRAN, using this algorithm, was much smaller than the (ideal) problems which the investigators desired to use. Being "input-bound" by this method meant either abandoning the procedure, or reducing the size of the problems to be studied. Ultimately, the decision made was to reduce the problem size and (at least) use this algorithm to acquire estimates of the "correct" eigenvalues.

The difficulty experienced in using the Determinant Method, in NASTRAN, was one of technique. Unless the "search region" (see Appendix B) could be rather carefully, and precisely, defined the chance of "finding" roots was somewhat remote. That is, the method (apparently) is quite sensitive in its search pattern, and does not "home-in" on the root if there is an extensive search area to work on. This can be interpreted as follows: If roots are to be found, by the Determinant Method, the search region must be small; and, generally, in order to find "several roots", the region should be segmented into several (sub) regions of search.

A root finding procedure was developed but not, unfortunately, until after a number of trial operations had been conducted, and where only a degree of success

was achieved. That is, roots were obtained by the Determination Method, for appropriately described search regions, but only after first finding approximate roots using the Hessenberg Method. This meant solving problems twice -- once, using a crude model and the Hessenberg algorithm, followed by a more exacting procedure using the Determinant algorithm, Incidentally, this technique was not discovered until fairly late in the investigation. It came to light after it had been concluded that many of the difficulties being experienced probably resided with NASTRAN operations and not with the formulation, etc. of the model and other related developments.

Utlimately, and fortunately, the FEER algorithm became available and the results which are reported here were checked. Thus the confidence needed to support the other findings was gained, and the degree of success which has been shown was verified. Interestingly, the models concocted for this study served, in part, as test vehicles for the FEER program. Consequently FEER was used in this investigation well in advance of its availability to the NASTRAN user community.

Probably the most disappointing consequence of this study was the inability to achieve that degree of refinement expected from the eigenvalues extracted. That is, using the Laplace equations for the actual lakes, studied, including Coriolis effects, the roots which were extracted did not show the structure expected or desired. From a study of the eigenvalues one would tend to conclude that the problem had a dissipation term in the mathematical model. In essence, the NASTRAN roots showed a numerical difference, in modulus, for the real and imaginary parts of the eigenvalues. Such an occurrence was not expected (see the discussion on roots extracted, in the Test Problem section); and was most disconcerting to the investigators. Suffice it to say, quite an effort was given in search of the causes for this deviation of the roots. Unfortunately, no concrete explanation or remedy was forthcoming. In attempting to trace the causes for this, the analysis was checked, carefully, and cross-checked; the input data were re-examined -- even the scaling of the problem was changed in an effort to ascertain whether or not an error could have been committed there. The Coriolis parameter was altered, thinking that this might be causing something akin to frequency beating --

and, other possibilities were explored in search for a cause to the dilemma. Needless to say, the problem was not resolved; the roots obtained (with the Coriolis parameter included) were felt to be not sufficiently clean to represent the problem, hence these eigenvalues are not reported here. Only those roots acquired with negligible Coriolis influence are listed in the body of this report.

There were other possibilities, which might have been explored, as candidate remedies for this situation; unfortunately, there was not sufficient time or resources, available, to pursue these tasks. Basically, these alternatives would have necessitated major revisions to the program; these would have involved a new development for input, and (of course) a complete rerun of all cases studied, to acquire new output information.

This finding is not felt to represent a negative result, rather it points to a condition or conditions which were not foreseen and which could not be rectified under the limitations which existed. Once the more obvious possible causes were discarded, the remaining candidate conditions could only be examined insofar as existing constraints would allow. Necessarily the quick response approaches were pursued first; when these were eliminated then the more difficult ones were studied, but only to the limits allowed.

All things considered, the investigators believe that while the present approach is a valid one, and that it does have the potential for success, it cannot be recommended as a procedure for the study of ocean tides. At least not by the methods used herein. The procedure employed in this investigation was too time consuming (however, the advent of the complex FEER routine considerably reduced the running time for all such problems). In addition, the machine requirements (in core size needed) strains the capabilities of even large machines. (This analysis was carried out on the GSFC 360 series machines; there the core requirements, for the macro-sized Lake Erie problem, put severe limitations on the running of any typical case study. Incidentally, this constraint, and the fact that the input was an extensive as it was, hampered operations throughout the entire time of the investigation).

The present use of a finite element approach (for the tides problem, generally) clear indicates its feasibility and its versatility as a viable candidate procedure for solutions. There is good reason to believe that the existing difficulties can be overcome -- and that these do not reside with the approach used here. What is needed to complete this work is adequate time and resources to diligently continue the effort and to resolve the present dilemmas. Likewise, it would be prudent to alter this approach, to something other than the direct assault procedure used here, if these ideas are to be employed in an analysis of large tidal basins.

Much has been learned from this investigation; the areas where new and renewed efforts should be directed are (hopefully) indicated in this report. Some of the pitfalls and problems which can exist with large programs (like NASTRAN), have been encountered and noted -- these should tell other investigators what cautions they should take; and should point to the possible areas where difficulties are likely to be found.

VI. REFERENCES

(1) ABBOT, M.B., et al, "System 21, Jupiter", Journal of Hydraulic Research, Vol. 11, No. 1, 1973.

(2) BAKER, A.J. and S.W. Zelaxyn, "Predictions in Environmental Hydrodynamics Using the Finite Element Method", AIAA 12th Aerospace Sciences Meeting, Paper No. 74-7, 1974.

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

(4) DRONKERS, J.J., "Tidal Computations for Rivers, Coastal Areas, and Seas", Journal Hydraulics Division, ASCE, HY1, January 1969.

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

(6) HANSEN, W., "Theorie zur Errechnung des Wasserstandes und der Stromungen in Randmeeren Nebst Anwendungen", Tellus, Vol. 8, No. 3, August 1956.

(7) HESS, K.W. and F.M. White, "A Numerical Tidal Model of Narragansett Bay", University of Rhode Island, Marine Technical Report No. 20, 1974.

(8) HUEBNER, K.H., The Finite Element Method for Engineers, John Wiley & Sons, 1975.

(9) LAMB, Sir H., Hydrodynamics, Dover, 1949.

(10) LEENDERTSE, J.J., "Aspects of a Computational Model for Long-Period Water-Wave Propagation", Memorandum, RM-5294-PR. Rand Corporation, May 1967.

(11) McIVER, D.B., "The Prediction of Pollutant Transport in Estuaries: Variational Principles". (An unpublished report from the Dept. of Naval Architecture, University of Strathclyde, October 1973).

(12) ODEN, J.T. and E.R.A. Oliveira, Lectures on Finite Element Methods in Continuum Mechanics. The University of Alabama Press, 1973.

(13) PLATZMAN, George W., Two-Dimensional Free Oscillations in Natural Basins, Journal of Physical Oceanography, Vol. 2, No. 2, April 1972.

(14) TRACOR, "Estuarine Modeling: An Assessment", A report prepared for EPA, February 1971.

(15) WANG, J.D. and J.J. Connor, "Mathematical Modeling of Near Coastal Circulation", Massachusetts Institute of Technology, Report No. 200, April 1975.

APPENDIX A

GOVERNING EQUATIONS

A.1 DISCUSSION

The two basic mathematical expressions, used to describe a fluid flow, are conservation statements; one for mass and one for linear momentum. In rather general terms these equations may be written (in Einstein (summation) notation) as:

(a) for the conservation of mass:

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = s; \tag{1}$$

(b) for the conservation of momentum:

$$\rho\left[\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j}\right] = -\frac{\partial p}{\partial x_i} + F_i + \frac{\partial \tau_{ij}}{\partial x_j} + m_i, \tag{2}$$

wherein

x_i, u_i ≡ are cartesian position and velocity components

ρ, p ≡ mass density and hydrostatic pressure (for the fluid)

s, m_i ≡ fluid and momentum sources (sinks)

F_i ≡ "distance acting" forces (in component form)

τ_{ij} ≡ body "contact" forces (local stress in component* form)

In the second expression above the τ_{ij} terms are nominally expressed in terms of viscosity. Frequently, for natural circulation problems, there is included, in with the direct visosity effects, terms which describe turbulent momentum transfer (between

*The stress components, denoted as τ_{ij}, include both normal and shear stresses. These can be recognized by the designation given to the subscripts (i, j). In writing τ_{ij} the "i" indicates a direction for the unit normal to the stressed cube face, while the "j" indicates the direction of the stress component. Therefore, repeated subscripts (i=j) signify normal stresses; the subscripts ($i \neq j$) denote shear stress components.

fluid layers). In addition, for earth circulation models the cartesian reference frame is attached to the geoid; thus there is added to the local and convective acceleration terms, noted in Equation (2) above, the so-called accelerations due to Coriolis and centrifugal effects. Incidentally, Equation (2), here, is frequently called the Navier-Stokes equation. It has a similar form in elasticity; however, there the τ_{ij} are expressed in terms of the Lamé (after Gabriel Lamé (1795-1870)) coefficients (μ, λ). These completely characterize the (isotropic) material (body) and are explicitly related to Young's modulus and Poisson's ratio. In fluid dynamics problems the coefficients (μ, λ) are regarded as the first and second coefficients of viscosity; they do not have a direct correlation to their elasticity counter-parts. (For example, in studies of a monotonic gas, $\lambda \equiv -\frac{2}{3}\mu$; a physically improbable relationship, but one which seems to work well in practice.)

Incidentally, the "form" of these equations assumes an Eulerian representation for the motion (vice the Lagrangian form). Also, the the case(s) at hand it will be presumed that the chemical composition of the fluid mass is constant throughout the region under investigation; and, that the fluid mass is (thermally) adiabatic. In this regard the two expressions shown above adequately represent the fluid motion - however, it is understood that these expressions will require some modification (based on additional assumptions and considerations).

As a second concern the subject of boundary conditions has not yet been addressed; this will place added constraints (or conditions) on the solution, as is frequently the case in real world applications.

A.2 OSCILLATION IN TIDAL BASINS

In the determination of oscillatory modes for tidal basins it is necessary to infer a rotating system of (cartesian) coordinates. Thus, included is an influence for Corolis acceleration, compatible with other assumptions and conditions. Also, since the fluid (here) is water is it not unreasonable (physically, and otherwise) to assume it to be incompressible; and, subsequently, to neglect its viscosity.

Taking, as an orientation, a coordinate system with the (x,y)-axes tangent to the geoid surface, and the z-axis vertical (or outward from the surface); and, neglecting all source (sink) mechanisms; then, the equations which describe the problem, to this point are:

(a) for the conservation of mass:

$$\rho \frac{\partial u_i}{\partial x_i} = 0; \tag{3}$$

(b) for the conservation of momentum:

$$\rho\left[\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j}\right] = -\frac{\partial p}{\partial x_i} + F_i + \frac{\partial \tau_{ij}}{\partial x_j} + \rho(f_j u_k - f_k u_j), \tag{4}$$

wherein the f_α-terms are those appropriate to describe the Coriolis acceleration. (These terms are noted below, for the earth as a rotating geoid. The origin of Coriolis acceleration follows from the work of Gaspard Gustane de Coriolis (1792-1843); quite frequently this term (in applications work) is deemed negligible in comparison to other terms present. In the present case the Coriolis (specific) force is retained (in part) for motion in the tangent plane, but is neglected otherwise). The influence, here, of the centrifugal action is disregarded throughout.

A.3 CORIOLIS ACCELERATION

Consider the earth in rotation about its polar axis, and assume a locally fixed cartesian frame as indicated below.

