DOE/ER/25187--T1

Computational Physical Oceanography – A Comprehensive Approach based on Generalized CFD/Grid Techniques for Planetary Scale Simulations of Oceanic Flows

Murali Beddhu, Min–Yee Jiang, David L Whitfield,

Lafayette K. Taylor and Abdollah Arabshahi

CFD Laboratory, NSF/ Engineering Research Center

Mississippi State University

Grant No. DEFG05–93ER25187

Final Report : 09/01/1995 – 08/31/1996

Feb. 20, 1997

DISCLAIMER

Computational Physical Oceanography– A Comprehensive Approach Based on Generalized CFD/Grid Techniques for Planetary Scale Simulations of Oceanic Flows

By

Murali Beddhu, Min-Yee Jiang, David L. Whitfield, Lafayette K. Taylor and Abdollah Arabshahi

Computational Fluid Dynamics Laboratory
NSF Engineering Research Center for Computational Field Simulation
March 1997

MSSU-EIRS-ERC-97-5

Mississippi State University
P.O. Box 9627
Mississippi State, MS 39762

Abstract

The original intention for this work was to impart the technology that was developed in the field of "computational aeronautics" to the field of computational physical oceanography. This technology transfer involved grid generation techniques and solution procedures to solve the governing equations over the grids thus generated. Specifically, boundary fitting non–orthogonal grids would be generated over a sphere taking into account the topography of the ocean floor and the topography of the continents. The solution methodology to be employed involved the application of an upwind, finite volume discretization procedure that uses higher order numerical fluxes at the cell faces to discretize the governing equations and an implicit Newton relaxation technique to solve the discretized equations. This report summarizes the efforts put forth during the past three years to achieve these goals and indicates the future direction of this work as it is still an ongoing effort.

Acknowledgement

This work was supported by the Department of Energy under Grant DE–FG05–93ER25187. This line of work was initiated by Dr. Gary M. Johnson (who was the first monitor of this computational physical oceanography grant) through discussions with Dr. Joe F. Thompson of the Engineering Research Center, Mississippi State University. The idea was to investigate the applicability of methods being developed in the field of computational aeronautics to the field of ocean modeling. Dr. Fred Howes, Program Manager, Office of Scientific Computing, is the current monitor. This support is greatly acknowledged.

Table of Contents

List of Figures

Introduction

Two of the widely used methods for predicting ocean flows are the σ–coordinate approach and the z–coordinates approach. The basis for the existing technology is at least a decade old in the case of the σ–coordinates and more than three decades in the case of the z–coordinates. With the then available computer resources in mind, various simplifying assumptions were made based on physical judgements and/or intuitive reasonings and a simplified set of equations were solved to predict the ocean flows. Aptly, this approach is called the ocean modeling approach. While advances have been made over the years in terms of improving the solution methodology, using parallel computers for example, the basic physical premises have remained unaltered. However, there are questions about the validity of these premises when they are applied to situations other than those for which they were originally intended. For example, the use of the hydrostatic equation for coordinates other than spherical or Cartesian is not well justified. In other cases, equations which do not strictly confirm to the assumptions of shallow water theory are routinely being used to compute flows under the name of shallow water theory and they are used in situations for which shallow water theory does not apply.

In computational physical oceanography, the full Navier–Stokes equations are solved thus avoiding the many pitfalls of trying to model ocean flows. Application of the Navier–Stokes equations offer several advantages: (1) The equations can be written in conservative form and in cases where that is not possible such as when including the effects of buoyancy etc., the number of source terms can be kept to a minimum, (2) since the Navier–Stokes equations are tensor invariant, numerical methods can be generated in a general setting. These methods appear to be much more stable and robust compared to the methods developed for the approximate equations (example, viscous–inviscid interactions) and have a wider range of applicability, (3) expressing the Navier–Stokes equations in general curvilinear coordinates allows one to solve problems involving complicated geometries with accurate representation of geometries

as well as offers the flexibility of packing grid lines in regions of interest and sparsely distributing them in regions with smaller gradients, (4) specification of boundary conditions on curvilinear coordinate surfaces is straight forward. The main disadvantages are that it might take more computer memory and longer CPU time to solve the problem. With modern computers these disadvantages are less stringent today compared to the past decade and it is anticipated that the situation in the next decade will be much better, particularly with parallel processing. In addition, techniques such as the multigrid method have been developed that are useful in accelerating the convergence of a Navier–Stokes solver for both steady and unsteady flows.

From another point of view, even though flow in a compressor or turbine of a turbomachine is entirely different from atmospheric and oceanic flows, it also falls in the category of flows in a rotating frame and typically involves Reynolds numbers of the order of 10^5 to 10^9. It is becoming increasingly common to use Navier–Stokes equations to solve the flow field in a turbomachine, even at the design stage, even though various models with various simplifying assumptions were in widespread use about a decade earlier. If one were to follow this path and uses the viscosity of water as the reference value in computing the Reynolds number for planetary scale ocean flows, the Reynolds number turns out to be of the order of 10^{15}. In order to resolve the boundary layer properly at such a high Reynolds number one needs to have a few grid points in the viscous sublayer and that implies a grid resolution of the order of millimeters. While such a flow is obviously difficult to compute, it is not impossible. An example calculation is presented later.

The present work is not concerned with ocean modeling per se. Rather, the objective is to use concepts from the field of "computational aeronautics" where appropriate and apply them to the field of ocean flow simulations/predictions. An additional objective is to extend and develop new tools dealing with the appropriate equations as well as numerical methods needed to solve the equations. Consequently, the present work is referred to herein as computation-

al physical oceanography (CPO) as opposed to ocean modeling. In the next section the state of the art in ocean modeling is reviewed which is followed by a discussion of the present approach, computational physical oceanography, in Section 3.

2. Review of Ocean Modeling

Historically, the first coordinate system that was used for computing ocean flows is the so called z–coordinates. In this coordinate system, in basin level models, for example, an f–plane or β–plane approximation is made for the Coriolis force and the equations are expressed in local Cartesian coordinates, with the x–direction increasing positive along the East direction, y–direction increasing positive along the North direction and the z–direction increasing positive in the vertical direction (opposite to the local gravity vector). The important property of this coordinate system is that it is orthogonal. To be more precise, it should be noted that the vertical z is truly orthogonal to the horizontal directions x and y. For global level models the spherical polar coordinates are used with the latitude θ, longitude λ and the vertical $z = r - a$ where, r is the radial distance of the point under consideration and 'a' is the radius of earth, as the coordinates. It can again be noted that the vertical coordinate is truly orthogonal to the planes containing the horizontal coordinates which are spherical surfaces with constant radii. It is well documented in the literature that the topography of the ocean floor, which is highly irregular, plays a crucial role in determining the ocean currents. The difficulty in using the z–coordinate system, for solving the ocean flow problem, is that the ocean floor can not be represented by a single coordinate surface. This led to the so called stair case representation of the ocean floor which introduces considerable error in representing the bottom topography. In addition, the shape of the continents is not preserved properly because again a stair case representation is used. The staircase representation results in a grid in which the number of points along different grid lines of the same family is different. In addition, selectively packing the grid lines near regions of interest is very difficult, if not impossible. Difficulties in the specification of boundary conditions as well as in coding make this approach unattractive. Thus, the simplicity of the equations are almost lost in the complexity of coding them over such a grid.

The best example of a z–coordinate model is the Bryan–Cox–Semtner model first introduced by Bryan [1] and later coded by Cox [2]. A finite difference formulation suitable for vector processors can be found in Semtner [3]. In this model, the free surface is approximated by the rigid lid assumption by which it is meant that the free surface is not allowed to evolve in time. As satellite altimetry data is becoming increasingly available, the sea surface height is becoming a reliable dataset that can be used to compare with models. So, many models now include some sort of approximations to predict the free surface evolution. Killworth et al [4], for example, have incorporated the free surface capability in the Bryan–Cox model.

Sigma coordinates were introduced by Blumberg and Mellor [5] in order to alleviate the stair–case problem in the vertical. Thus, in this system the region from the free surface to the bottom, in other words the vertical coordinate z is mapped to the coordinate σ in the interval [–1, 0] with the bottom at –1 and the free surface at 0. In this coordinate system some authors have preferred to vertically integrate the equations while others have retained the equations after applying all the appropriate approximations. It needs to be noted that even though the equation system that is not vertically integrated is called a three dimensional system, it is not truly three dimensional in the sense that the vertical momentum equation has been replaced by the hydrostatic equation. Thus, the number of equations that are actually solved to update the flow variables is reduced by one. Since both the free surface and the bottom topography vary in the θ and λ directions, where θ and λ represent latitudes and longitudes respectively, their derivatives with respect to θ and λ appear in the θ–momentum and λ–momentum equations. There is sufficient evidence in the literature that the σ–coordinates suffer the same difficulties as the previous stair–case approach near steeply varying bottom topography. In the σ–coordinate system "horizontal" surfaces are nonorthogonal to the vertical coordinate, especially in regions having steep bottom and/or free surface gradients. Bryan [1], in his now classic paper, states that the hydrostatic approximation may be shown to be highly accurate as long as the

aspect ratio of bottom topography is much less than unity. However, the aspect ratio of the bottom topography is comparable to unity in regions having steep bottom and/or free surface gradients. Haney [6] examines the accuracy of the "horizontal" pressure gradient over steep bottom topography in the σ–coordinate system without questioning the hydrostatic equation. It must be emphasized that the hydrostatic equation is not a coordinate system independent approximation and is strictly valid only for a spherical coordinate system or a Cartesian coordinate system in which one of the coordinate lines is aligned in the direction (opposite to that) of the gravity vector and the planes containing the other two coordinates are orthogonal to this direction. A natural approach to see whether the hydrostatic equation is valid for the σ–coordinate system would be to write down the complete vertical momentum equation in the σ–coordinate system and then introduce the order of magnitude approximation along the lines of Pedlosky [7]. When this approach is taken, it appears that the hydrostatic equation seems to be invalid for the σ–coordinate system in regions with steep bottom and/or free surface gradients. This issue is further discussed in Section 4.6.

An interesting approach with the vertically integrated equations in the σ–coordinates was taken by Borthwick and Kaar [8], in which they introduce curvilinear coordinates in the horizontal after integrating the equations in the vertical. This approach allows one to conform to the boundaries better and allows greater flexibility in spacing the grid lines as well as allows one to use the same number of grid points on all the grid lines representing a particular coordinate. While the application of this approach is not straight forward in the case of the "three dimensional" σ–coordinates, the important message of this work is the use of the curvilinear coordinate system in the horizontal to represent the side boundaries. Simple test cases where such accurate representation of the side boundaries are needed are presented by them. It must be noted here that when one uses a nonorthogonal curvilinear grid in the "horizontal" with the

σ–coordinates in the "vertical", one is actually using a general nonorthogonal coordinate system, similar to the one used in this work.

There is growing doubts within the ocean modeling community about the validity of the approximations made in deriving the shallow water equations. Especially, the approximation under more scrutiny is the hydrostatic approximation. The vertical component of velocity, however small it may be, is needed in order to transport mass (both water and salt) as well as heat in the vertical direction. In the field of Aerospace Engineering, even though the classical boundary layer theory treated the component of velocity normal to a viscous surface as being small, it still had to take that velocity component into account. Even in Ekman theory, in the field of Ocean Engineering, the Ekman pumping is obtained by not neglecting the vertical component of velocity. So the obvious inference is that the normal component of velocity from a viscous surface in a complicated three dimensional flow has to be accounted for, however small it may be.

Experience with boundary layers have again suggested that the best way to include the effects of the normal component of the velocity is to integrate the complete set of Navier–Stokes equations with highly packed grid points near the viscous wall rather than trying to make approximations to the governing equations and solving them as a viscous–inviscid interaction problems. The advent of very powerful computers makes it possible to integrate the Navier–Stokes equations over very complicated geometries involving a variety of length and time scales. So, it is opined that such applications of the Navier–Stokes equations are possible for the planetary scale ocean problems as well.

3. Computational Physical Oceanography – Present Approach

Grid generation and solution methodology over nonorthogonal curvilinear grids constitute the two basic elements of the present approach. Grid generation is a fairly mature area and standard packages are available for generating grids, and talented and experienced grid generators can produce good grids over complicated geometries. The earliest work in grid generation using elliptic equations was done by Winslow [9]. Algebraic grid generation techniques, and elliptic grid generation techniques, further developed by Thompson et al [10], Thompson [11], and Warsi [12] among others, are the most common grid generation techniques used today. Further details about these and related topics may be obtained from the book by Thompson, Warsi and Mastin [13]. One of the more widely used grid generation packages is the EAGLEView [14] which was developed in–house at the ERC. This package provides a user friendly graphics interface to the original EAGLE code developed by Thompson and co–workers [15].

Even though standard packages are available for the purpose of generating grids, considerable user interaction is needed for generating usable grids, mainly for the purpose of defining the bounding surfaces, or in other words, for the purpose of geometric definition. For the purpose of generating ocean grids, points were read from the ETOPO5 dataset along the coastal lines of the continents with prescribed resolution. Then cubic splines are used to generate smooth curves that represent the continents. Points along these curves are interconnected by cross lines and a surface grid is generated and projected onto the ETOPO5 dataset. This surface grid represents the ocean bottom surface. Once this surface grid is obtained it is projected radially outwards onto a sphere of radius one to generate the ocean free surface. Once these surfaces are generated all the intermediate surfaces are generated with a chosen point distribution in the radial direction. Thus, a volume grid is obtained by joining all the surface grids in the radial direction.

While the process of generating nonorthogonal curvilinear grids for the oceans was fairly straight forward, the process of obtaining numerical solutions over those grids evoked a series of fundamental questions: (1) What form of the governing equations is to be used, (2) What level of approximations is to be introduced in those equations, (3) What amount of grid resolution is appropriate for resolving the turbulent flows if one were to choose the viscosity of water as the reference value, (4) What are the appropriate boundary conditions to be used, etc. Thus, in addition to the task of the development of a numerical scheme as originally envisioned, it was felt that some theoretical work also needed to be carried out to answer some of these fundamental questions. In total, there are five new theoretical developments that have taken place in the past three years during the course of this research effort. They are listed below:

(1) Since the time of Coriolis, the Coriolis force has always been expressed as a source term. Recently, Beddhu, Taylor and Whitfield [16] have shown that using a simple tensor identity the Coriolis force can be expressed as the divergence of a tensor, thus, providing a fully conservative form of the momentum equation in a rotating frame. This opens up new possibilities of building alternate numerical approaches to solving the governing equations in a rotating frame. From a numerical point of view, one of the advantages of expressing the Coriolis term in a conservative form, in other words, in a divergence form, is that it fits in a natural manner in a finite volume scheme. Note that fluxes are evaluated at the cell faces in such a scheme whereas source terms need to be evaluated at the cell centers. A more important advantage is that in a higher order numerical approximation of the fluxes the Coriolis term naturally enters the flux Jacobian matrix, as can be seen from Section 5.2, and a higher order representation for the Coriolis term is thus possible.

(2) In order to account for free surface flows in a rotating frame the usual approach adopted in the literature is to introduce a time dependent coordinate transformation in the classical mo-

mentum equation for the rotating frame. Beddhu [17] took the approach of transforming the momentum equation directly from the inertial to the rotating and deforming frame, and, then resolving the resulting vector momentum equation with respect to a set of basis vectors fixed in the rotating frame. This approach also leads to the same momentum equation reported by others. This aspect is further discussed in Section 4.2.

(3) The classical momentum equation as presented in the introductory chapters of books dealing with flows in a rotating frame, Greenspan [18] and Pedlosky [7] , for example, assume that the coefficient of viscosity is a constant. Thus, the contribution of rotation to the viscous term in the momentum equation is not included in this formulation. However, when the coefficient of viscosity is a variable in space as in turbulent flows or even in laminar compressible flows with thermal gradients, Beddhu [17] showed that an extra term appears in the momentum equation that includes the contribution of rotation to the viscous term. This work is motivated by the observation that Stokes used the absolute velocity vector in his classical formulation of the Stokes tensor, which accounts for the pressure and viscous forces.

(4) When including the effects of buoyancy in the momentum equation the usual approach is to invoke Boussinesq's approximation in which the density appearing in all the terms of the momentum equation is treated as a constant except for the buoyancy term. Alternate formulation to Boussinesq's hypothesis are beginning to be examined by ocean modelers for various reasons. In this connection, Beddhu et al [19] have introduced an alternate formulation in which the momentum equation is first divided by density throughout. This results in an equation in which the pressure term and the viscous term are multiplied by $1/\rho$. Then $1/\rho$ is replaced by $(1/\rho_0)(1 + \rho')^{-1}$ where $\rho' = \Delta\rho/\rho_0$, ρ_0 is a reference density and $\Delta\rho$ is the change in density from the reference value. This approach leads to an approximation from which Boussinesq's approximation can be recovered as a lower order case. In oceanographic applications, which are the main thrust area of this research, the maximum change in density is

about 6 percent of the standard value, see Bryan and Cox [20]. Assuming that the maximum change in density is about 10 percent and writing ρ as $\rho = \rho_0 + \Delta\rho = \rho_0(1 + \rho')$ it can be easily checked that the error involved in writing $1/\rho \approx (1 - \rho')/\rho_0$, where higher order powers in ρ' are neglected, is less than 1 percent.

(5) Since the coordinate system employed in the present study is a nonorthogonal curvilinear coordinate system, the viscous stress boundary condition at the ocean surface has to be applied with respect to such a coordinate system. However, the viscous free surface boundary conditions are a complicated set of equations in a general curvilinear coordinate system. On the other hand, in an orthonormal coordinate system, these equations take the simplest form. This fact is taken advantage of by introducing a local orthonormal coordinate system at every grid point on the free surface in Beddhu and Whitfield [21]. The unit tangent vector (say, $\underline{t}$) to one of the coordinate lines lying on the free surface at the point of interest, the local unit normal (say, $\underline{n}$) to the free surface at the point of interest and the vector $\underline{t} \times \underline{n}$ constitute the orthonormal system. Velocity components and the derivatives in this orthonormal system are expressed in terms of the velocity components and the derivatives in the curvilinear coordinate system using transformation relations which results in a set of matrix equations that are solved to update the free surface velocity components as well as pressure.

As discussed in detail in Beddhu, Taylor and Whitfield [16], the momentum equation for flows in a rotating frame can be formulated in various ways. It can either be formulated with respect to an observer who is stationary in an inertial frame or with respect to an observer who is stationary with respect to the rotating frame. It can either be formulated with the absolute velocity components or the relative velocity components. Irrespective of which form one chooses, the governing equations need to be written in terms of a general curvilinear coordinates and finally expressed in the so called numerical vector form in order to be discretized. For the implicit scheme, one needs the flux Jacobians and the eigensystem of the flux Jaco-

bians. In Beddhu et al [19] the observer was positioned in the inertial frame and the governing equations were written using the absolute velocity components. The eigensystem for the complete system of six equations viz., the continuity equation, the three momentum equations, the temperature equation and the salinity equation, was presented. However, one has to subtract the term $\underline{\Omega} \times (\underline{\Omega} \times \underline{r})$, as discussed in [16], from the momentum equation presented in [19] in order to make it applicable to ocean flows. Since the term $\underline{\Omega} \times (\underline{\Omega} \times \underline{r})$ does not depend on any flow variable, it does not alter the eigensystem. In Beddhu, Taylor and Whitfield [16] two formulations of the momentum equation were presented with respect to an observer in the rotating frame, one using the absolute velocity vector in the local time derivative term and the other using the relative velocity vector in the local time derivative term where the local time derivative itself is formulated with respect to the rotating frame. The eigensystem for the set of four equations including the continuity equation and the three momentum equations was presented for both the formulations.

As can be seen from the above discussion, considerable theoretical progress has been made to answer some of the questions raised earlier. In terms of code development, three different categories of codes are being developed for various applications. They are as follows: (1) A set of codes for solving the continuity equation and the three momentum equations. This set is the basic set and is called the 4x4 set corresponding to the dimension of the flux Jacobians. (2) The second set includes the temperature equation in addition to the continuity and momentum equations and is called the 5x5 set. (3) The complete set of six equations is included in the third set and is called the 6x6 set. In each set of codes various formulations are included which are determined by the position of the observer and the velocity components used, i.e., relative or absolute. Depending on the particular application in hand, a particular code can be chosen that will result in minimal memory usage and CPU usage. This approach is needed since the problems of interest to the authors are very large scale problems. However, within each set,

since the eigensystems are remarkably similar for the various formulations, minimal code changes are required to go from one formulation to another. Most of the theoretical developments introduced earlier are already incorporated in the codes and the remaining are being implemented. Among the features that are yet to be implemented is the free surface formulation in the rotating frame as the theory is very recently developed. The generic name for the codes developed in the CFD Lab of the ERC is UNCLE (UNsteady Computation of fieLd Equations). The particular name for the codes developed for ocean applications is UNCLE.O-MAS (UNsteady Computation of fieLd Equations – Old Man And the Sea).

The original incompressible UNCLE code was developed by Taylor [22] as part of his doctoral degree requirement and belongs to the 4x4 category. In this work the observer is positioned in the inertial frame and he is observing the flow taking place in a general non–inertial frame. This allows the governing equations to be expressed in the so called strong conservative form. The artificial compressibility formulation is used to recast the continuity equation as a hyperbolic equation and the entire set of governing equations is expressed in terms of general curvilinear coordinates fixed in the non–inertial frame. The so called partial transformation is used to write the momentum equation in its component form in order to maintain its strong conservative nature in the component form as well. The resulting governing equations are discretized using the Roe scheme [23] for the first order fluxes and the MUSCL scheme of van Leer [24] for the higher order fluxes in a finite volume formulation. The discretized equations are then solved implicitly using Newton's method coupled with symmetric Gauss–Seidel passings in a time accurate manner where the flux Jacobians are calculated using numerical differentiation (see Whitfield [25]). Time accuracy is introduced by multiplying the residual in the Newton's method with a suitable preconditioning matrix as prescribed by Pan and Chakravarthy [26]. The resulting numerical algorithm is called the DNR (Discretized Newton Relaxation) scheme. Further details of this scheme can be found in Whitfield

and Taylor [27]. A detailed exposition of this scheme applicable to two dimensional flows can be found in Whitfield and Taylor [28].

One of the common methods of reducing the problem size when dealing with large scale computational problems is the domain decomposition technique in which a given large physical domain is decomposed into smaller sub–domains and the solution algorithm is applied in each sub–domain as if it is the only domain of interest. These sub–domains are also called blocks. This approach creates artificial block to block boundaries and information needs to be passed across these boundaries at each time step to correctly solve the global problem. General multiblock capability to the original UNCLE solver was added by Arabshahi and is reported in Arabshahi, Taylor and Whitfield [29]. In this approach, blocks of arbitrary sizes can be oriented in any arbitrary manner and an arbitrary number of boundary patches can be used on the block boundaries for specifying various boundary conditions. Even though an arbitrary orientation is allowed, grid lines across the block to block boundaries have to be continuous.

An important technique that has gained widespread popularity for accelerating the convergence of Navier–Stokes solvers is the multigrid method. A nonlinear multigrid method has been added to the original UNCLE solver by Sheng [30] and Sheng, Taylor and Whitfield [31] for steady flows and by Sheng, Taylor and Whitfield [32] for unsteady flows. This approach uses the Galerkin coarse grid approximation for restricting the fine grid matrix operator to the coarse grid and applies the implicit correction smoothing technique when prolonging the corrections from the coarse grid to the fine grid. Beddhu et al [33] have added the combined capability of multiblock and unsteady multigrid approaches to the original UNCLE code developed by Taylor [22] with some modification to the coding of Arabshahi [29] and Sheng et al [32]. For various other applications Arabshahi [34] and Nichols [35] have independently added the same capability to the original UNCLE code.

The strong conservative form of the governing equations in a rotating frame under the 4x4 category has been formulated in two different manners with the absolute velocity components and the relative velocity components respectively, and both have been coded by Beddhu and reported in Beddhu, Taylor and Whitfield [16]. The governing equations of the 5x5 category have been coded using a simple multiblock strategy by Siong [36] following the approach of Beddhu et al [19]. These equations are cast with respect to an observer in the inertial frame and uses the absolute velocity components. Simple multiblock strategy means that the blocks need to be of the same size, arbitrary orientation of individual blocks is not allowed, and, the boundary conditions are applied in a restricted manner. Starting from the original UNCLE code of Taylor [22], Beddhu wrote the code for the governing equations of the 6x6 category positioning the observer in the inertial frame and utilizing the absolute velocity components and this work has been reported in Beddhu et al [19]. This code uses the general multiblock strategy. Efforts are underway to include the multigrid capability in this code. A rotating frame formulation of this category has also been coded and is currently being used to compute the flow in the Atlantic ocean.

As mentioned earlier the implementation of the free surface capability in a rotating frame is underway using the equations proposed in [17]. However, Beddhu, Taylor and Whitfield [37] have introduced the modified artificial compressibility method for solving the free surface flows in an inertial frame using deforming grids. This method is capable of predicting un-steady free surface flows as demonstrated in [37] and [38].

Various strands of the UNCLE code have been successfully tested against various cases ranging from text book examples to real world engineering applications and are too numerous to list here individually. The basic methodology has proved to be sound and the code is robust. Interested readers can refer to the references cited above for applications. However, a set of specific cases were designed to demonstrate the capability of the UNCLE solver towards the

goal of applying a Navier–Stokes solver for computing oceanographic flows. These are discussed in the results section.

The rest of this report is organized as follows: In Section 4 the governing equations are presented. In Section 5, the numerical algorithm is presented. Section 6 deals with the grid generation strategy employed for generating ocean grids. In Section 7, the results of chosen test cases are presented which clearly demonstrate the capabilities of the UNCLE solver. Finally, in Section 8, conclusions are drawn and a possible course of future work is suggested as this is an ongoing effort.

4. Governing Equations

This section is further sub–divided into five subsections. In Section 4.1 strong conservative formulation of the momentum equation for geophysical applications is presented which is followed by a discussion on the formulation of the momentum equation for free surface flows in the presence of rotation in Section 4.2. In Section 4.3, the derivation of the contribution of rotation to the viscous term in the momentum equation is presented. In Section 4.4, the complete set of the governing equations for oceanographic applications is presented with the proposed modified Boussinesq's approximation. In Section 4.5, the formulation of the viscous stress boundary condition is presented for a general curvilinear coordinate system as outlined earlier. The validity of the hydrostatic approximation in a general curvilinear coordinate system is examined in Section 4.6.

4.1. Strong Conservative Formulation of the Momentum Equation in a Rotating Frame.

The momentum equations governing the (oceanic) flows over earth, which is a self–rotating gravitational body, in non–dimensional tensor invariant form is given by (see for example, Gill [39])

$$\frac{\hat{\partial}\underline{v}}{\partial\tau} + \nabla \cdot \left[\underline{v}\,\underline{v} + p\tilde{I} - \frac{1}{Re_0}\tilde{\sigma}\right] + 2\,\underline{\Omega} \times \underline{v} + \underline{b} = 0 \qquad (4.1.1)$$

where $\underline{v} = \underline{v}^*/U_0$, is the non–dimensional velocity vector with respect to the rotating frame, $\tau = tU_0/L$, is the non–dimensional time, $p = (p^* - p_0)/\rho_0 U_0^2$, is the non–dimensional pressure, $\underline{\Omega}$, is the angular velocity of the rotating frame, $\tilde{\sigma}$, is the Stokes tensor and $\underline{b}$, is the body force. Re_0 is the Reynolds number, $Re_0 = \rho_0 U_0 L / \mu_0$, where, ρ_0 is a reference density, U_0 is a reference velocity, L is a reference length, and, μ_0 is a reference coefficient of viscosity. A

tilde over a quantity denotes that it is a tensor and an underscore denotes that it is a vector. The Stokes tensor is given by

$$\tilde{\sigma} = \mu (\nabla \underline{v} + \nabla^T \underline{v}) \tag{4.1.2}$$

where, $\mu = \mu^*/\mu_0$, is the non–dimensional coefficient of viscosity. The superscript 'T' in Eq. (4.1.2) denotes the transpose operation. The only body force considered is that due to gravity and is given by $\underline{b} = \underline{n}/Fr^2$, where, Fr is the Froude number given by $Fr = U_0/\sqrt{aL}$, where, a is the acceleration due to gravity and $\underline{n}$ is the local normal to the earth's surface. In Eq. (4.1.1), $\frac{\hat{\partial}}{\partial \tau}$ denotes local time derivative with respect to the rotating frame. In other words if $\underline{i}_m$; m = 1, 2, 3 are the Cartesian base vectors in the rotating frame then, by definition,

$$\frac{\hat{\partial} \underline{v}}{\partial \tau} = \frac{\hat{\partial} (v_m \underline{i}_m)}{\partial \tau} = \frac{\hat{\partial} v_m}{\partial \tau} \underline{i}_m \tag{4.1.3}$$

It can be easily verified that

$$\underline{\Omega} \times \underline{v} = \underline{v} \cdot \nabla (\underline{\Omega} \times \underline{r}) \tag{4.1.4}$$

where, $\underline{r}$ is the radius vector from the origin of the rotating frame. Using the following tensor identity for any two vectors $\underline{a}$ and $\underline{b}$, (see, Morse and Feshbach [40])

$$\nabla \cdot (\underline{a}\, \underline{b}) = \underline{a} \cdot (\nabla \underline{b}) + \underline{b} (\nabla \cdot \underline{a}) \tag{4.1.5}$$

and the fact that $\nabla \cdot \underline{v} = 0$, which follows from the conservation of mass, it is seen that

$$\underline{\Omega} \times \underline{v} = - \nabla \cdot (\underline{v}\, \underline{w}) \tag{4.1.6}$$

where, $\underline{w} = - \underline{\Omega} \times \underline{r}$. Therefore, Eq. (4.1.1) can be written as

$$\frac{\hat{\partial} \underline{v}}{\partial \tau} + \nabla \cdot [\underline{v} (\underline{v} - 2 \underline{w}) + p' \tilde{I} - \frac{1}{Re_0} \tilde{\sigma}] = 0 \tag{4.1.7}$$

Note that the body force has been combined with the pressure by the use of the body force potential in the manner prescribed by Beddhu et al [37], that is, $p' = p + \chi/Fr^2$, where χ is

the body force potential due to gravity. Equation (4.1.7) is the strong conservative formulation of the Navier–Stokes equation for incompressible flows in a rotating frame. To the authors' best knowledge this is the first time the Navier–Stokes equations have been presented in a rotating frame without both source terms. Note that, such a formulation is not possible for compressible flows since $\nabla \cdot \underline{v} \neq 0$. The continuity equation in the modified artificial compressibility method [37] is given by

$$\frac{\hat{\partial} p'}{\partial \tau} + \beta \nabla \cdot \underline{v} = 0 \tag{4.1.8}$$

where β is the artificial compressibility parameter.

