UL

NUMERICAL SOLUTION OF THE THREE-DIMENSIONAL NAVIER-STOKES EQUATIONS

Stephen Michael Richardson

Department of Chemical Engineering and Chemical Technology,

Imperial College of Science and Technology, London, 1976.

Submitted for the Degree of Doctor of Philosophy

in the University of London.

Abstract

A numerical method for solving the three-dimensional Navier-Stokes equations is presented which combines efficiency with applicability to general problems of the laminar flow of incompressible Newtonian fluids. The method is also suitable for problems involving non-Newtonian fluids, but rather less so for compressible and turbulent flow problems. For problems involving incompressible fluids, the method entails the expression of velocity in terms of scalar and vector potentials which ensure that the equation of continuity is automatically satisifed, and the introduction of vorticity so as to eliminate pressure as a dependent variable. The method is applied to a test problem: uniform and linear-shear flow past a sphere, and drag, lift and moment coefficients of the sphere are predicted for a range of values of the flow parameters. Although a relatively large computational effort is required to obtain such three-dimensional solutions, application of the vorticity/potential method is straightforward, even in relatively complex flow problems, and results can be obtained whose accuracy is essentially limited only by the computer resources available. Use of the method in obtaining solutions of the three-dimensional Navier-Stokes equations, even for general flow problems, can, therefore, be recommended.

Acknowledgements

I wish to express my sincere thanks for the invaluable guidance, criticism and encouragement given to me during the course of the work desribed here, particularly by my family and Dr. Alan Cornish, and also by Steven Fiddes, Peter Hughes, Dr. Ignace Krizancic and Professor Anthony Pearson.

I dedicate this to Hilary.

S. M. Richardson

Contents

Notation

We list here the more commonly used symbols, and indicate their usual meanings. Less commonly used symbols, and alternative meanings which are used only occasionally, will be indicated in the text as they occur.

$\underline{A}$	vector potential field
C_D, C_{DP}, C_{DV}, C_L, C_{LP}, C_{LV}, C_M	drag, lift and moment coefficients defined in equations (3.30) - (3.36)
dR	element of R
dS	element of area
D	diameter of sphere
E_3	three-dimensional Euclidean space
$\underline{F}$	body forces
$\overset{\circ}{G}(R)$	subspace of $L_2(R)$ defined in sub-section 2.2.2
h_i, h, h^j	scalar potential fields
i, j, k, n	discretised r, θ, ϕ and t coordinates, respectively
$\overset{\circ}{J}(R)$	subspace of $L_2(R)$ defined in sub-section 2.2.2
l_j, l_j'	contours defined in sub-section 2.2.2
$L_2(R)$	space of Lebesgue square-integrable three-dimensional vector fields defined in equation (2.3)
m	constant defined under equation (2.12)
$\underline{n}$	unit outer normal to dS
p	pressure field
r	distance between two points in R; see also r, θ, ϕ
r, θ, ϕ	spherical polar coordinates
r_0, r^*	radius of sphere and outer envelope, respectively
R	three-dimensional subspace of E_3

Re	Reynolds number defined under equation (3.10)
S_i, S'_j	surfaces defined in sub-section 2.2.2
t	time
$\underline{u}$	velocity field
U^o_∞	centre-line velocity at infinity
$U_1(R)$, $U'(R)$, $U_2(R)$	subspaces of $L_2(R)$ defined in subsection 2.2.2
$\underline{w}$	vorticity field
$\underline{x}$, $\underline{y}$	points in R
x, y, z	Cartesian coordinates
α_i, β_j	constants defined in sub-section 2.2.2
δ_{ij}	Kronecker delta
δR, ∂R	boundary of R, outer boundary of R
θ	rate of dilatation; see also r, θ, ϕ
μ	viscosity
ν	kinematic viscosity
ρ	density
σ	magnitude of shear at infinity
ϕ	scalar potential field; see also r, θ, ϕ
∇, $\nabla .$, $\nabla_\wedge$, ∇^2, Δ	gradient, divergence, curl, scalar Laplacian and vector Laplacian operators, respectively
Δr, $\Delta\theta$, $\Delta\phi$, Δt	increments of r, θ, ϕ and t, respectively
$\in$, $\subset$	element of and subspace of, respectively

N.B. (i) all coordinate systems are right-handed unless otherwise stated;

(ii) the same symbol is used to denote continuous and discretised, dimensional and dimensionless variables. The context will make it clear which is intended.

Chapter 1 - Introduction and Literature Survey

1.1 Preliminaries

The Navier-Stokes equations are the mathematical expression of Newton's second law of motion applied to an element of an incompressible Newtonian fluid of constant viscosity moving in an Eulerian frame of reference in a region R of three-dimensional Euclidean space E_3. In vector form, the equations may be written(1):-

$$\frac{\partial \underline{u}}{\partial t} + (\underline{u}.\nabla)\underline{u} = -\frac{1}{\rho}\nabla p + \nu \Delta \underline{u} + \underline{F} \quad (1.1)$$

where $\underline{u}$ and p are, respectively, the velocity and pressure fields of the fluid,

ν and ρ are, respectively, the kinematic viscosity and density of the fluid, both of which are assumed to be constant at all points in the region R for all times t,

and $\underline{F}$ represents body forces which apply to the whole of a fluid element, and usually arise from (known) external fields such as gravity.

The Navier-Stokes equations define the dynamics of the flow; the kinematics of the flow, on the other hand, are defined by the equation of continuity which, for an incompressible fluid, reduces to a statement that the velocity field is everywhere solenoidal i.e.

$$\nabla.\underline{u} \equiv 0 \quad (1.2)$$

Together with conditions imposed on the boundary δR of the region R, and initial conditions at the time t = 0, the set of equations (1.1) - (1.2) is generally assumed to be sufficient to determine the velocity and pressure fields of the fluid in the region R for all times $t > 0$. The validity of this assumption will be considered in sub-section 1.2.1 below. If we accept it for the present, however, then the value of solving these equations is obvious. Our overall aim here is to obtain a general method for solving these equations (as opposed to particular solutions of them) principally for laminar flows. There are, however, three fundamental difficulties which we encounter when we attempt to solve them.

First, the expression for the acceleration of a fluid element in an Eulerian frame of reference (which is one that is fixed relative to some datum), $\frac{\partial \underline{u}}{\partial t} + (\underline{u}.\nabla)\underline{u}$, is essentially non-linear. This means that many of the useful properties of linear equations, such as superposition of solutions, cannot be applied to the Navier-Stokes equations. One obvious way of overcoming this difficulty is to eliminate it, and use a Lagrangian, instead of an Eulerian, frame of reference. A Lagrangian reference frame is essentially one that moves with the flow, so that the expression for the acceleration of a fluid element becomes just $\frac{\partial \underline{u}}{\partial t}$. Clearly, the equations of motion in such a reference frame are linear, but there is a drawback: we have to define a position vector $\underline{r}$ (relative to some datum) associated with every element of the fluid, so that we can follow the motion of each element. Thus, although a Lagrangian description is admirably suited to certain flows (for example, those with free surfaces, since the vector $\underline{r}$ immediately gives the shape and position of such surfaces), for many flows, especially those undergoing large distortions - for example, turbulent flows - a Lagrangian description is quite unsuitable. For most problems, this disadvantage is such that an Eulerian description is preferable, even allowing for its inherent non-linearity. We shall, therefore, confine our analysis to flows defined in an Eulerian frame of reference.

The second principal difficulty that we encounter when solving the Navier-Stokes equations is that they involve pressure as well as velocity. While the boundary conditions on velocity are generally straightforward (the no-slip condition, for example, means that the velocity vanishes on solid boundaries), those on pressure are not. It would, therefore, be an advantage if we could eliminate pressure as a dependent variable so that we could solve the resulting equations directly for the velocity field, and then obtain the pressure field by substitution back into the Navier-Stokes equations.

This difficulty over pressure is related to the third principal difficulty that we encounter when solving the Navier-Stokes equations, which is that they must be solved subject to the constraint of continuity. (These difficulties are related because we may regard the pressure field as providing the necessary degree of freedom in a flow to ensure that continuity is always satisfied.) If we can - efficiently - ensure that continuity is automatically satisfied, then we might be able to reduce the amount of work we must do to solve the Navier-Stokes equations.