Denoting the rotation vector as:

$$\bar{\Omega} = \Omega[\cos\phi\ (\hat{x}\sin\beta + \hat{y}\cos\beta) + \hat{z}\sin\phi],$$

where $\phi \equiv$ local longitude, β is the local aximuth angle and ($\hat{}$) represent unit vectors;

then the Coriolis acceleration can be expressed by:

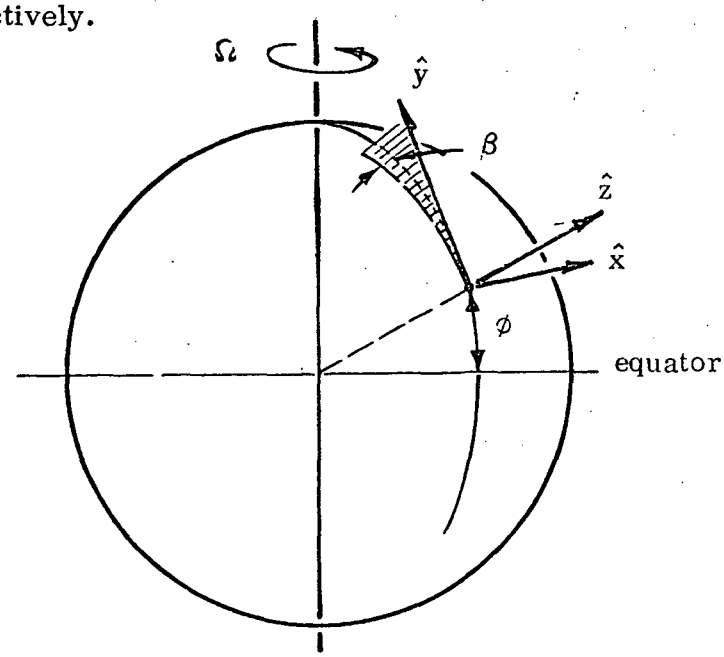
$$2\bar{\Omega} \times \bar{V} = 2\Omega \begin{bmatrix} \hat{x} & \hat{y} & \hat{z} \\ c\phi s\,\beta & c\phi c\,\beta & s\,\beta \\ u & v & w \end{bmatrix} \tag{5a}$$

$$= 2\Omega \left[\hat{x}(w\ c\phi c\ \beta - v s\beta)\right.$$

$$+\hat{y}\,(u s\,\phi - w c\phi s\ \beta) \tag{5b}$$

$$\left.+\hat{z}\ c\phi\ (v s\beta - u c\beta)\right],$$

wherein: ($\hat{}$) denotes unit vectors, for the local reference triad and u, v, w are components of velocity parallel to this triad. The notation (s, c) used here is shorthand for sine and cosine, respectively.



Sketch illustrating orientations, etc. applicable to the definition of Coriolis Acceleration.

Following the notation in Equation (4) it is apparent that the Coriolis acceleration components are:

$$(f_x, f_y, f_z) = 2\Omega\,[(w\cos\phi\cos\beta - v\sin\phi),\ (u\sin\phi - w\cos\phi\sin\beta),\ (v\cos\phi\sin\beta$$

$$- u\cos\phi\cos\beta)]. \qquad (5c)$$

Later, based on a relative order of magnitude of terms, these components will be reduced to the following approximation:

$$(f_x, f_y, f_z) \cong 2[(-v\sin\phi),\ (u\sin\phi),\ (-u\cos\phi)]\ \Omega, \qquad (5d)$$

wherein it has been assumed that $w \ll u, v$ and that $\beta \equiv 0$.

A.4 THE TWO-DIMENSIONAL (VERTICALLY INTEGRATED) MODEL

In this analysis of tidal oscillations the "vertically integrated" set of (reduced) Navier-Stokes equations are employed. The reduction, necessary to produce these expressions, from (say) Equations (4), will be outlined below.

Adding to the assumptions stated earlier, it is presumed (now) that:

(1) $w \ll u, v$ (a condition imposed on the velocity field);

(2) the Boussinesq assumption - for the fluid flow - holds true; i.e., the predominant effect on pressure variation is due to changes in fluid depth - a hydrostatic influence; and,

(3) the convective acceleration terms may be disregarded in contrast to the local variations.

Following from these conditions it is a simple task to reduce Equations (3) and (4) to the following set:

(a) for conservation of mass:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0. \qquad (6a)$$

(b) for conservation of momentum:

$$\rho \frac{\partial u}{\partial t} = - \frac{\partial p}{\partial x} + \frac{\partial}{\partial x} \tau_{xx} + \frac{\partial}{\partial y} \tau_{xy} + \frac{\partial}{\partial z} \tau_{xz} + 2\rho (\Omega \sin\phi)\, v, \tag{6b}$$

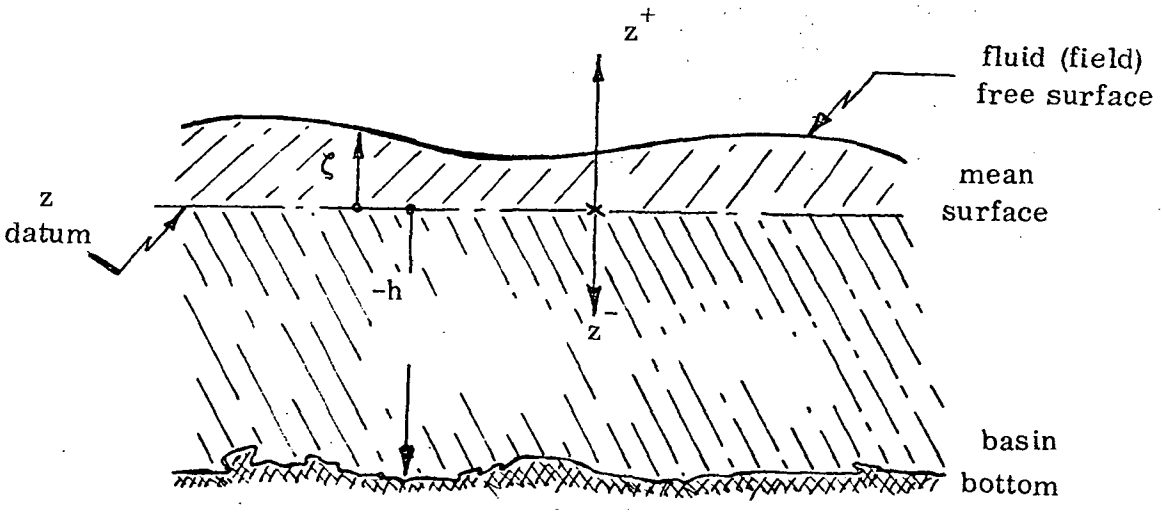
$$\rho \frac{\partial v}{\partial t} = - \frac{\partial p}{\partial y} + \frac{\partial}{\partial x} \tau_{yx} + \frac{\partial}{\partial y} \tau_{yy} + \frac{\partial}{\partial z} \tau_{yz} - 2\rho (\Omega \sin\phi)\, u, \tag{6c}$$

and

$$0 = - \frac{\partial p}{\partial z} - \rho g. \tag{6d}$$

Equations (6) are to be integrated ("vertically") through the fluid field to provide an appropriate set of (what is now apparent) two-dimensional equations for the motion. (The elimination of the convective acceleration terms is, in part, carried out to reduce the obvious non-linearity which would otherwise exist. Also, this aids in maintaining numerical stability for the solutions, as well as simplifying the subsequent statement formats).

The sketch, below, will aid in visualizing the meaning attached to various integrals which will follow:



Sketch Depicting a General Vertical Plane Bathymetry

When Equation (6d) is integrated (vertically) the result is

$$p\,(x,y,z;t) = -\rho g\, z + f(x,y;t).$$

However, the surface condition on pressure requires that

$$p\,(x,y,z;t) - p_o = -\rho g\, z + f(x,y;t),$$

wherein $p_o \equiv$ surface pressure (at $z = +\zeta$).

Combining these results leads directly to:

$$p\,(x,y,z;t) = p_o - \rho g\,(\zeta - z), \tag{7}$$

an expression describing the vertical pressure variation according to Boussinesq's assumption.

Making use of Equation (7), especially to describe pressure gradients, Equations (6b, 6c) can be recast as:

$$\rho \frac{\partial u}{\partial t} = -\rho g \frac{\partial \zeta}{\partial x} + \frac{\partial}{\partial x}\tau_{xx} + \frac{\partial}{\partial y}\tau_{xy} + \frac{\partial}{\partial z}\tau_{xz} + \rho f v\,, \tag{8a}$$

$$\rho \frac{\partial v}{\partial t} = -\rho g \frac{\partial \zeta}{\partial y} + \frac{\partial}{\partial x}\tau_{yx} + \frac{\partial}{\partial y}\tau_{yy} + \frac{\partial}{\partial z}\tau_{yz} - \rho f u\,, \tag{8b}$$

where $f(\equiv 2\Omega \sin\phi)$ is the so-called Coriolis parameter.

In integrating Equations (6a), (8a) and (8b), the governing equations for fluid motion, as specified here, use is made of Liebnitz's Rule*. Thus, e.g., the integration of Equation (6a) is written as

*Liebnitz's Rule states (for a single variation operation):

$$\frac{d}{dx}\int_{a(x)}^{b(x)} f(x,s)\,ds = \int_a^b \frac{\partial f}{\partial x}\,ds + f(x,b)\frac{db}{dx} - f(x,a)\frac{da}{dx}\,.$$

$$\int_{-h(x,y;t)}^{\zeta(x,y;t)} \frac{\partial u}{\partial x}\,dz + \int_{-h(x,y;t)}^{\zeta(x,y;t)} \frac{\partial v}{\partial y}\,dz + \int_{-h(x,y;t)}^{\zeta(x,y;t)} \frac{\partial w}{\partial z}\,dz = 0.$$

Introducing Liebnitz's Rule, the integrals are rewritten as:

$$\frac{\partial}{\partial x}\int_{-h}^{\zeta} u\,dz - u(x,y,\zeta;t)\frac{\partial \zeta}{\partial x} + u(x,y,-h;t)\frac{\partial(-h)}{\partial x}$$

$$+ \frac{\partial}{\partial y}\int_{-h}^{\zeta} v\,dz - v(x,y,\zeta;t)\frac{\partial \zeta}{\partial y} + v(x,y,-h;t)\frac{\partial(-h)}{\partial y}$$

$$+ w(x,y,\zeta;t) - w(x,y,-h;t) = 0. \qquad (9)$$

Note that the operation here is a vertical integration of (one of the) equations for the motion. Similar operations are to be carried out on the momentum expressions.

Now, defining the <u>vertically</u> averaged velocity components ($\bar{u}$, $\bar{v}$) as:

$$\bar{u} \equiv \frac{1}{\zeta + h}\int_{-h}^{\zeta} u\,dz,$$

and (10)

$$\bar{v} \equiv \frac{1}{\zeta + h}\int_{-h}^{\zeta} v\,dz,$$

then Equation (9) can be replaced by:

$$\frac{\partial}{\partial x}[(\zeta + h)\bar{u}] + \frac{\partial}{\partial y}[(\zeta + h)\bar{v}] - [u(x,y,\zeta;t)\frac{\partial \zeta}{\partial x} + v(x,y,\zeta;t)\frac{\partial \zeta}{\partial y} - w(x,y,\zeta;t)]$$

$$+ [u(x,y,-h;t)\frac{\partial(-h)}{\partial x} + v(x,y,-h;t)\frac{\partial(-h)}{\partial y} - w(x,y,-h;t)] = 0. \qquad (11)$$

In consideration of the physical problem, here, there are certain kinematic conditions which must be accounted for in the solution. For example, and in regard to the expression immediately above, the kinematics at the basin bottom and at the free surface dictate that:

(a) at the surface:

$$\frac{D}{Dt}\left[\zeta(x,y;t)\right] \equiv w(x,y,\zeta;t) = \left[\frac{\partial \zeta}{\partial t} + u\frac{\partial \zeta}{\partial x} + v\frac{\partial \zeta}{\partial y}\right]_{z=\zeta}, \tag{12a}$$

(b) at the bottom:

$$\frac{D}{Dt}\left[-h(x,y;t)\right] \equiv w(x,y,-h;t) = \left[\frac{\partial(-h)}{\partial t} + u\frac{\partial(-h)}{\partial x} + v\frac{\partial(-h)}{\partial y}\right]_{z=-h}, \tag{12b}$$

wherein the bottom contour will be assumed to be invariant in time.