Even though Eqs. (4.1.8) and (4.1.7) form a complete set of governing equations for solving the oceanic flow problems, further insight into the alternative formulations of the momentum equation appropriate for solving geophysical flow problems, can be gained by looking at an alternate derivation, starting from the governing equations with respect to an arbitrary non–inertial frame. The momentum equation for viscous, incompressible flows in a non–inertial frame of reference, in a gravitational field, in non–dimensional, vector invariant form is given by (see, for example, Warsi [41])

$$\frac{1}{\sqrt{g}} \frac{\partial (\sqrt{g}\, \underline{u})}{\partial \tau} + \nabla \cdot \left[\underline{v}\, \underline{u} + p \tilde{I} - \frac{1}{Re_0} \tilde{\sigma} \right] + \underline{b} = 0 \tag{4.1.9}$$

where $\sqrt{g}$ is the Jacobian of the coordinate transformation, $\underline{u} = \underline{u}^* / U_0$ is the non–dimensional velocity vector in the absolute frame, $\underline{v} = \underline{u} + \underline{w}$ is the non–dimensional velocity vector relative to the moving frame, $\underline{w}$ is the non–dimensional grid speed vector, and other quantities are as defined previously. The Stokes tensor is given by

$$\tilde{\sigma} = \mu \left(\nabla \underline{u} + \nabla^T \underline{u} \right) \tag{4.1.10}$$

It must be noted here that Warsi [41] follows the linear transformation representation (see, Truesdell and Noll [42]) for representing tensors whereas this work has adopted the dyadic

product representation (see, Morse and Feshbach [40]) for representing tensors. Hence, the equations found in Ref. [41] are suitably modified to fit the representation adopted in this work.

Since a rotating frame is a particular case of non–inertial frames for which Eq. (4.1.9) is applicable, it must be possible to derive the momentum equation in a rotating frame from Eq. (4.1.9). However, the concept of grid speed is not valid with respect to an observer situated in the rotating frame since the grid does not move with respect to him/her. Following Warsi [41], instead of considering $\underline{w}$ as the grid speed, one poses the question of what form of $\underline{w}$ in Eq. (4.1.9) would result in the Navier–Stokes equations in a rotating frame. It is an exercise problem in Ref. [41] to show that substituting $\underline{w} = -\underline{\Omega} \times \underline{r}$, where, $\underline{\Omega}$ is the angular velocity of the rotating frame and $\underline{r}$ is the distance from the origin of the rotating frame in Eq. (4.1.9), results in the classical rotating frame equation in a gravitational field, i.e., the centrifugal force term $\underline{\Omega} \times (\underline{\Omega} \times \underline{r})$ has to be added to the left hand side of Eq. (4.1.1). Hence, in order to arrive at Eq. (4.1.1) this term has to be subtracted from Eq. (4.1.9) to obtain

$$\frac{1}{\sqrt{g}} \frac{\partial(\sqrt{g}\,\underline{u})}{\partial \tau} + \nabla \cdot \left[\underline{v}\,\underline{u} + p\tilde{I} - \frac{1}{Re_0}\tilde{\sigma} \right] + \underline{b} - \underline{\Omega} \times (\underline{\Omega} \times \underline{r}) = 0 \qquad (4.1.11)$$

It is emphasized that now Eq. (4.1.11) is applicable only to self–gravitating, rotating bodies like the earth. The vector momentum equation, Eq. (4.1.11), can be resolved into components with respect to a set of basis vectors which can either be fixed in the inertial frame or in the non–inertial frame. This aspect is further discussed in Section 4.2. The basis vectors are fixed in the rotating frame for the following discussion. Thus, for the case of a rotating frame with a constant angular velocity $\underline{\Omega}$ using the relations given in Section 3.10B of Ref. [41], it can be proved easily that

$$\frac{1}{\sqrt{g}} \frac{\partial(\sqrt{g}\,\underline{u})}{\partial \tau} = \frac{\hat{\partial}\underline{u}}{\partial \tau} + \underline{\Omega} \times \underline{u} \qquad (4.1.12)$$

where, $\frac{\hat{\partial}}{\partial \tau}$, denotes local time derivative with respect to the rotating frame (See Eq. (4.1.3)). Equation (4.1.11) can now be re–written, using Eq. (4.1.12), as follows:

$$\frac{\hat{\partial}\underline{\underline{u}}}{\partial \tau} + \nabla \cdot \left[\underline{v}\,\underline{u} + p'\tilde{I} - \frac{1}{Re_0}\tilde{\sigma} \right] + \underline{\Omega} \times \underline{v} = 0 \tag{4.1.13}$$

where, $\underline{v} = \underline{u} + \underline{w} = \underline{u} - \underline{\Omega} \times \underline{r}$ is the velocity with respect to the rotating frame. Note that the body force has been combined with the pressure by the use of the body force potential as before. Substitution of Eq. (4.1.6) in Eq. (4.1.13) results in,

$$\frac{\hat{\partial}\underline{\underline{u}}}{\partial \tau} + \nabla \cdot \left[\underline{v}\,(\underline{u} - \underline{w}) + p'\tilde{I} - \frac{1}{Re_0}\tilde{\sigma} \right] = 0 \tag{4.1.14}$$

Equation (4.1.14) is an alternate strong conservative form of the Navier–Stokes equations in a rotating frame, applicable for a self–gravitating, rotating body like the earth. Note that Eq. (4.1.7) can be recovered from Eq. (4.1.14) by substituting, $\underline{u} = \underline{v} - \underline{w} = \underline{v} + \underline{\Omega} \times \underline{r}$. The main difference between Eqs. (4.1.7) and (4.1.14) is that in a time marching approach, one would solve for the relative velocity components using Eq. (4.1.7), whereas one would solve for the absolute velocity components using Eq. (4.1.14). The continuity equation in the modified artificial compressibility method [37] is given by

$$\frac{\hat{\partial}p'}{\partial \tau} + \beta \, \mathrm{div}\, \underline{u} = 0 \tag{4.1.15}$$

where β is the artificial compressibility parameter.

The fully conservative formulation of the momentum equation is given in the compact vector and tensor notations, thus far. However, in order to solve the equations, numerically or otherwise, one has to write the momentum equation in its component forms. When resolving the momentum equation into component forms one is presented with many choices. These choices arise due to the fact that the vector and tensor quantities can be expressed with respect to any set of coordinates independent of the coordinates one chooses to express the divergence

operator itself. Traditionally, however, the set of coordinates chosen for resolving the vector and tensor quantities is the same as the one chosen for expressing the divergence operator. Thus, Cartesian velocity components are chosen when the divergence operator is expressed in Cartesian coordinates, cylindrical components are chosen when the divergence operator is expressed with respect to cylindrical coordinates and so on. The problem, for example, with choosing cylindrical components of the vector and tensor quantities when expressing the divergence operator in cylindrical coordinates is that Christoffel symbols appear explicitly, thereby preventing the conservative formulation in the component form. Therefore, if one wants to come up with a conservative formulation in the component form also then there is only one choice. That, is to express the vector and tensor quantities in Cartesian components, no matter what coordinates are chosen to express the divergence operator. Since, nonorthogonal curvilinear coordinates are the most general coordinates, the divergence operator is expressed with respect to such a coordinate system in this report. The resulting equations are given in Section 5.1 (see also Section 4.6). The code UNCLE.OMAS (UNsteady Computation of fieLd Equations – Old Man And the Sea) is written to solve Eqs. (5.1.1). Thus, suppose one is interested in the flow over a sphere. Then one can construct a grid based on spherical coordinates and the appropriate metrics will automatically be computed. However, the real advantage of this approach is that one does not have to create a grid based on spherical coordinates. As long as the body shape is maintained spherical, any set of coordinate lines can be created, analytically or numerically, and the same code can be used to solve the flow field.

Even though Eq. (4.1.9) (after expressing the body force in terms of the body force potential) and Eq. (4.1.14) are in fully conservative form and a time marching scheme in both cases would solve for the absolute velocity components, the important difference between them is the position of the observer. While in the case of Eq. (4.1.9) the observer is situated in the inertial frame, he/she is situated in the rotating frame in the case of Eq. (4.1.14). Thus, the grid

remains stationary in the case of Eq. (4.1.14) whereas the grid has to be moved and all the metrics need to be recomputed at each time step in the case of Eq. (4.1.9). Steady flows in the rotating frame can be computed using time inaccurate schemes using Eq. (4.1.14) whereas they require time accurate computation of Eq. (4.1.9).

A time marching upwind scheme for the set of equations (4.1.14) and (4.1.15) would typically solve for the pressure and the Cartesian components of the absolute velocity vector. Either one can solve the set of equations (4.1.14) and (4.1.15), or, the set (4.1.7) and (4.1.8), by the numerical method presented later in this report. For both sets remarkably similar sets of eigensystems are derived. These eigensystems again differ from that derived by Taylor [22] for Eq. (4.1.9) only slightly which results in minimum code modifications.

The solution procedure for the set of equations (4.1.14) and (4.1.15) is called Absolute–Velocity Procedure and that for the set of equations (4.1.7) and (4.1.8) is called Relative–Velocity Procedure. An important element in the present formulation is the construction of the inviscid fluxes at the cell interfaces. The theory behind the construction of the inviscid fluxes has been well established by Roe [23], van Leer [24] and others. The tools needed for constructing the inviscid fluxes are provided in Section 5.2.

The strong conservative form of the momentum equation in a rotating frame for non–geophysical applications, turbomachinery problems for example, is presented in Beddhu, Taylor and Whitfield [16].

4.2. The momentum equation for incompressible free surface flows in the presence of rotation.

Actual ocean flows involve a dynamically evolving free surface. The effect of a free surface has been neglected altogether in early studies of the ocean flows. However, in recent times free surface computations are included at least for numerical reasons, if not for physical rea-

sons. It should be emphasized that proper inclusion of a free surface in a rotating frame complicates the governing equations further by bringing in the effects of deformation in addition to rotation. Besides, one has to deal with the non–linearity of the free surface kinematic equation in addition to the non–linearity of the Navier–Stokes equations themselves. In this section, a consistent formulation of the governing equations is presented when a dynamically evolving free surface is present in a rotating frame.

In order to compare the end result of this section with the classical form of the momentum equation in a rotating frame, the approach taken in this section is to formulate the momentum equation in a non–conservative form. First, the momentum equation applicable for a general arbitrary non–inertial frame is derived starting from first principles. This equation is then recast in terms of the relative velocity vector, and the grid velocity vector, and the local time derivative is expressed in terms of the relative frame. The relative velocity vector that appears in the resulting equation is described with respect to the rotating and deforming frame and the grid velocity vector includes both the rotational and deformation contributions. In order to cast the governing equations in a familiar form, appropriate variables are introduced. At the end of the section a strong conservative formulation of the momentum equation is presented.

The non–dimensional, non–conservative form of the momentum equation with respect to an inertial frame is given by

$$\frac{\partial \underline{u}}{\partial t} + \underline{u} \cdot \nabla \underline{u} = - \nabla p + \frac{1}{Re_0} \nabla \cdot \tilde{\sigma} + \underline{b} \qquad (4.2.1)$$

where the symbols are defined in the previous section. The deviatoric part of the Stokes tensor, $\tilde{\sigma}$, is given in Eq. (4.1.10).

Now, let x_1, x_2 and x_3 be a set of Cartesian coordinates attached to the inertial frame and consider an arbitrary time dependent coordinate transformation,

$$\left.\begin{aligned} x^i &= x^i(x_1, x_2, x_3, t),\ i = 1,2,3 \\ \tau &= t \end{aligned}\right\} \qquad (4.2.2)$$

where x^i are a set of general curvilinear coordinates attached to the non–inertial frame. It is assumed that the inverse of the transformation in Eq. (4.2.2) exists. In order to transform Eq. (4.2.1) to the non–inertial frame, first the local time derivative in Eq. (4.2.1) which is obtained by keeping x_i , i=1,2,3, fixed, needs to be expressed in terms of the time derivative obtained keeping x^i, i=1,2,3, fixed. This results in,

$$\left.\frac{\partial \underline{u}}{\partial t}\right|_{x_i \text{ fixed}} = \left.\frac{\partial \underline{u}}{\partial \tau}\right|_{x^i \text{ fixed}} + \frac{\partial \underline{u}}{\partial x^i}\frac{\partial x^i}{\partial t} = \left.\frac{\partial \underline{u}}{\partial \tau}\right|_{x^i \text{ fixed}} + \underline{w} \cdot \nabla \underline{u} \qquad (4.2.3)$$

where, the vector $\underline{w}$ is given by, $\underline{w} = \frac{\partial x^i}{\partial t}\,\underline{a}_i$, where, $\frac{\partial x^i}{\partial t}$ is the contravariant components of the grid velocity vector, and, $\underline{a}_i$, are the covariant base vectors of the curvilinear coordinate system. Substituting Eq. (4.2.3) in Eq. (4.2.1), one gets

$$\frac{\partial \underline{u}}{\partial \tau} + \underline{v} \cdot \nabla \underline{u} = -\nabla p + \frac{1}{Re_0}\nabla \cdot \tilde{\sigma} + \underline{b} \qquad (4.2.4)$$

where the local time derivative is evaluated by keeping x^i fixed, and, $\underline{v}$, is the relative velocity as observed from the non–inertial frame given by

$$\underline{v} = \underline{u} + \underline{w} \qquad (4.2.5)$$

Note that the sum in Eq. (4.2.5) is a vector sum and should not be misconstrued to mean that the relative velocity is greater than absolute velocity. It is emphasized here that the transformation implied in Eq. (4.2.2) has nothing to do with choosing a set of base vectors to resolve Eq. (4.2.4) into components. Without specifying the frame (which will be done shortly) let , m = 1, 2, 3, be a set of base vectors. Then the local time derivative in Eq. (4.2.4) can be written as:

$$\frac{\partial \underline{u}}{\partial \tau} = \frac{\partial \left(u^m \underline{a}_m \right)}{\partial \tau} = \frac{\partial u^m}{\partial \tau} \underline{a}_m + u^m \frac{\partial \underline{a}_m}{\partial \tau} \tag{4.2.6}$$

where, u^m, m = 1, 2, 3, are the contravariant components of the absolute velocity vector with respect to the base vectors $\underline{a}_m$. The first term on the right hand side of Eq. (4.2.6) represents the local time derivative with respect to the non–inertial frame and the second term on the right accounts for the rate of change of the base vectors with respect to time. It can be easily checked that if the base vectors are fixed in an inertial frame then $\partial \underline{a}_m / \partial \tau = 0$. Similarly, for the base vectors fixed in the rotating frame one has

$$\frac{\partial \underline{a}_m}{\partial \tau} = \underline{a}_m \cdot \nabla \left(\underline{\Omega} \times \underline{r} \right) = \underline{\Omega} \times \underline{a}_m. \tag{4.2.7}$$

Finally, for the base vectors fixed in a rotating and deforming frame one has (See Warsi [41], Eq. 3.134)

$$\frac{\partial \underline{a}_m}{\partial \tau} = - \frac{\partial \underline{w}}{\partial x^m} \tag{4.2.8}$$

Resolving Eq. (4.2.4) with respect to the base vectors fixed in the rotating frame means that one should use Eqs. (4.2.6) and (4.2.7) in Eq. (4.2.4) which results in

$$\frac{\hat{\partial} \underline{u}}{\partial \tau} + \underline{\Omega} \times \underline{u} + \underline{v} \cdot \nabla \underline{u} = - \nabla p + \frac{1}{Re_0} \nabla \cdot \tilde{\sigma} + \underline{b} \tag{4.2.9}$$

A hat is placed over the local time derivative in Eq. (4.2.9) to indicate that it is evaluated with respect to the relative frame (see Eq. (4.1.3) for the definition). Equation (4.2.9) is the momentum equation for a rotating and deforming frame expressed in terms of the base vectors fixed in a rotating frame alone. In other words, an observer situated in the rotating frame would use Eq. (4.2.9) to predict the flow in a deforming coordinate system. (Note that an observer situated in the rotating frame can only see deformation). In order to cast Eq. (4.2.9) in a familiar form one proceeds as follows. For a rotating and deforming frame the vector $\underline{w}$ can be decomposed into a vector sum of two components, one due to rotation and the other due to deformation. Let, $\underline{w} = \underline{W} + \underline{w}'$, where, $\underline{W} = - \left(\underline{\Omega} \times \underline{r} \right)$, is the part due to rotation and $\underline{w}'$

is the part due to deformation which has to be obtained numerically or otherwise at each instant in time. Then Eq. (4.2.5) becomes

$$\underline{v} = \underline{u} + \underline{w} = \underline{u} + \underline{W} + \underline{w}' = \underline{V} + \underline{w}' \tag{4.2.10}$$

where, by definition, $\underline{V} = \underline{u} - (\underline{\Omega} \times \underline{r})$ is the relative velocity with respect to the rotating frame. Substituting Eq. (4.2.10) in Eq. (4.2.9) and writing the resulting equation in terms of $\underline{V}$ results in

$$\frac{\hat{\partial}\underline{V}}{\partial\tau} + \underline{V} \cdot \nabla \underline{V} + \frac{\hat{\partial}\underline{\Omega}}{\partial\tau} \times \underline{r} + 2\,\underline{\Omega} \times \underline{V} + \underline{\Omega} \times (\underline{\Omega} \times \underline{r}) + \underline{w}' \cdot \nabla \underline{V} = -\nabla p + \frac{1}{Re_0} \nabla \cdot \tilde{\sigma} + \underline{b} \tag{4.2.11}$$

Equation (4.2.11) is the final form of the momentum equation governing a flow in a rotating and deforming frame cast with respect to an observer in the rotating frame and expressed in terms of the relative velocity vector with respect to the rotating frame.

In the traditional approach to deriving the momentum equation in terms of the rotating and deforming coordinates one first transforms to the rotating frame alone from the inertial frame. Thus, the transformation implied in Eq. (4.2.2) is from the inertial to the rotating frame alone and Eq. (4.2.4) becomes

$$\frac{\hat{\partial}\underline{u}}{\partial\tau} + \underline{\Omega} \times \underline{u} + \underline{V} \cdot \nabla \underline{u} = -\nabla p + \frac{1}{Re_0} \nabla \cdot \tilde{\sigma} + \underline{b} \tag{4.2.12}$$

The difference between Eqs. (4.2.12) and (4.2.9) is in the third term on the left hand side. In Eq. (4.2.9) the relative velocity is with respect to the rotating and deforming frame whereas in Eq. (4.2.12) the relative velocity is with respect to the rotating frame alone. Now, let x_1, x_2 and x_3 be a set of Cartesian coordinates attached to the *rotating frame* and consider an arbitrary time dependent coordinate transformation of the type given in Eq. (4.2.2) to account for deformation. The time derivative in Eq. (4.2.12) has to be replaced using Eq. (4.2.3) which results in

$$\hat{\frac{\partial \underline{u}}{\partial \tau}} + \underline{\Omega} \times \underline{u} + (\underline{V} + \underline{w}') \cdot \nabla \underline{u} = -\nabla p + \frac{1}{Re_0} \nabla \cdot \tilde{\sigma} + \underline{b} \tag{4.2.13}$$

where, $\underline{w}'$ is the deformation velocity vector. It can be seen that Eq. (4.2.13) and Eq. (4.2.9) are the same. Thus the traditional approach and the present formulation yield the same momentum equation.

Using the continuity equation and the geometric conservation law [45] in Eq. (4.2.11) yet another form of the momentum equation can be obtained as follows:

$$\hat{\frac{\partial (\sqrt{g}\,\underline{V})}{\partial \tau}} + \nabla \cdot [(\underline{V} + \underline{w}')\,\underline{V}] + \hat{\frac{\partial \underline{\Omega}}{\partial \tau}} \times \underline{r} + 2\,\underline{\Omega} \times \underline{V} + \underline{\Omega} \times (\underline{\Omega} \times \underline{r}) = -\nabla p + \frac{1}{Re_0} \nabla \cdot \tilde{\sigma} + \underline{b} \tag{4.2.14}$$

Next, for the sake of completeness, the momentum equation with respect to an observer situated in the rotating and deforming frame is derived. Substituting Eqs. (4.2.6) and (4.2.8) in Eq. (4.2.4), one obtains

$$\hat{\frac{\partial \underline{u}}{\partial \tau}} - \underline{u} \cdot \nabla \underline{w} + \underline{v} \cdot \nabla \underline{u} = -\nabla p + \frac{1}{Re_0} \nabla \cdot \tilde{\sigma} + \underline{b} \tag{4.2.15}$$

Replacing $\underline{u}$ in terms of $\underline{v}$ and $\underline{w}$, using Eq. (4.2.5), results in

$$\hat{\frac{\partial \underline{v}}{\partial \tau}} + \underline{v} \cdot \nabla \underline{v} - \hat{\frac{\partial \underline{w}}{\partial \tau}} - 2\,\underline{v} \cdot \nabla \underline{w} + \underline{w} \cdot \nabla \underline{w} = -\nabla p + \frac{1}{Re_0} \nabla \cdot \tilde{\sigma} + \underline{b} \tag{4.2.16}$$

Equation (4.2.16) is the final form of the momentum equation governing an incompressible flow in an arbitrary non–inertial frame, expressed in terms of the relative velocity, and is valid for an observer situated in the arbitrary non–inertial frame.

Substituting Eq. (4.2.10) in Eq. (4.2.16), and splitting $\underline{w}$ as mentioned above, one obtains

$$\frac{\hat{\partial}\underline{V}}{\partial\tau} + \underline{V}\cdot\nabla\underline{V} + \frac{\hat{\partial}\underline{\Omega}}{\partial\tau}\times\underline{r} + 2\,\underline{\Omega}\times\underline{V} + \underline{\Omega}\times(\underline{\Omega}\times\underline{r}) + \underline{w}'\cdot\nabla\underline{V}$$

$$\boxed{-\underline{V}\cdot\nabla\underline{w}' - (\underline{\Omega}\times\underline{r})\cdot\nabla\underline{w}'} = -\nabla p + \frac{1}{Re_0}\nabla\cdot\tilde{\sigma} + \underline{b} \qquad (4.2.17)$$

Notice that two additional terms appear on the left had side of Eq. (4.2.17) when compared with the left hand side of Eq. (4.2.11).

In Eqs. (4.2.11) and (4.2.17) the centrifugal force term, $\underline{\Omega}\times(\underline{\Omega}\times\underline{r})$, appears explicitly. Thus, these equations are valid for flows in a rotating frame in an external gravity field, turbomachinery flows for example. For the case of bodies such as the earth which are self–rotating and self–gravitating, the centrifugal force is implicitly accounted for in the definition of acceleration due to gravity. Hence, for geophysical applications the centrifugal force needs to be deleted from Eqs. (4.2.11) and (4.2.17) which leads to the following two equations respectively

$$\frac{\hat{\partial}\underline{V}}{\partial\tau} + \underline{V}\cdot\nabla\underline{V} + 2\,\underline{\Omega}\times\underline{V} + \underline{w}'\cdot\nabla\underline{V} = -\nabla p + \frac{1}{Re_0}\nabla\cdot\tilde{\sigma} + \underline{b} \qquad (4.2.18)$$

$$\frac{\hat{\partial}\underline{V}}{\partial\tau} + \underline{V}\cdot\nabla\underline{V} + 2\,\underline{\Omega}\times\underline{V} + \underline{w}'\cdot\nabla\underline{V}$$

$$\boxed{-\underline{V}\cdot\nabla\underline{w}' - (\underline{\Omega}\times\underline{r})\cdot\nabla\underline{w}'} = -\nabla p + \frac{1}{Re_0}\nabla\cdot\tilde{\sigma} + \underline{b} \qquad (4.2.19)$$

Note that in Eqs. (4.2.18) and (4.2.19) $\underline{\Omega}$ is assumed to be independent of time as well.

Following the procedure outlined in the Section 4.1, it is possible to cast the momentum equation for flows in a rotating and deforming frame in a strong conservative formulation. However, this formulation needs to be in terms of the absolute velocity vector appearing in the local time derivative. Thus, for an observer situated in the rotating frame one can obtain from Eq. (4.2.9)

$$\frac{1}{\sqrt{g}}\frac{\hat{\partial}(\sqrt{g}\,\underline{u})}{\partial\tau} + \nabla\cdot\left[\underline{v}\,\underline{u} - \underline{u}\,\underline{W} - \underline{W}\,\underline{W} + p'\tilde{I} - \frac{1}{Re_0}\tilde{\sigma}\right] = 0 \qquad (4.2.20)$$

and for an observer situated in the rotating and deforming frame one can obtain from Eq. (4.2.15)

$$\frac{1}{\sqrt{g}}\frac{\hat{\partial}(\sqrt{g}\,\underline{u})}{\partial\tau} + \nabla\cdot\left[\underline{v}\,\underline{u} - \underline{u}\,\underline{w} - \underline{W}\,\underline{W} + p'\tilde{I} - \frac{1}{Re_0}\tilde{\sigma}\right] = 0 \qquad (4.2.21)$$

where, $\sqrt{g}$ is the Jacobian of the coordinate transformation from the inertial frame to the non–inertial frame introduced in Eq. (4.2.2). The difference between Eqs. (4.2.20) and (4.2.21) is in the second term within the divergence term.

4.3. Effect of rotation on flows with a spatially varying viscosity field.

The discussion in this section is valid for both geophysical and non–geophysical flows. For non–geophysical flows one needs to add the centrifugal force term explicitly to the left hand side of Eq. (4.1.1). From a historical perspective, the classical momentum equation, Eq. (4.1.1), in a rotating frame was first written to be applicable for laminar flows. Thus, the coefficient of viscosity was treated as a constant and the viscous stress tensor was assumed to be given by Eq. (4.1.2). On the other hand, according to Stokes hypothesis, whether the frame is inertial or non–inertial (rotating frame, for example) the viscous stress tensor is given only by Eq. (4.1.10). However, when the coefficient of viscosity is a constant, the divergence of the viscous stress tensor as given by Eq. (4.1.2) turns out to be the same as given by Eq. (4.1.10). Since, only the divergence of the stress tensor appears in the momentum equation and the divergence of Eqs. (4.1.2) and (4.1.10) is the same, text books have been written stating that the viscous stress tensor in a rotating frame is given by Eq. (4.1.2). However, it is stressed here that even for a laminar flow in a rotating frame the viscous stress tensor is still given by Eq. (4.1.10). This understanding is essential for properly accounting for turbulence in a rotating

frame. Since only the divergence of Eq. (4.1.10) enters the momentum equation one can, however, compute the stress tensor using Eq. (4.1.2) for laminar flows.

When the understanding that many practical flows are turbulent in nature developed, attempts were made to cast the momentum equation in a rotating frame valid for turbulent flows. In all these attempts the standard Reynolds averaging procedure was introduced in Eq. (4.1.1). This led to a turbulent stress tensor which is modeled using Eq. (4.1.2) albeit with a spatially varying viscosity field. However, the correct approach is to use Eq. (4.1.10) for modeling the Reynolds stress tensor. Alternately, one can first introduce the Reynolds averaging procedure in the momentum equation for an inertial frame in which there is no ambiguity about modeling the Reynolds stress tensor and then transform the resulting equation to a rotating frame (or a rotating and deforming frame) as outlined in this report. Thus, one would again have the Reynolds stress tensor modeled by Eq. (4.1.10). When this approach is adopted, it is shown in the following that rotation has an effect on the viscous term as well.