Since, as we noted above, the non-linearity in the Navier-Stokes equations is implicit in the Eulerian description of a flow, we can do very little to ease this difficulty, except in certain special cases(2). On the other hand, we can eliminate pressure as a dependent variable, and ensure that continuity is satisfied automatically. The method that we shall develop to do this involves, in effect, the three-dimensional generalisation of the well-known two-dimensional streamfunction/vorticity approach(3) (although, as we would expect, non-trivial differences are involved in the change of dimensionality). The basis of our method is a decomposition of the velocity field into mutually orthogonal components. This decomposition is a generalisation of results obtained from classical potential theory (which, incidentally, is a kinematical theory, so it is not surprising that we can use it to ensure that the kinematical condition of continuity is automatically satisfied). The theory of this decomposition involves only minor assumptions about the velocity field and the flow region. In particular, it makes no assumptions about the smoothness of the velocity field so that, for example, the method can be used in compressible flow problems, even when there are shock waves present. This means that, to all intents and purposes, if the method is applied to the Navier-Stokes equations, no assumptions additional to those implicit in the equations themselves need be made. The method is, therefore, applicable to general flow problems. Thus although our main purpose is to study the flow of incompressible Newtonian fluids, the decomposition method can equally well be applied to flows of compressible and/or non-Newtonian fluids.

A brief outline of what is to follow is appropriate at this stage. In the remainder of this chapter, we will review methods of solving the Navier-Stokes equations. Then, in Chapter 2, which is fundamental to the development of our whole analysis, we will discuss the theory of "orthogonal decomposition" and show how it may be applied in two different but related ways to give a general method of solution of arbitrary flow problems. One involves the representation of velocity in terms of scalar and vector potentials, while the other involves the projection of the Navier-Stokes equations onto the space of vector fields which satisfy the continuity equation (1.2). We will show that the former method can be applied without ambiguity to general flow problems, so that it will, in this sense, become our preferred method for solving the Navier-Stokes equations. In Chapter 3, we will apply this (preferred) method to a particular problem: uniform and linear-shear flow at infinity past a stationary, neutrally buoyant, non-rotating sphere. Our reasons for choosing this problem are first that it is important in its own right, and secondly that it will illustrate how our general method of solution of the Navier-Stokes equations can be applied in a particular case. This problem will be solved numerically, since, because of the non-linearity of the Navier-Stokes equations, it cannot be solved analytically. Finally, in Chapter 4, we will discuss the results which we obtain for this problem, and compare them with other analytical, experimental and computational results which exist. This will enable us to draw conclusions, not only about the computational results we have obtained, but also about the method of solution of the three-dimensional Navier-Stokes equations that we have developed.

1.2 Methods of Solution of the Three-Dimensional Navier-Stokes Equations

In this section, we will first briefly examine the question of the existence and uniqueness of solutions of the three-dimensional Navier-Stokes equations. As we shall see, no definite answer can yet be given to this question, although there are certain indications that a unique solution does in fact exist. We shall then go on to review methods which are currently available for solving the three-dimensional Navier-Stokes equations. A common feature of all of these methods is that solutions are obtained numerically using a computer. The reason for this is that, as we have already noted in section 1.1, the essential non-linearity of the Navier-Stokes equations in an Eulerian frame of reference means that analytical solutions cannot be obtained in general. (We note, incidentally, that it has only been possible to obtain computational solutions to three-dimensional, as opposed to two-dimensional, flow problems relatively recently. This is because it is only recently that computers have had both the speed and the storage facilities to enable them to handle such problems.) A critical assessment of these currently available methods will enable us to conclude this section by identifying two particular methods which for various reasons appear to be suitable for use in general three-dimensional flow problems.

1.2.1 Existence and Uniqueness of Solutions

The general question of proving the existence and uniqueness of solutions of the Navier-Stokes equations usually involves applying the techniques of functional analysis to what is essentially a set of non-linear partial differential equations. (In fact, the techniques used are almost exclusively based on the orthogonal decomposition theory which we shall present in Chapter 2. It is not surprising, therefore, that we can also base methods of solution on this theory.) Although we cannot yet prove whether or not a unique solution of the three-dimensional Navier-Stokes equations (1.1), satisfying the continuity equation (1.2) for arbitrary boundary and initial conditions, given an arbitrary body force $\underline{F}$, actually exists for all times $t > 0$, we can prove several weaker results.

Thus we can show that the two-dimensional Navier-Stokes equations have a classical solution (i.e. a solution which is sufficiently differentiable for the various terms in the Navier-Stokes equations to be defined) which is unique in the large (i.e. $0 \leqslant t < \infty$), assuming the boundary conditions etc are smooth enough[4]. The three-dimensional equations, on the other hand, have a classical solution only if the initial conditions are small enough or, for less specific initial conditions, in the small (i.e. $0 \leqslant t < T < \infty$)[5]. More generally, the three-dimensional Navier-Stokes equations can be shown to have a weak solution (i.e. a solution which is not necessarily classical), and this solution can be shown to be unique and smooth only in the small[6].

Thus we see that, although the general existence and uniqueness question of solutions of the three-dimensional Navier-Stokes equations is still open, there are (weak) indications that unique classical solutions do in fact exist. Of course, it may be that the question is open because of some indeterminacy in the description of three-dimensional flows by the Navier-Stokes equations[7]. In the absence of definite results one way or the other, however, we will assume, not unreasonably, that the three-dimensional Navier-Stokes equations do have solutions which are classical and unique in the large.

1.2.2 Solution Methods

Solutions of the Navier-Stokes equations must, as we have already noted, generally be obtained numerically on a computer. This involves solving numerical analogues of the Navier-Stokes (or derived) equations in a discretised, as opposed to a continuous, flow field. Because three-dimensional problems involve more complex equations, as well as dimensionally larger flow fields than two-dimensional problems, methods applicable to three-dimensional problems must be relatively efficient, both in terms of speed and storage requirements, if they are to be at all satisfactory. Thus a method involving direct solution of the (discretised) Navier-Stokes equations is not viable, because the use of (a priori) unknown boundary conditions on pressure and the imposition of the continuity constraint mean that the method is very inefficient[8]. Other, less direct, methods must be used instead.

Several such indirect methods are currently available, some of which are closely related to one another, being based on the theory of orthogonal decomposition which we will develop in Chapter 2. The remaining methods are more difficult to classify, being based on a wide variety of theories and techniques. It is interesting to note that it is only methods in the first (related) group which overcome the difficulties concerning pressure and continuity in solving the Navier-Stokes equations. Methods in the second (unrelated) group, on the other hand, make no attempt to overcome these difficulties, and rely for their efficiency on some other technique. We will discuss each group in turn, starting with the group of closely related methods. Note that we will be examining each method with the aim of using it to solve the Navier-Stokes equations primarily for laminar flows in an Eulerian frame of reference. This means that although methods do exist for solving the equations of motion of turbulent flows[(9),(10)], as well as flows in a Lagrangian frame of reference[(11)], we will not discuss them.

An obvious way to eliminate pressure as a dependent variable and to ensure that continuity is automatically satisfied in three-dimensional flow problems is to draw an analogy with the streamfunction/vorticity approach of two-dimensional (and axisymmetric three-dimensional) flow problems. Use of the streamfunction ensures that continuity is satisfied automatically, while the introduction of vorticity eliminates pressure as a dependent variable[(12)]. The generalisation to three dimensions follows from the fact that we can represent any solenoidal vector field as the curl of a so-called "vector potential" field (this will be shown in Chapter 2), so that we can represent the velocity field $\underline{u}$ of an incompressible fluid thus:-

$$\underline{u} = \nabla\wedge \underline{A} \tag{1.3}$$

$\underline{A}$ being the vector potential of the velocity field. Use of the vector potential means that the velocity field automatically satisfies the equation of continuity (1.2), since the divergence of the curl of any vector field vanishes identically. Moreover, as in the two-dimensionel case, the introduction of vorticity, by taking the curl of the terms in the Navier-Stokes equations, eliminates pressure as a dependent variable, since the curl of the gradient of any scalar field vanishes identically. Thus we have the basis of a method for solving the three-dimensional Navier-Stokes equations which, in a relatively simple manner, eliminates pressure as a dependent variable and ensures that continuity is automatically satisfied. (We note in passing that, until recently, there has been some controversy over the boundary conditions appropriate to the vector potential. This controversy has now been completely resolved, the essential point being that the boundary conditions can be considerably simplified in flow-through problems by the use of a scalar, as well as a vector, potential thus:-

$$\underline{u} = \nabla\phi + \nabla_\wedge \underline{A} \qquad (1.4)$$

where $\nabla^2\phi = 0$. This point will be discussed in Chapter 2.) This method of solving the three-dimensional Navier-Stokes equations has been used by Aziz and Hellums[13], who studied three-dimensional thermal convection in a confined fluid heated from below. It has since been used by Holst and Aziz[14] in the study of thermal convection in confined porous media, although they applied it to equations other than the Navier-Stokes equations.