As a consequence of these added conditions, Equation (11) reduces to:

$$\frac{\partial}{\partial x}\left[(\zeta + h)\bar{u}\right] + \frac{\partial}{\partial y}\left[(\zeta + h)\bar{v}\right] + \frac{\partial \zeta}{\partial t} = 0; \tag{13a}$$

and, if a linearization of this expression is introduced (as is necessary for the eigen-analysis) a further reduction is achieved; namely:

$$\frac{\partial}{\partial x}(h\bar{u}) + \frac{\partial}{\partial y}(h\bar{v}) + \frac{\partial \zeta}{\partial t} \equiv 0. \tag{13b}$$

When the momentum expressions are (vertically) integrated, and Liebnitz Rule is employed, similarly reduced expressions are acquired. For convenience (here) only Equation (8a) will be manipulated - the results for both expressions will, however, be noted below.

Symbolically, the integration of the x-momentum equation is represented by:

$$\rho \int_{-h(x,y;t)}^{\zeta(x,y;t)} \frac{\partial u}{\partial t}\, dz = -\rho g \int_{-h}^{\zeta} \frac{\partial \zeta}{\partial x}\, dz + \int_{-h}^{\zeta} \frac{\partial \tau_{xx}}{\partial x}\, dz + \int_{-h}^{\zeta} \frac{\partial \tau_{xy}}{\partial y}\, dz$$

$$+ \int_{-h}^{\zeta} \frac{\partial \tau_{xz}}{\partial z}\, dz + \int_{-h}^{\zeta} f v\, dz.$$

Apply Liebnitz's Rule, it can be shown that:

$$\rho \frac{\partial}{\partial t} \int_{-h}^{\zeta} u\, dz - \rho u(x,y,\zeta;t) \frac{\partial \zeta}{\partial t} = -\rho g \frac{\partial \zeta}{\partial t}(h+\zeta) + \frac{\partial}{\partial x} \int_{-h}^{\zeta} \tau_{xx}\, dz + \frac{\partial}{\partial y} \int_{-h}^{\zeta} \tau_{xy}\, dz$$

$$- [\tau_{xx}(x,y,\zeta;t) \frac{\partial \zeta}{\partial x} - \tau_{xx}(x,y,-h;t) \frac{\partial(-h)}{\partial x}] - [\tau_{xy}(x,y,\zeta;t) \frac{\partial \zeta}{\partial x}$$

$$- \tau_{xy}(x,y,-h;t) \frac{\partial(-h)}{\partial y}] + \tau_{xz}(x,y,\zeta;t) - \tau_{xz}(x,y,-h;t) + \rho f \int_{-h}^{\zeta} v\, dz.$$

Representing the (collected) stress terms by P_x (for convenience) and introducing the vertically averaged parameters, write:

$$\rho \frac{\partial}{\partial t}[(\zeta+h)\bar{u}] - \rho u(x,y,\zeta;t) \frac{\partial \zeta}{\partial t} = -\rho g(\zeta+h) \frac{\partial \zeta}{\partial x} + \rho f(\zeta+h)\bar{v} + P_x; \qquad (14)$$

recall that $\bar{u}$, $\bar{v}$ are the vertically averaged velocity components, f is the Coriolis parameter, ζ is the free surface height (above datum), ρ is the fluid mass density and P_x represents the (integrated) fluid stress parameters.

A linearization of this expression leads directly to the following result:

$$\frac{\partial \bar{u}}{\partial t} = -g \frac{\partial \zeta}{\partial x} + f\bar{v} + \frac{P_x}{\rho h}. \qquad (15)$$

For the eigenanalysis, to obtain the natural frequencies of a "closed" tidal basin, it is necessary to assume the fluid field to be both incompressible and inviscid. This

added assumption supposes that the internal (viscous) stressing can be neglected, thus $P_x \equiv 0$. (It is recognized that in eigenvalue problems a solution is obtained for the homogeneous equations of motion). As a consequence of the linearization and the removal of the stress terms, then the x-momentum (homogeneous) equation is:

$$\frac{\partial \bar{u}}{\partial t} + g \frac{\partial \zeta}{\partial x} - f\bar{v} = 0. \tag{16a}$$

A similar treatment of the y-momentum equation leads to the following reduced (homogeneous) form:

$$\frac{\partial \bar{v}}{\partial t} + g \frac{\partial \zeta}{\partial y} + f\bar{u} = 0. \tag{16b}$$

Finally, writing (again) Equation (13b):

$$\frac{\partial \zeta}{\partial t} + \frac{\partial}{\partial x}(h\bar{u}) + \frac{\partial}{\partial y}(h\bar{v}) = 0 \tag{16c}$$

provides a full list of those differential equations to be employed in the eigenvalue solution.

These three differential equations describe the fluid motion in terms of ($\bar{u}$, $\bar{v}$, ζ), as dependent variables, and in terms of time (t) and horizontal (planar) displacements (x, y) as the independent parameters.

A.5 EQUATIONS OF MOTION, ALTERNATE FORMS

Digressing, for the moment, other forms of the vertically integrated hydrodynamic equations are noted and discussed, briefly, for purposes made obvious during the discussions.

One form of the conservation of mass and linear momentum equations, used in hydrodynamics studies, are as follows - these are found, developed, in Reference (2, 4):

(a) the continuity (mass conservation) expression:

$$\frac{\partial \zeta}{\partial t} + \frac{\partial}{\partial x}(q_x) + \frac{\partial}{\partial y}(q_y) = s, \tag{17a}$$

wherein: $q_j = \int_{-h}^{\zeta} v_j \, dz$ $(j = x, y)$ represents the local velocity flux, while "s" represents a local source (sink) horizontal flow rate (e.g., a rainfall or sub-surface discharge); all other parameters have been identified previously.

(b) the conservation of momentum (in a horizontal plane) expressions:

$$\frac{\partial}{\partial t}(q_x) + \frac{\partial}{\partial x}(\bar{u} q_x) + \frac{\partial}{\partial y}(\bar{u} q_y) = f q_y - g H \frac{\partial}{\partial x}(\zeta) + \{k \frac{\rho_a}{\rho} W_x |W_x| - g \frac{q_x (q_x^2 + q_y^2)^{\frac{1}{2}}}{C^2 H^2}\}, \tag{17b}$$

and

$$\frac{\partial}{\partial t}(q_y) + \frac{\partial}{\partial x}(\bar{v} q_x) + \frac{\partial}{\partial y}(\bar{v} q_y) = -f q_x - g H \frac{\partial}{\partial y}(\zeta) + \{k \frac{\rho_a}{\rho} W_y |W_y| - g \frac{q_y (q_x^2 + q_y^2)^{\frac{1}{2}}}{C^2 H^2}\}. \tag{17c}$$

Here:

$$\bar{u} \equiv \frac{1}{H} \int_{-h}^{\zeta} u \, dz = \frac{q_x}{H}, \qquad \bar{v} \equiv \frac{1}{H} \int_{-h}^{\zeta} v \, dz = \frac{q_y}{H},$$

and H is the local fluid column depth $(\Sigma(\zeta, -h))$. Also, as before:

f ≡ Coriolis parameter $(2\Omega \sin\phi)$

g ≡ gravitational constant.

In addition:

$k \equiv$ dimensionless (surface) drag, or friction, coefficient

ρ_a, $\rho \equiv$ air, water mass density (respectively)

W_x, $W_y \equiv$ (local) surface (or near surface) wind velocity components

$C \equiv$ Chezy coefficient (used to describe bottom friction).

Note: The terms enclosed in curly brackets, $\{\ -\ \}$, are substitutions introduced to account for (local) viscous or "body stress" terms.

The above stated equations for fluid motion are most useful in the calculation of flow direction(s) at cornors on the boundaries of the fluid field.

There is a second form of these fluid flow equations, readily constructed from the above, used only for cases of no mass source(s) or sink(s). These are:

$$\frac{\partial \zeta}{\partial t} + \frac{\partial}{\partial x}(H\bar{u}) + \frac{\partial}{\partial y}(H\bar{v}) = 0, \tag{18a}$$

$$\frac{\partial \bar{u}}{\partial t} + \bar{u}\frac{\partial \bar{u}}{\partial x} + \bar{v}\frac{\partial \bar{u}}{\partial y} = f\bar{v} - g\frac{\partial \zeta}{\partial x} + \left\{k\frac{\rho_a}{\rho}\frac{W_x|W_x|}{H} - g\frac{\bar{u}(\bar{u}^{-2}+\bar{v}^{-2})^{\frac{1}{2}}}{C^2 H}\right\}, \tag{18b}$$

and

$$\frac{\partial \bar{v}}{\partial t} + \bar{u}\frac{\partial \bar{v}}{\partial x} + v\frac{\partial \bar{v}}{\partial y} = -f\bar{u} - g\frac{\partial \zeta}{\partial y} + \left\{k\frac{\rho_a}{\rho}\frac{W_y|W_y|}{H} - g\frac{\bar{v}(\bar{u}^{-2}+\bar{v}^{-2})^{\frac{1}{2}}}{C^2 H}\right\}. \tag{18c}$$

The parameters used in these expressions have been described above, and earlier.

It should be mentioned that both these sets of equations should give the same results for the case of a constant depth (-h).

The primary difference between the expressions in (q_x, q_y) and $(\bar{u}, \bar{v})$ (above) is that the equations in (q_x, q_y) contain lumped product forms for $(\bar{u}H, \bar{v}H)$. As a conse-

quence of this lumping (of parameters) there may be a selection made as to how these terms are introduced into the finite element method (FEM) equations.

As an aside, the reader is directed to the forms used for convective acceleration terms in the above expressions. For Equations (17) the q_j terms may be handled differently in the finite element method formulations. Since these parameters are "lumped", their representation, in an appropriate finite element function form, can vary. That is, the $\bar{u}_i \bar{q}_j$ quantities may be retained as single parameters; or, they may be considered as a multiple of parameters. The second consideration would (obviously) lead to higher ordered integration(s) for the FEM formulations. To reduce the mathematical complexity, a general procedure, at least in first studies, is to make use of the lumped parameters in the convective acceleration terms, and in the representation for bottom friction where (obvious) non-linearities are present.

The equations discussed here are, necessarily, not applicable to the eigenanalysis performed in this investigation. It is instructive, however, to review and study these various forms, for the hydrodynamic equations, in order to view the kinds, levels and degrees of approximations which have been introduced into water circulation and tidal prediction studies. What we have not mentioned, or seen, in this review is the multi-layered modelling of tidal ponds or other bodies of water. Basically, this topic has not been brought up because it does not "fit" the concepts used here. It is not necessary, or even advisable to consider multi-layered modelling unless there are vertical variations in fluid field parameters which cannot be ignored; or, for cases where it is felt that this degree of refinement and complexity will be needed for prediction accuracy. To date only limited analysis on multi-layering of the fluid field has been carried out. For most studies the single (vertically integrated) layer model -- called a "shadow water" model - has been adequate. Primarily this approach has been deemed sufficient in view of the small-vertical-variations in parameters (or the assumption of such) for the models. Until more experience has been gained, or the need arises, it is not likely that multi-layer models will become widely used.