The velocity vector with respect to the absolute frame, $\underline{u}$, can be written as, using Eq. (4.2.5)

$$\underline{u} = \underline{v} - \underline{w} = \underline{V} + \underline{w}' - \underline{W} - \underline{w}' = \underline{V} - \underline{W} \tag{4.3.1}$$

Hence, the Stokes tensor, $\tilde{\sigma}$, given by Eq. (4.1.10) becomes

$$\tilde{\sigma} = \mu \left[\nabla (\underline{V} - \underline{W}) + \nabla^T (\underline{V} - \underline{W}) \right] \tag{4.3.2}$$

For the purpose of the following discussion, $\tilde{\sigma}$ is decomposed into two parts, $\tilde{\sigma}'$ and $\tilde{\sigma}''$, as follows:

$$\left.\begin{aligned} \tilde{\sigma}' &= \mu \left[\nabla \underline{V} + \nabla^T \underline{V} \right] \\ \tilde{\sigma}'' &= - \mu \left[\nabla \underline{W} + \nabla^T \underline{W} \right] \end{aligned}\right\} \tag{4.3.3}$$

Thus, the viscous term in Eq. (4.2.17) is given by

$$\nabla \cdot \tilde{\sigma} = \nabla \cdot \tilde{\sigma}' + \nabla \cdot \tilde{\sigma}'' \tag{4.3.4}$$

It can easily be shown that when the (non–dimensional) coefficient of viscosity μ is constant, then $\nabla \cdot \tilde{\sigma}'' = 0$. Thus, in the introductory chapters of [7] or [18], for example, the viscous term $\nabla \cdot \tilde{\sigma}$ in the classical momentum equation for a rotating frame, Eq. (4.2.14) with $\underline{w}' = 0$, is replaced by $\nabla \cdot \tilde{\sigma}'$, assuming the coefficient of viscosity μ to be a constant. However, when μ is *not* a constant it is inappropriate to replace $\nabla \cdot \tilde{\sigma}$ by $\nabla \cdot \tilde{\sigma}'$ in Eq. (4.2.14) or in Eq. (4.2.17), since $\nabla \cdot \tilde{\sigma}'' \neq 0$. In fact,

$$\nabla \cdot \tilde{\sigma}'' = 2\ \underline{\Omega} \times \nabla\mu \tag{4.3.5}$$

Thus, when one is dealing with fluid flows in a rotating frame which result in a spatially varying viscosity field, such as turbulent flows, one can not neglect the term given by Eq. (4.3.5) without justification. Note that this term arises due to rotation and is not influenced by deformation. Since the formulation in Eq. (4.1.10) is valid for compressible flows, so is Eq. (4.3.5). Laminar compressible flows in a rotating frame, with or without free surfaces, with strong thermal gradients are other applications for which the viscous term formulation is appropriate.

4.4. Governing equations of the ocean flows with the modified Boussinesq's approximation.

In this section the complete set of governing equations for thermohaline ocean flows are given with respect to an observer at rest in an inertial frame. However, the observer is observing the flow taking place in a rotating frame. The momentum equation is cast using the absolute velocity vector and the local time derivative is expressed with respect to the absolute frame. Note that by appropriate choice of the vector $\underline{w}$ one can either consider a rotating frame alone ($\underline{w} = \underline{W}$) or a rotating and deforming frame ($\underline{w} = \underline{W} + \underline{w}'$). These symbols are defined in Section 4.1. To obtain the modified Boussinesq's approximation, the momentum equation is first divided by density throughout. This results in an equation in which the pres-

sure term and the viscous term are multiplied by $1/\rho$. Then $1/\rho$ is replaced by $(1/\rho_0)(1 + \rho')^{-1}$ where $\rho' = \Delta\rho/\rho_0$, ρ_0 is a reference density and $\Delta\rho$ is the change in density from the reference value. In oceanographic applications, which are the main thrust area of this research, the maximum change in density is about 6 percent of the standard value, see Bryan and Cox [20]. Assuming that the maximum change in density is about 10 percent and writing ρ as $\rho = \rho_0 + \Delta\rho = \rho_0(1 + \rho')$ it can be easily checked that the error involved in writing $1/\rho \approx (1 - \rho')/\rho_0$, where higher order powers in ρ' are neglected, is less than 1 percent. The resulting governing equations are

Continuity

$$\nabla \cdot \underline{u} = 0 \tag{4.4.1}$$

Momentum

$$\frac{1}{\sqrt{g}}\frac{\partial(\sqrt{g}\,\underline{u})}{\partial\tau} + \nabla \cdot [\,\underline{v}\,\underline{u} + p'\tilde{I} - \frac{1}{Re_0}\tilde{\sigma} - \underline{W}\,\underline{W}\,] =$$

$$\rho'[\,-\nabla \cdot (\frac{\chi}{Fr^2}\tilde{I}) + \nabla \cdot (p'\tilde{I}) - \frac{1}{Re_0}\nabla \cdot \tilde{\sigma}\,] \tag{4.4.2}$$

Temperature

$$\frac{1}{\sqrt{g}}\frac{\partial(\sqrt{g}\,T)}{\partial\tau} + \nabla \cdot (\underline{v}T) = \nabla \cdot (\frac{\kappa}{Pe}\nabla T) \tag{4.4.3}$$

Salinity

$$\frac{1}{\sqrt{g}}\frac{\partial(\sqrt{g}\,S)}{\partial\tau} + \nabla \cdot (\underline{v}S) = \nabla \cdot (\alpha_S \nabla S) \tag{4.4.4}$$

where $T = T^*/T_0$ is the non–dimensional temperature and $S = S^*/S_0$ is the non–dimensional salinity. T_0 and S_0 are the reference values of temperature and salinity respectively. The new non–dimensional parameters are Prandtl number, $Pr = \mu_0 c_p/\kappa_T$, Peclet number, $Pe = Re_0 Pr$ and the Schmidt number $\alpha_S = \kappa_S/U_0 L$ where κ_T is the coefficient of thermal

conductivity and κ_S is the coefficient of saline diffusivity. All other symbols are defined in the previous sections. The last two terms on the right hand side of Eq. (4.4.2) are the proposed modification to the Boussinesq's approximation. Note that computation of these terms do not involve any extra overhead since all the terms need to be computed anyway for the left hand side. Terms involving ρ' are not included in the temperature and salinity equations since these terms are assumed to be of the same order of magnitude as the other terms that have been neglected from these equations. An important difference between the present set of equations and the set obtained using Boussinesq's approximation [43] is in the treatment of the momentum equations. Boussinesq's approximation proceeds by assuming the density to be constant in every term in the original momentum equation except the body force term, whereas no such assumption is made in obtaining Eq (4.4.2).

4.5. Viscous free surface boundary conditions in general curvilinear coordinates.

The dynamic boundary condition at an interface is that the stress vector be continuous across the interface neglecting the effects of surface tension. These conditions were first introduced by Hirt and Shannon [44]. These conditions are important when a wind shear is imposed on the free surface. As mentioned earlier, the coordinate system used in this work is a general curvilinear coordinate system in which one of the coordinate surfaces always coincides with the evolving free surface. The viscous free surface boundary conditions become quite complicated under these circumstances and it is not quite obvious how to solve them. It is the objective of this section to present a useful form of the viscous free surface boundary conditions in such a general setting and to deduce systematically from them a set of diagonally dominant matrix equations that are easy to solve.

The dimensional viscous boundary condition at the free surface is given by

$$\underline{n} \cdot \tilde{T}^* = \underline{\tau}_0^* \tag{4.5.1}$$

+where, $\tilde{T}^* = -p^* \tilde{I} + \mu^* (\nabla \underline{u}^* + \nabla^T \underline{u}^*)$ is the dimensional Stokes tensor, $\underline{n}$ is the local normal to the free surface and $\underline{\tau}_0^*$ is the dimensional applied wind (shear) stress vector. Non–dimensionalizing the other variables an in Section 4.1 and $\underline{\tau}_0^*$ as $\underline{\tau}_0 = \frac{L\underline{\tau}_0^*}{\mu_o U_o}$, Eq. (4.5.1) becomes

$$\underline{n} \cdot [-p \tilde{I} + \frac{\mu}{Re_o} (\nabla \underline{u} + \nabla^T \underline{u})] = \frac{\underline{\tau}_o}{Re_o} + p_o \underline{n} \tag{4.5.2}$$

Now, introduce a Cartesian coordinate system $\bar{x}_1, \bar{x}_2$ and $\bar{x}_3$ locally, with unit vectors $(\bar{i}_1, \bar{i}_2, \bar{i}_3)$ and velocity components $(\bar{u}_1, \bar{u}_2, \bar{u}_3)$. Since Eq. (4.5.2) is a vector equation, resolving it into three components in the above coordinate system, taking $\bar{i}_2 = \underline{n}$, yields

$$\mu \left[\frac{\partial \bar{u}_2}{\partial \bar{x}_1} + \frac{\partial \bar{u}_1}{\partial \bar{x}_2} \right] = \bar{\tau}_{o1} \tag{4.5.3}$$

$$-p + \frac{2\mu}{Re_o} \frac{\partial \bar{u}_2}{\partial \bar{x}_2} = \frac{\bar{\tau}_{o2}}{Re_o} + p_o \tag{4.5.4}$$

$$\mu \left[\frac{\partial \bar{u}_2}{\partial \bar{x}_3} + \frac{\partial \bar{u}_3}{\partial \bar{x}_2} \right] = \bar{\tau}_{o3} \tag{4.5.5}$$

The continuity equation in this system becomes

$$\frac{\partial \bar{u}_1}{\partial \bar{x}_1} + \frac{\partial \bar{u}_2}{\partial \bar{x}_2} + \frac{\partial \bar{u}_3}{\partial \bar{x}_3} = 0 \tag{4.5.6}$$

Note that such a local Cartesian coordinate can be introduced in the following manner. Let $\bar{i}_1 = \frac{\underline{a}_1}{|\underline{a}_1|}$, $\bar{i}_2 = \frac{\underline{a}^2}{|\underline{a}^2|}$ and $\bar{i}_3 = \bar{i}_1 \times \bar{i}_2$, where, $\underline{a}_1$ is the covariant base vector $\frac{\partial \underline{r}}{\partial \xi}$, and $\underline{a}^2$ is the contravariant base vector grad η for the given curvilinear coordinate system. Here it is assumed that the free surface is represented by a η = constant surface. It is quite easy to express the

various partial derivatives and the velocity components in Eqs. (4.5.3) – (4.5.6) in terms of the quantities in the given (or global) Cartesian and curvilinear systems. Using the relationships between the derivatives in the Cartesian and curvilinear coordinates Eq. (4.5.3) becomes

$$(\eta_x \beta_{11} + \eta_y \beta_{12} + \eta_z \beta_{13})\, u_\eta + (\eta_x \beta_{21} + \eta_y \beta_{22} + \eta_z \beta_{23})\, v_\eta + (\eta_x \beta_{31} + \eta_y \beta_{32} + \eta_z \beta_{33})\, w_\eta = \frac{\bar{\tau}_{o1}}{\mu} - R_1 \qquad (4.5.7)$$

where R_1 contains the terms involving u_ξ, u_ς, v_ξ, v_ς, w_ξ and w_ς, $\beta_{lk} = \alpha_{1l}\,\alpha_{2k} + \alpha_{2l}\,\alpha_{1k}$ and $\alpha_{ml} = \bar{i}_m \cdot i_l$. Similarly, Eqs. (4.5.4) and (4.5.6) are written as

$$\frac{\partial u_l}{\partial x_k}\,\gamma_{lk} = \frac{\bar{\tau}_{o3}}{\mu} \qquad (4.5.8)$$

$$\frac{\partial u_l}{\partial x_k}\,\epsilon_{lk} = 0 \qquad (4.5.9)$$

where $\gamma_{lk} = \alpha_{2l}\,\alpha_{3k} + \alpha_{3l}\,\alpha_{2k}$ and $\epsilon_{lk} = \alpha_{1l}\,\alpha_{1k} + \alpha_{2l}\,\alpha_{2k} + \alpha_{3l}\,\alpha_{3k}$. Equations (4.5.8) and (4.5.9) can also be cast in the same form as Eq. (4.5.7) and can be obtained from Eq. (4.5.7) by replacing β by γ and ϵ respectively. Equation (4.5.7) along with the two equations obtained from Eqs. (4.5.8) and (4.5.9) are solved simultaneously for u_η, v_η, and w_η. The velocity components at the free surface are obtained from these derivatives and their values in the cell just below the free surface. Then, pressure is updated using Eq. (4.5.4).

4.6. Validity of the hydrostatic equation in general curvilinear coordinates.

It was noted in Section 2 that when one uses either a Cartesian or a spherical coordinate system, the vertical direction is orthogonal to the horizontal directions. Thus, among the contravariant components of the metric tensor $\tilde{g}$ that involve the z–direction (the vertical direction), only the g^{zz} component is non–zero and the other two components involving the z–direction, viz. g^{xz} and g^{yz} (or $g^{\theta z}$ and $g^{\lambda z}$), are zero. The same holds true for the covariant components also. For these coordinate systems, using the assumptions that (1) the vertical

length scale is very much smaller than the horizontal length scales, and (2) the vertical velocity component is very much smaller than the horizontal velocity components, in the complete vertical momentum equation leads to the hydrostatic equation in a straightforward manner (see Pedlosky [7], for example). It was also mentioned in Section 2 that when one uses a non-orthogonal curvilinear grid in the "horizontal" and the σ–coordinates in the "vertical", one is actually using a general nonorthogonal coordinate system, similar to the one used in this work. For such a coordinate system, all the cross metric terms are, in general, non–zero. Hence, the question whether one can use the hydrostatic equation in such a coordinate system naturally arises. The approach taken in the literature is to transform the hydrostatic equation from the Cartesian or spherical coordinate system to the general curvilinear coordinate system. Obviously, this approach does not answer the question raised above. The correct approach one should take is that one should start with the complete vertical momentum equation and introduce all the assumptions used to obtain the hydrostatic equation in the Cartesian or spherical coordinate system, and see whether one recovers the hydrostatic equation in the nonorthogonal coordinate system also. It is the purpose of this section to show that when the cross metric terms involving the vertical direction are non–zero one can not recover the hydrostatic equation, in general. In order to cast the governing equations in a form that is available in the literature, for the purpose of comparison, it is sufficient to consider only the inviscid part of the momentum equation. Equation (4.2.14), with the viscous term neglected reduces to:

$$\frac{1}{\sqrt{g}}\frac{\hat{\partial}(\sqrt{g}\,\underline{V})}{\partial\tau} + \nabla\cdot\left[\underline{V}\,\underline{V} + \underline{w'}\,\underline{V} + p'\tilde{I}\right] + 2\,\underline{\Omega}\times\underline{V} = 0 \qquad (4.6.1)$$

Note that Eq. (4.6.1) is not in a strong conservative form. However, it is in a form appropriate for the current discussion.

Using the so called partial transformation (see the appendix) in which the velocity vector $\underline{V}$ is resolved with respect to the underlying Cartesian coordinates, whereas the divergence

operator itself is expressed with respect to the general curvilinear coordinates (ξ, ϕ and σ in the following), the three components of the momentum equation (4.6.1) are as follows: For want of better symbols, the symbols (u, v, w) – without the underscore – are used to denote the Cartesian components of $\underline{V}$.

$$\frac{\hat{\partial}(\sqrt{g}\,u)}{\partial\tau} + \frac{\partial}{\partial\xi}\Big[\sqrt{g}\,\{u(u\,\xi_x + v\,\xi_y + w\,\xi_z + \xi_t) + p'\,\xi_x\}\Big] + \frac{\partial}{\partial\phi}\Big[\sqrt{g}\,\{u(u\,\phi_x + v\,\phi_y + w\,\phi_z + \phi_t) + p'\,\phi_x\}\Big] + \frac{\partial}{\partial\sigma}\Big[\sqrt{g}\,\{u(u\,\sigma_x + v\,\sigma_y + w\,\sigma_z + \sigma_t) + p'\,\sigma_x\}\Big] - \sqrt{g}\,f\,v = 0 \qquad (4.6.2)$$

$$\frac{\hat{\partial}(\sqrt{g}\,v)}{\partial\tau} + \frac{\partial}{\partial\xi}\Big[\sqrt{g}\,\{v(u\,\xi_x + v\,\xi_y + w\,\xi_z + \xi_t) + p'\,\xi_y\}\Big] + \frac{\partial}{\partial\phi}\Big[\sqrt{g}\,\{v(u\,\phi_x + v\,\phi_y + w\,\phi_z + \phi_t) + p'\,\phi_y\}\Big] + \frac{\partial}{\partial\sigma}\Big[\sqrt{g}\,\{v(u\,\sigma_x + v\,\sigma_y + w\,\sigma_z + \sigma_t) + p'\,\sigma_y\}\Big] + \sqrt{g}\,f\,u = 0 \qquad (4.6.3)$$

$$\frac{\hat{\partial}(\sqrt{g}\,w)}{\partial\tau} + \frac{\partial}{\partial\xi}\Big[\sqrt{g}\,\{w(u\,\xi_x + v\,\xi_y + w\,\xi_z + \xi_t) + p'\,\xi_z\}\Big] + \frac{\partial}{\partial\phi}\Big[\sqrt{g}\,\{w(u\,\phi_x + v\,\phi_y + w\,\phi_z + \phi_t) + p'\,\phi_z\}\Big] + \frac{\partial}{\partial\sigma}\Big[\sqrt{g}\,\{w(u\,\sigma_x + v\,\sigma_y + w\,\sigma_z + \sigma_t) + p'\,\sigma_z\}\Big] = 0 \qquad (4.6.4)$$

where, $f = 2\,|\,\underline{\Omega}\,|$, and it is assumed that the vector $\underline{\Omega}$ is aligned in the $\underline{k}$ direction for convenience, and subscripts denote partial differentiation. The quantities ξ_t, ϕ_t and σ_t denote the contravariant components of the vector $\underline{w}'$. Note that $\underline{w}'$ is the deformation velocity vector defined in Section 4.2.

Now, in order to be specific, the well known σ–coordinates are introduced as follows:

$$\xi = x\,;\ \phi = y \text{ and } \sigma = \frac{z - \eta}{H + \eta} \qquad (4.6.5)$$

where, $\eta = \eta(x, y, t)$ denotes the free surface and $H = H(x, y)$ denotes the bottom topography. Using Eq. (4.6.5), the various metric quantities that appear in Eqs (4.6.2) – (4.6.4) are obtained as follows:

$$x_\xi = 1 \; ; \; x_\phi = 0 \; ; \; x_\sigma = 0 \tag{4.6.6}$$

$$y_\xi = 0 \; ; \; y_\phi = 1 \; ; \; y_\sigma = 0 \tag{4.6.7}$$

$$\sqrt{g} = x_\xi (y_\phi z_\sigma - y_\sigma z_\phi) + y_\xi (z_\phi x_\sigma - z_\sigma x_\phi) + z_\xi (x_\phi y_\sigma - x_\sigma y_\phi) = z_\sigma \tag{4.6.8}$$

$$\left.\begin{aligned} \xi_x &= (y_\phi z_\sigma - y_\sigma z_\phi)/\sqrt{g} = 1 \\ \xi_y &= (z_\phi x_\sigma - z_\sigma x_\phi)/\sqrt{g} = 0 \\ \xi_z &= (x_\phi y_\sigma - x_\sigma y_\phi)/\sqrt{g} = 0 \end{aligned}\right\} \tag{4.6.9}$$

$$\left.\begin{aligned} \phi_x &= (y_\sigma z_\xi - y_\xi z_\sigma)/\sqrt{g} = 0 \\ \phi_y &= (z_\sigma x_\xi - z_\xi x_\sigma)/\sqrt{g} = 1 \\ \phi_z &= (x_\sigma y_\xi - x_\xi y_\sigma)/\sqrt{g} = 0 \end{aligned}\right\} \tag{4.6.10}$$

$$\left.\begin{aligned} \sigma_x &= (y_\xi z_\phi - y_\phi z_\xi)/\sqrt{g} = - z_\xi/z_\sigma \\ \sigma_y &= (z_\xi x_\phi - z_\phi x_\xi)/\sqrt{g} = - z_\phi/z_\sigma \\ \sigma_z &= (x_\xi y_\phi - x_\phi y_\xi)/\sqrt{g} = 1/z_\sigma \end{aligned}\right\} \tag{4.6.11}$$

Also,

$$\left.\begin{aligned} \sigma_x &= - (\eta_x + \sigma D_x)/D \\ \sigma_y &= - (\eta_y + \sigma D_y)/D \\ \sigma_z &= 1/D \end{aligned}\right\} \tag{4.6.12}$$

where, $D = H + \eta$. Note that Eq. (4.6.11) is obtained from the general relations between contravariant and covariant base vectors whereas Eq. (4.6.12) is obtained from the explicit relationship for σ given in Eq. (4.6.5). From Eqs. (4.6.8), (4.6.11) and (4.6.12) one obtains

$$\sqrt{g} = z_\sigma = 1/\sigma_z = D \tag{4.6.13}$$

Again, from the definition of the σ–coordinate system, it follows that

$$\left.\begin{aligned} \xi_t &= 0 \\ \phi_t &= 0 \\ \sigma_t &= -(\eta_t + \sigma D_t)/D \end{aligned}\right\} \tag{4.6.14}$$

Now, from the Geometric Conservation Law (GCL) [45], one obtains

$$\frac{1}{\sqrt{g}}\frac{\partial \sqrt{g}}{\partial \tau} = -\nabla \cdot \underline{w} = -\nabla \cdot \underline{w}' \tag{4.6.15}$$

since $\nabla \cdot \underline{W} = 0$. Using Eq. (4.6.14) in Eq. (4.6.15), it can be verified that

$$\frac{\partial \sqrt{g}}{\partial \tau} = -\frac{\partial (\sqrt{g}\,\xi_t)}{\partial \xi} - \frac{\partial (\sqrt{g}\,\phi_t)}{\partial \phi} - \frac{\partial (\sqrt{g}\,\sigma_t)}{\partial \sigma}$$

$$= -\frac{\partial}{\partial \sigma}\left[-D(\eta_t + \sigma D_t)/D \right] = \frac{\partial D}{\partial t}$$

Therefore,

$$\frac{\partial \sqrt{g}}{\partial \tau} = \frac{\partial \eta}{\partial t} \tag{4.6.16}$$

The contravariant components of the metric tensor involving the vertical direction are

$$\left.\begin{aligned} g^{\xi\sigma} &= (\nabla\xi) \cdot (\nabla\sigma) = \xi_x \sigma_x + \xi_y \sigma_y + \xi_z \sigma_z = \sigma_x \\ g^{\phi\sigma} &= (\nabla\phi) \cdot (\nabla\sigma) = \phi_x \sigma_x + \phi_y \sigma_y + \phi_z \sigma_z = \sigma_y \\ g^{\sigma\sigma} &= (\nabla\sigma) \cdot (\nabla\sigma) = \sigma_x^2 + \sigma_y^2 + \sigma_z^2 \end{aligned}\right\} \tag{4.6.17}$$

Substituting the various quantities in Eqs. (4.6.2) – (4.6.4), one obtains

$$\frac{\hat{\partial}(Du)}{\partial\tau} + \frac{\partial}{\partial x}\left[Du^2\right] + \frac{\partial}{\partial y}[Duv] + \frac{\partial(Du\omega)}{\partial\sigma} - Dfv$$

$$+ \frac{\partial}{\partial x}[Dp'] - \frac{\partial}{\partial\sigma}[p'(\eta_x + \sigma D_x)] = 0 \qquad (4.6.18)$$

$$\frac{\hat{\partial}(Dv)}{\partial\tau} + \frac{\partial}{\partial x}[Dvu] + \frac{\partial}{\partial y}\left[Dv^2\right] + \frac{\partial(Dv\omega)}{\partial\sigma} + Dfu$$

$$+ \frac{\partial}{\partial y}[Dp'] - \frac{\partial}{\partial\sigma}\left[p'(\eta_y + \sigma D_y)\right] = 0 \qquad (4.6.19)$$

$$\frac{\hat{\partial}(Dw)}{\partial\tau} + \frac{\partial}{\partial x}[Dwu] + \frac{\partial}{\partial y}[Dwv] + \frac{\partial(Dw\omega)}{\partial\sigma} + \frac{\partial p'}{\partial\sigma} = 0 \qquad (4.6.20)$$

where, ω is given by

$$\omega = u\sigma_x + v\sigma_y + w\sigma_z + \sigma_t$$

$$= \frac{1}{D}\left[w - u(\eta_x + \sigma D_x) - v(\eta_y + \sigma D_y) - (\eta_t + \sigma D_t)\right] \qquad (4.6.21)$$

Note that, all the corresponding terms that appear in Eqs. (4.6.18) – (4.6.20) agree with the derivation of Blumberg and Mellor [5]. Equation (4.6.20) is the complete inviscid vertical momentum equation.

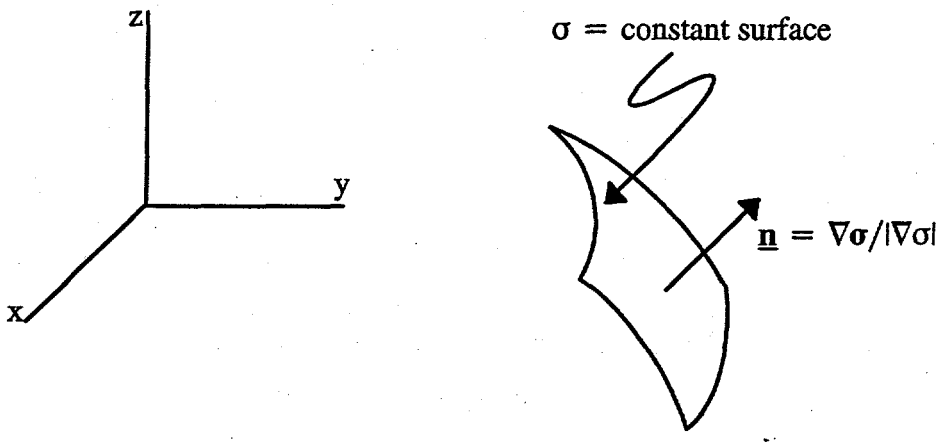


Fig. 1. Schematic Diagram of a σ–surface with Steep Gradient

In order to show that one can not recover the hydrostatic equation in a general curvilinear coordinate system (through the example of the σ–coordinate system) it is enough to show that in Eq. (4.6.20) there is at least one more term in addition to the pressure term that can not be neglected. For this purpose, it is necessary to consider the dimensional form of the equations whereas Eqs. (4.6.18) – (4.6.20) are in non–dimensional form. The dimensional form can be shown to be the same equations except that the pressure term is multiplied by $1/\rho$. Note that, in Eq. (4.6.21), σ_x, σ_y and σ_z are components of $\nabla\sigma$. For a σ = constant surface, $\nabla\sigma/|\nabla\sigma|$ denotes the normal to the surface. In regions of steep (bottom or free surface) gradients σ_x and σ_y are of the same order or even greater than that of σ_z, depending upon the steepness (see Fig. 1). Hence, the contravariant velocity component $D\omega$ can be of the same order of magnitude as the horizontal velocities u and v. Looking at Eq. (4.6.17), it can be readily inferred that this is due to the fact that the cross metrics involving the "vertical" direction are non–zero. Under these circumstances the order of magnitude analysis similar to the one that can be found in Pedlosky [7] (page 60), for Eq. (4.6.20) yields:

$$\frac{\hat{\partial}(Dw)}{\partial\tau} + \frac{\partial}{\partial x}[Dwu] + \frac{\partial}{\partial y}[Dwv] + \frac{\partial(Dw\omega)}{\partial\sigma} + \frac{1}{\rho}\frac{\partial p'}{\partial\sigma} = 0 \qquad (4.6.22)$$

$$\frac{DW}{T} \qquad \frac{DWU}{L} \qquad \frac{DWU}{L} \qquad WU \qquad \frac{P}{\rho}$$

Or

$$\frac{W}{T} \qquad \frac{WU}{L} \qquad \frac{WU}{L} \qquad \frac{WU}{D} \qquad \frac{P}{\rho D}$$

An analysis of the horizontal momentum equations, Eqs. (4.6.18) and (4.6.19), still yields the same order of magnitude for the pressure as in Pedlosky [7]. That is,

$$P = \rho U\left[\frac{L}{T},\ U,\ fL\right]_{max} \qquad (4.6.23)$$

From Eq. (4.6.22), the ratio of convective terms to the pressure gradient term is given by

$$\frac{\text{convective terms}}{\text{pressure gradient term}} = \frac{\delta\left[\frac{\delta}{T}, \frac{U\delta}{L}, \frac{U}{L}\right]_{max}}{\left[\frac{1}{T}, \frac{U}{L}, f\right]_{max}} \tag{4.6.24}$$

The ratio in Eq. (4.6.24) is of the order of δ where $\delta = D/L$, D is the vertical scale length and L is the horizontal scale length. Terms of the order of δ can not be neglected from the governing equations since otherwise one would end up with a strictly two dimensional (in x and y) equations. Hence, the term $\frac{\partial (D w \omega)}{\partial \sigma}$ can not be dropped from Eq. (4.6.20). Thus, Eq. (4.6.20) simplifies to

$$\frac{\partial (D w \omega)}{\partial \sigma} + \frac{\partial p'}{\partial \sigma} = 0 \tag{4.6.25}$$

Note that one can not simplify Eq. (4.6.25) further since Eq. (4.6.24) contains the 'max' operator. Thus, using the same assumptions that are used to obtain the hydrostatic equation in a Cartesian (or spherical) coordinate system one can not obtain the hydrostatic equation in a general curvilinear coordinate system. On the other hand, suppose $\sigma = z$ in Eq. (4.6.5). Then, $D = 1$, $g^{xz} = 0$, $g^{yz} = 0$ and $\omega = w$, and Eq. (4.6.25) becomes

$$\frac{\partial (w w)}{\partial z} + \frac{\partial p'}{\partial z} = 0 \tag{4.6.26}$$

It can be easily shown that the first term on the left hand side of Eq. (4.6.26) is of the order of δ^2 which can be dropped. Thus, one obtains the hydrostatic equation in a Cartesian coordinate system. So far, the discussion has not included the viscous terms. Including the viscous terms and using the same assumptions that led to Eq. (4.6.25) leads one to

$$\frac{\partial (D w \omega)}{\partial \sigma} + \frac{\partial p'}{\partial \sigma} - \frac{\partial}{\partial \sigma}\left(\sigma_x \tau_{xz} + \sigma_y \tau_{yz}\right) = 0 \tag{4.6.27}$$

where τ_{xz} and τ_{yz} are the shear stresses. Again, it can be seen that one can not obtain the hydrostatic equation in a general coordinate system. It should be mentioned here that the discus-

sion in this Section does not invoke the effects of variable density which are addressed by Haney [6].