A method which is closely related to the vorticity/potential method also introduces vorticity so as to eliminate pressure from the equations of motion. But then, instead of ensuring that continuity is automatically satisfied using a potential representation, the velocity field $\underline{u}$ is obtained directly from the vorticity field $\underline{\omega}$ by an integral relation which is a generalisation of the well-known Biot-Savart law[15]:-

$$\underline{u}(\underline{x}) = \frac{1}{4\pi}\iiint_R \frac{\underline{\omega}(\underline{y})_\wedge(\underline{x}-\underline{y})}{|\underline{x}-\underline{y}|^3}\, dR \qquad (1.5)$$

$$+ \left[\begin{array}{l}\text{various (known) terms incorporating boundary}\\ \text{conditions on } \underline{u} \text{ and the field } \underline{u} \text{ at infinity}\\ \text{if the flow region R is unbounded}\end{array}\right]$$

where $\underline{x}$ is a point in the flow region R, and integration is over points $\underline{y}$ in R. In fact, obtaining the velocity directly from the vorticity is analytically equivalent to the use of a single vector potential for the velocity (as in equation (1.3)), since equation (1.5) is nothing more than the analytical solution for the curl of the vector potential, as we shall show in Chapter 2. On the other hand, the integral relation method does differ from the potential representation method in one important respect. While the latter method involves solution for the potential(s) at all points in the flow field - which presents certain difficulties if the flow field is unbounded – the integral relation method involves solution for the vorticity field only in regions of non-zero vorticity. This can be a decided advantage, especially in problems with unbounded flow fields, because we can confine our calculations to the regions of effective non-zero vorticity – say, where the magnitude of the vorticity $|\underline{w}|$ is greater than a small number δ. Then, with an accuracy determined by the size of δ, we can solve a flow problem which may be smaller than that resulting from a potential representation of the velocity field. The drawback with the method is that a large amount of computer time is required to evaluate the integral in equation (1.5), and a large amount of storage is necessary for the geometric factors $(\underline{x} - \underline{y})/|\underline{x} - \underline{y}|^3$ etc. Obviously, some sort of balance is involved, but it does seem that, except for small values of time when the non-zero vorticity field is relatively small in extent, the balance is against such an integral relation method. The method has been used, nevertheless, by Wu and Thompson[16], who studied two-dimensional uniform flow past a cylinder in an unbounded region, and by Thompson, Shanks and Wu[17], who studied three-dimensional uniform flow past a slab at various angles of attack, also in an unbounded region.

Both of the above methods use derived variables (vorticity, with or without one or more potentials) in the solution of the Navier-Stokes equations. Two methods related to these use the primitive variables velocity and pressure instead. The second of these methods essentially grew out of the first, which is an approximate method of eliminating pressure p as a dependent variable, and satisfying continuity automatically, at one and the same time. In essence, it involves putting:-

$$\nabla \cdot \underline{u} = \varepsilon p \qquad (1.6)$$

where ε is very small. Clearly, pressure can be eliminated immediately from the Navier-Stokes equations (1.1) by means of this expression. This so-called (for obvious reasons) "artificial compressibility" method can then be used to give an approximate solution of the Navier-Stokes equations which should be better the smaller we make ε. Témam has shown that, in two-dimensional problems, the solution of the approximate equations exists, is unique, and converges to the true solution of the Navier-Stokes equations in the limit as ε tends to zero. In three-dimensional problems, on the other hand, the solution, which can be shown to exist, is not necessarily unique, and only tends to the true solution of the Navier-Stokes equations under rather specific conditions(18),(19). (This, not surprisingly, is precisely analogous to the existence and uniqueness question of solutions of the Navier-Stokes equations themselves, which we examined in sub-section 1.2.1 above.) The disadvantage with the artificial compressiblity method is that in practice ε has to be very small (as also does the size of the time-step involved in the discretisation of unsteady flow problems) to obtain an effectively incompressible solution(20). Nevertheless, Chorin(21) and Plows(22) have both applied the artificial compressibility method to two-dimensional steady-state thermal convection in a layer of fluid of infinite lateral extent heated from below.

The artificial compressibility method essentially involves an approximation to the solution of the Navier-Stokes equations, the accuracy of which is determined by the parameter ε. It is clear that we should be able to increase the accuracy of the solution by iterating to effective incompressibility. In other words, we relax the constraint of continuity in the manner of the artificial compressibility method, and obtain approximations $\underline{u}_0$ and p_0 to the velocity and pressure fields, respectively. Now $\underline{u}_0$ will not generally be solenoidal, but if we update our estimate of p_0 by putting:-

$$p_1 = p_0 + \lambda \nabla \cdot \underline{u}_0 \qquad (1.7)$$

λ being a suitably chosen parameter, we should obtain an improved estimate $\underline{u}_1$ to the velocity field, which should in turn give successively better estimates $\underline{u}_2$, $\underline{u}_3$, $\underline{u}_4$.... which will satisfy continuity more and more closely. We may, therefore, look on this artificial compressibility – plus – iteration scheme as a numerical method for projecting the approximate solution $\underline{u}_0$ onto the space of solenoidal vector fields, in other words onto the space of vector fields that satisfy the continuity equation (1.2). Thus we arrive at a rather different method for solving the Navier-Stokes equations. It essentially involves obtaining some solution $\underline{u}_0$ which satisfies the Navier-Stokes equations and boundary conditions, but not, in general, continuity so that $\underline{u}_0$ is not solenoidal. We project $\underline{u}_0$, therefore, onto the space of solenoidal vector fields (numerically), and hence obtain the true solution of the equations (to within some small error). This is the basis of the so-called "projection" method, whose theory has been developed by Chorin(23),(24) and Témam(25) out of the theory of the artificial compressibility method. As with the artificial compressiblity method, no completely general proofs of the uniqueness and convergence of solutions to the true solution of the Navier-Stokes equations can be given, although certain weaker results are available. In spite of this, the projection method has been used widely in the solution of the Navier-Stokes equations. Chorin(26), and Veltishchev and Zelnin(27), have used it to study three-dimensional thermal convection in a confined layer of fluid heated from below. Crane(28) has used the projection method to study axisymmetric flow development in a circular pipe, and Peskin(29) has used it to study flow patterns around heart valves. A slightly modified method has been used by Somerville(30) to study small-scale thermal convection in the atmosphere, and a differently modified method has been used by Padmanabhan, Ames, Kennedy and Hung(31) to study the middle stages of the collapse of a fluid mass surrounded by a linearly stratified fluid. A slightly different method again, more closely related to a predictor-corrector approach, has been developed by Patankar and Spalding(32).

This completes our discussion of the group of four related methods. The theoretical basis of two of them, the vorticity/potential and the projection methods, will be discussed in Chapter 2. The basis of the vorticity/integral relation method will not be discussed explicitly, however, and that of the artificial compressibility method not at all, because the disadvantages associated with each mean that their immediate alternatives (the vorticity/potential and the projection methods, respectively) are much more suitable as general methods for solving the Navier-Stokes equations. We will return to this point at the end of this sub-section.

The second group of (unrelated) methods, which we will now discuss, attempt neither to eliminate pressure as a dependent variable, nor to ensure that continuity is satisfied automatically: the methods rely for their efficiency on some other special technique instead. Because these methods are all (apparently) unrelated, there is no natural order in which to discuss them. So, to emphasise the difference between this group of methods and the group of related methods, we will start with two methods which, far from eliminating pressure as a dependent variable, actually use it, rather than velocity, as the principal variable.

If we take the divergence of the terms in the Navier-Stokes equations (1.1), we obtain a Poisson equation for pressure:-

$$\nabla^2 p = -\rho \nabla . \left[(\underline{u} . \nabla) \underline{u} \right] \qquad (1.8)$$

(We have assumed, without loss of generality, that all body forces derivable from a single-valued potential are incorporated into the pressure term, so that $\underline{F}$ is solenoidal; the reason why this is so is given in sub-section 2.3.2.) As a rule, Poisson equations are notoriously difficult equations to solve numerically. Moreover, the Poisson equation (1.8) involves unknown Neumann boundary conditions on pressure which, as we have already noted, form one of the main difficulties in solving the Navier-Stokes equations. Thus if a method of solving the Poisson equation is to be at all satisfactory, it must be really efficient. For flow regions of relatively simple geometry, an obvious way to solve the Poisson equation (1.8) is to use the Green's function[33] appropriate to the region to give us the solution directly. Unless the flow region has a relatively simple geometry, however, determining the appropriate Green's function is not at all straightforward, so that this approach is not really suitable as a method of solving the Navier-Stokes equations for general flow problems. It has, nevertheless, been used by Lyczowski and Gidaspow[34] in the study of two-dimensional flow and reaction in rectangular fuel cells, although they applied it to equations other than the Navier-Stokes equations.