APPENDIX B

EIGENVALUE EXTRACTION METHODS

B.1 INTRODUCTION

Three methods for eigenvalue extraction are generally provided in NASTRAN. This multiple of procedures is provided because no single method, or pair of methods, has been found fully satisfactory with regard to efficiency, reliability, and (general) applicability, for all situations. Today, the growth in technology is rather explosive, particularily when one considers the digital computer. One consequence of this device is that there are a variety and capability of eigenvalue extraction routines presently available. In addition, it can be expected that new methods will be added (to NASTRAN) as time goes on, and that old methods will be improved or discarded as new discoveries are made. (Incidentally, one new procedure - known as the FEER method - is presently being developed for incorporation into the NASTRAN system).

Most methods for algebraic eigenvalue extraction belong to one of two groups; these are: transformation methods and "tracking" methods. In the transformation method a matrix of coefficients is (first) transformed, while preserving the eigenvalues, into one of several special forms (diagonal, tridiagonal, or Upper Hessenberg) from which the eigenvalues may be readily extracted. On the other hand, in a "tracking" method the roots are extracted, one at a time, by iterative procedures applied to the original (dynamic) matrix.

Of the procedures present in NASTRAN one is classed as a transformation method (the Tridiagonal Method), while the others (the Determinant Method, and the Inverse Power Method with Shifts) are tracking methods.

The "effort" expended in tracking methods is linearly proportional to the number of eigenvalues extracted. Consequently, the tracking methods are more efficient when only a few eigenvalues are required; but are less efficient otherwise.

The general characteristics of those methods used by NASTRAN, for eigenvalue extraction, are compared in Table 1. The Tridiagonal Method, due to restrictions, is available only for vibration modes of conservative systems; it is not to be used for complex eigenvalue analysis*. The other two methods are useable for all real and complex eigenvalue problems currently "solved" by NASTRAN. Also, the Determinant Method is available for future problem types where the coefficients are general functions of the eigenvalues.

It is noted (see Table 1) that a narrow bandwidth, and a small proportion of extracted roots, favors the tracking methods. One example of this is the evaluation for the lowest (few) modes of a system. When bandwidth is relatively large, and/or when a high proportion of the eigenvalues is required, the Tridiagonal Method will likely be more efficient (subject to the foregoing restrictions).

The Determinant Method, and the Inverse Power Method with Shifts, have the same general characteristics in regards to current NASTRAN problems. The Inverse Power Method is, however, a more efficient procedure except when the bandwidth is extremely narrow. The main advantage in including both methods here is a redundancy in procedures. (This can be most appreciated in those cases where one or the other of the methods fails, as sometimes happens for any eigenvalue extraction procedure.

*For this reason the Tridiagonal Method is not discussed further in the following descriptions.

Table 1. Comparison of General NASTRAN Eigenvalue Extraction Methods

Characteristic \ Method	Tridiagonal Method	Inverse Power Method with Shifts	Determinant Method
Most general form of the problem (matrix)	$[A - pI]$	$[Mp^2 + Bp + K]$	$[A(p)]$
Restrictions on the matrix	Must be a real, sym., constant matrix	Coefficients M, B and K, are constant	None
Eigenvalues obtained:	All at once	Nearest to shift point	(Usually) nearest to starting points
Does method take advantage of bandwidth?	No	Yes	Yes
Number of calculations required (by order of magnitude)	$O(n^3)$	$O(nb^2E)$	$O(nb^2E)$

Note: n = number of equations

b = semi-bandwidth

E = number of eigenvalues extracted

B.2 COMPLEX EIGENVALUE ANALYSIS, FOR NASTRAN

The form of a complex eigenvalue problem, using the direct formulation, is:

$$[M_{dd}p^2 + B_{dd}p + K_{dd}]\{u_d\} = 0.$$

The vector $\{u_d\}$ includes the set, u_a, degrees of freedom (at structural grid points), and the set of "extra points", u_e, described later. The mass matrix, $[M_{dd}]$, the damping matrix, $[B_{dd}]$, and the stiffness matrix, $[K_{dd}]$, may be real or complex. All matrices may be symmetric or unsymmetric, singular or nonsingular. In any case the eigenvalues, p_j, are acquired from a homogeneous solution of the form:

$$\{u_d\} \equiv \{\phi_{dj}\} e^{p_j t},$$

which is equivalent to

$$\{u_d\} = \{\phi_{dj}\} e^{\alpha_j t} \sin(\omega_j t),$$

where (α_j, ω_j) are the real and imaginary parts of p_j, respectively.

The form of a complex eigenvalue problem, using the modal formulation, is:

$$[M_{hh}p^2 + B_{hh}p + K_{hh}]\{u_h\} = 0.$$

The components $\{u_h\}$ contain modal coordinates (ξ_i) and the set of extra points (η_c). As in the direct formulation, there are no restrictions on these matrices.

In all solution procedures the eigenvectors are normalized to a maximum element value of unity; or, to a value of unity for a specified element, depending on user option.

B.3 THE DETERMINANT METHOD

B.3.1 Fundamentals of the Method

The basic ideas used here, for eigenvalue extraction, are quite simple. If the matrix elements are polynomial functions of an operator (p) then the determinant can be expressed as:

$$D(A) = (p - p_1)(p - p_2)(p - p_3) \ldots . (p - p_n),$$

where the p_i are eigenvalues of the matrix.

In this method, the determinant is evaluated using trial values for p (selected by some iterative procedure); and, a criterion is established to determine when D(A) is sufficiently small, or p is "close" to an eigenvalue. Finally, an eigenvector is found from a solution to the equation:

$$[A]\{u\} = 0,$$

with one of the elements for $\{u\}$ preset.

A most convenient procedure, used for evaluating the determinant of a matrix, employs triangular decomposition. For this procedure let

$$[A] \equiv [L][U],$$

where $[L]$ is a lower unit triangular matrix (unity on the diagonal), and $[U]$ is an upper triangular matrix. (Recall that the determinant of $[A]$ is the product of the diagonal terms in $[U]$).

Two versions of triangular decomposition are provided. In the standard version row interchanges are used to improve numerical stability. An optional version, available

for real eigenvalue extraction only, does not use row interchanges. This version is approximately four times faster than the standard (for banded matrices), but it suffers the risk of numerical failure in that [A] is seldom a positive definite matrix.

The matrix [A] can be expressed as:

$$[A] \equiv -p\,[M] + [K],$$

for real eigenvalue problems, and as

$$[A] \equiv p^2[M] + p[B] + [K],$$

for complex eigenvalue problems.

The determinant method is not particularly efficient because of the large numbers of triangular decompositions taken on the [A] matrix (when more than a few eigenvalues are desired). The main strength of this method lies in its insensitivity to functional form for the elements in the [A] matrix. Such a form could, for example, contain poles and zeroes; or, be a transcendental function of p.

B.3.2 The Iterative Algorithm

Wilkinson's recent, but now (considered) standard, treatise* includes an authoritative discussion of polynomial curve-fitting schemes useful in tracking the roots of a determinant. He shows that little is gained from the use of polynomials higher than second degree. Consequently, Muller's quadratic method (Wilkinson, p. 435) is used in NASTRAN. This algorithm's form, in our notation, is described below:

A series of determinants, D_{k-2}, D_{k-1}, D_k, are evaluated for trial values of an eigenvalue (say $p = p_{k-2}$, p_{k-1}, p_k). A better approximation to an eigenvalue is obtained by the following calculations: First, let

*Wilkinson, J.H., "The Algebraic Eigenvalue Problem", Clarendon Press, Oxford, 1965.

$$h_k \equiv p_k - p_{k-1},$$

$$\lambda_k \equiv \frac{h_k}{h_{k-1}}$$

and

$$\delta_k \equiv 1 + \lambda_k .$$

Then, denote

$$h_{k+1} = \lambda_{k+1} h_k ,$$

so that

$$p_{k+1} = p_k + h_{k+1},$$

wherein

$$\lambda_{k+1} = \frac{-2D_k \delta_k}{g_k \pm [g_k^2 - 4D_k \delta_k \lambda_k (D_{k-2} \lambda_k - D_{k-1} \delta_k + D_k)]^{\frac{1}{2}}} ,$$

with

$$g_k \equiv D_{k-2} \lambda_k^2 - D_{k-1} \delta_k^2 + D_k (\lambda_k + \delta_k) .$$

(The (±) sign in the above expression is chosen so as to minimize the absolute value of λ_{k+1}). For those cases where p_k, p_{k-1}, and p_{k-2} are <u>all</u> arbitrarily selected initial values (starting points), the starting points are arranged so that

$$D_k \leq D_{k-1} \leq D_{k-2} .$$

Consequently the (±) sign is chosen to minimize the distance from p_{k+1} to the <u>closest</u> starting point (rather than to p_k).

In the real eigenvalue analysis, it is possible to calculate a complex value for λ_{k+1} from above. In order to preclude the occurrence of complex arithmetic, in a real eigenvalue analysis, only the real part of λ_{k+1} is used to estimate p_{k+1}. (The real part corresponds to the minimum absolute value of a parabolic approximation).

B.3.3 Scaling

In calculating the determinant of [A], some form of scaling must be employed since the accumulated product will rapidly overflow or underflow (the floating point size) in a digital computer. Accordingly, the accumulated product of the diagonal terms in [U] is computed, and stored, as a scaled number

$$D \equiv d \times 10^{n},$$

wherein

$$\frac{1}{10} \leq d < 1.$$

The arithmetic operations indicated in the equations for λ_{k+1} and g_k are carried out in scaled arithmetic. The quantity λ_{k+1} is then reverted to its unscaled form.

B.3.4 The Sweeping Out of Previously Extracted Eigenvalues

Once an eigenvalue has been found (to a specified accuracy) a return to it, by the iteration, can be prevented if the determinant is divided by $(p - p_i')$, where p_i' is the approximation to p_i.

Wilkinson states that the sweeping procedure is satisfactory provided all p_i' are calculated to an accuracy limited only by round-off error.

In some instances there are known eigenvalues; thus calculations need not be made. Also, the user may know other eigenvalues (e.g., those extracted previously, or those resulting from transfer functions), which should be avoided. [A special data card (EIGP) is available (in the complex analysis) to specify a location for any and all such roots. These are immediately eliminated.]

For problems with conjugate complex eigenvalues the conjugates (of extracted eigenvalues) are also swept from the determinant.

A danger exists in this procedure when roots are very near the axis of reals; there the exact eigenvalue may be real. To avoid such a situation a test is applied to the imaginary part of p_i; the result being that the conjugate of p_i' is swept out only if

$$|\text{Im } p_i'| \geq 1000.0(R_{max})(\epsilon).$$

B.3.5 The Search Procedures

Three initial values of p (starting points) are needed to initiate the iteration algorithm. The determinant method is, basically, a root-tracking method that finds nearby roots easily, and remote roots with difficulty. Consequently, it is not advisable to use the same three starting points for all eigenvalues since eigenvalues are usually distributed throughout a region of the p-plane.

In a real eigenvalue analysis, the starting points are distributed uniformly over some interval of p. The user specifies lowest and highest expected eigenvalues (called R_{min} and R_{max}). In addition, he estimates the number of roots in this range; thus the starting points are located to coincide with R_{min}, and with R_{max}; all others are located uniformly between R_{min} and R_{max}. As eigenvalues within this "range" are extracted, some points are dropped and others added; thus the "search" is repeated.

Search procedures for complex eigenvalues are more complicated since the roots are distributed throughout a region rather than along a line. Fortunately, in the present analysis roots are found along a 45^o line, in the imaginary plane, so that a set of starting points are readily provided. In more general problems, rectangular search regions should be located in regions as indicated on Figure B.1. It is supposed that all eigenvalues within a search region will be extracted (within limits specified by the maximum number of desired roots).