5. Numerical Procedure

The governing equations that are presented in tensor invariant form in Section 4 need to be expressed in component form in order to be solved numerically or otherwise. As mentioned in Section 4.1, in order to maintain the strong conservative nature of the governing equations in component form also, one effects the so called partial transformation. An example of this procedure was presented in Section 4.6 in which the tensor invariant form presented in Eq. (4.6.1) is presented in component form in Eqs. (4.6.2) – (4.6.4) where the divergence operator is written with respect to a general curvilinear coordinate system (see also the appendix). In this section the numerical procedure used to solve the governing equations is described in detail. The set of governing equations chosen are those corresponding to the absolute and relative velocity procedures discussed in Section 4.1 and the set of equations belonging to Section 4.4 with Eq. (4.4.1) replaced by Eq. (4.1.15). The continuity equation in the original artificial compressibility method proposed by Chorin [46] contained the time derivative of the static pressure only. Recently, Beddhu, Taylor and Whitfield [37] have proposed the modified artificial compressibility method in which the body force potential is also added to the pressure in the continuity equation. This is the approach adopted in this work as well. Thus, the governing equations are hyperbolic and a time marching approach is adopted to solve them.

In Section 5.1, the governing equations are first cast in the so called numerical vector form. Then the equations in the numerical vector form are discretized in an implicit manner and then linearized which result in the flux Jacobians. In Section 5.2, the eigensystem of the flux Jacobians are presented. In Section 5.3, an approximate, one dimensional Riemann problem is solved at each cell face following the approach of Roe [23] to obtain first order accurate fluxes and then the MUSCL scheme of van Leer [24] is used to obtain higher order fluxes (up to third order). The Newton–relaxation procedure [25] is introduced in Section 5.4 and the numerical implementation of the boundary conditions is discussed in Section 5.5.

5.1. Numerical vector form.

All the governing equations of Section 4 can be expressed in the following manner after the introduction of the partial transformation

$$\frac{\partial Q}{\partial \tau} + \frac{\partial F}{\partial \xi} + \frac{\partial G}{\partial \eta} + \frac{\partial H}{\partial \zeta} + \frac{\partial F^v}{\partial \xi} + \frac{\partial G^v}{\partial \eta} + \frac{\partial H^v}{\partial \zeta} + M = 0 \qquad (5.1.1)$$

In Eq. (5.1.1) ξ, η and ς denote the curvilinear coordinates, F, G and H denote the inviscid fluxes on a ξ = constant face, η = constant face and ς = constant face respectively. Similarly, F^v, G^v and H^v denote the viscous fluxes. M is the source term which is non–zero only for the 6x6 set. The various quantities in Eq. (5.1.1) for the different sets of governing equations are presented below.

Absolute velocity procedure (4x4 set)

This set consists of Eqs (4.1.14) and (4.1.15) and the various quantities in Eq. (5.1.1) are:

$$Q = \sqrt{g}\begin{bmatrix} p' \\ u \\ v \\ w \end{bmatrix} \quad F = \sqrt{g}\begin{bmatrix} \beta u^1 \\ u'(u^1 + \xi_t) + p'\xi_x \\ v'(u^1 + \xi_t) + p'\xi_y \\ w'(u^1 + \xi_t) + p'\xi_z \end{bmatrix} \quad F^v = \sqrt{g}\begin{bmatrix} 0 \\ \sigma_{xx}\xi_x + \sigma_{xy}\xi_y + \sigma_{xz}\xi_z \\ \sigma_{xy}\xi_x + \sigma_{yy}\xi_y + \sigma_{yz}\xi_z \\ \sigma_{xz}\xi_x + \sigma_{yz}\xi_y + \sigma_{zz}\xi_z \end{bmatrix} \qquad (5.1.2)$$

$$u^1 = u\,\xi_x + v\,\xi_y + w\,\xi_z$$

where, u, v and w are the components of the absolute velocity vector with respect to a Cartesian coordinate system and u′, v′ and w′ are the Cartesian components of the vector $\underline{u} - \underline{w}$, σ_{xx}, etc., are the Cartesian components of the Stokes tensor, ξ_x, ξ_y and ξ_z are the Cartesian components of the contravariant base vector grad ξ. Expressions for G and H are similar to F and can be obtained from F by replacing ξ by η and ζ respectively. Similarly G^v and H^v can be obtained from F^v.

Relative velocity procedure (4x4 set)

Equations (4.1.7) and (4.1.8) are the governing equations for this set.

$$Q = \sqrt{g}\begin{bmatrix} p' \\ u \\ v \\ w \end{bmatrix} \qquad F = \sqrt{g}\begin{bmatrix} \beta u^1 \\ u' u^1 + p'\xi_x \\ v' u^1 + p'\xi_y \\ w' u^1 + p'\xi_z \end{bmatrix} \qquad F^v = \sqrt{g}\begin{bmatrix} 0 \\ \sigma_{xx}\xi_x + \sigma_{xy}\xi_y + \sigma_{xz}\xi_z \\ \sigma_{xy}\xi_x + \sigma_{yy}\xi_y + \sigma_{yz}\xi_z \\ \sigma_{xz}\xi_x + \sigma_{yz}\xi_y + \sigma_{zz}\xi_z \end{bmatrix} \quad (5.1.3)$$

$$u^1 = u\,\xi_x + v\,\xi_y + w\,\xi_z$$

where, u, v and w are the components of the relative velocity vector with respect to a Cartesian coordinate system and u′, v′ and w′ are the Cartesian components of the vector $\underline{v} - 2\,\underline{w}$. Other quantities are as in absolute velocity procedure.

Governing equations of a thermohaline ocean with respect to an observer in the absolute frame (6x6 set)

The governing equations for this set are Eqs. (4.4.1) – (4.4.4).

$$Q = \sqrt{g}\begin{bmatrix} p' \\ u \\ v \\ w \\ T \\ S \end{bmatrix} \qquad F = \sqrt{g}\begin{bmatrix} \beta u^1 \\ u(u^1 + \xi_t) + p'\xi_x \\ v(u^1 + \xi_t) + p'\xi_y \\ w(u^1 + \xi_t) + p'\xi_z \\ T(u^1 + \xi_t) \\ S(u^1 + \xi_t) \end{bmatrix}$$

$$F^v = \sqrt{g}\begin{bmatrix} 0 \\ \sigma_{xx}\xi_x + \sigma_{xy}\xi_y + \sigma_{xz}\xi_z \\ \sigma_{xy}\xi_x + \sigma_{yy}\xi_y + \sigma_{yz}\xi_z \\ \sigma_{xz}\xi_x + \sigma_{yz}\xi_y + \sigma_{zz}\xi_z \\ \kappa(T_x\xi_x + T_y\xi_y + T_z\xi_z)/Pe \\ \alpha_S(S_x\xi_x + S_y\xi_y + S_z\xi_z) \end{bmatrix} \qquad M = \rho'\begin{bmatrix} 0 \\ M_1 \\ M_2 \\ M_3 \\ 0 \\ 0 \end{bmatrix} \qquad (5.1.4)$$

$$u^1 = u\,\xi_x + v\,\xi_y + w\,\xi_z$$

Note that, the tensor $\underline{W}\,\underline{W}$ appearing in Eq. (4.4.2) can also be written as $(\Omega^2 R^2/2)\tilde{I}$ where R is the normal distance from the axis of rotation to the point under consideration, and can be added to the definition of p'. Thus, it does not explicitly appear in Eq. (5.1.4). The expressions for M_1, M_2 and M_3 are involved and are written in compact tensor notation as follows::

$$M_i = -\frac{\partial}{\partial x^k}\left[\sqrt{g}\,(\hat{p}\frac{\partial x^k}{\partial x_i} - \sigma_{ij}\frac{\partial x^k}{\partial x_j})\right] + \frac{1}{Fr^2}\frac{\partial}{\partial x^k}\left[\sqrt{g}\,\chi\frac{\partial x^k}{\partial x_i}\right] \qquad i = 1,\ 2,\ 3$$

In the expression for M_i, the summation convention is used over repeated indices.

Discretization

In the finite volume approach, the physical domain of interest is divided into a finite number of cells and the variables are defined at the centers of these cells. Note that these cells can be created in a curvilinear coordinate system in a natural manner using standard grid generation packages. A simple one dimensional case is schematically shown in Fig. 2. Cell centers designated from 2 through NI are the field points and the cell centers 1 and NI+1 of the fictitious cells (not shown) are the so called phantom points used for prescribing the boundary conditions. The governing equations are solved only at the field points. In what follows, the ξ, η and ς coordinates are indicated by the symbols i, j and k respectively in the discrete sense. Thus, i and i+1 represent two consecutive points in the increasing direction of ξ and so on.

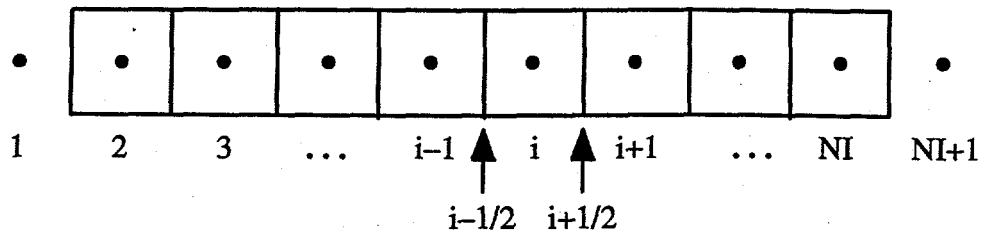


Fig. 2. Schematic Showing the Cell Volumes and the Variable Locations for a One Dimensional Case

Denoting the discrete time by the symbol n (thus, $\tau_n = n\,\Delta\tau$), Eq. (5.1.1) is discretized about the cell center (i, j, k) as follows (refer to Fig. 2):

$$\frac{3\,Q^{n+1} - 4\,Q^{n} + Q^{n-1}}{2\,\Delta\tau} + F^{n+1}_{i+\frac{1}{2}} - F^{n+1}_{i-\frac{1}{2}} + G^{n+1}_{j+\frac{1}{2}} - G^{n+1}_{j-\frac{1}{2}} + H^{n+1}_{k+\frac{1}{2}} - H^{n+1}_{k-\frac{1}{2}} +$$

$$F^{v\,n+1}_{i+\frac{1}{2}} - F^{v\,n+1}_{i-\frac{1}{2}} + G^{v\,n+1}_{j+\frac{1}{2}} - G^{v\,n+1}_{j-\frac{1}{2}} + H^{v\,n+1}_{k+\frac{1}{2}} - H^{v\,n+1}_{k-\frac{1}{2}} + M^{n+1} = 0 \quad (5.1.5)$$

Note that, in Eq. (5.1.5), $\Delta\xi$, $\Delta\eta$ and $\Delta\varsigma$ are all taken to be unity and the time derivative is approximated using the second order accurate backward Euler formula. Note that the fluxes have only one subscript instead of three. This is just a notational simplification and it is understood that $F^{n+1}_{i+\frac{1}{2}}$, in fact, denotes $F^{n+1}_{i+\frac{1}{2},j,k}$ and so on.

Equation (5.1.5) is in the implicit form in which the unknown solution is at time level n+1. Direct solution of Eq. (5.1.5) is difficult since the inviscid fluxes are non–linear. One method of solving Eq. (5.1.5) is to consider it as a system of non–linear algebraic equations in the unknowns Q^{n+1} and use Newton's method to find the root of this system. Note that this is an iterative procedure. Strictly speaking, the fluxes are functions of the Q variables. However, since the metrics are known at time n+1, no linearization needs to be done with respect to them. Hence, the fluxes are considered only as functions of the q–variables where $q = Q/\sqrt{g}$.

The process of solving Eq. (5.1.5) using Newton's method requires the evaluation of inviscid flux Jacobians, $\frac{\partial F}{\partial q}$, $\frac{\partial G}{\partial q}$ and $\frac{\partial H}{\partial q}$ which is discussed next.

5.2. Flux Jacobians and their eigensystems.

Absolute velocity procedure (4x4 set)

Defining the flux Jacobians as $A = \frac{\partial F}{\partial q}$; $B = \frac{\partial G}{\partial q}$; $C = \frac{\partial H}{\partial q}$ and denoting the generic flux Jacobian by K, one obtains

$$K = \sqrt{g}\begin{bmatrix} 0 & \beta k_x & \beta k_y & \beta k_z \\ k_x & \theta_k + u'k_x & u'k_y & u'k_z \\ k_y & v'k_x & \theta_k + v'k_y & v'k_z \\ k_z & w'k_x & w'k_y & \theta_k + w'k_z \end{bmatrix} \qquad (5.2.1)$$

where $\theta_k = k_t + u\,k_x + v\,k_y + w\,k_z$ and $k_t = \underline{w} \cdot \underline{a}^k$ where $\underline{a}^k$ is the contravariant base vector on the k = constant face. When $k = \xi$, K = A; $k = \eta$, K = B and $k = \varsigma$, K = C. In order to find the eigenvalues of K, the following matrix M and its inverse M^{-1} are used to form the matrix $\kappa = MKM^{-1}$.

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{u'}{\beta} & 1 & 0 & 0 \\ \frac{v'}{\beta} & 0 & 1 & 0 \\ \frac{w'}{\beta} & 0 & 0 & 1 \end{bmatrix} \qquad M^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{u'}{\beta} & 1 & 0 & 0 \\ -\frac{v'}{\beta} & 0 & 1 & 0 \\ -\frac{w'}{\beta} & 0 & 0 & 1 \end{bmatrix}$$

The matrix κ is given by

$$\kappa = \sqrt{g}\begin{bmatrix} \theta_k - 2k_t & \beta k_x & \beta k_y & \beta k_z \\ k_x + \dfrac{u'\theta_k}{\beta} & \theta_k & 0 & 0 \\ k_y + \dfrac{v'\theta_k}{\beta} & 0 & \theta_k & 0 \\ k_z + \dfrac{w'\theta_k}{\beta} & 0 & 0 & \theta_k \end{bmatrix} \tag{5.2.2}$$

The eigenvalues of K and κ are the same since they are similar matrices. However, it is much easier to find the eigenvalues of κ rather than that of K, and are found to be

$$\left.\begin{aligned} \lambda_{1,2} &= \theta_k \\ \lambda_3 &= \theta_k - k_t + c \\ \lambda_4 &= \theta_k - k_t - c \end{aligned}\right\} \tag{5.2.3}$$

where,

$$c = \sqrt{(\theta_k - k_t)^2 + \beta\,(k_x^2 + k_y^2 + k_z^2)}$$

Following Taylor [22], in order to obtain the left and right eigenvectors of K, first the left and right eigenvectors of κ are obtained. They are the columns and rows of the following matrices respectively

$$P_k = \begin{bmatrix} 0 & 0 & -\tilde{c}^- & -\tilde{c}^+ \\ \tilde{x}1 & \tilde{x}2 & \phi 1 & \phi 1 \\ \tilde{y}1 & \tilde{y}2 & \phi 2 & \phi 2 \\ \tilde{z}1 & \tilde{z}2 & \phi 3 & \phi 3 \end{bmatrix}$$

$$P_k^{-1} = \frac{1}{\phi}\begin{bmatrix} 0 & 2\tilde{c}\phi 4 & -2\tilde{c}\phi 5 & 2\tilde{c}\phi 6 \\ 0 & -2\tilde{c}\phi 7 & 2\tilde{c}\phi 8 & -2\tilde{c}\phi 9 \\ \frac{\phi}{2\tilde{c}} & 2k_x\tilde{c}^+ & 2k_y\tilde{c}^+ & 2k_z\tilde{c}^+ \\ -\frac{\phi}{2\tilde{c}} & -2k_x\tilde{c}^- & -2k_y\tilde{c}^- & -2k_z\tilde{c}^- \end{bmatrix}$$

where

$$\phi 1 = \tilde{k}_x + \frac{u'\,\tilde{\theta}_k}{\beta} \qquad \phi 2 = \tilde{k}_y + \frac{v'\,\tilde{\theta}_k}{\beta} \qquad \phi 3 = \tilde{k}_z + \frac{w'\,\tilde{\theta}_k}{\beta}$$

$$\phi 4 = \tilde{y2}\phi 3 - \tilde{z2}\phi 2 \qquad \phi 5 = \tilde{x2}\phi 3 - \tilde{z2}\phi 1 \qquad \phi 6 = \tilde{x2}\phi 2 - \tilde{y2}\phi 1$$

$$\phi 7 = \tilde{y1}\phi 3 - \tilde{z1}\phi 2 \qquad \phi 8 = \tilde{x1}\phi 3 - \tilde{z1}\phi 1 \qquad \phi 9 = \tilde{x1}\phi 2 - \tilde{y1}\phi 1$$

$$\tilde{x1} = \frac{x1}{\sqrt{g|\nabla k|}} \qquad \tilde{y1} = \frac{y1}{\sqrt{g|\nabla k|}} \qquad \tilde{z1} = \frac{z1}{\sqrt{g|\nabla k|}}$$

$$\tilde{x2} = \frac{x2}{\sqrt{g|\nabla k|}} \qquad \tilde{y2} = \frac{y2}{\sqrt{g|\nabla k|}} \qquad \tilde{z2} = \frac{z2}{\sqrt{g|\nabla k|}}$$

$$|\nabla k| = \sqrt{k_x^2 + k_y^2 + k_z^2} \qquad \text{and} \qquad \phi = \frac{4\tilde{c}}{\beta}\left(\beta + \tilde{\theta}_k\left(\tilde{\theta}_k - 2\,\tilde{k}_t\right)\right)$$

(x1, y1, z1) and (x2, y2, z2) are the diagonal vectors on the k = constant face [22]. In this section, a tilde over a quantity denotes that the metrics used in computing that quantity are normalized with the area of the cell face. The left and right eigenvectors of the flux Jacobians K are obtained as follows: $T_k = M\,P_k$ and $T_k^{-1} = P_k^{-1}\,M^{-1}$ where the left eigenvectors are given by the rows of T_k^{-1}, and the right eigenvectors are given by the columns of T_k respectively. The matrices T_k and T_k^{-1} are the following:

$$T_k = \begin{bmatrix} 0 & 0 & -\tilde{c}^- & -\tilde{c}^+ \\ \tilde{x1} & \tilde{x2} & \tilde{k}_x + \frac{u'\tilde{\lambda}_k^3}{\beta} & \tilde{k}_x + \frac{u'\tilde{\lambda}_k^4}{\beta} \\ \tilde{y1} & \tilde{y2} & \tilde{k}_y + \frac{v'\tilde{\lambda}_k^3}{\beta} & \tilde{k}_y + \frac{v'\tilde{\lambda}_k^4}{\beta} \\ \tilde{z1} & \tilde{z2} & \tilde{k}_z + \frac{w'\tilde{\lambda}_k^3}{\beta} & \tilde{k}_z + \frac{w'\tilde{\lambda}_k^4}{\beta} \end{bmatrix}$$

$$T_k^{-1} = \frac{1}{\phi}\begin{bmatrix} \frac{2\tilde{c}}{\beta}(-u'\phi4 + v'\phi5 - w'\phi6) & 2\tilde{c}\phi4 & -2\tilde{c}\phi5 & 2\tilde{c}\phi6 \\ \frac{2\tilde{c}}{\beta}(u'\phi7 - v'\phi8 - w'\phi9) & -2\tilde{c}\phi7 & 2\tilde{c}\phi8 & -2\tilde{c}\phi9 \\ \frac{2}{\beta}(\beta + \tilde{\lambda}_k^4(\tilde{\theta}_k - 2\tilde{k}_t)) & 2\tilde{c}^+\tilde{k}_x & 2\tilde{c}^+\tilde{k}_y & 2\tilde{c}^+\tilde{k}_z \\ -\frac{2}{\beta}(\beta + \tilde{\lambda}_k^3(\tilde{\theta}_k - 2\tilde{k}_t)) & -2\tilde{c}^-\tilde{k}_x & -2\tilde{c}^-\tilde{k}_y & -2\tilde{c}^-\tilde{k}_z \end{bmatrix}$$

The quantity $T\Lambda^{-}T^{-1}\delta q$ which is required in the Roe flux formulation [23] is given by

$$T\Lambda^{-}T^{-1}\delta q = \begin{bmatrix} \lambda_k^4 r_{14} R_4 \\ \lambda_k^4 r_{24} R_4 - \lambda_k^1 (r_{24} R_4 + r_{23} R_3 - \delta u) \\ \lambda_k^4 r_{34} R_4 - \lambda_k^1 (r_{34} R_4 + r_{33} R_3 - \delta v) \\ \lambda_k^4 r_{44} R_4 - \lambda_k^1 (r_{44} R_4 + r_{43} R_3 - \delta w) \end{bmatrix}$$

where

$R_3 = l_{31}\,\delta p + l_{32}\delta u + l_{33}\delta v + l_{34}\delta w,$ $\qquad$ $R_4 = l_{41}\,\delta p + l_{42}\delta u + l_{43}\delta v + l_{44}\delta w,$

$(l_{31}, \ldots, l_{34})$ and $(l_{41}, \ldots, l_{44})$ are the 3rd and 4th left eigenvectors (that is, 3rd and 4th rows of T_k^{-1}), and, $(r_{13}, \ldots, r_{43})^T$ and $(r_{14}, \ldots, r_{44})^T$ are the 3rd and 4th right eigenvec-

tors (that is, 3rd and 4th columns of T_k) . The quantity δq is given by $\delta q = q_R - q_L$ where q_R and q_L are defined using a MUSCL type approach ([24], [28]).

A requirement of the theory behind this numerical scheme is that the first order fluxes satisfy the property U defined by Roe [23]. It can be easily verified, by direct substitution, that the Roe averages, defined by $\overline{\phi} = (\phi_i + \phi_{i+1})/2$ where ϕ is any flow variable, and the components of $\underline{w}$ at cell centers i and i+1 taken to be the same as that at the cell face i+1/2, satisfy the relation $F_{i+1} - F_i = A(\overline{\phi})\,[Q_{i+1} - Q_i]$ where F is the flux and A is the flux Jacobian.

Relative velocity procedure (4x4 set)

The analysis of the absolute velocity procedure carries through and the flux Jacobians as well as eigenvectors retain the same form as given by the matrices K, κ, T_k, and T_k^{-1} respectively. The eigenvalues of this system are given by

$$\left.\begin{aligned} \lambda_{1,2} &= \theta_k \\ \lambda_3 &= \theta_k - k_t + c \\ \lambda_4 &= \theta_k - k_t - c \end{aligned}\right\} \qquad (5.2.4)$$

where,

$$\theta_k = u\,k_x + v\,k_y + w\,k_z \quad \text{and} \quad c = \sqrt{(\theta_k - k_t)^2 + \beta\,(k_x^2 + k_y^2 + k_z^2)}$$

Note that the definition of, θ_k, is different from that of the absolute velocity procedure. k_t is the same as before.

Governing equations of a thermohaline ocean with respect to an observer in the absolute frame (6x6 set)

The software Mathematica has been used extensively to obtain the eigensystem for this case. The generic flux Jacobian is given by

$$K = \sqrt{g}\begin{bmatrix} 0 & \beta k_x & \beta k_y & \beta k_z & 0 & 0 \\ k_x & \theta_k + uk_x & uk_y & uk_z & 0 & 0 \\ k_y & vk_x & \theta_k + vk_y & vk_z & 0 & 0 \\ k_z & wk_x & wk_y & \theta_k + wk_z & 0 & 0 \\ 0 & Tk_x & Tk_y & Tk_z & \theta_k & 0 \\ 0 & Sk_x & Sk_y & Sk_z & 0 & \theta_k \end{bmatrix}$$

where $\theta_k = k_t + u\,k_x + v\,k_y + w\,k_z$. The eigenvalues of K are found to be

$$\left.\begin{aligned} \lambda_k^i &= \sqrt{g}\,\theta_k\,;\ i = 1, \ldots, 4 \\ \lambda_k^5 &= \sqrt{g}\,(\theta_k - c^-) \\ \lambda_k^6 &= \sqrt{g}\,(\theta_k - c^+) \end{aligned}\right\} \qquad (5.2.5)$$

where,

$$c^- = k_t/2 - c\ ;\ c^+ = k_t/2 + c \text{ and}$$

$$c = \sqrt{(\theta_k - k_t/2)^2 + \beta(k_x^2 + k_y^2 + k_z^2)}.$$

A set of normalized right eigenvectors are given by the columns of the following matrix T

$$T = \frac{1}{|\nabla k|}\begin{bmatrix} 0 & 0 & 0 & 0 & -c^- & -c^+ \\ \tilde{x}1 & \tilde{x}2 & 0 & 0 & k_x + u\lambda_k^5/\beta' & k_x + u\lambda_k^6/\beta' \\ \tilde{y}1 & \tilde{y}2 & 0 & 0 & k_y + v\lambda_k^5/\beta' & k_y + v\lambda_k^6/\beta' \\ \tilde{z}1 & \tilde{z}2 & 0 & 0 & k_z + w\lambda_k^5/\beta' & k_z + w\lambda_k^6/\beta' \\ 0 & 0 & 1 & 0 & -T\phi 1/c'^- & -T\phi 2/c'^+ \\ 0 & 0 & 0 & 1 & -S\phi 1/c'^- & -S\phi 2/c'^+ \end{bmatrix}$$

where $|\nabla k| = \sqrt{k_x^2 + k_y^2 + k_z^2}$, $\phi 1 = \sqrt{g}\,(k_x^2 + k_y^2 + k_z^2) + \lambda_k^5(\theta_k - k_t)/\beta$,

$\phi 2 = \sqrt{g}\,(k_x^2 + k_y^2 + k_z^2) \quad + \lambda_k^6(\theta_k - k_t)/\beta$, $\beta' = \sqrt{g}\,\beta$, $c'^- = \sqrt{g}\,c^-$, $c'^+ = \sqrt{g}\,c^+$.