An alternative to using the Green's function to solve the Poisson equation (1.8) is to use an eigenfunction expansion for the pressure field. Suppose we put:-

$$p = p^* + p' \qquad (1.9)$$

where p^* is any reasonable scalar field which satisfies the non-homogeneous Neumann boundary conditions on the pressure field p. Given p^*, we can determine $\nabla^2 p^*$, and hence obtain a Poisson equation for p':-

$$\nabla^2 p' = -\rho\nabla\cdot\left[(\underline{u}\cdot\nabla)\underline{u}\right] - \nabla^2 p^* \qquad (1.10)$$

Because p' has homogeneous Neumann boundary conditions, we can expand p' in terms of an infinite set of eigenfunctions f_i:-

$$p' = \sum_{i=1}^{\infty} c_i f_i \qquad (1.11)$$

where the c_i are constants. Clearly, the use of an infinite set of eigenfunctions is going to present problems when we solve the Poisson equation (1.10) numerically, but, because numerical solution involves discretisation of the flow field, only a finite number of eigenfunctions will (and, indeed, can) be involved (N, say):-

$$p' = \sum_{i=1}^{N} c_i f_i \qquad (1.12)$$

The N eigenfunctions f_i are determined by the geometry of the flow field; so constructing the eigenfunction expansion (1.12) appropriate to a particular problem involves determining the set of N constants c_i, and this can be performed quite efficiently. The drawback with the method is that obtaining the appropriate eigenfunctions is difficult except in flow fields with a relatively simple geometry (which is precisely analogous to the drawback with the Green's function method). The method is, therefore, unsuitable for general flow problems, but it has been used by Williams[35] in the study of three-dimensional thermal convection between two coaxial circular cylinders of equal length which rotate relative to one another.

Several methods for solving the three-dimensional Navier-Stokes equations use a Galerkin type of (approximate) representation for the velocity field $\underline{u}$:-

$$\underline{u} = \sum_{i=1}^{M} c_i \underline{v}_i \qquad (1.13)$$

where the M coefficients c_i are constants (spatially), and the M $\underline{v}_i$ are (known) vector fields which satisfy the boundary conditions on $\underline{u}$. Such methods rely for their efficiency on the rapid determination of the M coefficients c_i, and can be used in a variety of ways, the most common being in conjunction with variational techniques in a so-called "finite-element" formulation. This formulation involves the use of elements of finite size and simple geometrical shape which (together) define the flow field. The geometry of each element, which may vary between elements, is characterised by a finite number of nodes, and the local flow properties in each element are weighted averages of the flow properties at the nodes of that element. Oden and Wellford[36] have given a general formulation of the method for flow problems, and have applied it to several relatively simple two-dimensional examples, while Bratanow and Ecer[37] have used the method to study three-dimensional flow about oscillating aerofoils. The method does not, however, overcome any of the difficulties associated with pressure and continuity involved in solving the Navier-Stokes equations, and so offers no real advantages for general problems.

A method which we may regard as being a cross between a finite-element and a Lagrangian method is the so-called "marker and cell" method, which has two distinctive features. First, it involves the use of special markers, or inertia-less particles, whose motion is determined in a Lagrangian manner by the flow:-

$$\underline{x}(t) = \underline{x}(0) + \int_0^t \underline{u}(\tau)d\tau \qquad (1.14)$$

where $\underline{x}$ denotes the position of the marker, $\underline{u}$ its (Eulerian) velocity, and t the time. Secondly, it involves the use of a finite network of cells (which we might equally well call finite-elements). The great advantage of the marker and cell method is in problems involving free surfaces, since the markers can be used to locate the surfaces as the calculation proceeds. On the other hand, the method has the same drawbacks as the Galerkin and related methods and, in any case, the "marker" feature can be incorporated into other methods of solution of the Navier-Stokes equations without difficulty. The method has, nevertheless, been used widely; Welch, Harlow, Shannon and Daly(38), for example, have used it to study a variety of flow problems.

The final method which we will discuss for solving the three-dimensional Navier-Stokes equations has been developed by Roache(39) for the solution of steady-state flow problems using iterative techniques which, in contrast to common iterative techniques, are not time-like in any sense. So far, the method has only been applied to two-dimensional problems, but extensions to three dimensions should present no difficulty. The restriction to steady-state problems, however, means that the method is clearly unsuitable for use as a general method of solving the Navier-Stokes equations.

This completes our discussion of currently available methods of solving the three-dimensional Navier-Stokes equations. The immediate conclusion that we can draw is that none of the second group of (unrelated) methods is entirely suitable for solving the Navier-Stokes equations for general flow problems, although each offers certain advantages in particular types of problem. We are thus led back to the first group of four related methods for use in general flow problems. We can eliminate two of these, the vorticity/integral relation and artificial compressibility methods, because their immediate alternatives, the vorticity/potential and projection methods, respectively, are, as we have already discussed, more suitable for use in general problems. Our conclusion, therefore, is that general problems involving the three-dimensional Navier-Stokes equations should be solved using either the vorticity/potential method or the projection method. It is difficult at this stage to comment further on these methods, or on their relative merits; to do this, we need to look at each of them more closely. We shall do so in Chapter 2.

Chapter 2 – The Orthogonal Decomposition of Lebesgue Square-Integrable Vector Fields and Applications to Hydrodynamics

2.1 Introduction

In this chapter, we will show how three-dimensional Lebesgue square-integrable vector fields may be decomposed into mutually orthogonal components, and how we may apply this decomposition to obtain solutions of the Navier-Stokes equations. The decomposition, which is formalised in the Orthogonal Decomposition Theorem in sub-section 2.2.2, is essentially derived from Helmholtz's theorem, which states that if $\underline{v}$ is an arbitrary finite continuously differentiable vector field defined throughout three-dimensional Euclidean space E_3, and vanishing at infinity, then we can put[40]:-

$$\underline{v} = \nabla \phi + \nabla \wedge \underline{A}$$

where ϕ is a scalar potential and $\underline{A}$ is a vector potential. It was, in fact, Helmholtz who first employed the term "velocity potential" used in the so-called potential flows of classical hydrodynamics, in which the velocity field is expressed as the gradient of a scalar potential field. The theory of such flows was brought to a relatively high degree of sophistication[41] using the results of classical potential theory[42]. This theory, which generalised Helmholtz's theorem to regions $R \subset E_3$, still required, however, that the field $\underline{v}$ be continuously differentiable and everywhere finite. The problem of extending the results of potential theory to more general vector fields was effectively solved by Weyl[43] for fields defined in subspaces $R \subset E_3$, and by Kodaira[44] and Hodge[45] for fields defined in Riemannian manifolds. The complete extension to quite general vector fields defined in $R \subset E_3$ was performed by Bykhovski and Smirnov[46] using the techniques of functional analysis.

In functional analysis, whole spaces or sets, rather than particular elements, are considered, and very general results can be obtained. The Orthogonal Decomposition Theorem of sub-section 2.2.2, which is due to Bykhovski and Smirnov, thus requires only that the vector field $\underline{v}$ be Lebesgue square-integrable (two other minor assumptions are also required, concerning behaviour at infinity if R is partly or wholly unbounded, and smoothness of the boundary δR of R).

As we shall see in section 2.2, the theory of orthogonal decomposition is of such generality that it can be applied without essential restriction to obtain solutions of the equations of motion of a fluid; in particular, of the Navier-Stokes equations. In other words, the requirement of Lebesgue square-integrability, together with the two other minor assumptions noted above, adds nothing essential to the assumptions inherent in the hypotheses used to derive the equations of motion of a fluid. Thus orthogonal decomposition theory is applicable quite generally to all situations in which the equations of motion themselves are valid. And the reason why the theory is so valuable in this context is that for flows of incompressible fluids, whether Newtonian or non-Newtonian, the use of such a decomposition will, as we shall see in section 2.3, automatically ensure that the equation of continuity is satisfied, and also eliminate pressure as a dependent variable. Thus two of the principal problems involved in solving the equations of motion are removed for incompressible flows. The theory can also be applied to compressible flows, although its utility is then less apparent.

In the next section, 2.2, we shall introduce the concept of orthogonality of elements of arbitrary spaces, and the concept of orthogonal decomposition. We shall then state the Orthogonal Decomposition Theorem, the assumptions underlying it, and give it a physical interpretation. Finally, in section 2.3, we shall show how the decomposition can be applied to give two essentially different general methods of solution of the equations of motion of a fluid, with particular reference to the Navier-Stokes equations.