There can be as many search regions as needed; and search regions may overlap. Each region is established by specifying coordinates for the end points (A_j, B_j), and by

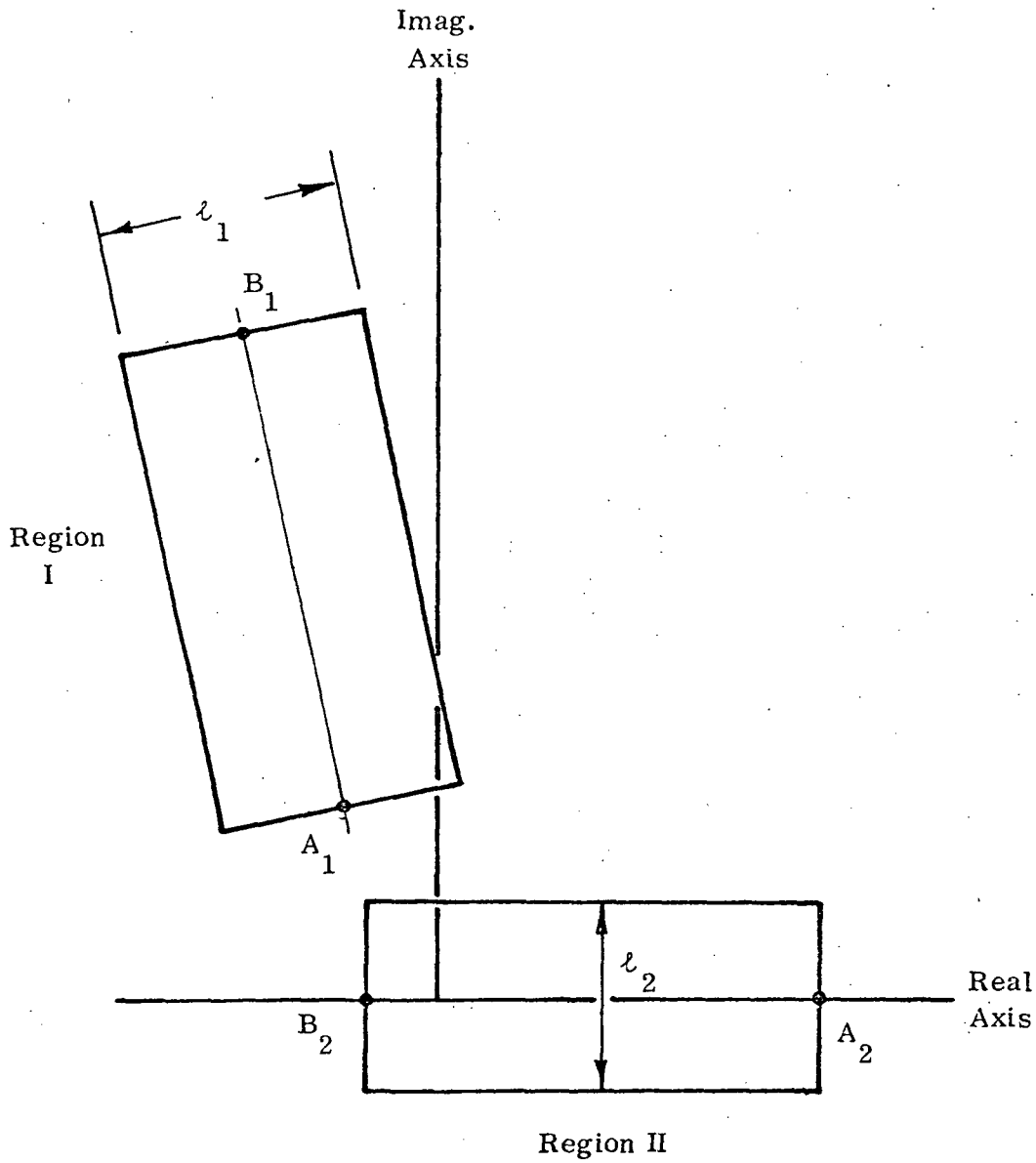


Fig. B.1. Search Region, DETERMINANT METHOD

giving a width to the region (ℓ_j). Problems with real coefficients have roots as either "reals" or "conjugate complex pairs". Consequently it is inefficient to specify "regions" in the lower half of the complex plane due to the existence of conjugate pairs.

The steps, in a search procedure for complex eigenvalues, are as follows:

(1). Select starting points equally distributed along A_j, B_j. Note: points A_j and B_j are not starting points.

(2). Find that starting point, in Region I, nearest the origin (see Figure B.2), it is denoted as p_{s1}. A line perpendicular to A_1B_1, midway between p_{s1} and p'_{s1}, (a point next nearest the origin) divides Region I into Regions IA and IB.

(3). Selecting three starting points in Region IA, nearest line a - a', as an initial set, proceed to extract roots. Next, go to Region IB; alternate (back and forth) between the two regions until all starting points have been used once; or, until termination occurs (for some other reason).

(4). When all starting points have been used once, go to Region II, etc. Sweep out all extracted roots (in each region) before evaluating determinants.

(5). When all starting points have been used once, return to Region I, II, etc., and repeat the procedure above. Continue these operations until no new roots are found in any pass through all regions, or until termination occurs.

When searching for either real or complex roots, the search is terminated if a root is predicted to lie outside of the "local" search region.

The search for eigenvalues is terminated when all roots are found, through all regions, or when the desired maximum number of roots have been extracted.

Failure to find additional roots generally occurs when all roots within the region(s) have been extracted. Situations can occur for which some roots will be missed. The most common of these, in a complex eigenvalue analysis, is that one or more of the desired roots is at a large distance from the region's centerline and several other roots are just

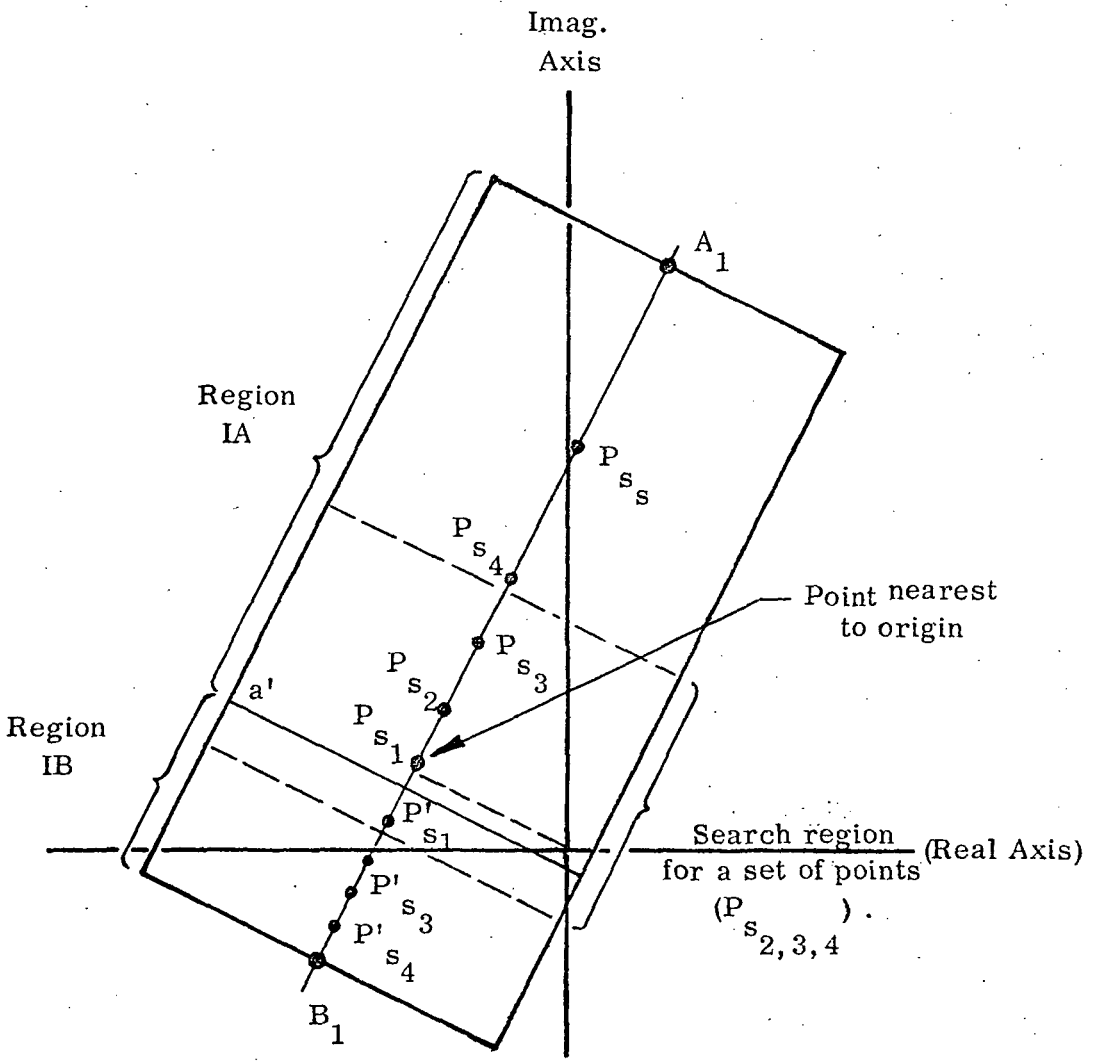


Fig. B.2. Location of Starting Points, in a Region; DETERMINANT METHOD

beyond the region's ends. The search procedure is likely to be attracted toward these latter roots, in lieu of the former. Another possible cause of missed roots is a failure of the iteration algorithm to converge.

B.3.6 Convergence Criteria

Convergence criteria are based on successive values of the increment, h_k, for estimated eigenvalues. No tests on determinant magnitude, or on any diagonal terms in the triangular decomposition, are necessary or desired.

Wilkinson shows that for h_k sufficiently small, the magnitude (of h_k) is approximately squared for each successive iteration (when using Muller's method to find isolated roots). This leads to a very rapid rate of convergence.

If the number of iterations becomes excessively large, without satisfying a convergence criterion, it is best to give up the search and proceed to a new set of starting points.

B.3.7 Test for Roots Close to a Starting Point

Once an eigenvalue has been extracted it is tested for closeness to a starting point. When found to be too close, the starting point is shifted. The reason being that the value of the determinant, near to such a root, is small and contains considerable round-off error. As a consequence the value of the swept determinant may be in considerable error.

B.3.8 Determination of Eigenvectors

Once an eigenvalue has been accepted, the eigenvector is determined by substituting into the previously computed triangular decomposition of $[A(p_j)]$. Since, for an eigenvalue,

$$[A(p_j)]\{u\} \equiv [L(p_j)][U(p_j)]\{u\} = 0,$$

and with $[L(p_j)]$ nonsingular, only $[U(p_j)]$ is used for the eigenvector description. The last diagonal term in $[U(p_j)]$ is normally the only one with a very small value.

B.4 THE INVERSE POWER METHOD WITH SHIFTS

B.4.1 Introduction

The Inverse Power Method with Shifts is a particularly effective eigenvalue extraction routine, for problem formulated using the displacement approach, when only a fraction of the eigenvalues is required. The rudiments of this method are described in Wilkinson*; there it is touted as a powerful method for refining the accuracy of eigenvalues (and eigenvectors) which have been approximately located by other methods. In NASTRAN the method is used as a stand-alone procedure for finding all eigenvalues within a domain specified by the user.

It is a well known fact that the standard inverse power method has a number of defects particular to the solution of structural problems formulated by the displacement approach. These include: awkwardness (of procedure) in the presence of zero eigenvalues (rigid body modes); slow convergence for closely spaced roots; and a deterioration in accuracy, for higher modes, as more roots are found. These defects are eliminated, or minimized, by a modification to the method, introduced and incorporated into NASTRAN.

Note: This procedure was found to be too difficult to use in the extraction of eigenvalues for the tides problems. As a consequence the method was abandoned, early in this work, and, consequently, it will not be described, further, at this time.

*Wilkinson, J. H., THE ALGEBRAIC EIGENVALUE PROBLEM. Clarendon Press, Oxford, 1965.