Following Taylor [22], (x1, y1, z1) and (x2, y2, z2) are the diagonal vectors on the k = constant cell face; $\tilde{x1} = x1\sqrt{|\nabla k|}$ and so on. The corresponding set of left eigenvectors are the row vectors of the inverse of T and after considerable algebra are obtained to be

$$T^{-1} = \frac{1}{|\nabla k| D}\begin{bmatrix} l_{11} & l_{12} & l_{13} & l_{14} & 0 & 0 \\ l_{21} & l_{22} & l_{23} & l_{24} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & l_{34} & |\nabla k| D & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} & 0 & |\nabla k| D \\ l_{51} & l_{52} & l_{53} & l_{54} & 0 & 0 \\ l_{61} & l_{62} & l_{63} & l_{64} & 0 & 0 \end{bmatrix}$$

where

$$l_{11} = \frac{2c}{\beta}\left[\tilde{x2}\,(k_z v - k_y w) + \tilde{y2}\,(k_x w - k_z u) \quad + \tilde{z2}\,(k_y u - k_x v)\right]$$

$$l_{12} = \frac{2c}{\beta}\left[\beta\,(k_z \tilde{y2} - k_y \tilde{z2}) + \theta_k\,(w \tilde{y2} - v \tilde{z2})\right]$$

$$l_{13} = \frac{2c}{\beta}\left[\beta\,(k_x \tilde{z2} - k_z \tilde{x2}) + \theta_k\,(u \tilde{z2} - w \tilde{x2})\right]$$

$$l_{14} = \frac{2c}{\beta}\left[\beta\,(k_y \tilde{x2} - k_x \tilde{y2}) + \theta_k\,(v \tilde{x2} - u \tilde{y2})\right]$$

$$l_{21} = -\frac{2c}{\beta}\left[\tilde{x1}\,(k_z v - k_y w) + \tilde{y1}\,(k_x w - k_z u) \quad + \tilde{z1}\,(k_y u - k_x v)\right]$$

$$l_{22} = -\frac{2c}{\beta}\left[\beta\,(k_z \tilde{y1} - k_y \tilde{z1}) + \theta_k\,(w \tilde{y1} - v \tilde{z1})\right]$$

$$l_{23} = -\frac{2c}{\beta}\left[\beta\,(k_x \tilde{z1} - k_z \tilde{x1}) + \theta_k\,(u \tilde{z1} - w \tilde{x1})\right]$$

$$l_{24} = -\frac{2c}{\beta}\left[\beta\,(k_y \tilde{x1} - k_x \tilde{y1}) + \theta_k\,(v\,\tilde{x1} - u\,\tilde{y1})\right]$$

$$l_{31} = \frac{2T}{\beta}\left[\beta\left(\frac{\phi 1}{c^-} - \frac{\phi 2}{c^+}\right) + (\theta_k - k_t)\left(\frac{\phi 1\,\lambda_k^6}{c^-} - \frac{\phi 2\,\lambda_k^5}{c^+}\right)\right]$$

$$l_{32} = 2T\,k_x\left[\frac{c^+\phi 1}{c^-} - \frac{c^-\phi 2}{c^+}\right]$$

$$l_{33} = 2T\,k_y\left[\frac{c^+\phi 1}{c^-} - \frac{c^-\phi 2}{c^+}\right]$$

$$l_{34} = 2T\,k_z\left[\frac{c^+\phi 1}{c^-} - \frac{c^-\phi 2}{c^+}\right]$$

$$l_{41} = \frac{2S}{\beta}\left[\beta\left(\frac{\phi 1}{c^-} - \frac{\phi 2}{c^+}\right) + (\theta_k - k_t)\left(\frac{\phi 1\,\lambda_k^6}{c^-} - \frac{\phi 2\,\lambda_k^5}{c^+}\right)\right]$$

$$l_{42} = 2\,S\,k_x\left[\frac{c^+\phi 1}{c^-} - \frac{c^-\phi 2}{c^+}\right]$$

$$l_{43} = 2\,S\,k_y\left[\frac{c^+\phi 1}{c^-} - \frac{c^-\phi 2}{c^+}\right]$$

$$l_{44} = 2\,S\,k_z\left[\frac{c^+\phi 1}{c^-} - \frac{c^-\phi 2}{c^+}\right]$$

$$l_{51} = \frac{2}{\beta}\left[\beta + \lambda_k^6\,(\theta_k - k_t)\right]$$

$$l_{52} = 2\,k_x\,c^+$$

$$l_{53} = 2\,k_y\,c^+$$

$$l_{54} = 2\,k_z\,c^+$$

$$l_{61} = \frac{-2}{\beta}\left[\beta + \lambda_k^5(\theta_k - k_t)\right]$$

$$l_{62} = -2k_x c^-$$

$$l_{63} = -2k_y c^-$$

$$l_{64} = -2k_z c^-$$

and

$$D = \frac{4c}{\beta}\left[\beta + \theta_k(\theta_k - k_t)\right]$$

In the Roe [23] flux formulation used here to compute the flux at the cell face, one needs the quantity $T\Lambda^- T^{-1}\Delta q$ at the cell face, where Λ^- is the diagonal matrix containing only the negative eigenvalues of K. After considerable algebra and using the fact that $T\,T^{-1} = I$, it can be shown that $T\Lambda^- T^{-1}\Delta q$ can be written only in terms of the 5th and 6th left and right eigenvectors as follows:

$$T\Lambda^- T^{-1}\Delta q = \begin{bmatrix} \lambda_k^6 r_{16} R_6 \\ \lambda_k^6 r_{26} R_6 - \lambda_k^1 (r_{26} R_6 + r_{25} R_5 - \Delta u) \\ \lambda_k^6 r_{36} R_6 - \lambda_k^1 (r_{36} R_6 + r_{35} R_5 - \Delta v) \\ \lambda_k^6 r_{46} R_6 - \lambda_k^1 (r_{46} R_6 + r_{45} R_5 - \Delta w) \\ \lambda_k^6 r_{56} R_6 - \lambda_k^1 (r_{56} R_6 + r_{55} R_5 - \Delta T) \\ \lambda_k^6 r_{66} R_6 - \lambda_k^1 (r_{66} R_6 + r_{65} R_5 - \Delta S) \end{bmatrix}$$

5.3. Numerical flux formulation.

From Eq. (5.1.5) and Fig. 2, it can be seen that the fluxes are evaluated at the cell faces whereas the dependent variables are stored at cell centers. In order to evaluate the fluxes at the cell faces one can use some kind of interpolation technique to obtain the dependent variables at the cell faces. Roe [23], on the other hand, treated the problem of obtaining the inviscid flux

at a cell face as an approximate Riemann problem. Thus, suppose one is interested in obtaining the inviscid flux F, at the cell face i+1/2. Roe assumed that the solution at the interface is discontinuous and the solution just to the left of the cell face i+1/2 is given by q_i and the solution just to the right of the cell face is given by q_{i+1}, thus setting up a Riemann problem. In this approach, at any instant only one cell face is assumed to exist and the effect of the adjacent cell faces are ignored. Then, Roe showed that the following equation holds for the flux difference across the cell face

$$F_{i+1} - F_i = A(\bar{q})\,(q_{i+1} - q_i) \tag{5.3.1}$$

provided the flux Jacobian A is evaluated with the Roe average variable $\bar{q}$ where $\bar{q}$ is a function of q_i and q_{i+1}. Pan and Chakravarthy [26] have shown that for the form of the incompressible flow equations used in this study the Roe average turns out to be just the simple arithmetic average. Equation (5.3.1) can be used to find the flux at cell center i+1 in terms of the flux at i. However, one is actually interested in the flux at cell face i+1/2. Roe's approach can still be used to find the flux at the cell face i+1/2 and yields the following expression.

$$\hat{F}_{i+1/2} = \left[F(q_i)\right]_{i+1/2} + \sum_{j=1}^{n} \alpha_{j,i+1/2}\, \lambda^{-(j)}_{i+1/2}\, r^{(j)}_{i+1/2} \tag{5.3.2}$$

where, $\alpha_{j,i+1/2} = l^{(j)}_{i+1/2} \cdot (q_{i+1} - q_i)$. $\lambda^{-(j)}$ is the j–th negative eigenvalue of the Roe matrix $A(\bar{q})$, $r^{(j)}$ is the corresponding right eigenvector of the Roe matrix and, $l^{(j)}$ is the corresponding left eigenvector of the Roe matrix. Note that α_j is the corresponding jump in the characteristic variable. The subscript i+1/2 on the right hand side of Eq. (5.3.2) denotes that the metrics used in evaluating the various quantities are evaluated at the cell face i+1/2. In Eq. (5.3.2) $\hat{F}$ denotes the numerical flux at the cell face i+1/2 and $\left[F(q_i)\right]_{i+1/2}$ denotes the actual flux given by Eq. (5.1.2), (5.1.3) or (5.1.4) as the case may be, evaluated using q_i and the metrics at i+1/2. The formulation in Eq. (5.3.2) is not unique and other equivalent formula-

tions can be found in Whitfield, Janus and Simpson [47]. Note that the formulation in Eq. (5.3.2) requires the hyperbolicity of the governing equations. Equation (5.3.2) can also be written as

$$\hat{F}_{i+1/2} = [F(q_i)]_{i+1/2} + T_{i+1/2}\Lambda^-_{i+1/2}T^{-1}_{i+1/2}\Delta q \tag{5.3.3}$$

where T and T^{-1} and Λ^- are defined in the previous section. Note that a clever way of evaluating the second term on the right hand side of Eq. (5.3.3) is also given in the previous section. It can be shown that the formulation in Eqs. (5.3.2) or (5.3.3) is first order accurate in space. The reason for this is that the formulation in Eq. (5.3.2) assumes that the solution variables q's are constant in each cell. Higher order flux formulas are given in Taylor [22] and Taylor and Whitfield [48] that closely follows the formulation of higher order flux formulas for compressible flows. In a departure from this approach, Whitfield and Taylor [28] observe that the flux formulation of the governing equations of incompressible flows do not require limiters and adopt the MUSCL scheme of van Leer [24] for higher order flux formulation. Note that in deriving Eq. (5.3.2) Roe used q_i as the dependent variable to the left of the cell face and q_{i+1} as the the dependent variable to the right of the cell face. Instead of this approach, Anderson, Thomas and van Leer [49] uses the following formulas for left (denoted by $q^L_{i+1/2}$) and right (denoted by $q^R_{i+1/2}$) dependent variables.

$$\begin{aligned} q^L_{i+1/2} &= q_i + \frac{\phi}{4}\left[(1-\kappa)(q_i - q_{i-1}) + (1+\kappa)(q_{i+1} - q_i)\right] \\ q^R_{i+1/2} &= q_{i+1} - \frac{\phi}{4}\left[(1-\kappa)(q_{i+2} - q_{i+1}) + (1+\kappa)(q_{i+1} - q_i)\right] \end{aligned} \tag{5.3.4}$$

The higher order flux is obtained from Eq. (5.3.2) or from Eq. (5.3.3) by replacing q_i by $q^L_{i+1/2}$ and q_{i+1} by $q^R_{i+1/2}$. Thus, when $\phi = 0$ in Eq. (5.3.4) one recovers the first order formulation. So, for higher orders ϕ needs to be taken as unity. With $\kappa = -1$ one obtains a sec-

ond order accurate scheme in which only values to the left of the cell face (i.e. q_i and q_{i-1}) are used to evaluate $q^L_{i+1/2}$ and only values to the right of the cell face (i.e. q_{i+1} and q_{i+2}) are used to evaluate $q^R_{i+1/2}$. With $\kappa = 1/3$ one obtains a third order upwind biased scheme [49] which uses two points to the left and one point to the right of the cell face to obtain $q^L_{i+1/2}$ and two points to the right and one point to the left of the cell face to obtain $q^R_{i+1/2}$.

5.4. Discretized Newton–relaxation scheme.

It was mentioned in Section 5.1 that the implicit discrete equation (5.1.5) needs to be solved in an iterative manner in order to obtain the solution at time level n+1. Equation (5.1.5) is considered as a set of non–linear algebraic equations and Newton's method is employed to find the root of this system. In operator form Eq. (5.1.5) can be written as

$$N(q^{n+1}) = 0 \tag{5.4.1}$$

where

$$\begin{aligned} N(q^{n+1}) = {} & \frac{3\,Q^{n+1} - 4\,Q^n + Q^{n-1}}{2\,\Delta\tau} + \hat{F}^{n+1}_{i+\frac{1}{2}} - \hat{F}^{n+1}_{i-\frac{1}{2}} \\ & + \hat{G}^{n+1}_{j+\frac{1}{2}} - \hat{G}^{n+1}_{j-\frac{1}{2}} + \hat{H}^{n+1}_{k+\frac{1}{2}} - \hat{H}^{n+1}_{k-\frac{1}{2}} \\ & + F^{v\,n+1}_{i+\frac{1}{2}} - F^{v\,n+1}_{i-\frac{1}{2}} + G^{v\,n+1}_{j+\frac{1}{2}} - G^{v\,n+1}_{j-\frac{1}{2}} \\ & + H^{v\,n+1}_{k+\frac{1}{2}} - H^{v\,n+1}_{k-\frac{1}{2}} + M^{n+1} \end{aligned} \tag{5.4.2}$$

In Eq. (5.4.2) the numerical fluxes introduced in Section 5.3 are used for the inviscid fluxes. Note that the function N includes the time derivative. Thus, time accuracy is inherently built into the scheme. This fact is further explained in Whitfield [25]. Newton's method [50] applied to Eq. (5.4.1) results in

$$\left(\frac{\partial N}{\partial q^{n+1}}\right)_{q^{n+1,m}} \cdot \left(q^{n+1,m+1} - q^{n+1,m} \right) = - N(q^{n+1,m}) \qquad (5.4.3)$$

where m = 1,2,3, ... The iteration is started by taking $q^{n+1,1} = q^n$. The generated sequence $q^{n+1,m}$ converges to q^{n+1}, in principle. Thus, Eq. (5.4.1) is satisfied as $m \rightarrow \infty$.

From Eq. (5.4.3) it can be seen that the Jacobian matrix $\left(\frac{\partial N}{\partial q^{n+1}}\right)_{q^{n+1,m}}$, hereafter denoted as $N'(q^{n+1,m})$, is needed in Newton's method. In order to obtain $N'(q^{n+1,m})$ one needs to differentiate each term on the right hand side of Eq. (5.4.2) with respect to the solution vector q^{n+1} and add all the terms together. In order to differentiate the right hand side of Eq. (5.4.2) one needs to know the functional dependence of the various terms on q^{n+1}. Strictly speaking the higher order fluxes at the cell faces depend on two points on either side of the cell face. However, in order to contain the band width of the resulting matrix the approach taken by Whitfield [25] was to evaluate the Jacobian matrix $N'(q^{n+1,m})$ assuming that the fluxes in Eq. (5.4.2) are given by first order formulae and to obtain the residual on the right hand side in Eq. (5.4.3) using higher order fluxes. However, it can be seen from Eq. (5.3.3) that, for the MUSCL scheme, the same equation is used to obtain either the first order fluxes or the higher order fluxes depending upon what is substituted for the dependent variables (q's). Referring to Eq. (5.3.4), Whitfield [25] used $\phi = 0$ to evaluate the Jacobian matrix $N'(q^{n+1,m})$ on the left hand side of Eq. (5.4.3) and $\phi = 1$ to evaluate the residual $N(q^{n+1,m})$ on the right hand side. Thus, Whitfield [25] considered the flux $\hat{F}^{n+1}_{i+1/2}$ at the cell face i+1/2 to be functions of q_i^{n+1} and q_{i+1}^{n+1} alone which results in

$$\left[\frac{\partial \hat{F}_{i+1/2}^{n+1}}{\partial q^{n+1}}\right]_{q^{n+1,m}} \cdot \left(q^{n+1,m+1} - q^{n+1,m}\right) = \left[\frac{\partial \hat{F}_{i+1/2}^{n+1}}{\partial q_i^{n+1}}\right]_{q_i^{n+1,m}} \cdot \left(q_i{}^{n+1,m+1} - q_i^{n+1,m}\right)$$

$$+ \left[\frac{\partial \hat{F}_{i+1/2}^{n+1}}{\partial q_{i+1}^{n+1}}\right]_{q_{i+1}^{n+1,m}} \cdot \left(q_{i+1}{}^{n+1,m+1} - q_{i+1}^{n+1,m}\right)$$

$$= D\hat{F}_{i+1/2,i}\,\Delta q_i^{n+1,m} + D\hat{F}_{i+1/2,i+1}\,\Delta q_{i+1}^{n+1,m} \qquad (5.4.4)$$

Note that the left hand side of the first equality in Eq. (5.4.4) does not contain any spatial location. Thus, it is in functional form. In Eq. (5.4.4),

$$D\hat{F}_{i+1/2,i} = \left[\frac{\partial \hat{F}_{i+1/2}^{n+1}}{\partial q_i^{n+1}}\right]_{q_i^{n+1,m}} \qquad (5.4.5)$$

and

$$\Delta q_i^{n+1,m} = q_i{}^{n+1,m+1} - q_i^{n+1,m}$$

In a similar manner the other flux terms in Eq. (5.4.2) can be differentiated. The evaluation of the flux Jacobian, defined in Eq. (5.4.5), is discussed later. The first subscript on the left hand side of Eq. (5.4.5) indicates the location where the metrics are evaluated and the second subscript indicates the location of the solution vector.

Whitfield and Taylor [28], on the other hand, consider the numerical flux to be dependent on $q_{i+1/2}^{L}$ and $q_{i+1/2}^{R}$ thereby using $\phi = 1$ (see Eq. (5.3.4)) for the evaluation of the Jacobian matrix $N'(q^{n+1,m})$. Using this approach one obtains

$$\left[\frac{\partial \hat{F}_{i+1/2}^{n+1}}{\partial q^{n+1}}\right]_{q^{n+1,m}} \cdot \left(q^{n+1,m+1} - q^{n+1,m}\right)$$

$$= D\hat{F}_{i+1/2,L}\,\Delta q_i^{n+1,m} + D\hat{F}_{i+1/2,R}\,\Delta q_{i+1}^{n+1,m} \qquad (5.4.6)$$

The difference between Eqs. (5.4.4) and (5.4.6) can be seen in the subscripts of the flux Jacobians. After substituting the various quantities, Eq. (5.4.3) becomes

$$
\begin{aligned}
&- D\hat{F}_{i-1/2,L}\,\Delta q_{i-1}^{n+1,m} - D\hat{G}_{j-1/2,L}\,\Delta q_{j-1}^{n+1,m} - D\hat{H}_{k-1/2,L}\,\Delta q_{k-1}^{n+1,m} \\
&+ \left(\frac{3\,I}{2} + D\hat{F}_{i+1/2,L} + D\hat{G}_{j+1/2,L} + D\hat{H}_{k+1/2,L}\right. \\
&\left. - D\hat{F}_{i-1/2,R} - D\hat{G}_{j-1/2,R} - D\hat{H}_{k-1/2,R}\right)\Delta q_{i}^{n+1,m} \\
&+ D\hat{F}_{i+1/2,R}\,\Delta q_{i+1}^{n+1,m} + D\hat{G}_{j+1/2,R}\,\Delta q_{j+1}^{n+1,m} + D\hat{H}_{k+1/2,R}\,\Delta q_{k+1}^{n+1,m} \\
&= -\left[\frac{3\,Q^{n+1,m} - 4\,Q^{n} + Q^{n-1}}{2\,\Delta\tau}\,I_a + \hat{F}_{i+\frac{1}{2}}^{n} - \hat{F}_{i-\frac{1}{2}}^{n} + \hat{G}_{j+\frac{1}{2}}^{n} - \hat{G}_{j-\frac{1}{2}}^{n}\right. \\
&\quad + \hat{H}_{k+\frac{1}{2}}^{n} - \hat{H}_{k-\frac{1}{2}}^{n} + F^{vn}_{i+\frac{1}{2}} - F^{vn}_{i-\frac{1}{2}} + G^{vn}_{j+\frac{1}{2}} - G^{vn}_{j-\frac{1}{2}} \\
&\quad \left. + H^{vn}_{k+\frac{1}{2}} - H^{vn}_{k-\frac{1}{2}} + M^{n}\right] \qquad (5.4.7)
\end{aligned}
$$

The matrix $I_a = \mathrm{diag}(0,\ 1,\ 1,\ 1)$ multiplying the time derivative on the right hand side of Eq. (5.4.7) is a conditioning matrix used to introduce time accuracy into the scheme [26]. Note that only the inviscid flux Jacobians are shown in Eq. (5.4.7). In an analogous manner the viscous flux Jacobians can be obtained using the thin layer approximation and added to the left hand side of (5.4.7). When this is done it can be seen that for every inviscid term on the left hand side of Eq. (5.4.7) there is a corresponding viscous term. The Jacobian of the source term can be obtained using the Boussinesq's approximation and when added to the left hand side of Eq. (5.4.7) it results in the strengthening of the block diagonal matrix since the source term is only a function of $q_{i,\,j,\,k}$.

Analytical derivation of the flux Jacobian matrices, viz. $D\hat{F}_{i+1/2,L}$ etc. is not straight forward. Hence, they are obtained numerically in this work. In order to obtain the derivative of the m–th element of the vector $\hat{F}_{i+1/2}$ with respect to the n–th element of the vector $q^L_{i+1/2}$ one proceeds as follows:

$$\left(D\hat{F}_{i+1/2,L}\right)_{m,n} = \frac{\hat{F}_m\left(q^L_{i+1/2} + he_n\right) - \hat{F}_m\left(q^L_{i+1/2}\right)}{h} \tag{5.4.8}$$

where e_n is the n–th unit vector and

$$h \approx \sqrt{machine\ \epsilon} \tag{5.4.9}$$

It should be mentioned that the metrics used to evaluate the right hand side of Eq. (5.4.8) are evaluated at the cell face i+1/2. In a similar manner all other flux Jacobians appearing in Eq. (5.4.7) are obtained including the viscous Jacobians which are not shown. Equation (5.4.7) with the Jacobians obtained as in Eq. (5.4.8) is called the discrete Newton's method (see Ortega and Rheinboldt [50]).

The method used in this work to solve Eq. (5.4.7), for a fixed value of m, is the relaxation technique which is described by Whitfield and Taylor [28]. It follows the method proposed by Chakravarthy [51] and uses the symmetric block Gauss–Seidel iterative method found in Hageman and Young [52].

In a global sense, Eq, (5.4.7) can be written as

$$(L + B + U)\,x = b \tag{5.4.10}$$

where L is a lower block triangular matrix with zeros on the diagonal which is composed of the first three terms on the left hand side of Eq. (5.4.7), B is a block diagonal matrix which is composed of the terms multiplying $\Delta q_i^{n+1,m}$ on the left hand side of Eq. (5.4.7), U is an upper block triangular matrix with zeros on the diagonal which is composed of the last three terms

on the left hand side of Eq. (5.4.7), x is the global solution increment vector ($\Delta q's$) and, b is the global residual vector containing the right hand side of Eq. (5.4.7).

In the symmetric block Gauss–Seidel iterative method, for each iteration one employs a global forward sweep which is followed by a global backward sweep (hence the name symmetric). Thus, in p iterations there are a total of 2p sweeps. The forward and backward sweeps can be described as follows:

Forward sweep:

$$B\, x^{(2p-1)} = b - U\, x^{(2p-2)} - L\, x^{(2p-1)} \tag{5.4.11}$$

Backward sweep:

$$B\, x^{(2p)} = b - L\, x^{(2p-1)} - U\, x^{(2p)} \tag{5.4.12}$$

where p = 1, 2, 3 ... . The iterative process is started with the forward sweep with p = 1 and $x^{(0)} = 0$. Suppose one is at a point whose indices are i, j, k during the forward sweep. Since one is in the forward sweep, the solution at level (2p–1) has already been computed at all points whose vertices are either less than i or less than j or less than k. In other words the solution at level (2p–1) is known at all points corresponding to the lower block triangular matrix L. This is the reason the vector $L\, x^{(2p-1)}$ is moved to the right hand side of Eq. (5.4.11). Thus, in order to find the solution at level (2p–1) at the point (i,j,k) all one needs to do is to multiply the right hand side of Eq. (5.4.11) by the inverse of B. Here it is understood that the right hand side of Eq. (5.4.11) and the matrix B should correspond to the point (i,j,k). Note that B is either a 4x4 matrix or a 6x6 matrix depending upon the equation set one is solving. The solution of Eq. (5.4.11) is obtained by Dolittle's method. For the backward sweep the solution is known at points corresponding to the upper block triangular matrix U and Dolittle's method is again used for solving (5.4.12). Note that Dolittle's method is a direct method and is a compact scheme for Gaussian elimination [53]. For further details, see Whitfield and Taylor [28]. Thus, for each value of m+1 the corresponding solution vector $q^{n+1,m+1}$ of Eq. (5.4.7)

is obtained through an iterative process. The resulting numerical scheme is called Discretized Newton–relaxation (DNR) scheme (see [25], [27] and [28]).

5.5. Boundary conditions.

The description of a numerical scheme is incomplete without a discussion of the implementation of the boundary conditions. For oceanographic problems one needs to specify inflow, outflow, viscous wall and free surface boundary conditions. For cases involving temperature and salinity, in addition to velocity conditions (no slip or free surface), one needs to specify either adiabatic or source conditions for temperature and salinity. Since the present method employs boundary conforming coordinates, the boundary conditions are specified on k = constant surfaces where k could be either ξ, η or ς. For the specification of inflow and outflow conditions, characteristic variables are employed as described by Whitfield and Janus [54] and are derived from the Euler equations as follows:

Neglecting every term other than the terms involving the time derivative and the derivative with respect to k, the inviscid portion of Eq. (5.1.1) can be written in quasi–linear form as follows:

$$\frac{\partial Q}{\partial \tau} + \overline{K}\frac{\partial Q}{\partial k} = 0 \tag{5.5.1}$$

where $\overline{K} = \frac{\partial F}{\partial Q}$ if $k = \xi$; $\overline{K} = \frac{\partial G}{\partial Q}$ if $k = \eta$ and $\overline{K} = \frac{\partial H}{\partial Q}$ if $k = \varsigma$. Note that $A = \frac{\partial F}{\partial q} = \frac{\partial F}{\partial Q}\frac{\partial Q}{\partial q} = \sqrt{g}\,\overline{A}$ and so on. In other words, $K = \sqrt{g}\,\overline{K}$. Note that, $\sqrt{g}$ is the Jacobian of the coordinate transformation and is a scalar. It can be shown following the description in Section 5.2 that the eigenvalues of $\overline{K}$ are given by $\Lambda = \sqrt{g}\,\overline{\Lambda}$ where $\overline{\Lambda}$ is the diagonal matrix containing the eigenvalues of $\overline{K}$ and Λ is the diagonal matrix containing the eigenvalues

of K. Thus, one can choose the left and right eigenvectors of $\overline{K}$ to be the same as that of K. Therefore,

$$\overline{K} = T_k \overline{\Lambda} T_k^{-1} \tag{5.5.2}$$

and

$$K = \sqrt{g}\,\overline{K} = \sqrt{g}\,T_k \overline{\Lambda} T_k^{-1} = T_k (\sqrt{g}\,\overline{\Lambda}) T_k^{-1} = T_k \Lambda T_k^{-1}$$

Substituting Eq. (5.5.2) in Eq. (5.5.1) and re–arranging assuming the matrices T_k and T_k^{-1} are constant matrices, one gets

$$\frac{\partial W_k}{\partial \tau} + \overline{\Lambda}\frac{\partial W_k}{\partial k} = 0 \tag{5.5.3}$$

where $W_k = T_{k,0}^{-1} Q$. The subscript '0' in $T_{k,0}^{-1}$ denotes that it is a constant matrix. Elements of the vector W_k are called the characteristic variables. Depending upon the sign of the corresponding eigenvalue, the characteristic variables are either prescribed or extrapolated from the solution domain in a consistent manner. This procedure is described for the various types of boundary conditions below.

Inflow boundary:

An inflow boundary, k = constant, is called a codirectional inflow boundary if the flow entering the boundary does so in the direction of increasing k. Otherwise, if the flow enters in the direction of decreasing k, it is called a contradirectional inflow boundary. The terminology of codirectional and contradirectional boundaries was first used by Janus [55]. In the case of codirectional inflow boundary, referring to Section 5.2 (Eqs. (5.2.3), (5.2.4) and (5.2.5)), it can be seen that only one eigenvalue is negative and the rest are positive. The characteristic lines corresponding to the positive eigenvalues run from outside the computational domain towards the boundary and the corresponding characteristic variables are constant along these lines. Similarly, the characteristic line corresponding to the negative eigenvalue runs from inside the

computational domain towards the boundary and the corresponding characteristic variable is constant along this line. Thus, the characteristic variables corresponding to the positive eigenvalues need to be prescribed and the one corresponding to the negative eigenvalue needs to be extrapolated from the solution domain. Thus, for the absolute velocity procedure one has

$$\left.\begin{aligned} \left(w_{k,1}\right)_b &= \left(w_{k,1}\right)_a \\ \left(w_{k,2}\right)_b &= \left(w_{k,2}\right)_a \\ \left(w_{k,3}\right)_b &= \left(w_{k,3}\right)_a \\ \left(w_{k,4}\right)_b &= \left(w_{k,4}\right)_l \end{aligned}\right\} \qquad (5.5.4)$$

In Eq. (5.5.4) the first three equations correspond to the positive eigenvalues and the last one corresponds to the negative eigenvalue (see Eq. (5.2.3)). The subscript 'b' represents the boundary, 'a' represents the approaching direction (outside the computational domain) and 'l' represents the leaving direction (inside the computational domain). These are marked in

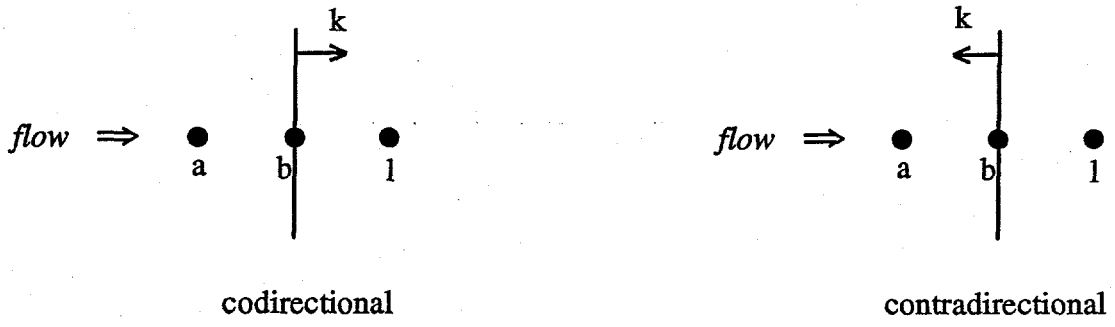


Fig. 3. Schematic for Inflow / Outflow Boundary Condition

Fig. 3. In Eq. (5.5.4) the four solution variables at the boundary (p_b, u_b, v_b and w_b) are the only unknowns which can be obtained by simultaneously solving the set of algebraic equations. For contradirectional inflow boundaries one has three negative eigenvalues and one positive eigenvalue. Thus, one has

$$\left.\begin{aligned} \left(w_{k,1}\right)_b &= \left(w_{k,1}\right)_a \\ \left(w_{k,2}\right)_b &= \left(w_{k,2}\right)_a \\ \left(w_{k,3}\right)_b &= \left(w_{k,3}\right)_1 \\ \left(w_{k,4}\right)_b &= \left(w_{k,4}\right)_a \end{aligned}\right\} \qquad (5.5.5)$$

In Eq. (5.5.5) note that the subscripts on the right hand side of the last two equations differ from that of Eq. (5.5.4).