2.2 Orthogonal Decomposition Theory

2.2.1 The Orthogonal Decomposition of Arbitrary Hilbert Spaces

Consider an arbitrary Hilbert space X with typical elements x and y. The definition of a Hilbert space(47) implies that it has:-

(i) a structure, or topology;

(ii) a metric, $\rho(x,y)$, which defines the distance between elements x and $y \in X$;

(iii) a norm, $\|x\|$, which defines the size of elements $x \in X$; and

(iv) an inner product, (x,y), which defines the relative orientation of elements x and $y \in X$.

Suppose, for example, that X is the space of n-dimensional real-valued vectors $\underline{x} = (x_1 \ldots x_n)$. Then we could define the metric thus:-

$$\rho(\underline{x},\underline{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}; \quad \underline{x},\ \underline{y} \in X.$$

The norm could be defined thus:-

$$\|\underline{x}\| = \sqrt{\sum_{i=1}^{n} x_i^2} \quad ; \underline{x} \in X$$

and the inner product thus:-

$$(\underline{x},\underline{y}) = \sum_{i=1}^{n} x_i y_i \quad ; \underline{x},\ \underline{y} \in X.$$

These definitions are familiar as the Euclidean metric, norm, and scalar product.

The property which is relevant to the theory of orthogonal decomposition is the existence of an inner product in a Hilbert space. By analogy with the concept of orthogonality in vectors, we say that two elements x and $y \in X$ are "mutually orthogonal" if their inner product (x,y) vanishes identically. Consider a subspace Y of X, and denote all elements of Y by y. Let z denote the difference $x - y$, where $x \in X$, and

$y \in Y \subset X$. Suppose, with a suitable definition of the inner product, that $(y,z) \equiv 0$ for all $y \in Y$. We then say that the subspaces Y and Z, where $z \in Z$, are "mutually orthogonal", and put:-

$$X = Y \oplus Z \tag{2.1}$$

and:-

$$x = y + z \quad (x \in X,\ y \in Y,\ z \in X \ominus Y) \tag{2.2}$$

the circle round the operations sign denoting orthogonality. Equation (2.1) now represents the "orthogonal decomposition" of the Hilbert space X. Moreover, given X and Y, and any element $x \in X$, we can easily show that the decomposition (2.2) is unique[48].

2.2.2 <u>The Orthogonal Decomposition of Lebesgue Square-Integrable Vector Fields</u>

We now apply the concepts of the preceding sub-section to the Hilbert space of Lebesgue square-integrable vector fields, in particular to three-dimensional vector fields defined in a region $R \subset E_3$. We denote this space by $L_2(R)$, and say that a vector field $\underline{v} \in L_2(R)$ if and only if:-

$$\|\underline{v}\|^2 = \iiint_R \sum_{i=1}^{3} v_i^2 \, dR < \infty \tag{2.3}$$

The inner product is defined for vectors $\underline{u}, \underline{v} \in L_2(R)$ by:-

$$(\underline{u},\underline{v}) = \iiint_R \sum_{i=1}^{3} u_i v_i \, dR$$

where the integrals are understood in the sense of Lebesgue[49]. We see immediately that Lebesgue square-integrability (also referred to as "square-summability") does not imply that $\underline{v}$ needs to be smooth. It may be piecewise continuous, and even point-, line-, or area-wise infinite, provided only that the (Lebesgue) integral of its magnitude squared is bounded. Physically, this concept is both reasonable and meaningful. If, for example, we identify $\underline{v}$ with the velocity field of an incompressible fluid, then Lebesgue square-integrability implies finite kinetic energy. More generally, it implies finite quantities, which is a natural condition for any physical system.

One further consideration is necessary before we state the Orthogonal Decomposition Theorem. We must describe the topology of a general three-dimensional region $R \subset E_3$. The boundary of R, which we denote by δR, comprises:-

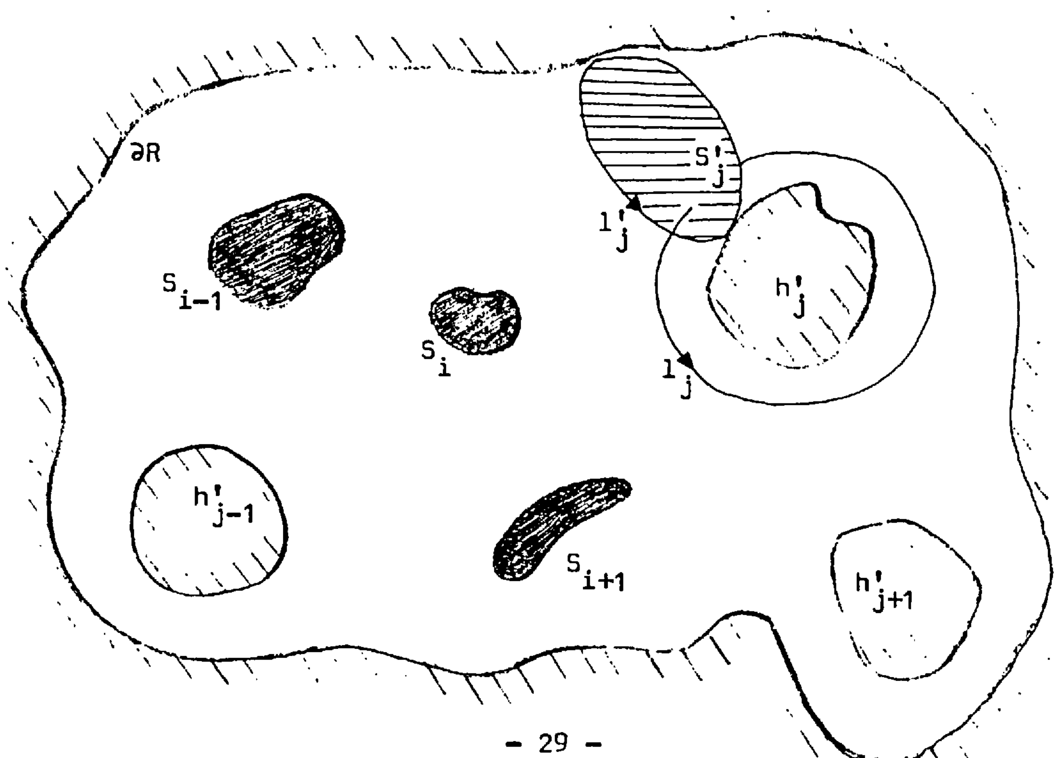
(i) the outer boundary ∂R of R, part or all of which may be at infinity; and

(ii) m surfaces S_i contained entirely within R, and thus disconnected from ∂R.

Thus we may write:-

$$\delta R \quad = \quad \partial R \cup S_1 \cup S_2 \cup \ldots \cup S_m \qquad (2.4)$$

where $\cup$ implies (Boolean) union. The region R may be multiply-connected, and contain n holes of type h'_j like the hole in a torus, so that R is, in fact, (n+1)-ply connected. Such holes h'_j are characterised by closed contours l_j and by surfaces S'_j. The contours of type l_j are contained within R and completely encircle the hole h'_j; thus they cannot be continuously shrunk to a point without leaving R. The surfaces of type S'_j are bounded by closed contours l'_j on δR which cannot be continuously shrunk to a point without leaving δR. (Note that the surfaces S_i can also contain holes of type h'_j.)



We can now state the Orthogonal Decomposition Theorem for the space $L_2(R)^{(50)}$:-

$$L_2(R) = \mathring{G}(R) \oplus U_1(R) \oplus U'(R) \oplus U_2(R) \oplus \mathring{J}(R) \qquad (2.5)$$

where:-

$\mathring{G}(R)$ is the closure of the space of infinitely differentiable vector fields of the form $\nabla\phi$, with ϕ vanishing identically on δR.

$U_1(R)$ is the closure of the space of infinitely differentiable vector fields of the form $\sum_{i=1}^{m} \alpha_i \nabla h_i$, such that the (scalar) Laplacian of h_i, $\nabla^2 h_i$, $\equiv 0$ in R; m is the number of surfaces S_i; α_i is a constant; $h_i = \delta_{ik}$ (the Kronecker delta) on S_k and vanishes on ∂R; and the two-period of ∇h_i, $\iint_S \nabla h_i . \underline{n}\, dS$, where $\underline{n}$ is the unit outer normal to the element dS of S, vanishes identically unless S is a closed surface completely enclosing S_i.

$U'(R)$ is the closure of the space of infinitely differentiable vector fields of the form ∇h, such that $\nabla^2 h \equiv 0$ in R; and the two-period of ∇h over any closed surface S vanishes identically.