B.5 THE UPPER HESSENBERG METHOD

B.5.1 Introduction

The Upper Hessenberg method can be used to extract eigenvalues (and describe eigenvectors) for any general, real or complex, system of matrices.

The fundamental reference for this procedure is Wilkinson*. Algorithms, due to Wilkinson, for complex matrices, have been automated and are available (in a modified form) within NASTRAN.

The following outline shows a logical order of the calculations: Reduction to Canonical Form, Reduction to Upper Hessenberg Form, the QR Iteration, Convergence Criteria, Shifting, Deflation, and Eigenvector Description.

B.5.2 Reduction to Canonical Form

The Upper Hessenberg Method requires the eigenvalue problem to be set down in canonical form; i.e.,

$$[A - \lambda I]\{\phi\} = 0.$$

First, Matrix A is reduced to Upper Hessenberg form (by means of transformation techniques). This is performed, automatically, in the NASTRAN module CEAD.

Two equation forms are considered:

(1) $[Mp^2 + Bp + K]\{U\} = 0$ (A general form)

wherein,

$$A \equiv \left[\begin{array}{c|c} 0 & I \\ \hline -M^{-1}K & -M^{-1}B \end{array}\right],$$

$$p \equiv \lambda,$$

(continued)

*Wilkinson, J. H., THE ALGEBRAIC EIGENVALUE PROBLEM. Clarendon Press, Oxford, 1965.

$U \equiv$ The upper half of ϕ. (The lower half of ϕ contains the velocity vector, it is discarded).

(2) $$[Mp^2 + K]\{U\} = 0$$ (the matrix B is missing, now).

Here:

$$A \equiv [-M^{-1}K],$$

$$p \equiv \sqrt{\lambda}\ , \text{ with } Im(p) > 0,$$

and

$$U = \phi.$$

In (2) the decomposition of M is bypassed if it is an identity matrix. In order to reduce to canonical form it is necessary that the matrix be nonsingular. (The order of the problem is doubled when B is not a null matrix).

B.5.3 Reduction to the Upper Hessenberg Form

A given matrix [A] can be reduced to an Upper Hessenberg matrix $[A_0]$ by means of elementary transformations. The basic algorithm, and two alternatives, are shown in Wilkinson (pp. 354-355). The total number of multiplications needed, in the complete reduction, is approximately $(5/6)n^3$; this is half the number used in Householder's reduction, and one-quarter of the number used in Givens' reduction.

B.5.4 The QR Iteration

The QR iteration [Francis*] is defined by Wilkinson, p. 515):

$$[A_o^{(s)}] = [Q^{(s)}]\,[R^{(s)}]\ ,$$

and

$$[A_o^{(s+1)}] = [R^{(s)}][Q^{(s)}].$$

Here $[Q^{(s)}]$ is the product of (n-1) elementary unitary transformations (needed to reduce

*Francis, J.G.F., "The QR Transformation - a Unitary Analogue to the LR Transformation". Parts 1 & 2, Computer Journal, Vol. 4, No. 3 (10/61) & No. 4 (01/62).

$[A_o^{(s)}]$ to an upper triangular form $[R^{(s)}]$ with positive and real diagonal elements); thus,

$$[Q^{(s)}] = [T_s^{(1)}][T_s^{(2)}] \ldots [T_s^{(n-2)}][T_s^{(n-1)}],$$

with

$$[R^{(s)}] = [Q^{(s)}]^{-1}[A^{(s)}].$$

The transformation matrices $[T_s^{(j)}]$ are Givens' rotations (discussed in Wilkinson (p. 239-240), but in complex form for complex matrices). Iterations continue until the n^{th} diagonal element $|a_{n,n-1}^{(s)}| < \epsilon$, the convergence test; at this point the smallest eigenvalue $\lambda_1 \equiv a_{n,n}^{(s)}$. If the convergence proceeds so that $|a_{n-1,n-2}^{(s)}| < \epsilon$, before $|a_{n,n-1}^{(s)}| < \epsilon$, then the two smallest eigenvalues are the roots of

$$\begin{vmatrix} a_{n-1,n-2}^{(s)} - \lambda & a_{n-1,n}^{(s)} \\ a_{n,n-1}^{(s)} & a_{n,n}^{(s)} \end{vmatrix} = 0.$$

The roots will be complex for complex matrices; and will be either real or complex conjugates for real matrices.

B.5.5 Convergence Criteria

The convergence criteria suggested by Wilkinson (p. 526) is based on the Euclidean norm of matrix $||A_o||_E$; it is:

$$\epsilon \equiv 2^{-t}||A_o||_E,$$

for floating-point calculations with a mantissas of t binary bits. The Euclidean norm (Wilkinson, p. 57) is obtained as

$$||A||_E^2 = \sum_{i=1}^{n} \sum_{j-1}^{n} |a_{ij}|^2.$$

Decimal equivalents of the convergence criters (ϵ) are used in NASTRAN.

B.5.6 Shifting

Since the QR iteration converges to a smallest eigenvalue, the convergence can be accelerated by shifting; i.e., by subtracting selected scalar matrices from the original matrix.

The matrix $[A_o^{(s)}]$ is replaced by the difference $[A_o^{(s)}] - k_s [I]$ after each iteration. In this difference k_s is an estimate of the eigenvalue. The shift eigenvalue (k_s) is the root which makes $|a_{n,n}^{(s)} - p_s|$ or $|a_{n,n}^{(s)} - q_s|$ a minimum.

B.5.7 Deflation

When convergence to a single eigenvalue occurs, the Hessenberg matrix $[A_o]$ is "deflated" by the elimination of its last row and column, thus the principal submatrix $[A_1]$, (order one less) is the Hessenberg form used in seeking a next eigenvalue. If convergence occurs to a pair of eigenvalues, the matrix $[A_0]$ is deflated by deleting the last two rows and columns; then the principal submatrix $[A_2]$ (order two less) becomes the basis for seeking a new eigenvalue. Each deflation removes either one or two eigenvalues depending on the convergence tests.

B.5.8 Eigenvectors

The algorithm for discribing the eigenvectors, corresponding to each shift eigenvalue, is the same (here) as that for the inverse power method. The interested reader should consult the NASTRAN Theoretical Manual, or seek more descriptive information from Wilkinson (Reference 1).

B.6 THE FEER METHOD FOR EIGENVALUE EXTRACTION

B.6.1 Introduction

The complex Tridiagonal Reduction Method is an extension of the FEER* algorithm for real eigenvalue analysis to the complex, algebraic eigenproblem formulation.

This method is used to find a specified number of eigenvalues located in the immediate vicinity of a selected "point" in the complex plane; and, in addition, to describe the associated eigenvectors. The eigensolutions are extracted from a symmetric, tridiagonal eigenmatrix whose order is much less than that of the corresponding full-size problem. As a matter of fact, the size of this canonical, reduced matrix is of the order of the number of roots desired (even when the discretized system model possesses thousands of degrees of freedom).

With regard to computational speed, the complex FEER method is somewhat slower than the Hessenberg procedure, for small problems (of the order of one hundred or less degrees of freedom), when all existing eigensolutions are to be obtained. However, this procedure is more efficient than the Hessenberg when the number of requested eigensolutions is much less than full problem size. It should be noted that for very large problems, the required central memory, using the Hessenberg method, exceeds the capability of most large computers; such a restriction does not exist for the Tridiagonal Reduction Method, however.

The complex FEER method uses a single initial "shift point"; hence only one matrix decomposition is required for each such neighborhood chosen (in the complex plane). As a consequence, the FEER procedure is more efficient than the Complex Inverse Power Method, which nominally performs many shifts and decompositions for each chosen search region.

The theory, and computational procedure, for a complex eigenanalysis differs from that for real analysis as follows:

*FEER is an acronym for Fast Eigenvalue Extraction Routine.

1. Both left and right bi-orthogonal vectors must be described during the process of constructing a reduced tridiagonal matrix.

2. The reduced tridiagonal matrix, though symmetric in form is, generally, complex rather than real.

3. The calculated theoretical errors, for computed eigenvalues, are estimates rather than upper bounds.

4. Eigensolutions which are closest to one or more specified (shift) points, in the complex plane, are those found. All eigensolutions acquired from previous shift points are swept out from the solution; this prevents their regeneration while dealing with a subsequent (or current) shift point.

B.6.2 Problem Formulation, Complex Eigenvalue Analysis

A general complex eigenvalue problem begins with an expression of the form:

$$[Mp^2 + Bp + K]\{u\} = 0, \tag{1}$$

where [M], [B] and [K] can be real or complex, symmetric or unsymmetric, singular or non-singular matrices. A specified number of eigenvalues (p) located closest to the specified shift point, λ_o, in the complex plane, are to be found; in addition, their associated eigenvectors {u} are to be described. Incidentally, there are no restrictions regarding the multiplicity of eigenvalues in this procedure.

First, defining a "velocity vector":

$$\{v\} \equiv p\{u\}, \tag{2a}$$

and, a "shift eigenvalue":

$$\lambda \equiv p - \lambda_o, \tag{2b}$$

these are substituted into the expressions above. Then, after inverting, the resulting form would appear as the expression:

$$[A]\{x\} = \Lambda\{x\}\,, \tag{3}$$

wherein

$$[A] \equiv \left[\begin{array}{c|c} K & B+\lambda_o M \\ \hline -\lambda_o I & I \end{array}\right]^{-1} \left[\begin{array}{c|c} O & -M \\ \hline I & O \end{array}\right]\,, \tag{4a}$$

$$\{x\} \equiv \left\{\begin{array}{c} u \\ \hline v \end{array}\right\}\,, \tag{4b}$$

and, where:

$$\Lambda \equiv \frac{1}{p-\lambda_o}\,. \tag{4c}$$

This development shows the eigenvalue problem in standard form. One implication here is that the order of the eigenvalue problem is doubled when the [B] matrix is present in the formulation statement.

For those cases where [B] is a null matrix (i.e., damping is absent) the formulation reduces to:

$$[Mp^2 + K]\{u\} = 0. \tag{5}$$

For this statement the double-size eigenvalue problem is avoided when the mathematical eigenvalue is defined as p^2. Thus, in place of Equation (2b) write:

$$\lambda^2 \equiv p^2 - \lambda_o^2\,, \tag{6a}$$

and, consequently,

$$\Lambda \equiv \frac{1}{\lambda^2}\,. \tag{6b}$$

Substituting these parameters into the reduced formula (above) and using the inverse form, it follows that:

$$[K+\lambda_o^2 M]^{-1}[-M]\{u\}=\Lambda\{u\}. \tag{7}$$

Comparing this expression with Equation (3) we see that the standard form, for a null [B] matrix, has, as coefficients,

$$[A] \equiv [K+\lambda_o^2 M]^{-1}[-M], \tag{8a}$$

and

$$\{x\} \equiv \{u\}. \tag{8b}$$

Since the eigenmatrix [A] is, generally, unsymmetric, then the eigenvectors, $\{x\}$, are orthogonal to the eigenvectors, $\{\bar{\bar{x}}\}$, of the transpose eigenproblem; thus the problem is also described by:

$$[A]^T\{\bar{\bar{x}}\} = \Lambda\{\bar{\bar{x}}\}. \tag{9}$$

As a consequence, when $\Lambda_i \neq \Lambda_j$, then

$$\{\bar{\bar{x}}_j\}^T\{x_i\} = 0; \quad i \neq j. \tag{10}$$

This relationship shows the biorthogonality condition; the eigenvectors, $\{x_i\}$ and $\{\bar{\bar{x}}_j\}$ are then referred to as the right and left eigenvectors, respectively.

B.6.3 The Reduction Algorithm

A reduction in the order of the eigenvalue problem, Equation (3), is effected through transformations which are selected to be biorthonormal. (See Reference (1)* for a detailed discussion).