Outflow boundary:

The development of the outflow boundary conditions is very similar to the inflow boundary conditions. In this case also one has codirectional and contradirectional boundaries. For an outflow boundary, the approaching direction is from within the computational domain and the leaving direction is towards the exterior of the computational domain (see Fig. 3). For co-directional outflow one has three positive eigenvalues and one negative eigenvalue. The characteristic variables corresponding to the positive eigenvalues are extrapolated from inside the computational domain and instead of specifying the characteristic variable corresponding to the negative eigenvalue one usually specifies the pressure. The remaining three equations are solved for the velocity components. Thus, for a codirectional outflow boundary one has

$$\left.\begin{aligned} \left(w_{k,1}\right)_b &= \left(w_{k,1}\right)_a \\ \left(w_{k,2}\right)_b &= \left(w_{k,2}\right)_a \\ \left(w_{k,3}\right)_b &= \left(w_{k,3}\right)_a \\ p_b &= p_1 \end{aligned}\right\} \qquad (5.5.6)$$

and for a contradirectional boundary one has

$$\left.\begin{aligned} \left(w_{k,1}\right)_b &= \left(w_{k,1}\right)_a \\ \left(w_{k,2}\right)_b &= \left(w_{k,2}\right)_a \\ p_b &= p_1 \\ \left(w_{k,4}\right)_b &= \left(w_{k,4}\right)_a \end{aligned}\right\} \qquad (5.5.7)$$

Viscous wall:

The no slip condition at a viscous wall (ocean floor for example) implies that the fluid is at rest in the rotating frame. Thus, if the governing equations are solved using relative velocity components (relative velocity procedure) then they are set equal to zero at a viscous wall. On the other hand, if the governing equations are solved using the absolute velocity components then from Eq. (4.2.5) one obtains $\underline{u} = \underline{v} - \underline{w} = -\underline{w} = \underline{\Omega} \times \underline{r}$, since $\underline{v} = 0$. Thus, the components of the absolute velocity are set equal to the Cartesian components of the vector $\underline{\Omega} \times \underline{r}$. The condition for pressure is vanishing normal gradient at the wall. For a η = constant wall this condition is approximately implemented as $p_\eta = 0$ where the suffix denotes partial differentiation. Rigorously, vanishing normal gradient condition at a η = constant wall is implemented as follows:

$$\frac{\partial p}{\partial n} = 0$$

$$\Rightarrow \underline{n} \cdot \nabla p = 0$$

$$\Rightarrow \underline{a}^2 \cdot \frac{\partial p}{\partial x^k}\,\underline{a}^k = 0$$

$$\Rightarrow g^{12}\frac{\partial p}{\partial \xi} + g^{22}\frac{\partial p}{\partial \eta} + g^{23}\frac{\partial p}{\partial \varsigma} = 0$$

$$\Rightarrow \frac{\partial p}{\partial \eta} = -\frac{1}{g^{22}}\left(g^{12}\frac{\partial p}{\partial \xi} + g^{23}\frac{\partial p}{\partial \varsigma} \right)$$

where n is the normal direction, $\underline{a}^k$, k=1, 2, 3; are the contravariant base vectors, g^{12}, g^{22} and g^{23} are components of the contravariant metric tensor. The adiabatic condition for temperature and zero normal gradient condition for salinity are imposed analogously.

Free surface boundary:

The velocity and pressure conditions have already been discussed in Section 4.3 and their implementation is straightforward. For prescribed heat and salt flux one can follow the approach described above for the viscous wall case and obtain

$$\frac{\partial T}{\partial \eta} = -\frac{1}{g^{22}}\left(\frac{Q}{\kappa} + g^{12}\frac{\partial T}{\partial \xi} + g^{23}\frac{\partial T}{\partial \varsigma} \right)$$

where Q is the prescribed heat flux and κ is the thermal conductivity. A similar expression can be obtained for the η–derivative of salinity that involves the salt flux.

6. Grid Generation

In this work the governing equations of the ocean flows are solved in a curvilinear coordinate system generated around the earth. The results of ocean flows presented in this report uses the rigid lid condition though this is not an inherent restriction of the solution methodology as well as the grid generation methodology. Example calculations of free surface flows for other geometries have been included using the same solution and grid generation methodologies. The calculation of ocean flows with free surfaces is currently underway.

Basic ideas about generating general curvilinear grids over complicated geometries are dealt with in detail in the book by Thompson, Warsi and Mastin [13]. Based on the work of Thompson and his coworkers the original EAGLE code was written. Later a graphical user interface was added to EAGLE and the resulting software is called EAGLEView [14]. Jiang [56] used EAGLEView to generate the grids used in this work.

For generating the ocean grids Jiang used the ETOPO5 dataset. Note that the format used to write ETOPO5 dataset traverses the earth in latitudinal circles from the north pole to the south pole with an increment of 0.2 degrees. Along each latitudinal circle the depth is written for each longitude in 0.2 degrees increment. Depths are recorded as positive values. If a point is in the land mass then the height of that point above sea level is recorded as a negative value. Both the depth and height are given in feet. The original code supplied for reading this dataset was modified to write the output in terms of x, y and z values with respect to a Cartesian coordinate system with its origin at the center of the earth. One prescribes a range for longitude and latitude corresponding to the region of one's interest as inputs to this code. This code works as follows. Within the prescribed range, once a land point is encountered its height is reassigned to a constant value. Small islands are manipulated in the following way. Suppose one is at a point (i, j) where i corresponds to the longitudinal direction and j corresponds to the latitudinal direction. If this point is a land point and its neighbors (i+1, j), (i–1, j), (i, j+1) and (i, j–1) are

ocean points then one is dealing with a one point island. If the island is kept then one has to construct coordinate lines around it, and so it gets sunk to the average depth of its four neighbors. Similarly, along the coasts if three of the four neighboring points of a land point is water then that point is sunk and the average depth of its neighbors is assigned to it. In addition, if

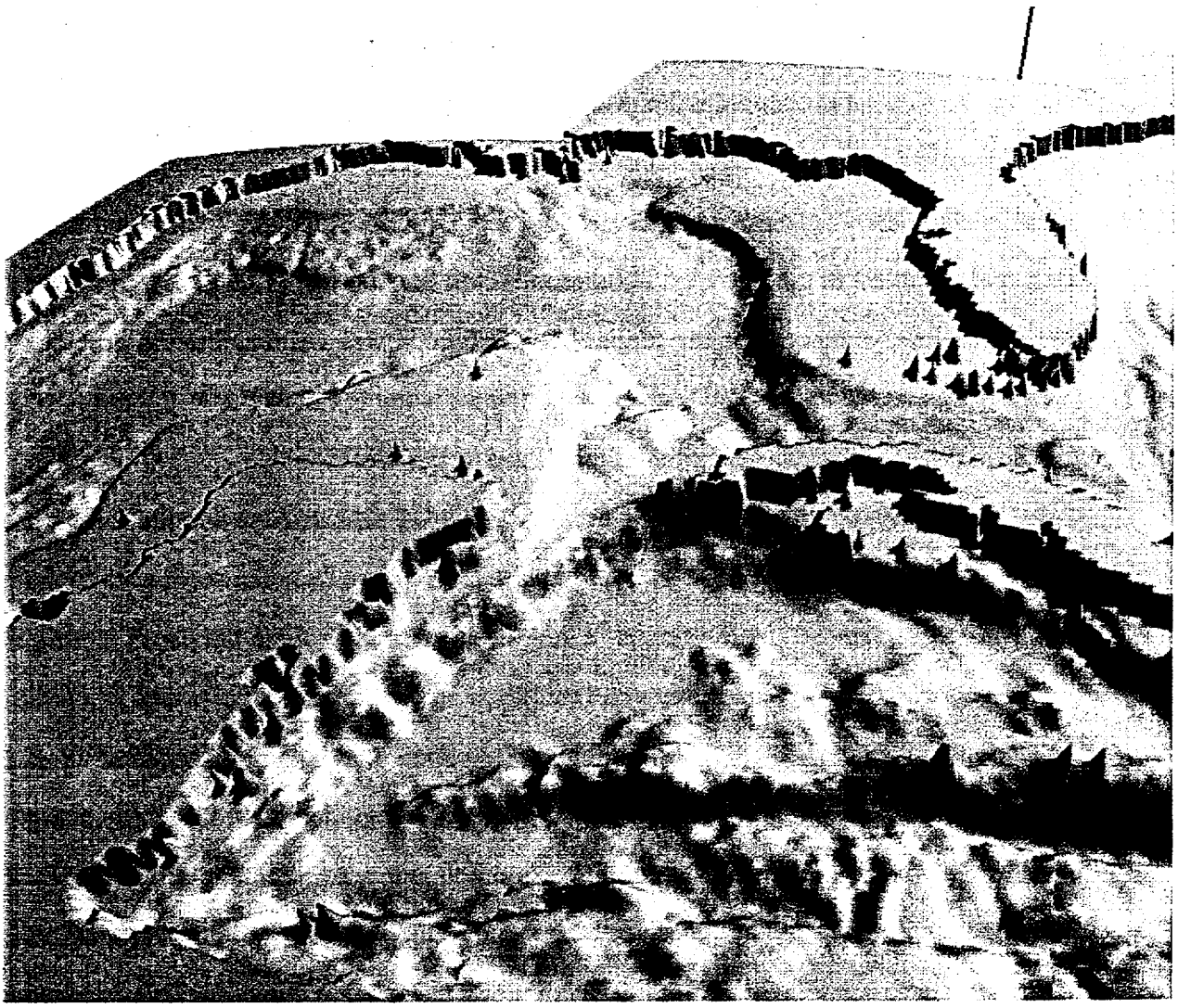


Fig. 4. Depth Enhanced View of the Gulf of Mexico

points (i, j) and (i+1, j) are land points and (i–1, j), (i, j+1), (i+1, j+1) and (i+2, j) are water points then the land points are sunk and replaced with the average depth of the above mentioned neighbors. In other words peninsulas with two point width (0.2 degrees) are sunk. Also

sunk are peninsulas with three point width (0.4 degrees). Similarly one point lakes, two point bays and three point bays are filled with land. Once this process is done, two output files are written out. One file has the constant land heights and the actual ocean floor data in terms of x,

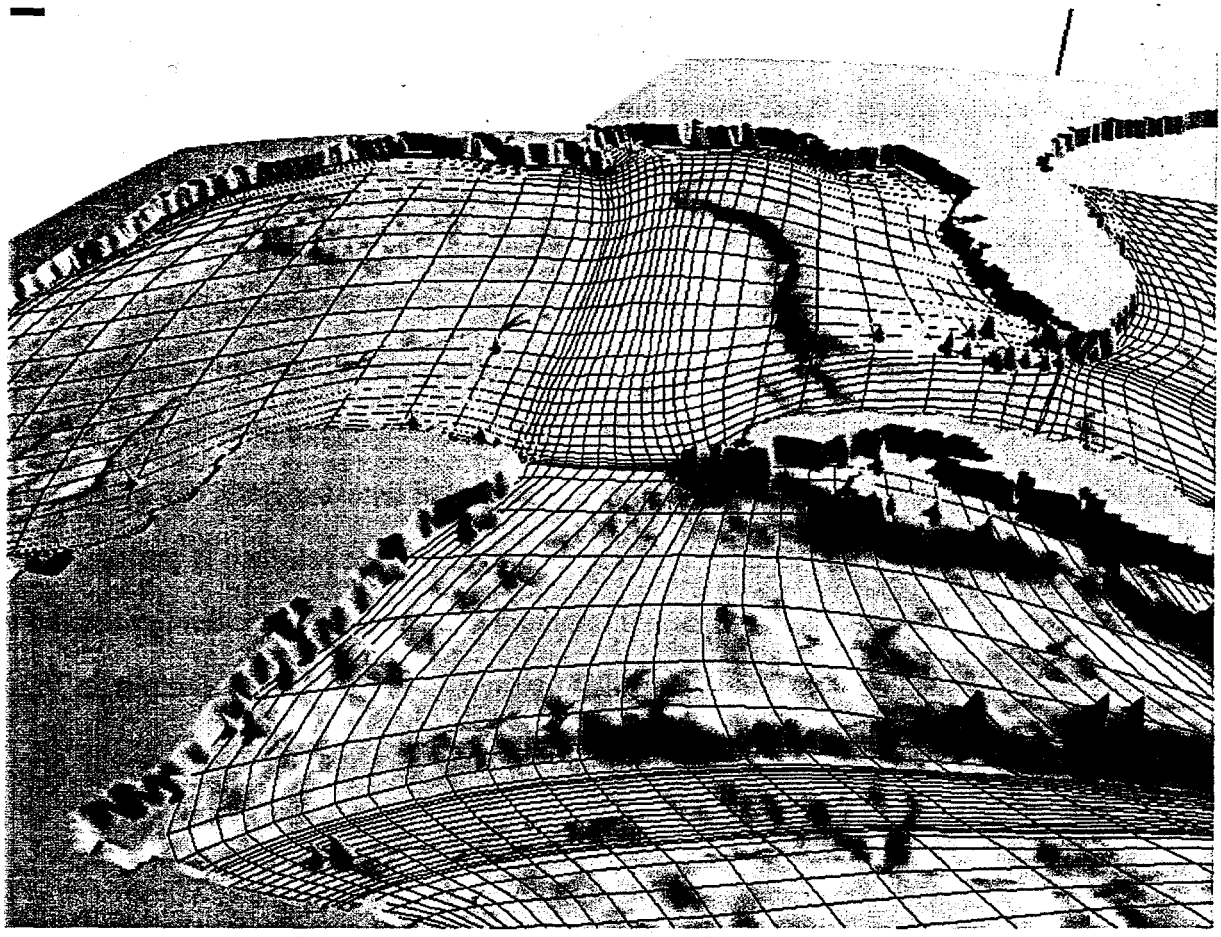


Fig. 5. Actual Bottom Surface Grid of the Gulf of Mexico

y and z values and the other file has the constant land height values and the ocean depth at every point increased by a constant value (say 5000 feet). The land points are assigned a constant value so that one can easily recognize the land area when the data is read into EA-GLEView. The file with enhanced ocean depth is used to view the ocean floor and the coastal area clearly. As an example, the Gulf of Mexico with a part of the Atlantic ocean is shown in Fig. 4 in which the land (especially the islands) and coastal areas and the ocean floor are all

clearly and distinctly visible. The other file is used to actually build the grid. It can be seen from Fig. 4 that the coastal lines are still not smooth. In order to smooth the coastal lines without destroying the shape of the geometry too much, points are manually picked along the coastlines using the depth enhanced file. When this EAGLEView script file is written out and reread with the actual surface file the points along the actual coastlines would have been picked up automatically. These points are connected using cubic splines and thus a smooth coastal line is created. Creation of smooth coastlines is the major task in the present method. Once this is done the next task is to decide how to block the region so that it does not result in highly skewed grid lines. Experience plays a major role in this decision making process. Once the block boundaries are identified, points are distributed along them and are interconnected to create a grid surface. Since only the points along the boundaries are used this surface will not lie on the ETOPO5 dataset. Hence, this grid is projected onto the ETOPO5 dataset radially thus generating the bottom surface. In the next step this bottom surface is projected radially outwards onto a sphere of radius one. All the intermediate surfaces are then generated using interpolation. The surface grid on the actual ocean floor for the region shown in Fig. 4 is shown in Fig 5.

7. Results

Two sets of results are presented here with different objectives. In Section 7.1 the test cases used to validate the code are dealt with. These include laminar flat plate results, laminar backward step solutions, driven cavity with buoyancy results and Ekman layer solutions. In addition, example calculations of free surface flows in an inertial frame around ship hulls are also included in this section. In section 7.2 results for ocean computations are presented. These include turbulent Ekman layer calculations, flow field in the Atlantic ocean and flow field in the entire world ocean. These results include only the hydrodynamic phenomena. The calculations with the thermodynamic variables are in progress.

7.1 Validation comparisons.

Classical boundary layer theory says that for the flow over a flat plate with a Prandtl number of unity and with adiabatic wall conditions, the temperature profile for the thermal boundary layer would be the same as the velocity profile for the velocity boundary layer. Since the equations for temperature and salinity are similar, except for the parameters, setting $\frac{\theta}{Sc} = \frac{\kappa}{Pe}$ and using the condition $\frac{\partial S}{\partial n} = 0$ at the wall, should result in a salinity profile which is also the same as the velocity profile. For laminar flows the velocity profile over the flat plate is known as the Blasius profile. The above observations can be seen to be correct from Fig. 6 in which the computed velocity, temperature and salinity profiles are plotted against the Blasius profile. Another interesting point to note from Fig. 6 is that there are only a few points in the viscous region and this is sufficient to resolve the Blasius profile. This is a consequence of the method used to obtain the numerical flux at cell faces.

Flow over a backward facing step has been an interesting problem that has received considerable attention for more than a decade now. The flow exhibits complex behaviors such as

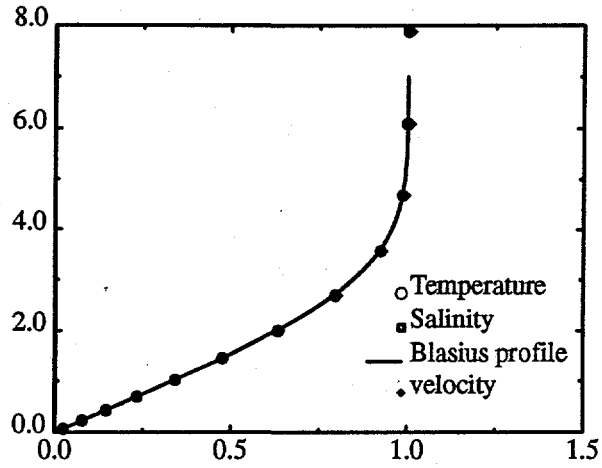
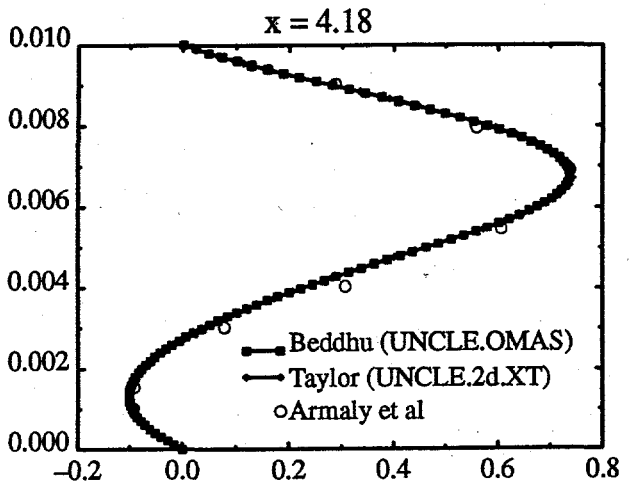
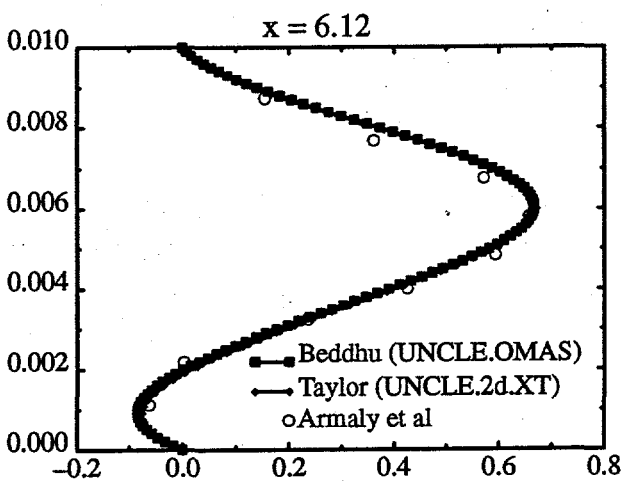


Fig. 6. Flow over a Flat Plate. Re = 10000, Pr =1.0

separation and reattachment and the geometry is simple. Armaly et al. [57] have done extensive experimental work for both laminar and turbulent flows over backward facing step. Usually the laminar results are used for code validation purposes and the turbulent results are used for validating turbulence models. However, the experimental work of Armaly et al. did not include any thermal effects. Rhodes and Acharya [58] have computed the laminar flow over a backward facing step with forced convection. They prescribed a parabolic profile for velocity and set the non–dimensional temperature to be zero as the inlet conditions. These conditions were prescribed at the mouth of the step. The temperature at the bottom wall was set to be unity and the adiabatic wall condition was used at the top wall. The temperature at the vertical wall section below the mouth of the step was set at zero. The flow Reynolds number was taken as 389 and the Prandtl number was taken as 0.71 which corresponds to air. Based on step height and maximum inlet velocity the Reynolds number was taken as 275 in the present study. This corresponds to the case of Re = 389 by Armaly et al. [57] based on their non–dimensionalization. A grid of 201x101 points in the axial and normal directions respectively has been chosen for the geometry described in Rhodes and Acharya. With a CFL number of 50 the residual drops to 10^{-10} in 1500 iterations. The converged velocity and temperature profiles are shown in Figs. 7 and 8. Using a different eigensystem compared to that prescribed in





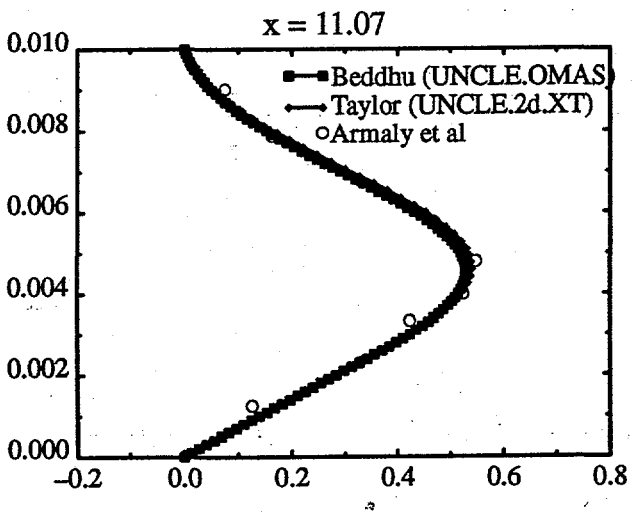
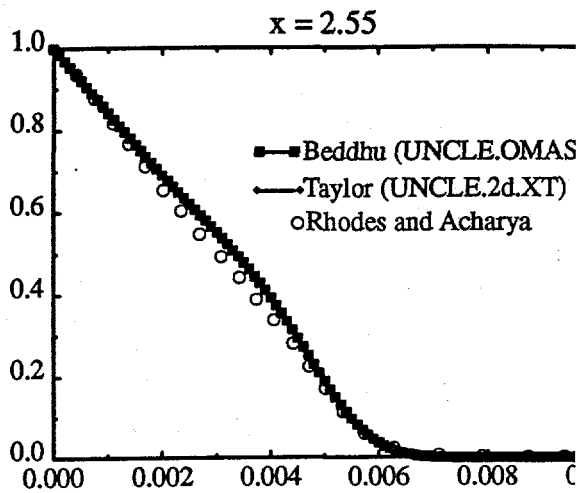
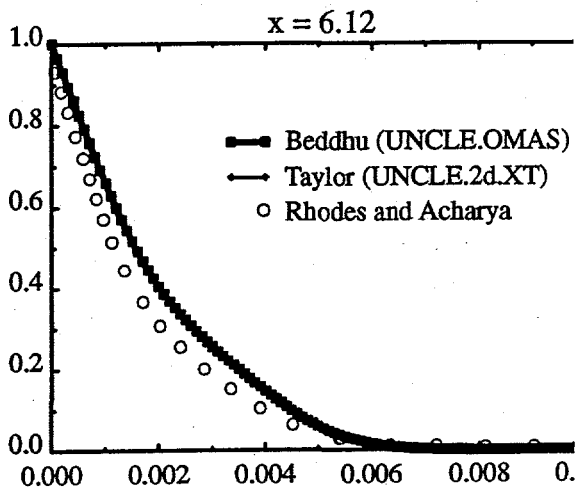


Fig. 7. Flow over a Backward Facing Step.





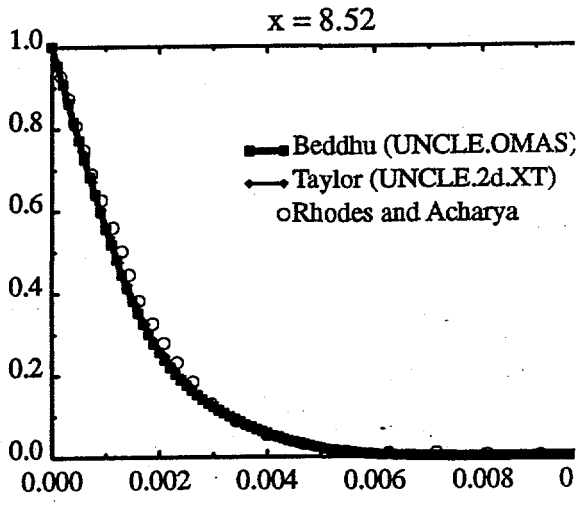


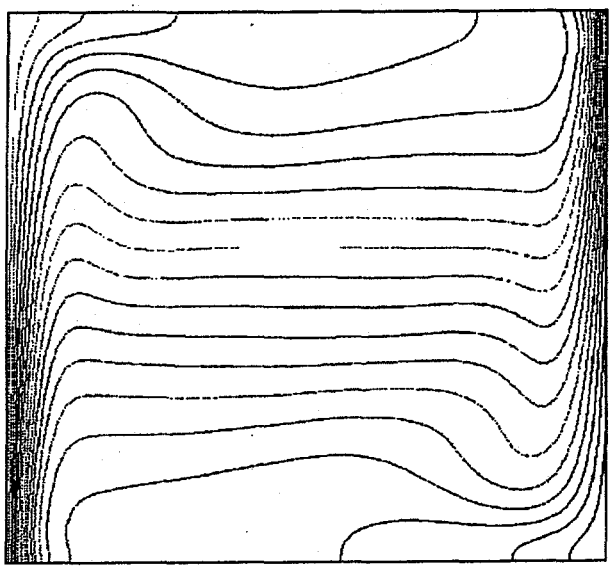
Fig. 8. Flow over a Backward Facing Ste

Beddhu et al [16], Taylor [59] has developed a two–dimensional flow code that includes temperature. The agreement between Taylor and Armaly et al., for the velocity profiles is excellent, whereas there is very good agreement between the present scheme and that of Taylor [59] for the temperature. Both schemes differ from that of Rhodes and Acharya [58]. Taylor made his computations on a 101x51, uniform grid. The present computations have used a grid with stretching from the mouth of the step and from both the upper and lower walls. Rhodes and Acharya used an adaptive grid strategy with a very coarse grid (47x38) and that could be the reason for the discrepancies seen in the temperature profiles.