$U_2(R)$ is the closure of the space of infinitely differentiable vector fields of the form $\sum_{j=1}^{n} \beta_j \nabla h^j$, such that $\nabla^2 h^j \equiv 0$ in R; n is the number of holes h'_j; β_j is a constant; and the one-period of ∇h^j, $\oint_{l_k} \nabla h^j . \underline{dl}$, vanishes on all closed contours l_k except for contours l_j encircling the hole h'_j, where the one-period equals unity (this implies that each h^j is many-valued). Also, the normal derivative of h^j vanishes identically on δR; and $\iint_{S'_k} \nabla h^j . \underline{n}\, dS$ vanishes identically except on the surface S'_j.

$\mathring{J}(R)$ is the closure of the space of infinitely differentiable vector fields of the form $\nabla_\wedge \underline{A}$, such that $\nabla . \underline{A} \equiv 0$ in R; the tangential components of $\underline{A}$ vanish identically on δR; and $\iint_{S'_k} \nabla_\wedge \underline{A} . \underline{n}\, dS'$ vanishes identically for all S'_k.

In the above definitions, we have used the term "closure" or, strictly, "closure in norm". By the closure of a space, we mean all the elements of the space plus their limit points. Thus the closure of the open interval $0 < x < 1$ is the closed interval $0 \leqslant x \leqslant 1$. Similarly, the closure of the space of all smooth scalar functions defined on the one-dimensional Euclidean space E_1 and vanishing at infinity includes the Dirac delta function. Closure "in norm" implies that x is the limit point of a sequence $\{x_1, x_2, \dots x_m, \dots\}$ of elements $\in X$ if $\|x - x_m\| \to 0$ as m tends to infinity, with the norm relevant to the space X being used (which, in the present context, means the $L_2(R)$ norm defined in equation (2.3)). The concept of closure means that all the subspaces defined above are closures (in norm) of "dense" spaces[51]. A space X is said to be dense if every element $x \in X$ has another element $y \in X$ near it such that the distance between x and y, $\rho(x,y)$, can be made arbitrarily small. We see, therefore, that an arbitrary vector field from $\mathring{G}(R)$, say, need not be expressible in the form $\nabla\phi$. Since, however, vector fields of the form $\nabla\phi$ are dense in $\mathring{G}(R)$, we can approximate any vector field in $\mathring{G}(R)$ as closely as we desire. Alternatively, we say that a vector field in $\mathring{G}(R)$ is of the form $\nabla\phi$ "almost everywhere". It follows from this that we can approximate an arbitrary Lebesgue square-integrable vector field $\underline{v}$ defined in a region $R \subset E_3$ as closely as we like thus:-

$$\underline{v} = \nabla\phi + \sum_{i=1}^{m} \alpha_i \nabla h_i + \nabla h + \sum_{j=1}^{n} \beta_j \nabla h^j + \nabla \wedge \underline{A} \quad (2.6)$$

where:-

$\phi \equiv 0$ on δR;

the α_i and β_j are constants;

m is the number of surfaces S_i, and n is the number of holes h'_j;

$\nabla^2 h_i \equiv 0$, $\nabla^2 h \equiv 0$, $\nabla^2 h^j \equiv 0$ in R for all i, j;

$h_i = \delta_{ik}$ on S_k, and $h_i \equiv 0$ on ∂R for all i;

$c^2(\nabla h_i)$, the two-period of ∇h_i, $\equiv 0$ except on surfaces completely surrounding S_i for all i;

$C^2(\nabla h) \equiv 0$;

$C^1(\nabla h^j)$, the one-period of ∇h^j, $= \delta_{jk}$ on contour l_k for all j;

$\nabla h^j.\underline{n} \equiv 0$ on δR for all j;

$\iint_{S'_k} \nabla h^j.\underline{n}\, dS' \equiv 0$ unless k = j, for all j;

$\underline{A}_\wedge \underline{n} \equiv \underline{0}$ on δR;

$\nabla.\underline{A} \equiv 0$ in R;

and $\iint_{S'_k} \nabla_\wedge \underline{A}.\underline{n}\, dS' \equiv 0$ for all S'_k.

Equation (2.6) gives the orthogonal decomposition of a Lebesgue square-integrable three-dimensional vector field into mutually orthogonal components. It is not, however, the only possible decomposition of the field $\underline{v}$, for we can show that we can entirely equivalently put $\nabla_\wedge \underline{H}_i$ instead of ∇h_i, $\nabla_\wedge \underline{H}$ instead of ∇h, and $\nabla_\wedge \underline{H}^j$ instead of ∇h^j, so that:-

$$\underline{v} = \nabla \phi + \sum_{i=1}^{m} \alpha_i \nabla_\wedge \underline{H}_i + \nabla_\wedge \underline{H} + \sum_{j=1}^{n} \beta_j \nabla_\wedge \underline{H}^j + \nabla_\wedge \underline{A} \qquad (2.7)$$

with the same conditions as before on ϕ and $\underline{A}$, and corresponding (but more complex) conditions of the m $\underline{H}_i$, $\underline{H}$, and the n $\underline{H}^j$. Since, however, the decomposition (2.6) is easier to manipulate than decomposition (2.7), we will develop further results for the former decomposition (unless otherwise stated), with the understanding that all results can be transferred with slight modifications to the latter.

We can give a physical interpretation of the decomposition (2.6) by considering the flow of a fluid in a region $R \subset E_3$, and identifying $\underline{v}$ with its velocity field. For other vector fields, the analogy with velocity will be useful. In any three-dimensional region, we can identify three types of motion, namely vibration, rotation, and translation. Moreover, we can identify three distinct types of translation. First, there is translation through the outer boundary ∂R. Secondly, there is translation through the m surfaces S_i contained entirely within R. Finally, there is translation round the n holes h'_j. We might expect, therefore, to be able to decompose a velocity field into a linear sum of motions of the types mentioned. In fact, this is precisely what equation (2.6) represents. The $\nabla\phi$ term represents the vibrational, or compressible, component of the flow, and $\nabla_\wedge \underline{A}$ represents its rotational component. The ∇h term represents translation through ∂R, the $\sum_{i=1}^{m} \alpha_i \nabla h_i$ term represents translation through the m surfaces S_i, and the $\sum_{j=1}^{n} \beta_j \nabla h^j$ term represents translation round the n holes h'_j.

Using this physical interpretation, it becomes an easy matter to determine which terms in the decomposition (2.6) are relevant to a particular velocity field. (Again, for other vector fields, the analogy with velocity will be useful.) Thus if the fluid is incompressible, it can have no vibrational, or compressible, component $\nabla\phi$. If it is irrotational, it can have no rotational component $\nabla_\wedge \underline{A}$. If the region R is bounded by a solid wall δR, then the normal component of the velocity must vanish on δR. It then follows that we can express $\underline{v}$ in the form $\nabla\phi + \sum_{j=1}^{n} \beta_j \nabla h^j + \nabla_\wedge \underline{A}$ almost everywhere. Similarly, if there is flow through the outer boundary ∂R, but not through any of the m surfaces S_i, then the decomposition of $\underline{v}$ will contain no $\sum_{i=1}^{m} \alpha_i \nabla h_i$ term. Again, if the flux of $\underline{v}$ through each of the n surfaces S'_j characterising the multiply-connected region R vanishes identically, the decomposition of $\underline{v}$ will contain no $\sum_{j=1}^{n} \beta_j \nabla h^j$ term, and so on.

We now list the assumptions implicit in the decomposition (2.6), so that they are quite clear:-

Assumption I:- the vector field $\underline{v}$ must be Lebesgue square-integrable.

The meaning and implications of this assumption have already been discussed.

Assumption II:- if the region R is unbounded, then we require that $\underline{v}$, $\nabla\phi$, etc are of order $1/|\underline{x}|^2$, and their first derivatives are of order $1/|\underline{x}|^3$, as the space coordinate $\underline{x}$ tends to infinity[52]. This condition is, in fact, consistent with the assumption that $\underline{v} \in L_2(R)$.

Assumption III:- the boundary δR of R must be piecewise continuous, so that it has a unique normal almost everywhere. This means that point and line discontinuities are generally allowable. Clearly, most surfaces encountered in physical problems are piecewise continuous, at least to a similar degree as the continuum hypothesis holds, and hence are satisfactory. (If we further require that $\underline{v}$, $\nabla\phi$, etc belong to the Sobolev space $W_2^p(R)$, by which we mean that $\underline{v}$, $\nabla\phi$, etc and their derivatives up to and including order p are Lebesgue square-integrable in R, then we can show that the boundary δR must be at least (p+1) times piecewise continuously differentiable[53].)