*(1) Newman, Malcolm and F.I. Mann, "Complex Eigenvalue Extraction in NASTRAN by the Tridiagonal Reduction (FEER) Method", AMA Report No. 77-17, September 1977.

As in the case of a real eigenvalue analysis (see Reference (2)*), the Lanczos algorithm is used to construct the transformation matrices, vector by vector. This will reduce the transformed matrix to a tridiagonal form with the eigenvalues accurately approximated by those roots from Equation (3) having a largest magnitude (or, equivalently, to the roots, p, which are closest to the specified point of interest, λ_o, located in the complex plane).

After the transformation has been affected (see Reference (1) for details) the reduced eigenmatrix is found to be a tridiagonal and symmetric matrix.

The eigenvalues, and eigenvectors from the transformed system, are extracted using the Q-R iteration algorithm; and, the eigenvector computational scheme is that described in the Upper Hessenberg method for NASTRAN.

Before proceeding, the velocity vector $\{v_i\}$ is discarded (prior to any further processing of the eigensolutions by NASTRAN). Additionally, any solution which fails the FEER error test is rejected. Nevertheless, the number of accepted solutions will, in all probability, equal or exceed the number requested by the user (when the reduced problem size is chosen according to the criteria described below).

B.6.4 Criteria for Establishing the Reduced Eigenvalue Problem Size

The maximum number of finite eigensolutions, including any existing rigid body modes, is equal to the rank, r, of the eigenmatrix [A] in the standard form. Thus, any massless degrees of freedom, appearing as zero diagonal terms in the [M] matrix, will result in singularities (and rank reduction). This could imply infinite valued physical eigenvalues. However, any such spurious roots are swept out of the problem in the complex FEER process. As a consequence there is a reduction to the available eigensolutions.

A further option, limiting the maximum problem size, is that the user is able to request eigensolutions in the neighborhood of several shift points, (e.g., λ_{01}, λ_{02}, ...)

(2) Newman, M. and P. F. Flanagan, "Eigenvalue Extraction in NASTRAN by the Tridiagonal Reduction (FEER) Method - Real Eigenvalue Analysis", NASA CR-2731, August 1976.

located in the complex plane. Recall that for the Tridiagonal Reduction Method, all eigensolutions obtained from a previous shift point are swept out of the problem to prevent their regeneration at a current point. This implies a limit for the maximum possible size of the reduced problem, also.

On the basis of numerical experiments, similar to those cited in Reference (1), for the real eigenvalue analysis, it has been found that there is a real lower limit to this maximum size of the reduced problem. As a practical lower limit on size, based on accuracy of the associated eigenvectors, a value of twelve is used. Consequently, when the user requests a total of (say) q eigenvalues, closest to a specified point in the complex plane, the order of the reduced problem is initially set to

$$m = \min[(2\bar{q}+10),\ (2n-f)]; \quad \text{for } [B] \neq [0],$$

or

$$= \min[(2\bar{q}+10),\ (n-f)]; \quad \text{for } [B] = [0],$$

where n is the order of the unreduced problem and f is the number of eigensolutions.

Although the total number of eigensolutions requested should not exceed the maximum, there is (usually) no simple way of discerning the upper limit for a complex problem. However, the reorthogonalization tests are designed to automatically establish this upper limit; and, if these tests fail there is an indication that a null vector has been generated.

B.6.5 Choice of the Initial Trial Vectors and Restart Vectors

Because of the inverse relationship between the computed eigenvalues, and the physical eigenvalues, spurious eigensolutions corresponding to a zero are equivalent to physical eigenvalues which approach infinity. Since these eigenvalues and their corresponding eigenvectors are of no interest, and could cause numerical instabilities, they are eliminated from the reduced tridiagonal problem. For more particulars on this operation, the reader is referred to Reference (1).

B.6.6 The Sweeping-out of Previously Obtained Eigenvectors and Reorthogonalization of the Trial Vectors

As in the real eigenvalue analysis, successive trial vectors tend to degrade rapidly as computations proceed in a finite digit machine. Consequently, the right vectors, generated later in the analysis, are far removed from the orthogonality related to earlier left vectors. Therefore, new vector pairs, as obtained, are reorthogonalized with respect to all the previously obtained vectors. This procedure is carried out iteratively.

B.6.7 Error Estimates for the Computed Eigenvalues

Following a development similar to that in Reference (1), for a real eigenvalue analysis, it can be shown that the difference between computed and true eigenvalue magnitude is proportional to the magnitude of the next off-diagonal term that would be generated, if the reduced tridiagonal matrix [H] had been increased by one in its order, times the last term in the reduced system eigenvector.

APPENDIX C

PROGRAM INPUT AND CONTROL CARDS

The use of NASTRAN to solve for eigenvalues in a tidal modes problem represents a departure from normal operations in utilizing the program. The fact that NASTRAN was designed to handle large sized analytically expressed problems, and that it included several algorithms for eigenvalue extractions, prompted the investigators to propose its use in this study. In addition, this program was designed and developed, primarily, to handle problems which employed the Finite Element Method as a solution technique.

The very nature of this application of the Laplace Tidal Equations, and the use of the FEM, suggested a need for NASTRAN and its computational capabilities. In this regard, however, it should be recognized that only a relatively small portion of the program's versatility is being called upon here. To some degree this lessens the input requirements, to the computer, even though the input stream (for the matrices to be manipulated) is rather extensive.

Basically, here, there were three versions of the NASTRAN system utilized during the course of this investigation. All of these were (obviously) available to users of the GSFC (360/95) system. Initially the version used was one of the NASTRAN 15 series. Later, a McNeil-Schwendler (38) scheme was utilized; and, during the terminal stages of the study, a modified 16.01 system was put to use. This (last) program was modified to accommodate the FEER complex-eigenvalue routines; here the investigators were able to take advantage of a most recent development in eigen-analysis, and did so prior to the release of those algorithms to the general user community.

In the next few paragraphs some explanations will be offered to acquaint the reader with the necessary JCL, etc. required to operate NASTRAN's eigenvalue extraction routines.

C.1 NASTRAN OPERATION MODES

Before launching into these discussions it may be well to mention that two "operating types" of NASTRAN were employed in this investigation.

One of these -- referred to as NASFAST -- is a less formal and all encompassing version of NASTRAN. This modified program was developed at GSFC and is useable on both the "usual" versions and the McNeil-Schwendler systems for NASTRAN.

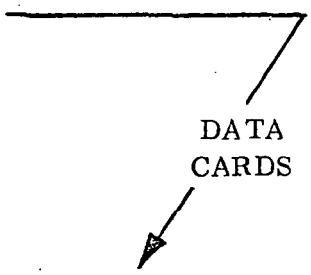
First the JCL-control card set up will be described, for both normal and NASFAST operations; then a discussion on input format and requirements will be given. These remarks will certainly acquaint the reader with the "needs" of an investigator in setting up an eigenvalue problem of this type.

(a) The NASFAST Program Operation. The "cards" needed for these run types are listed below; some comments on these will be included following the card notations. Incidentally, the numbers, at left, are for future reference; input cards are not numerically identified. The control deck, etc. is left justified.

```
 1.   A "JOB CARD"
 2.   // EXEC NASFAST, PARM='NAME=GODDARD', REGION=400K
 3.   //STEPLIB DD UNIT=2314, DISP=SHR,
      //      DSN=N2RSM.MSC38.LOAD.VOL=SER=DAGPAK
 4.   //FTØ7ØØ1 DD DUMMY
 5.   //SYSIN DD *
 6.   ID TIDAL, MODES
 7.   APP DISPLACEMENTS
 8.   SOL 7,Ø
 9.   TIME 3
10.   DIAG 7,8,13,19
11.   CEND
12.   TITLE CARD
13.   SUBTITLE CARD
14.   LINE=7Ø
15.   CMETHOD=1
16.   SPC = 1
17.   LABEL CARD
18.   M2PP=AMAT
19.   K2PP=BMAT
```

(continued on next page)

```
20.   OUTPUT
21.   DISPLACEMENT(PHASE)=ALL
22.   BEGIN BULK
23.
```




```
ENDDATA
/*
```

Comments:

(1) Card 2. locates the "NASFAST" program and requests 400K of core for operations.

(2) Card 3. identifies the operating program.

(3) Card 4. implies no decks are to be punched.

(4) Card 5. states control - cards are to follow.

(5) Cards 6, 7, 8 identify problem type and operation.

(6) Card 9. constrains the CPU time used prior to "shut-off".

(7) Card 10. denotes the diagnostics to be invoked.

(8) The next card signals an end to the controls.

(9) Titles and sub-titles provide printed statements, to be generated so that identifications can appear in the output strings.

(10) Card 14. controls the output lines (printed) per page of output.

(11) Cards 15. and 16. describe which method, and group of (SPC) cards are to be used in the solution. This suggests that input decks can be set-up to handle multi-solutions problems.

(12) Cards 18. and 19. allocate, or assign, space for the (input) matrices to be operated on.

(13) Next the output, to be generated, is identified here.

(14) Following the BEGIN BULK cards, the analyst should introduce the input to the system. These data include the "collected" (matrix) parameters plus the cards for geometry, boundary conditions, etc.

(15) Finally, the termination cards appear -- these are the ENDDATA, /* pair.

(b). The NASTRAN Program Operation. The more usual NASTRAN runs, and especially those operating under the CHECKPOINT options are illustrated by the next input string. (The CHECKPOINT option is used when it is uncertain, or desired, that the program's computations be interrupted. At the interruption, information, within the computational framework, is transferred to "storage" where it can be recalled, subsequently, for a continuation of the calculations procedure). Also, shown below, the input deck is introduced to the program from tape -- this is especially useful with the large sized input connected with the "big" models of the basins. Other features are noted by comments made for the control cards.

Once again, all cards are left justified and commence in column 1. A sample deck is shown below:

```
1.  JOB CARD
2.  // EXEC NASTRAN,REGION=700K,P1=30,
          SUNITS=CYL,SI=8,TI=10
3.  //STEPLIB DD UNIT=2314,DISP=SHR,VOL=SER=NAST52,DSN=N2RSM.NAST1601.LOAD
4.  //FT01F001 DD SPACE=(CYL,(17,1),RLSE)
5.  //NPTP DD UNIT=2400-9,DISP=(NEW,KEEP),LABEL=(,BLP),
6.  //          DCB=DEN=3,DSN=ZOJBE.ERIE,VOL=SER=33271
7.  //OPTP      DD UNIT=2400-4,DISP=(OLD,KEEP),LABEL=(1,SLP),
8.  //            DSN=ZOJBE.ERIE,VOL=SER=33040
9.  //SYSIN DD *
10. ID ERIE,MODES
11. APP DISPLACEMENT
12. SOL 7,0
13. CHKPNT YES
14. TIME 10
15. DIAG 7
16. DIAG 1,8,15,19
17. RESTART  ERIE     ,MODES    , 5/ 2/77,    86074,
.
.
.
18. CEND
19. TITLE = FINITE ELEMENT MODEL FOR TIDAL MODES IN LAKE ERIE
20. SUBTITLE = REPRESENTATION VIA TWO-DIMENSIONAL TRIANGULAR FLUID ELEMENTS
21. MAXLINES = 1000000
22. LINE = 70
23. ECHO = NONE
24. CMETHOD =1
25. SPC = 1
26. MPC = 101
27. SUBCASE 1
28. LABEL = CALCULATION OF LOWEST FREQUENCY MODES
29. M2PP = AMAT
30. K2PP = BMAT
31. $$$$$  DEPTH IN FATHOMS
32. $$$$$ PLANFORM DIMENSIONS IN 10,000 YARD UNITS
33. $$$$$  TIME IN SECONDS        DETERMINANT METHOD
34. OUTPUT
35. DISPLACEMENT(PRINT,PUNCH) = ALL
36. BEGIN BULK
37. /        23557    23559
38. EIGC     1        DET      MAX                                  1.0-9                +EIG
39. +EIG     5.0-3    5.0-3    1.0-2    1.0-2    1.0-4    3        3                    +EIG1
40. +EIG1    1.0-2    1.0-2    2.0-2    2.0-2    1.0-4    3        3
41. ENDDATA
42. /*
```

Comments: Those cards which are repeats of the foregoing ones will be noted by reference.