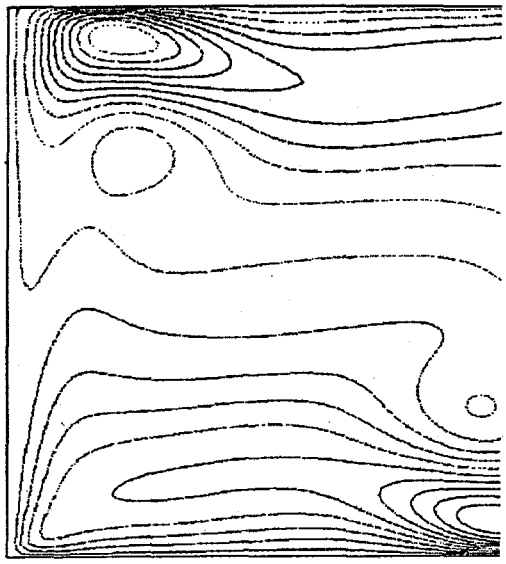
Natural convection in a square cavity is another important test case that has been used extensively for validating codes for which bench marking results are available. Based on the methodology presented in Beddhu et al. [19], Siong [36] has developed an algorithm for solving the five governing equations without the salinity equation. Computation of this test case using the algorithm developed by Siong for various Raleigh numbers has resulted in excellent agreement with the benchmarking results [60]. Figure 9 indicates the results for a Raleigh number of 10^6. Further details can be obtained from Siong [36].

So far, all the test cases mentioned are computed with no rotation. As can be seen they have been selected to specifically test various aspects of the scheme. In the following, test results are included that includes rotation but excludes temperature and salinity at the present time.

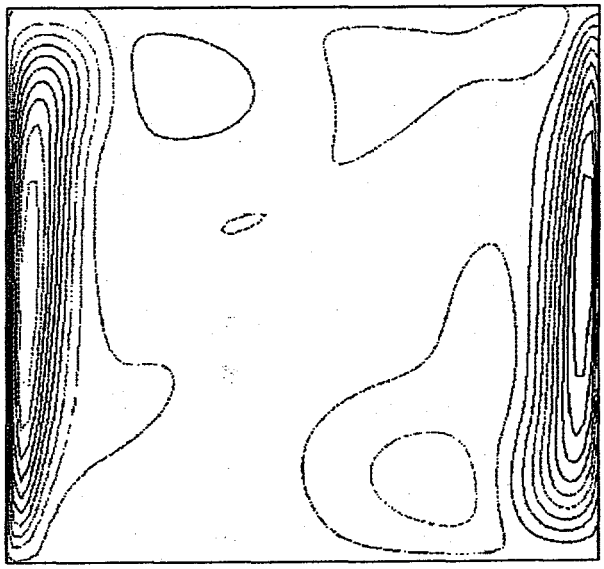
Ekman boundary layer profiles, for the governing equations of the geophysical flows, are the classical counterpart of Blasius boundary layer profiles for the Navier–Stokes equations in an absolute frame. Under suitable simplifying assumptions, closed form analytical expressions, the Ekman boundary layer profiles, can be obtained for a geostrophic flow over a flat viscous surface where the viscous surface can either be a solid wall or a free surface. A detailed discussion of the governing equations and their solutions is available in Pedlosky [7]. A schematic diagram of the physical domain is given in Fig. 10. A Cartesian coordinate system



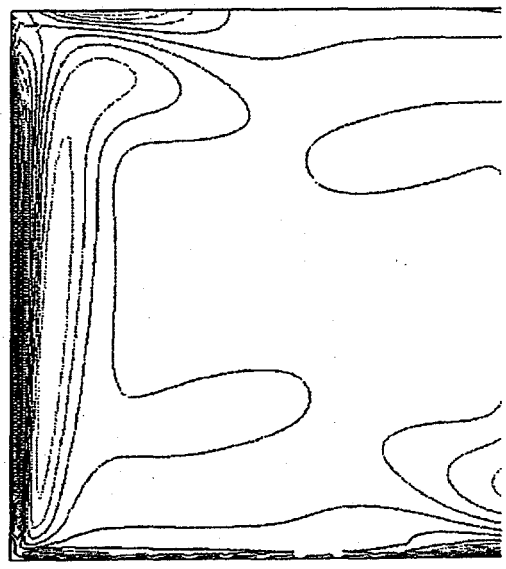
Temperature



u–component



v–component



vorticity

Fig. 9. Natural Convection in a Square Cavity
$Ra = 10^6$

xyz is introduced in a rotating frame such that the y–axis coincides with the axis of rotation. The Reynolds number was chosen to be 10000. The point distribution is 51, 101 and 2 in the

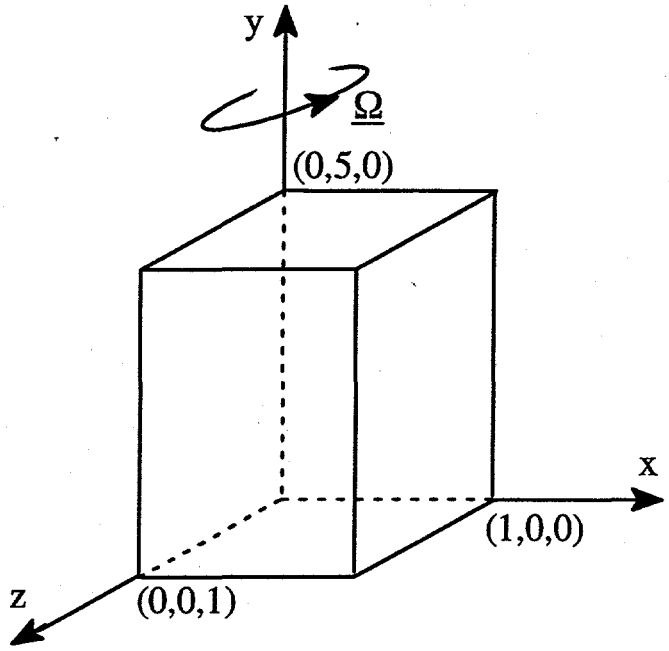


Fig. 10. Physical Domain (not to Scale)

i, j and k direction respectively. At all side boundaries extrapolation conditions were used and at the bottom boundary a geostrophic flow is imposed in the z (or k) direction, with the following values for the non–dimensional quantities: u = 0, v = 0, w = 1 and $\frac{\partial p}{\partial x} = -2$. The top wall is treated either as a viscous wall or as a free surface with a applied wind stress in the x (or i) direction. Starting from an initial condition of a geostrophic flow everywhere, converged solutions are obtained in 2000 cycles. The time step used is 0.05. Excellent agreement is seen between the computed and analytical results in Fig. 11 for the case of no–slip wall, and, Fig. 12 for the case of applied shear stress at the free surface. As stated in Beddhu, Taylor and Whitfield [16] these test cases were used to confirm the new formulation for the Coriolis force. Further details can be obtained from Beddhu, Taylor and Whitfield [16].

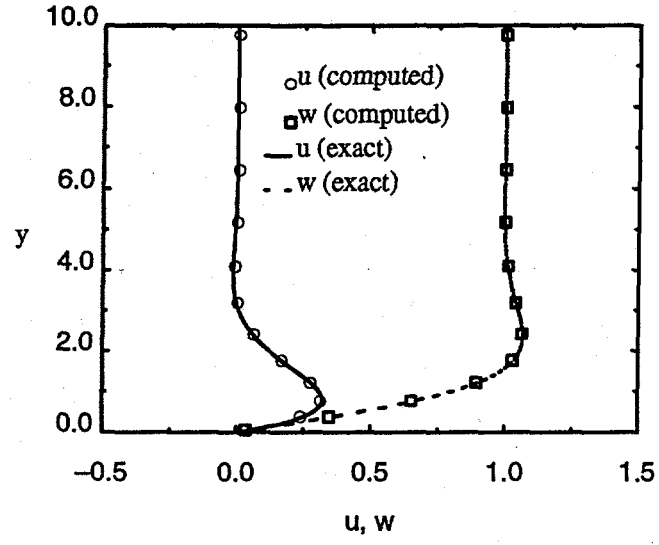


Fig. 11. Ekman Boundary Layer with a No Slip Wall

Re = 10000; Rotation about y – axis

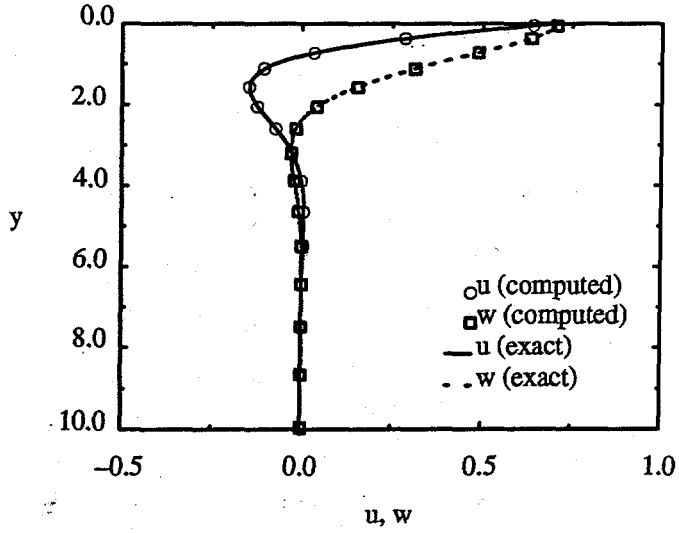


Fig. 12. Ekman Boundary Layer with Applied Shear Stress at the Free Surface

Re = 10000; Rotation about y – axis

Free surface flows are important in many other areas of applications in addition to oceanographic applications. The US Navy is interested in the free surface flows in the vicinity of ship hulls. These are treated as inertial flows. Both steady and unsteady flows around various ship hulls have been computed with the UNCLE solver (see [33], [37] and [38]) . In Fig. 13(a) the

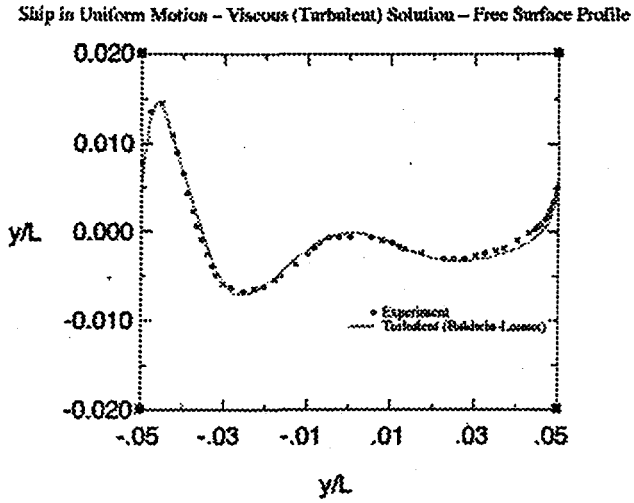


Fig. 13(a). Comparison of the Computed and Experimental Wave Profiles along the Wigley Hull. Fr = 0.289 ; Re = 1,000,000.

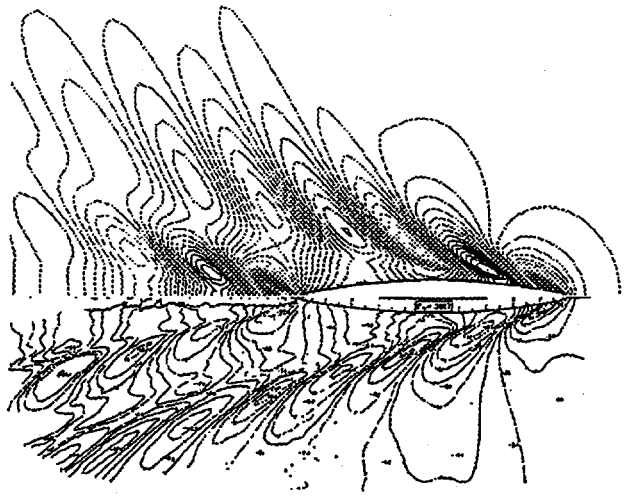


Fig. 13(b). Comparison of the Computed and Experimental Wave Contours for the Wigley Hull. Fr = 0.289 ; Re = 1,000,000.

computed wave profile along the Wigley hull is compared with the experimental wave profile. Figure 13(b) shows the comparison between computed and experimental wave contours. In both these figures the ship is stationary and a uniform flow is flowing past it. Figures 14(a) and 14(b) show the unsteady wave response due to a heaving Wigley hull placed in a uniform flow. The heaving is produced by oscillating the hull in a sinusoidal fashion in the xy–plane

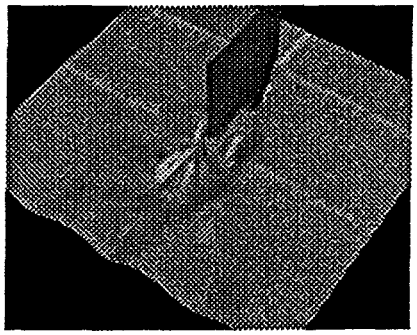


Fig. 14(a). Perspective View of the Unsteady Wave Pattern due to the Heaving Wigley Hull at time 5.25T. Fr = 0.289.

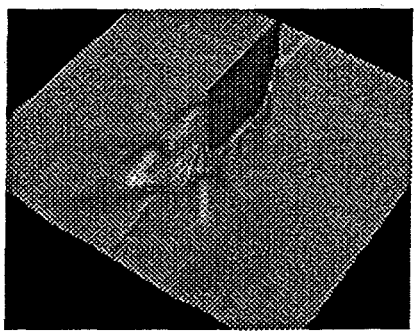


Fig. 14(b). Perspective View of the Unsteady Wave Pattern due to the Heaving Wigley Hull at time 6.0T. Fr = 0.289.

with a reduced frequency of 9.81 based on length. The wave patterns are shown at 5.25T and 6.0T. Further results are presented in Ref. [38]. In Fig. 15, the comparison is shown between the computed and experimental wave profiles along the DTMB Model 5415 hull placed in a uniform flow. Computed and experimental wavecuts off the body are compared in Fig. 16.

7.2 Ocean results.

Eddy viscosity

The viscous boundary conditions at the free surface used in the present study are derived in Section 4.5 (see Beddhu and Whitfield [21] for details). From Section 4.5, the non–dimen-

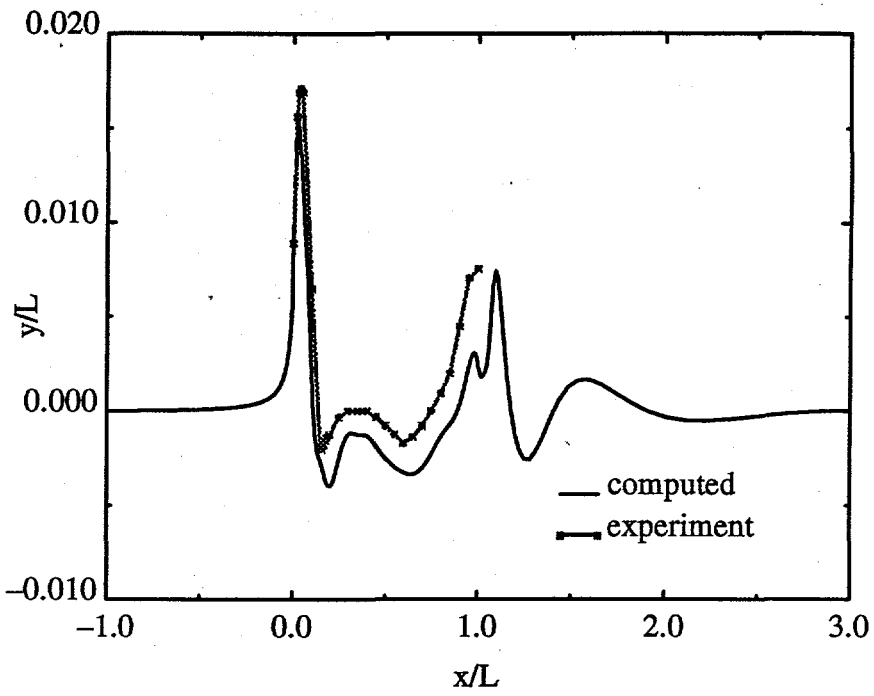


Fig. 15. Comparison of Computed and Experimental Hull Profiles for Model DTMB 5415. Fr = 0.2756; Re = 12021000.

sional viscous stress is given by $\underline{\tau} = \frac{L\tau^*}{\mu_o U_o}$, where τ^* is the dimensional viscous stress obtained from the ECMWF–data, Trenberth et al. [61]. For the planetary scale ocean flow problem the reference length was chosen as the radius of the earth, and the reference velocity was chosen as, $U_o = |\underline{\Omega}| L$ where $|\underline{\Omega}| = 2\pi/86400$ rad/s. The density of water was taken as 1035 kg/m^3. The only quantity over which uncertainty prevails in the ocean modeling community is the reference viscosity, μ_0, whose unit is kg/m–s. If the reference viscosity is chosen as the molecular viscosity for water (0.001 kg/m–s) then the Reynolds number for such a flow is 3.05×10^{15}. Also the non–dimensional value of the applied wind stress becomes very large. Choosing a higher value of the reference viscosity is equivalent to introducing a constant eddy viscosity. This will result in a lower value of the Reynolds number as well as lower the applied non–dimensional shear stress. The predominant trend in the ocean modeling community is to

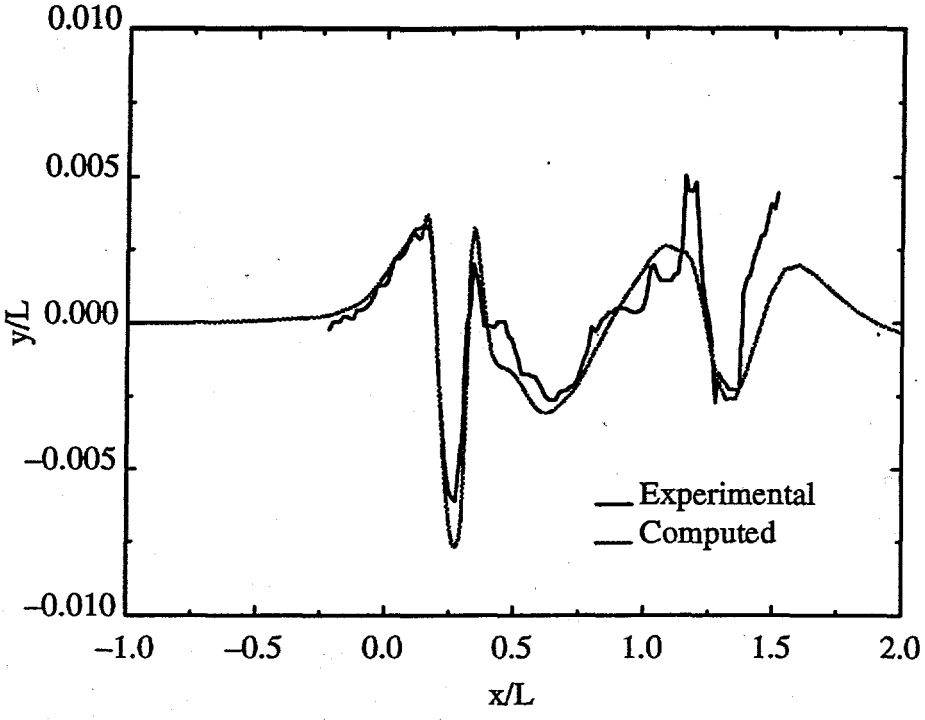


Fig. 16. Comparison of Computed and Experimental Stereophotographic Wave Cuts
Model 5415; z/L = 0.0965; Fr = 0.2756; Re = 12021000.

choose a constant eddy viscosity. In fact, different values of the constant are used depending upon the direction of the coordinate lines. Typically, the values chosen for the horizontal are much larger than the values chosen for the vertical.

In the oceans, large scale currents (or transports) occur in the horizontal whereas large scale gradients occur in the vertical (near the free surface and the bottom). This case is analogous to a high speed flow over an airfoil. To fix ideas, assume that the flow is from left to right along the x–axis, in the xy–plane, and the airfoil is at the origin. The y–axis is normal to the flow direction. In this case, even though large scale transport occurs in the x–direction, large scale gradients occur in the y–direction. In the field of Aerospace Engineering, in order to compute such a flow, one would normally ignore the viscous effects in the x–direction (the predominant flow direction) and retain the viscous effects only in the y–direction. This is the basic idea behind the thin layer Navier–Stokes equations approach proposed by Baldwin and Lomax

[62]. In contrast to this philosophy, in ocean modeling, large values of eddy viscosities are used in the horizontal, which is predominantly a transport direction, whereas smaller values of the eddy viscosity are used in the vertical direction, which is dominated by large gradients in the flow variables. It is not clear whether there is a physical basis for this apparently contradictory approach used widely in the ocean modeling community, or, whether the numerics involved in solving the various approximate equations necessitate such usage.

In calculating atmospheric flows it is frequently assumed that the planetary boundary profile is logarithmic in nature in the inner region from the sea–level upwards. In fact, the equations used in the ECMWF–dataset [61] to calculate the wind stresses from the wind speeds makes use of this assumption. These equations, for the speed range over 3 m/s, are first presented by Large and Pond [63]. Measurements made by Chriss and Caldwell [64] at the ocean floor indicates that the universal log–law is obeyed by the velocity profiles. Hence, it seems plausible to make the assumption that the log–law is valid in the inner layer of the ocean boundary layer from the sea–level downwards also.

Corresponding to a wind speed of 10 m/s, one obtains a wind stress (τ_{fs}) of roughly 0.17 N/m^2. This implies that the friction velocity is $u_\tau = \sqrt{\tau_{fs}/\rho} = 0.0128$ m/s. That is the friction velocity is of the order of one cm/s. The non–dimensional length $y^+$is given by $y^+ = yu_\tau/\nu_0$, where $\nu_0 = \mu_0/\rho_0$. Substituting the appropriate values and taking $\mu_0 = 0.001$ kg/m–s (molecular viscosity of water), one obtains $y^+ = 13265\, y$, where y is in meters. Thus, when $y^+ = 5$, which is typically in the sub–layer, the corresponding value of y is 0.38 mm. (0.00038 meters). In order to resolve the sub–layer near the ocean free surface one has to have a grid whose first grid point from the free surface has be less than 0.37 mm away. Compared to the radius of the earth this number is indeed very, very small.

Baldwin–Lomax model [62] has been widely used in Computational Fluid Dynamics area for predicting the mean flow quantities in a turbulent flow. Originally proposed for wall

bounded thin shear layers, this model has found successful applications in other areas as well. This model falls under the category of algebraic closure of the turbulent closure models. It attempts to provide the right turbulent viscosity from the mean flow itself. It also has a wake correction formulation to take into account the effects of wakes. For the ocean flows this wake correction is not needed. If the grid has sufficient resolution, this model resolves the flow structure including the sub–layer and reproduces the log–law quite well.

Since the reference viscosity used in the non–dimensionalization should have no bearing on the solution and since the only obvious choice that is physically meaningful is the molecular viscosity of water, it was decided to use the molecular viscosity of water as the reference viscosity, to obtain the turbulent Ekman boundary layer solution using the Baldwin–Lomax model, with an applied wind shear at the free surface. The Ekman layer is another example where even though the flow is predominantly in the horizontal, the viscous effects are important only in the vertical which is the direction of dominant flow gradients. The issue of horizontal viscosity coefficients does not arise in this test case.

A rectangular grid, as shown in Fig. 10, was constructed with 101, 201 and 2 points in the x, y and z (in other words i, j and k) directions respectively. The angular velocity vector is aligned along the y–direction (vertical) as before. A wind shear of 0.17 N/m^2 is applied along the x–direction. It was decided to focus only near the free surface region. So in order to simulate a deep ocean condition the bottom was kept at 12750 m depth. In non–dimensional lengths the bottom boundary is placed at 0.998 and the top boundary is placed at 1 (L = radius of earth = 6365000 m). Along the x–direction the boundaries are placed at x=0 and x=1. Only two grid points are used in the z–direction at 0 and 0.1 respectively. The grid in the x–direction is uniform and in the y–direction it is stretched from the free surface (y=1) with the first point from the free surface placed at 10^{-11}. This corresponds to a physical distance of 0.06 mm. So this point is well within the sub–layer. At the x=0 boundary inflow boundary condi-

tions are specified and at the x=1 boundary outflow boundary conditions are specified, both using characteristic variables. At both the boundaries in the z–direction extrapolation boundary conditions are used. At the bottom boundary the boundary condition depends upon whether the flow is coming into the boundary or leaving it. This is determined by taking the dot product of the local velocity vector with the normal to the surface. The Reynolds number in this case becomes 3.05×10^{15}. The main interest is to find out whether the log–law is reproduced, and, what kind of dimensional values of the velocity components are obtained at the the free surface. From observations it is known that a surface wind of 10 m/s would produce an ocean current in the order of 10 cm/s. The initial condition is rest everywhere and the shear stress is applied at the free surface. With a time step of 0.05 it take about 250 cycles for the solutions to converge. However, the run was continued up to 3000 cycles and the solutions presented are at the 3000th cycle.

Normally, the universal velocity profile is plotted such that the velocity is zero at the viscous surface and reaches its maximum in the interior. However, in the present case the maximum velocity occurs at the free surface. So in order to compare with the traditional log–law what is plotted in Fig. 17 is the quantity $U_s - u + \epsilon$, where U_s is the surface velocity and ϵ is the small correction applied so that $u^+ = 1$ at $y^+ = 1$ (i.e. $\log y^+ = 0$). The value of ϵ turns out to be 0.7752. The logarithmic portion is given by $u^+ = 1/\kappa \ln (y^+) + C$ where κ is the von Karman constant and is taken to be 0.41 and C = 4.9. It is clearly seen from Fig. 17 that the u–component of velocity obeys the universal velocity profile. The w–component of velocity is plotted in Fig. 18. Since the applied stress is in the x–direction only the derivative of w with respect to y should be zero at the free surface and it is indeed the case as can be seen from Fig. 18. The eddy viscosity is plotted in Fig. 19 as a function of y^+. The dimensional values of the velocity components at the free surface are 44.01 cm/s for u–component and 6.25 cm/s for the w–component. These values are not only reasonable but also show that the

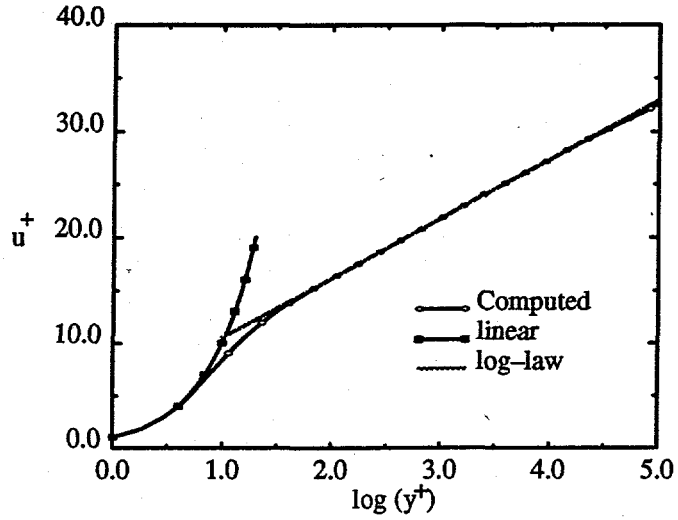


Fig. 17. Turbulent Ekman Layer due to Applied Shear Stress at the Free Surface

$Re = 3.05 \times 10^{15}$; $\tau_{fs} = 0.169$ N/m^2

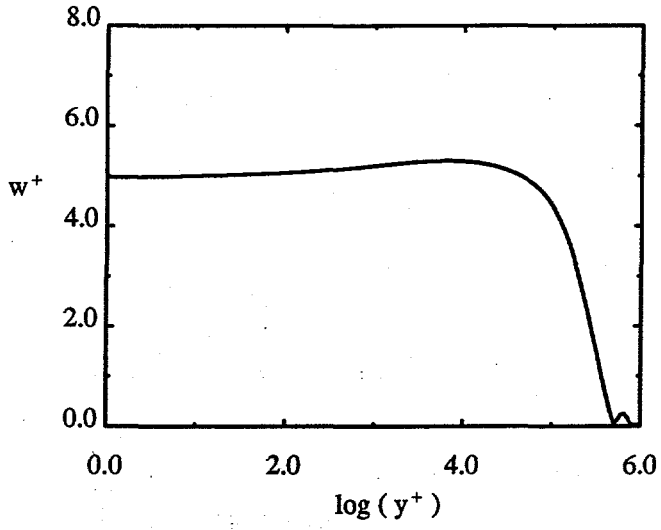


Fig. 18. Turbulent Ekman Layer due to Applied Shear Stress at the Free Sur-

$Re = 3.05 \times 10^{15}$; $\tau_{fs}^{face} = 0.169$ N/m^2

so called Ekman drift is present in the results. Further study is underway to plot the Ekman drift as a function of Reynolds number in the hodographic plane.