Finally, it should be noted that the decomposition (2.6) is essentially a decomposition in space, and not in time. Thus it does not matter whether the boundary δR is fixed in E_3 with respect to time or not. So, if the vector field $\underline{v}$ represents the velocity of a fluid, and the surfaces S_i represent particles suspended in the fluid, for example, we can use decomposition (2.6) for $\underline{v}$ whether or not the surfaces S_i move with time.

2.2.3 Analytical Projections Derived from the Orthogonal Decomposition of Lebesque Square-Integrable Vector Fields

Consider a Hilbert space X with mutually orthogonal subspaces Y and Z such that equations (2.1) and (2.2) hold. Then we define the "projection operator" P_Y from X onto Y by:-
$$P_Y x = y$$

and we call y the "orthogonal projection" of $x \in X$ onto Y. Clearly, $P_Y y = y$ and $P_Y z \equiv 0$. Also, $P_Y + P_Z \equiv I$, the identity operator, and $P_Y(P_Z x) = P_Y z \equiv 0$ (in other words, the projection operators P_Y and P_Z are mutually orthogonal). We now apply the concepts of projection operators and orthogonal projections to the decompositions (2.5) and (2.6). If $\underline{v}_{\mathring{G}}$ denotes the orthogonal projection of $\underline{v} \in L_2(R)$ onto $\mathring{G}(R)$, and $P_{\mathring{G}}$ denotes the projection operator from $L_2(R)$ onto $\mathring{G}(R)$, and so on, we may put:-

$$\underline{v} = \underline{v}_{\mathring{G}} + \underline{v}_{U_1} + \underline{v}_{U'} + \underline{v}_{U_2} + \underline{v}_{\mathring{J}} \tag{2.8}$$

$$\text{or } \underline{v} = P_{\mathring{G}}\underline{v} + P_{U_1}\underline{v} + P_{U'}\underline{v} + P_{U_2}\underline{v} + P_{\mathring{J}}\underline{v} \tag{2.9}$$

If we can find the form of the projection operators $P_{\mathring{G}}$ etc, then we can determine $\underline{v}_{\mathring{G}}$ etc, and hence the decomposition (2.6). Now it is clear that $\nabla\phi$ approximates $\underline{v}_{\mathring{G}}$ (to any desired accuracy), $\nabla\wedge\underline{A}$ approximates $\underline{v}_{\mathring{J}}$, and so on. So all we have to do is find the form of $\nabla\phi$, $\nabla\wedge\underline{A}$, etc, and to do this, all we require are some standard vector identities and integral theorems.

It follows from equation (2.6) and the conditions immediately following it that:-

$$\left.\begin{aligned}
\nabla\cdot\underline{v} &= \nabla\cdot\underline{v}_{\mathring{G}} = \nabla^2\phi \\
\nabla\wedge\underline{v} &= \nabla\wedge\underline{v}_{\mathring{J}} = \nabla\wedge\nabla\wedge\underline{A} = -\Delta\underline{A}, \text{ since } \nabla\cdot\underline{A} \equiv 0 \\
\nabla^2 h_i &\equiv 0 \text{ for all } i \\
\nabla^2 h &\equiv 0 \\
\nabla^2 h^j &\equiv 0 \text{ for all } j
\end{aligned}\right\} \tag{2.10}$$

where Δ is the vector Laplacian. It is easy to show, using Green's theorems[54], together with standard vector identities, that the principal value of the general solution of the equation:-

$$\nabla^2\psi = f \tag{2.11}$$

where ψ and f are scalar functions of position $\underline{x} \in E_3$

ψ has continuous first derivatives in R and on δR

ψ has continuous second derivatives in R and ψ can be single- or many-valued

is (to within at most an arbitrary constant):-

$$\psi(\underline{x}) = -\frac{1}{m\pi}\iiint\limits_{R}\frac{f(\underline{y})}{r}\,dR + \frac{1}{m\pi}\iint\limits_{\delta R}\frac{1}{r}\nabla\psi.\underline{n}\,dS - \frac{1}{m\pi}\iint\limits_{\delta R}\psi\nabla(1/r).\underline{n}\,dS$$

$$-\frac{1}{m\pi}\sum_{j=1}^{n}\left[\tfrac{1}{2}c_j\iint\limits_{S_j'^{+}-S_j'^{-}}\nabla(1/r).\underline{n}\,dS\right] \qquad (2.12)$$

where $m = 4$ if $\underline{x} \in R$, 2 if $\underline{x} \in \delta R$, 0 otherwise;

$\underline{y} \in R$;

r is the distance between $\underline{x}$ and the element dR of volume, or dS of area;

$S_j'^{+}$ and $S_j'^{-}$ denote opposite sides of the surface S_j';

$\underline{n}$ is the unit outer normal to the element dS;

and c_j is the one-period of $\nabla\psi$ around the closed contour l_j i.e.

$$c_j = C^1(\nabla\psi) = \psi\Big|_{S_j'^{+}} - \psi\Big|_{S_j'^{-}}$$

If ψ is harmonic (i.e. $f \equiv 0$), then the first integral above vanishes. If ψ is single-valued (i.e. $c_j \equiv 0$ for all j), then the final integral above vanishes.

From the general solution (2.12) to the equation (2.11), it is easy to obtain expressions for $\emptyset$, the m h_i, h, and the n h^j, using the conditions listed under equation (2.6). Then provided we can differentiate under the integral sign, which we can do if the result is bounded or sufficiently smooth[55], we obtain expressions for $\nabla\emptyset$ etc:-

$$\nabla\emptyset(\underline{x}) = \frac{1}{m\pi}\left[\iiint\limits_{R}\frac{\underline{r}}{r^3}\nabla.\underline{v}(\underline{y})\,dR - \iint\limits_{\delta R}\frac{\underline{r}}{r^3}\frac{\partial\emptyset}{\partial n}\,dS\right] \qquad (2.13)$$

$$\sum_{i=1}^{m}\alpha_i\nabla h_i(\underline{x}) = \sum_{i=1}^{m}\frac{\alpha_i}{m\pi}\left[-\iint\limits_{\delta R}\frac{\underline{r}}{r^3}\frac{\partial h_i}{\partial n}\,dS + \iint\limits_{S_i}\underline{q}\,dS\right] \qquad (2.14)$$

$$\nabla h(\underline{x}) = \frac{1}{m\pi}\left[-\iint\limits_{\delta R}\frac{\underline{r}}{r^3}\frac{\partial h}{\partial n}\,dS + \iint\limits_{\delta R}h\,\underline{q}\,dS\right] \qquad (2.15)$$

$$\sum_{j=1}^{n} \beta_j \nabla h^j(\underline{x}) = \sum_{j=1}^{n} \frac{\beta_j}{m\pi} \left[\iint_{\delta R} h^j \, \underline{q} \, dS + \frac{1}{2} \iint_{S_j'^{+} - S_j'^{-}} \underline{q} \, dS \right] \quad (2.16)$$

where $\underline{q} = -(1/r^3)\underline{n} + (3/r^5)(\underline{r}.\underline{n})\underline{r}$ and $\underline{r} = \underline{x} - \underline{y}$ (2.17)

To obtain an expression for $\nabla_\wedge \underline{A}$, we note that we cannot use equation (2.12) for each component of $\underline{A}$ since the k-th component of the Laplacian of $\underline{A}$ does not (except in rectangular Cartesian coordinates) equal the Laplacian of the k-th component of $\underline{A}$ i.e. $(\Delta\underline{A})_k \neq \nabla^2(A_k)$[56]. (It is for this reason that a careful distinction has been made throughout between the scalar Laplacian (∇^2) and the vector Laplacian (Δ).) Instead, we solve the equation $\nabla_\wedge\nabla_\wedge\underline{A} = \nabla_\wedge\underline{v}$ in R. Using vectorial equivalents of Green's theorems, together with standard vector identities, it is easy to show that the solution of the equation:-

$$\nabla_\wedge\nabla_\wedge\underline{\psi} = \underline{f} \quad (2.18)$$

is:-

$$\nabla_\wedge\underline{\psi}(\underline{x}) = \frac{1}{m\pi}\left[\iiint_R \frac{\underline{f}(\underline{y})_\wedge \underline{r}}{r^3} \, dR - \iint_{\delta R} (\nabla_\wedge \underline{\psi}.\underline{n}) \frac{\underline{r}}{r^3} \, dS + \iint_{\delta R} \frac{((\nabla_\wedge\underline{\psi})_\wedge \underline{n})_\wedge \underline{r}}{r^3} \, dS \right] \quad (2.19)$$

with m etc defined as above. Note that we have obtained $\nabla_\wedge\underline{\psi}$ directly, as opposed to $\underline{\psi}$. This is because $\nabla_\wedge\underline{\psi}$ can be obtained in a particularly elegant form, and also because we do not actually require an expression for $\underline{\psi}$. Thus the expression for $\nabla_\wedge\underline{A}$ is:-

$$\nabla_\wedge\underline{A}(\underline{x}) = \frac{1}{m\pi}\left[\iiint_R \frac{(\nabla_\wedge\underline{v}(\underline{y}))_\wedge \underline{r}}{r^3} \, dR + \iint_{\delta R} \frac{((\nabla_\wedge\underline{A}(\underline{y}))_\wedge \underline{n})_\wedge \underline{r}}{r^3} \, dS \right] \quad (2.20)$$