(1) Following the job card(s), the call to NASTRAN, and desired machine peripherals, plus core estimates are noted on 2.

(2) Card 3. identifies the NASTRAN version to be used; here operations are to employ version 16.01.

(3) Cards 5. and 6. identify the tape onto which the CHECKPOINT data are to be stored.

(4) Cards 7. and 8. note the 'INPUT TAPE" for initiating the run. In this case the raw data are called from the "ERIE" tape -- this was generated by the Pre-processor program.

(5) Cards 9. through 12. are the same as before.

(6) Card 13. denotes the use of the CHECKPOINT option in this run.

(7) Card 14. shows the requested CPU time (in minutes) to be expended before the CHECKPOINT option is enforced.

(8) Cards 15. and 16. indicate the diagnostics requested for the run.

(9) Cards 17. are the RESTART deck generated by the CHECKPOINT option. These cards, plus the tape called by cards 5. and 6. are needed to reinitiate the computations in a future run operation.

(10) Card 21. is inserted to assure that the program will not be terminated by machine limits on line printing.

(11) Card 22. is used to denote the line p rint per page output.

(12) Card 23. indicates that the input is not to be printed on output.

(13) Cards 24. through 27. are analogous to 15. and 16. of the former program. Here more input types are specified than previously.

(14) Cards 29. and 30. are the same as before.

(15) Cards 31. through 33. are comment identifiers.

(16) Cards 34. through 36. are as before.

(17) Card 37. indicates a range of deleted cards (denoted by count numbers); cards 38. through 40. are the replacements.

(18) The last two cards are the termination indications for the input stream.

C.2 INPUT DATA DECKS

To illustrate how the input data (raw data) are formatted, for NASTRAN, the following examples are shown and commented upon.

First, the "standard" matrix input will be shown. Second, the "expanded" format will be illustrated, with comments added.

(a) Standard Entries. The data string, below, was taken from a run made on the test (square) basin used in this study. This illustrates the required format (for matrices data and other input information).

SORTED BULK DATA ECHO

CARD COUNT	1	2	3	4	5	6	7	8	9	10
1-	CELAS2	13	.0	12	1	13	1			
2-	DMIG	AMAT	0	1	1	2				
3-	DMIG	AMAT	1	1		1	1	3.6E9		+AM1
4-	+AM1	2	1	9.CE8		9	1	1.8E9		+AM3
5-	+AM3	10	1	9.CE8						
6-	DMIG	AMAT	1	2		1	2	37.50E4		+AM5
7-	+AM5	2	2	93.75E3		9	2	18.75E4		+AM7
8-	+AM7	10	2	93.75E3						
9-	DMIG	AMAT	1	3		1	3	37.50E4		+AM9
10-	+AM9	2	3	93.75E3		9	3	18.75E4		+AM11
11-	+AM11	10	3	93.75E3						
12-	DMIG	AMAT	2	1		1	1	9.0E8		+AM13
13-	+AM13	2	1	3.6E9		3	1	9.0E8		+AM15
14-	+AM15	9	1	1.8E9		10	1	.0		
15-	DMIG	AMAT	2	2		1	2	93.75E3		+AM17
16-	+AM17	2	2	37.5E4		3	2	93.75E3		+AM19
17-	+AM19	9	2	18.75E4		10	2	.0		
18-	DMIG	AMAT	2	3		1	3	93.75E3		+AM21
19-	+AM21	2	3	37.50E4		3	3	93.75E3		+AM23
20-	+AM23	9	3	18.75E4		10	3	.0		
21-	DMIG	AMAT	3	1		2	1	9.0E8		+AM25
22-	+AM25	3	1	7.2E9		4	1	9.0E8		+AM27
23-	+AM27	7	1	1.8E9		8	1	1.8E9		+AM29
24-	+AM29	9	1	1.8E9						
25-	DMIG	AMAT	3	2		2	2	93.75E3		+AM31
26-	+AM31	3	2	75.0E4		4	2	93.75E3		+AM33
27-	+AM33	7	2	18.75E4		8	2	18.75E4		+AM35
28-	+AM35	9	2	18.75E4						

Comments: The input card count is shown at left; the data fields are listed to the right, with continuation counters shown at the far right edge. Note that the format here shows 10 fields of eight characters each -- this is a standard format; but, obviously, is restricted in the numerical accuracy for the input.

(1) Card 1. is a "spring" connecting nodes 1. for elements 12. and 13. This is a zero modules spring; the card is inserted to satisfy requirements in NASTRAN; some "physical data" are needed in the input sequence.

(2) Card 2. notes that Direct Matrix Input Groups are to follow. This identifies the A-matrix group -- later the B-matrix input will be indicated.

(3) Cards 3. through 5. describe the input elements, for the A-matrix's first column, with each row component indicated and the parameter value noted (see Equation (40a)). Here Column 1 has entries in rows (1,1), (2,1), (9,1) and (10,1) of the collected A-matrix. Cards 6. through 9. show the input data in the second column (of the A-matrix), where the second column is denoted as (1,2) -- the central element of the first grouped matrix -- again, see Equation (40a). (Values shown in the third and eighth fields are the collected matrix elements for the elements joined together from the sub-domain matrices). Note the last, and succeeding first, entries (per card) indicate the continuation counters for the row entries from the collected matrices.

This pattern is followed until the entire A-matrix is introduced to the program via the input format shown above. Next, the same procedure is followed for the B-matrix (see Equation (40b)). Below we find a sample of the B-matrix input. There the entry for node (25), column (3), is shown; note that contributions occur from nodes (hence rows) 16., 17., 24., and 25.. Formatting here is the same as previously described for the A-matrix.

```
653-    DMIG     BMAT    25      3                       16      1       24.0E5          +BM385
654-    +BM385   16      2       -5.625                  17      1       24.0E5          +BM386
655-    +BM386   17      2       -11.25                  24      2       -5.625          +BM387
656-    +BM387   25      1       24.0E5                  25      2       -22.50
657-    EIGC     1       DET     MAX                             1.0-11                  +EIG
658-    +EIG     .12     .12     .26     .26     .001    21      24
659-    GRDSET                                                           456
660-    GRID     1               .0      .0      .0
661-    GRID     2               .2500   .0      .0
662-    GRID     3               .5000   .0      .0
```

Comments: Following the B-matrix entries, on cards (657. and 658.), there is information regarding which eigenvalue extraction routine is to be used, and a notation on the "search region" to be examined.

(1) The EIGC card (control card) says this is METHOD 1 (from input control), and that the DETERMINANT Method (see Appendix B) is used. Also, noted is error limits for the eigenvalues.

(2) The EIG card sets the search regions, for this extraction procedure, and indicates the numbers of roots expected and sought.

(3) The GRDSET, and GRID, cards describe locations for the nodes in the solution domain. Here GRID (point) 1 is situated at the origin; point (3) is located at $x = 0$, $y = \frac{1}{2}$, with upper limits on (x, y) being set at 1.0.

```
685-    SPC1     1       2       6       10      11      15      16      20
686-    SPC1     1       3       2       3       4       22      23      24
687-    SPC1     1       23      1       5       21      25
        ENDDATA
```

(4) The SPC1 cards (see input control) specify boundary conditions on the u (#2 variable) and v (#3 variable) for this problem. Here $u = 0$ at nodes indicated by the last six entries on card 685; $v = 0$ for the nodes indicated on card 686; and both are zero at the nodes (cornors) shown on card 687.

(5) Finally, the end of data card is reproduced (echoed) here.

(b). Expanded Entries: Counter to the standard matrix entry format, there is an expanded (or extended) format which is illustrated below. The reproduced data string excerpts are shown, and comments offered.

```
BEGIN BULK
CELAS2    100       0.0       12        1         13        1
CMASS2*                   1  .1771027500E-05                  1         1
CMASS2*                   2  .3750000000E+00                  1         2
CMASS2*                   3  .3750000000E+00                  1         3
CMASS2*                   4  .3075147750E-05                  2         1
CMASS2*                   5  .7500000000E+00                  2         2
CMASS2*                   6  .7500000000E+00                  2         3
CMASS2*                   7  .1198753058E-04                  3         1
CMASS2*                   8  .1166666667E+01                  3         2
CMASS2*                   9  .1166666667E+01                  3         3
```

Comments: Following the "BEGIN BULK" card a CELAS2 (null spring) card appears, followed by a series of CMASS2 cards. These are, again, physical data entries which are needed for the NASTRAN input. Actually these "mass" cards are made up of numbers taken from the A-matrix diagonal elements -- the element loci are indicated by entries in the last two columns in each line of data. Necessarily this splitting of A-matrix numbers is properly accounted for by the subsequent values used in the A-matrix entries, per se.

To illustrate the nature of the expanded format an entry for one column of the B-matrix is shown below. Here the 124th node is implied (by the DMIG card) and column 3 of the collected matrix is indicated. Data entries from (adjacent) nodes are indicated by the numbers in the second and third columns; the appropriate B-matrix values are found in column 4; the last column lists the continuation identifier. The continuations cards are "matched" in the last and first columns, respectively.

```
DMIG*    BMAT                    124            3                         *B04816
*B04816        121               1              -.6518991667E-06          *B04817
*B04817        121               2              -.0                       *B04818
*B04818        122               1              -.1511634583E-05          *B04819
*B04819        122               2              0.                        *B04820
*B04820        123               1              -.8855137500E-06          *B04821
*B04821        123               2              -.0                       *B04822
*B04822        124               1              -.1735249167E-05          *B04823
*B04823        124               2              0.                        *B04824
*B04824
EIGC     1        FEER    MAX                                             +EIG
+EIG     1.-2     1.-2                                            30
GRDSET                                                            456
GRID     1                1.      6.      12.
GRID     2                4.      5.      6.
GRID     3                4.      8.      68.
GRID     4                7.      6.      7.
GRID     5                6.      10.     64.
```

Comments: The EIGC card (here) says this is the first solution type, that the FEER extraction procedure is to be used; and that 30 roots are expected in the extraction process. The location for a central point in the search region is given on the EIG card.

Following the EIG cards is the set of GRID (node) descriptors. The entries (here) are given (left to right) in terms of x, y, z expressed in the units chosen for the global coordinates.

After the data cards comes a set of MPC cards. These are entries which are needed to satisfy the oblique boundary condition of "no-flow over the boundary" (see the section on Boundary Conditions). Following the MPC data will be an SPC set (if such is needed); these too are boundary condition entries.

C.3 OUTPUT DATA

The basic output information is generally described as: (1) A listing of the eigenvalues -- listed in ascending order according to imaginary root components. (2) A collection of eigenvectors, one for each root extracted. The eigenvectors can be requested in various formats (the usual one called here listed amplitude and phase).

Of course, with the output can be an ECHO of the input data string (see input cards, above), and added printout according to the DIAGNOSTICS asked for in the control deck. These data are very useful in searching out errors, and in learning about the operations of various program procedures. The variety of output, its meaning and utility are beyond the scope of this effort. The interested reader should consult the operations and users manuals for NASTRAN.

It should be apparent that the NASTRAN system is one with a multitude of complexities, many virtures and uses, and one with a multitude of opportunities for making blunders and committing errors. To fully utilize the system requires much study and "hands-on" experience. The uninitiated is cautioned to be wary - seek the advice and guidance of the experts who know the system and its characteristics.