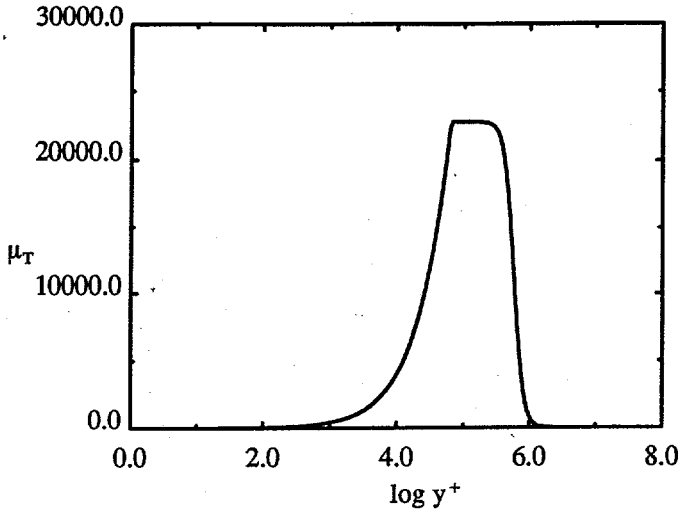


Fig. 19. Eddy viscosity vs log y^+

Atlantic ocean

This test case was chosen to demonstrate the capability of a Navier–Stokes code and the results must be considered as preliminary. Figure 20 shows a satellite view of the entire grid and Fig. 21 shows the view of the grid in the polar region (Arctic ocean).

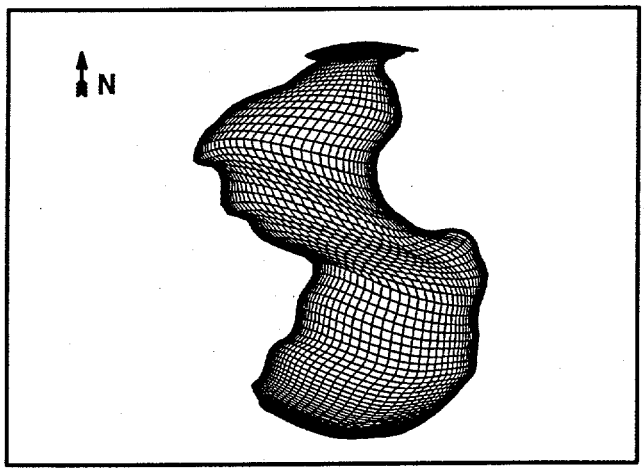


Fig. 20. Satellite View of the Atlantic Grid

The origin is at the center of the earth, the y–axis passes through the North pole, the x axis passes through the intersection of the Greenwich line with the equator. A set of points were

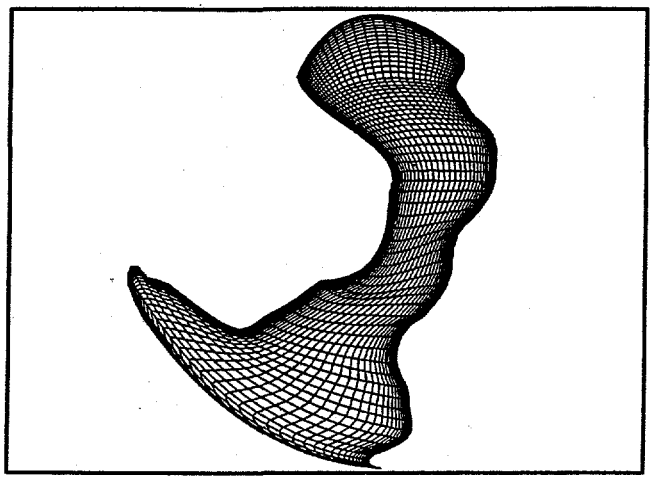


Fig. 21. Satellite View of the Polar Region

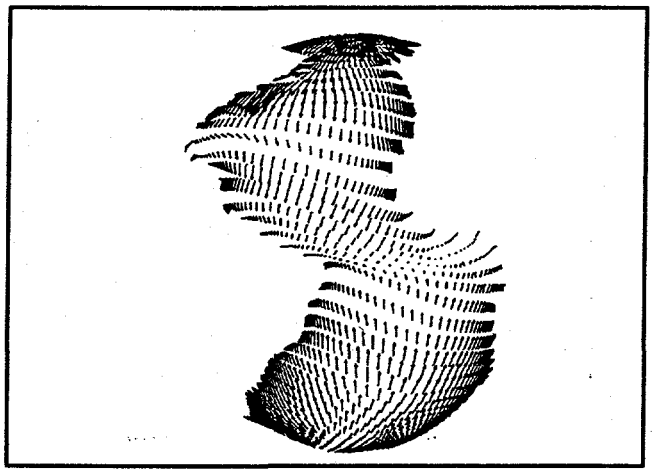


Fig. 22. Velocity Vectors at Mid Depth

chosen along the continents and they were smoothed using cubic splines. Thus, the two outer most S–shaped curves were generated. Realistic bottom topography is used to generate the surface grid on the ocean floor. From coast to coast, 51 grid points were distributed with packing near the coasts. This corresponds to the ζ (or k) direction. The η (or j) direction corresponds to the radial direction along which 41 points were distributed with packing near the bottom as well as near the free surface. However, this packing was not sufficient to resolve the viscous sub–layer as was done in the turbulent Ekman layer case. The ξ direction increases in the meridional direction from latitude 70 S up to the Asian continent in the north to include the

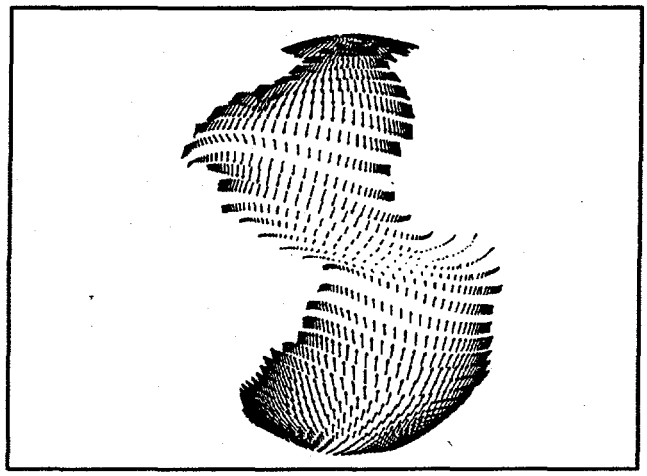


Fig. 23. Velocity Vectors at the Free Surface

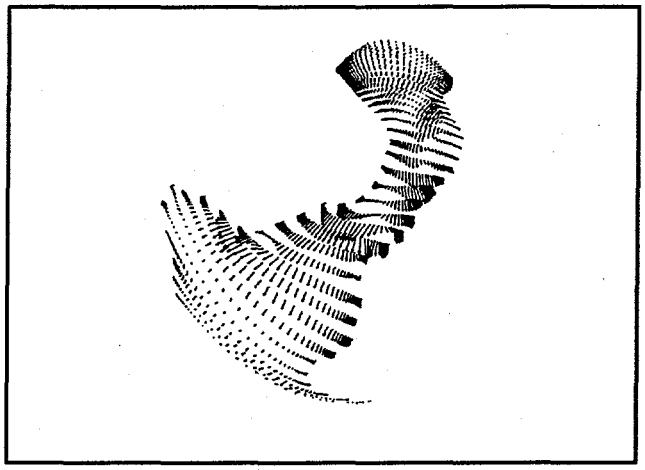


Fig. 24. Velocity Vectors in the Polar Region (Mid Depth)

Arctic Ocean. 131 points were distributed along this direction with packing near 70 S and near the Asian continent. Note that the inclusion of the poles does not need special treatment in the present approach. For viscous calculations a relative spacing of 0.001 was chosen.

The boundaries i=1, i=imax, j=1, k=1 and k=kmax were treated as viscous walls, j=jmax was treated as a rigid lid (using the impermeable wall boundary condition). Part of the k=1 boundary corresponding to the Drake pass was treated as an inflow boundary with a uniform inlet velocity of 1 m/s and part of the k=kmax boundary corresponding to the Agulhas pass was treated as an outflow boundary. The Reynolds number based on Earth's radius and the

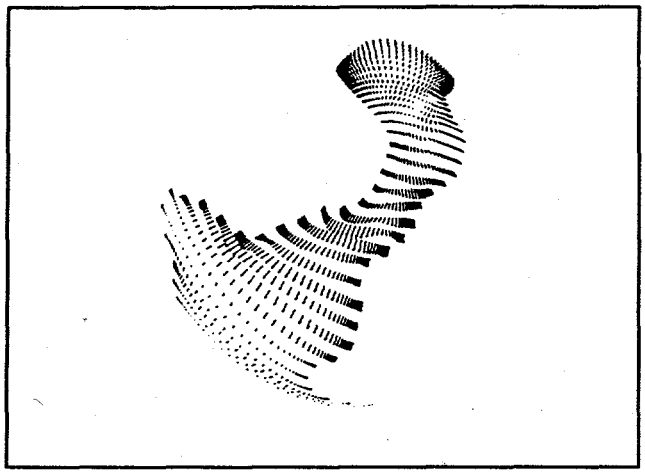


Fig. 25. Velocity Vectors in the Polar Region (Free Surface)

equatorial rotational speed turns out to be of the order of 10^{15}. A very highly refined grid would be needed to resolve such a flow. Hence, a Reynolds number of 10^6 was chosen for the viscous calculations. It is to be noted that neither wind stresses nor temperature and salinity were included in the calculations. Thus, the results are purely governed by the dynamics of the flow. Starting from the initial condition of rest in the relative frame, the Earth was rotated with one revolution per day through the use of the grid speed vector $\underline{w}$. For each complete rotation 628 time steps were used. Figure 22 shows the velocity vectors at mid depth and Fig. 23 shows the same at the free surface for the entire Atlantic ocean. The corresponding cases for the Arctic region are shown in Figures 24 and 25. These figures mainly show the effects of the curvature of the Earth, bottom topography and the shape of the continents.

World ocean

Since the motivation of this work has been to compute the planetary scale ocean flows, the final set of results are given on a model ocean which has true bottom topography but the side boundaries are approximated. Thus, the Gulf of Mexico is ignored and Australia and Asia are joined with each other, among other simplifications. There are 26 blocks each with varying

sizes with a total of approximately 3.53 million points. In each block the number of points in the vertical is kept the same at 41 points. Since it is not yet practical to use such a fine spacing near the free surface and the bottom boundary as is done in the turbulent Ekman layer, for the computations of the world ocean the first point from the free surface is adjusted according to the depth. Depending upon the depth the first point could be either a few cms away or as much as meter away from the free surface. As a rough estimate, from discussions above the value of y^+ at one meter depth can be as large as 13000 which means that major portions of the inner

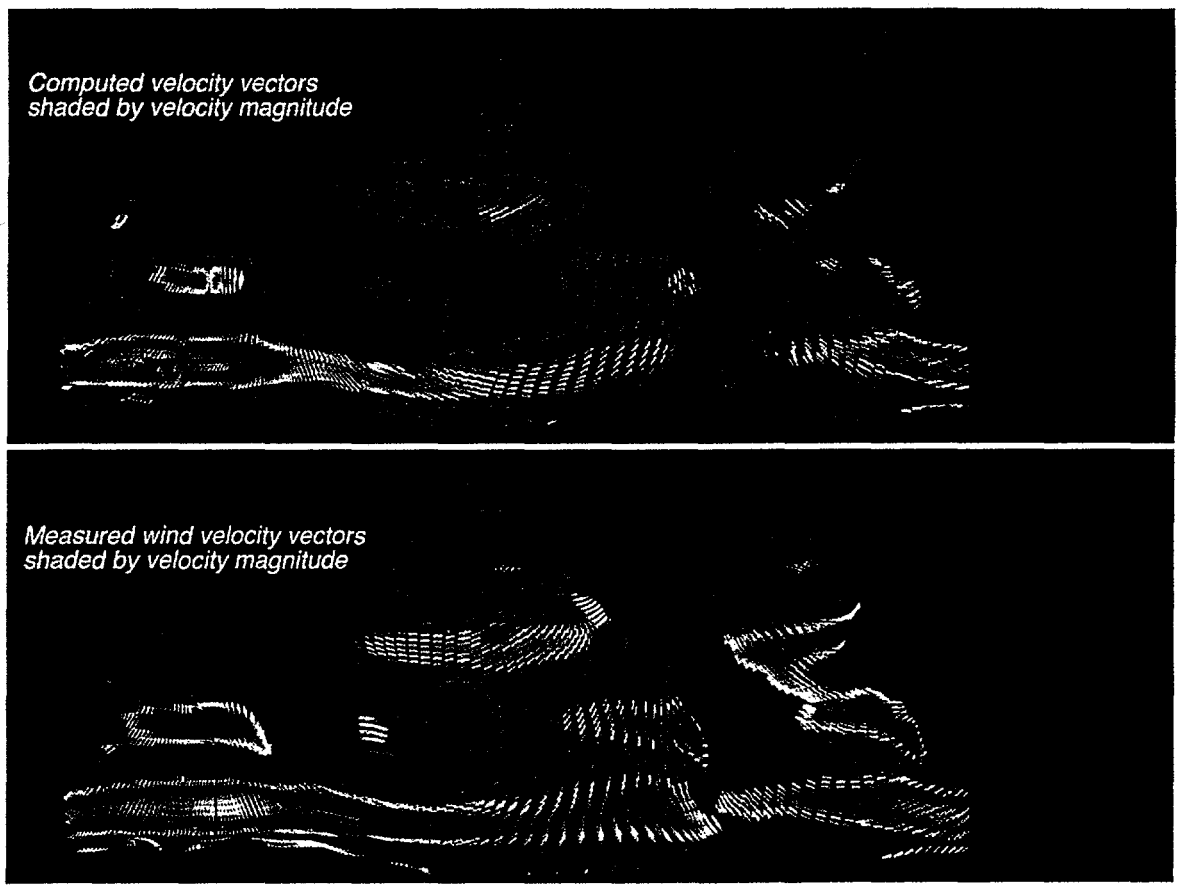


Fig. 26. Comparison of Computed World Ocean Surface Velocity Vectors with Wind Velocity Vectors Obtained from ECMWF Dataset

region will not be resolved. Also the gradients computed at the free surface will be incorrect. So, it was decided to use a constant eddy viscosity for this computation. From Fig. 19, it is

seen that the non–dimensional eddy viscosity reaches a maximum of 25000. In dimensional units this corresponds to 25 kg/m–s. So a value of 100 kg/m–s has been chosen as the eddy viscosity. Therefore, the kinematic eddy viscosity becomes 0.0966 m^2/s. The choice of this particular value is completely arbitrary. This value of μ_0, effectively reduces the Reynolds number to 3.05×10^{10}. The non–dimensional eddy viscosity is one and is kept the same in all directions.

Figure 26 shows the velocity vectors of the computed ocean currents at the free surface. These results are obtained starting from the initial condition of rest with applied wind stress at the free surface. The results presented are after a four day spin up. Figure 26 also shows the wind vectors as obtained from the ECMWF dataset. In both figures the vectors are colored according to the velocity magnitude. It can be seen that the computed surface currents exhibit all the major features of the wind data. There is correspondence in the magnitude of the velocity vectors as well, i.e., where the wind velocities are high (indicated by red) the ocean surface current velocities are also high and where the wind velocities are low (indicated by blue) the ocean surface current velocities are also low. There are discrepancies in the south east Pacific ocean where the directions of the wind and ocean currents seem to deviate from each other the most. This result is presented here only as a preliminary result and further investigations are underway.

Conclusions

This report presents an approach for solving ocean flows which is a total departure from the approaches currently being used. However, it should be noted that the proposed approach is routinely being used in other areas involving fluid flows. The approach of computational physical oceanography involves solving the complete set of (Reynolds Averaged) Navier–Stokes equations on nonorthogonal curvilinear grids. During the course of this investigation, a detailed analysis of the governing equations has been undertaken and many new theoretical results have been obtained. From a theoretical point of view, the new governing equations proposed in Section 4.2 and the new viscous term proposed in Section 4.3 are important. The extent to which these results will alter the flow field of the oceans remains to be seen. It is shown in Section 4.6 that the hydrostatic approximation is a coordinate system dependent approximation. The new formulation of the Coriolis force could be the basis for alternate numerical schemes one of which is the present scheme. The results of the turbulent Ekman layer shows that the modeling of the turbulent viscosity is still an unsettled problem. It also shows the grid resolution that is needed (in the "vertical") to solve the ocean flow field in a consistent manner. The results for the Atlantic ocean and the world ocean show encouraging trends and further investigation is needed. Computation of the ocean flows with an evolving free surface also needs to be carried out. Time dependent surface forcing fields are another area that needs to be investigated. The present method is formulated to handle such problems. The CPU–time per grid point per time step on an SGI R8000 architecture for the world grid is 0.000462 seconds.

Appendix

A. Coordinate form of the momentum equations in unsteady Eulerian Coordinates

In order to express the tensor invariant form in terms of a a particular coordinate system all one needs is the following identity:

$$\nabla \cdot F = \frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^k} \left[\sqrt{g}\, \underline{a}^k \cdot F \right] \tag{A.1}$$

where, F could be a vector or tensor; ξ^k, k=1,2,3 are the curvilinear coordinates; and $\underline{a}^k$, k=1,2,3 are the contravariant base vectors. Note that, $\underline{a}^k = \nabla \xi^k$. Summation over repeated indices is implied in Eq. (A.1). Note that, one can express the quantity F in either Cartesian or curvilinear coordinates. In the so called partial transformation, F is expressed in terms of Cartesian coordinates. Upon expressing F (assuming F to be a tensor) and $\underline{a}^k$ in Cartesian coordinates, Eq. (A.1) becomes

$$\nabla \cdot F = \frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^k} \left[\sqrt{g} \left(\frac{\partial \xi^k}{\partial x_n} F_{nm} \right) \right] \underline{i}_m \tag{A.2}$$

Equation (A.2) can be expanded for m = 1, 2 and 3 to obtain the three components. Note that, since F is assumed to be a tensor $\nabla \cdot F$ is a vector. Equations (4.6.2) – (4.6.4) can now easily be obtained using (A.2).

References

[1] K. Bryan, *J. Comp. Phys.*, **4**, 347.

[2] M. D. Cox, "A primitive Equation Three Dimensional Model of the Ocean," GFDL Ocean Group Tech. Rept. No. 1, GFDL/NOAA, Princeton University, Princeton, (1984).

[3] A. J. Semtner Jr., "Finite–Difference Formulation of a World Ocean Model," in *Advanced Physical Oceanographic Numerical Modelling*, Edited by J. J. O'Brian, (D. Reidel, Norwell, Mass., 1986), p. 187.

[4] P. D. Killworth, D. Stainforth, D. J. Webb and S. M. Paterson, *J. Phys. Oceano.*, **21**, 1333, (1991).

[5] A. F. Blumberg and G. L. Mellor, "A Description of a Three–Dimensional Coastal Ocean Circulation Model," in *Three Dimensional Coastal Ocean Models,* Edited by N. Heaps, (American Geophysical Union, Washington, DC, 1987), p. 1.

[6] R. L. Haney, *J. Phys. Oceano.*, **21**, 610, (1991).

[7] J. Pedlosky, *Geophysical Fluid Dynamics*, (2nd edition, Springer–Verlag, New York, 1987).

[8] A. G. L. Borthwick and E. T. Kaar, *Inter. J. Num. Meth. Fluids*, **17**, 417, (1993).

[9] A. M. Winslow, J. Comp. Phys. **2**, 149, (1966).

[10] J. F. Thompson, F. C. Thames and C. W. Mastin, J. Comp. Phys., **15**, 299, (1974).

[11] J. F. Thompson, "Numerical Solution of Flow Problems Using Body–Fitted Coordinate Systems," Lecture Series in Computational Fluid Dynamics, Von Karman Inst. for Fluid Dynamics, Belgium, 1978; in "Computational Fluid Dynamics" (W. Kollmann, Ed.), Hemisphere, 1980.

[12] Z. U. A. Warsi, Applied Math. Comp., **21**, 295, (1987).

[13] J. F. Thompson, Z. U. A. Warsi and C. W. Mastin, *Numerical Grid Generation – Foundation and Applications*, (North Holland, New York, 1986).

[14] M. G. Remotigue, E. T. Hart and M. L. Stokes, "EAGLEView: A Surface and Grid Generation Program and its Data Management," in Proceedings of Workshop on Software Systems for Surface Modeling and Grid Generation, NASA CP–3143, p. 243, April, 1992.

[15] J. F. Thompson and B. Gatlin, "Program EAGLE – User's Manual, Volumes I, II and III, AFATL–TR–88–117, September 1988.

[16] M. Beddhu, L. K. Taylor and D. L. Whitfield, *J. Comp. Phys.*, **128**, 427, (1996).

[17] M. Beddhu, The Momentum Equation for Incompressible Turbulent Free Surface Flows in the Presence of Rotation, Submitted to *J. Comp. Phys.*, 1996.

[18] Greenspan, H. P., *The Theory of Rotating Fluids,* (Cambridge University Press, London, 1968).

[19] M. Beddhu, M. Y. Jiang, L. K. Taylor and D. L. Whitfield, Towards Computations of Ocean Flows Using Navier–Stokes Equations, Invited Paper, *Proceedings, Supercomputing–94*, Washington D. C., pp. 144–153, 1994.

[20] K. Bryan and M. D. Cox, *J. Phys. Oceanogr.*, **2**, 510, (1972).

[21] M. Beddhu, and D. L. Whitfield, A New Formulation of the Three Dimensional, Viscous Free Surface Boundary Conditions in a Curvilinear Coordinate System, manuscript under preparation, 1996.

[22] L. K. Taylor, "Unsteady Three – Dimensional Incompressible Algorithm Based on Artificial Compressibility," Ph. D. Dissertation, Department of Aerospace Engineering, Mississippi State University, Mississippi State, May 1991.

[23] P. L. Roe, *J. Comp. Phys.*, **43**, 357, (1981).

[24] B. van Leer, *J. Comp. Phys.*, **32**, 101, (1979).

[25] D. L. Whitfield, "Newton–Relaxation Schemes for Nonlinear Hyperbolic Systems," *MSSU–EIRS–ASE–90–3,* Mississippi State University, Mississippi State, MS, October, 1990.

[26] D. Pan and S. Chakravarthy, "Unified Formulation for Incompressible Flows," *AIAA–89– 0122*, Jan., (1989).

[27] D. L. Whitfield and L. K. Taylor, "Discretized Newton–Relaxation Solution of High Resolution Flux–Difference Split Schemes," *AIAA–91–1539*, June (1991).

[28] D. L. Whitfield and L. K. Taylor, "Numerical Solution of the Two–Dimensional Time–Dependent Incompressible Euler Equations," MSSU–EIRS– ERC–93–14, Mississippi State University, Mississippi MS 39762 April, 1994.

[29] A. Arabshahi, L. K. Taylor and D. L. Whitfield, "UNCLE: Toward a Comprehensive Time–Accurate Incompressible Navier–Stokes Flow Solver", *AIAA–95–0050*, Jan., (1995).

[30] C. Sheng, "Development of a Multiblock Multigrid Algorithm for the Three–Dimensional Incompressible Navier–Stokes Equations," Ph. D. Dissertation, Department of Aerospace Engineering, Mississippi State University, Mississippi State, May 1994.

[31] C. Sheng, L. K. Taylor and D. L. Whitfield, "An Efficient Multigrid Acceleration for Solving the 3–D Incompressible Navier–Stokes Equations in Generalized Curvilinear Coordinates," *AIAA–94–2335*, June, (1994).

[32] C. Sheng, L. K. Taylor and D. L. Whitfield, "A Multigrid Algorithm for Unsteady Incompressible Euler and Navier–Stokes Flow Computations," Sixth Int. Symp. Comp. Fl. Dyn., Sept. (1995).

[33] M. Beddhu, A. Arabshahi, C. Sheng, M. Y. Jiang and D. L. Whitfield, "A Time Accurate Multigrid Algorithm for Free Surface Flow Calculations Using Domain Decomposition," Manuscript under preparation, 1996.

[34] A. Arabshahi, CFD Lab, NSF/ERC, Mississippi State Univ., Personal Communications, 1996.

[35] S. Nichols, CFD Lab, NSF/ERC, Mississippi State Univ., Personal Communications, 1996.

[36] E. C. Siong, An Algorithm for Solving the Three Dimensional, Incompressible Navier–Stokes Equations with Temperature Using the Modified Artificial Compressibility Method, Master's Thesis, Department of Aerospace Engineering, Mississippi State University, Mississippi State, May 1995.

[37] M. Beddhu, L. K. Taylor and D. L. Whitfield, *Applied Mathematics and Computation,* **65**, 33, (1994).

[38] M. Beddhu, M. Y. Jiang, L. K. Taylor and D. L. Whitfield, "Computation of Steady and Unsteady Flows with a Free Surface Around the Wigley Hull," To appear in *Applied Mathematics and Computation.*

[39] A. E. Gill, Atmosphere–Ocean Dynamics, International Geophysics Series, Vol. 30, (Academic Press Inc., Orlando, 1982).

[40] P. M. Morse and H. Feshbach, Methods of Theoretical Physics, (McGraw–Hill, New York, 1953).

[41] Z. U. A. Warsi, *Fluid Dynamics : Theoretical and Computational Approaches*, (CRC Press Inc., Boca Raton, 1993), p. 91.

[42] C. Truesdell and W. Noll, "The Non–Linear Field Theories of Mechanics," in Encyclopedia of Physics III/3, Ed: S. Flugge, (Springer–Verlag, Berlin, 1965), p.15.

[43] U. Bulgarelli, V. Casulli and D. Greenspan, "Pressure Methods for the Numerical Solution of Free Surface Fluid Flows," (Pineridge Press Limited, Swansea, U. K. 1984).

[44] C. W. Hirt and J. P. Shannon *J. Comp. Phys.*, **2**, 403, (1968).

[45] P. D. Thomas and C. K. Lombard, "The Geometric Conservation Law – A Link Between Finite–Difference and Finite Volume Methods of FLow Computation On Moving Grids," *AIAA Paper 78–1208*, (1978).

[46] A. J. Chorin, *J. Comp. Phys.*, **2**, 12, (1967).

[47] D. L. Whitfield, J. M. Janus, L. B. Simpson, "Implicit Finite Volume High Resolution Wave–Split Scheme for Solving the Unsteady Three Dimensional Euler and Navier–Stokes Equations on Stationary and Dynamic Grids," EIRS Report, *MSSU–EIRS–ASE–88–3*, Mississippi State University, Mississippi State, MS, Feb. 1988.

[48] L. K. Taylor and D. L. Whitfield, "Unsteady, Three–Dimensional Incompressible Euler and Navier–Stokes Solver for Stationary and Dynamic Grids," AIAA–91–1650, June 1991.

[49] W. K. Anderson J. L. Thomas and B. van Leer, *AIAA J.*, **24**, 1453 (1986).

[50] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, (Academic Press, New York, 1970).

[51] S. Chakravarthy, "Relaxation Methods for Unfactored Implicit Schemes," *AIAA–84–0165*, 1984.

[52] L. A. Hageman and D. M. Young, *Applied Iterative Methods*, (Academic Press, New York, 1981).

[53] G. Dahlquist and A. Björck, *Numerical Methods*, (Prentice–Hall, Englewood Cliffs, NJ, 1974).

[54] D. L. Whitfield and J. M. Janus, "Three Dimensional Unsteady Euler Equations Solutions Using Flux Vector Splitting," *AIAA–91–1539*, June 1991.

[55] J. M. Janus, "Advanced 3–D CFD Algorithm for Turbomachinery," Ph.D. Dissertation, Mississippi State University, May 1989.

[56] M. Y. Jiang, CFD Lab, NSF/ERC, Mississippi State Univ., Personal Communications, 1996.

[57] D. L Whitfield, "Numerical Solution of the Shallow Water Equations," *MSSU–EIRS–ERC–96–4*, Mississippi State University, Mississippi State, MS, July. 1996.

[58] B. F. Armaly, F. Durst, J. C. F. Pereira and B. Schonung, J. of Fluid Mech., **127**, 473, (1983).

[59] T. Rhodes and S. Acharya, *Num. Heat Trans., Part B*, **23**, 153, (1993).

[60] L. K. Taylor, CFD Lab, NSF/ERC, Mississippi State Univ., Personal Communications, 1993.

[61] G. D. V. Davis, Int. J. Numr. Meth. Fluids, **3**, 227, (1983).

[62] K. E. Trenberth, J. G. Olson and W. G. Large, A Global Ocean Wind Stress Climatology Based on ECMWF Analyses, *NCAR Technical Note, NCAR TN–338+STR*, Natl. Cent. for Atmos. Res., Boulder Colorado, Aug. 1989.

[63] B. S. Baldwin and H Lomax, Thin Layer Approximation and Algebraic Model for Separated Turbulent Flow, *AIAA– Paper 78–257*, Huntsville, Alabama, 1978.

[64] W. G. Large and S. Pond, J. of Phy. Oceanogr., **11**, 324, (1981).

[65] T. M. Chriss and D. R. Caldwell, *J. of Geophy. Res.*, **89**(C4), 6403, (1984).