We have thus (implicitly, admittedly) obtained the projections of $\underline{v}$ onto the five subspaces $\mathring{G}(R)$, $U_1(R)$, $U'(R)$, $U_2(R)$ and $\mathring{J}(R)$ - or, to be more precise, we have obtained approximations to the projections which may be as close as we desire. One point to note is that the projections onto $\mathring{G}(R)$ and $\mathring{J}(R)$ involve $\nabla.\underline{v}$ and $\nabla_\wedge \underline{v}$, respectively (c.f. equations (2.13) and (2.20), derived from equations (2.10)). Now we have assumed that $\underline{v} \in L_2(R)$, so it is quite possible that its derivatives are point-, line-, or area-wise undefined. To overcome this difficulty, we generalise the definitions of the divergence and curl of a vector field, so that no assumption is made about its differentiability.

Consider a volume V with smooth surface S. Let $\underline{n}$ denote the unit outer normal to an element dS of S. If the vector field $\underline{v}$ has continuous first derivatives in V, the following integral relations hold[57]:-

$$\left.\begin{aligned} \iiint_V \nabla.\underline{v}(\underline{x})\, dV &= \iint_S \underline{n}.\underline{v}\, dS \\ \iiint_V \nabla_\wedge \underline{v}(\underline{x})\, dV &= \iint_S \underline{n}_\wedge \underline{v}\, dS \end{aligned}\right\} \text{ for all } \underline{x} \in V \subset E_3$$

Then, using the mean value theorem for volume integrals[58], we can write:-

$$\nabla.\underline{v}(\underline{x}) = \lim_{V \to 0} \frac{1}{V} \iint_S \underline{n}.\underline{v}\, dS$$

$$\nabla_\wedge \underline{v}(\underline{x}) = \lim_{V \to 0} \frac{1}{V} \iint_S \underline{n}_\wedge \underline{v}\, dS$$

Let us now assume that $\underline{v} \in L_2(R)$. We define the generalised divergence and curl of $\underline{v}$, which we shall denote by $\nabla^*.\underline{v}$ and $\nabla^*_\wedge \underline{v}$, respectively, as follows:-

$$\nabla^*.\underline{v}(\underline{x}) = \lim_{V \to 0} \frac{1}{V} \iint_S \underline{n}.\underline{v}\, dS \qquad (2.21)$$

and

$$\nabla^*_\wedge \underline{v}(\underline{x}) = \lim_{V \to 0} \frac{1}{V} \iint_S \underline{n}_\wedge \underline{v}\, dS \qquad (2.22)$$

We note the equivalence of the generalised definitions and the usual definitions if $\underline{v}$ does have continuous first derivatives in V. As a result of this equivalence, we will henceforth always understand the divergence and curl in the generalised sense and will, therefore, drop the star superscript. Thus the use of $\nabla.\underline{v}$ and $\nabla_\wedge \underline{v}$ in equations (2.13) and (2.20), respectively, is perfectly valid when understood in the sense of equations (2.21) and (2.22), again respectively.

2.3 Applications of Orthogonal Decomposition Theory to Hydrodynamics

The equations of motion of a Newtonian fluid with constant kinematic viscosity in an Eulerian frame of reference are, ignoring thermal effects:-

$$\frac{\partial \underline{u}}{\partial t} + (\underline{u}.\nabla)\underline{u} = -\frac{1}{\rho}\nabla p + \nu(\Delta \underline{u} + \tfrac{1}{3}\nabla(\nabla.\underline{u})) + \underline{F}$$

where $\underline{u}$ and p are, respectively, the velocity and pressure fields of the fluid, t is the time, $\underline{F}$ represents body forces, and ρ and ν are, respectively, the density and kinematic viscosity of the fluid. The equation of continuity is:-

$$\frac{\partial \rho}{\partial t} + \nabla.(\rho \underline{u}) \equiv 0 \qquad (2.23)$$

and the equation of state is:-

$$\rho = \rho(p)$$

Given the body forces $\underline{F}$ for all $\underline{x} \in R$ and all $t \geqslant 0$, and boundary conditions:-

$$\underline{u}(\underline{x}) = \underline{U}\,; \qquad \underline{x} \in \delta R$$

and initial conditions:-

$$\underline{u}(t{=}0) = \underline{u}_0$$

we are in general required to determine the velocity field $\underline{u}$ and pressure field p at all points $\underline{x} \in R$ for all times $t > 0$.

We shall consider first a method of solution of the system (2.23) based on a potential representation of the velocity field. We shall then consider a method based on projection of either the equations of motion, or the velocity field, onto each of the subspaces $\mathring{G}(R)$, $U_1(R)$, etc defined in decomposition (2.5). All methods will be generally applicable to both compressible and incompressible, Newtonian and non-Newtonian, fluids. Unless otherwise stated, however, we shall assume that the fluid is incompressible and Newtonian with constant kinematic viscosity, and that thermal effects are negligible, so that the equations to be solved are:-

$$\left.\begin{aligned} \frac{\partial \underline{u}}{\partial t} + (\underline{u}.\nabla)\underline{u} &= -\frac{1}{\rho}\nabla p + \nu\Delta\underline{u} + \underline{F} \\ \nabla.\underline{u} &\equiv 0 \quad \text{in } R \\ \underline{u}(\underline{x}) &= \underline{U};\ \underline{x} \in \delta R \\ \underline{u}(t=0) &= \underline{u}_0 \end{aligned}\right\} \qquad (2.24)$$

Extensions to compressible and/or non-Newtonian fluids will be discussed at the end of the appropriate sub-section.

2.3.1 <u>The Vorticity/Potential Method</u>

We will assume that the velocity field $\underline{u}$, which is the solution of equations (2.24), $\in L_2(R)$. Now continuity implies that, in the generalised sense of equation (2.21), the divergence of the velocity field vanishes identically, so $\underline{u}$ can have no $\nabla\phi$ (vibrational, or compressible) component in the decomposition (2.6). Thus we may put:-

$$\underline{u} = \sum_{i=1}^{m}\alpha_i\nabla h_i + \nabla h + \sum_{j=1}^{n}\beta_j\nabla h^j + \nabla_\wedge\underline{A}$$

or, letting $\nabla h' = \sum_{i=1}^{m}\alpha_i\nabla h_i + \nabla h$

and $\nabla_\wedge\underline{A}' = \sum_{j=1}^{n}\beta_j\nabla h^j + \nabla_\wedge\underline{A}$

$$\underline{u} = \nabla h' + \nabla_\wedge\underline{A}' \qquad (2.25)$$

almost everywhere in R. (We use this reduced decomposition since it gives us sufficient generality combined with simplicity. Recall that we can put $\nabla h^j = \nabla_\wedge\underline{H}^j$; c.f. equation (2.7).) Together with suitable conditions on h' and $\underline{A}'$, equation (2.25), substituted into the equations of motion (here, the Navier-Stokes equations, since ρ and ν are constant) will automatically ensure that continuity is satisfied. If we now take the curl (in the generalised sense of equation (2.22)) of the Navier-Stokes equations, we obtain the vorticity transport equation:-

$$\frac{\partial \underline{\omega}}{\partial t} + ((\nabla h' + \nabla_\wedge \underline{A}').\nabla)\underline{\omega} - (\underline{\omega}.\nabla)(\nabla h' + \nabla_\wedge \underline{A}')$$
$$= \nu \triangle \underline{\omega} + \nabla_\wedge \underline{F} \qquad (2.26a)$$

where $\underline{\omega}$ is the vorticity. We see immediately that, not only have we satisfied continuity, but we have also eliminated pressure as a dependent variable. Together with the vector potential Poisson equation:-

$$\triangle \underline{A}' = -\nabla_\wedge \nabla_\wedge \underline{A}' = -\underline{\omega} \qquad (2.26b)$$

the scalar potential Laplace equation:-

$$\nabla^2 h' \equiv 0 \qquad (2.26c)$$

and suitable boundary and initial conditions on h', $\underline{A}'$ and $\underline{\omega}$, equations (2.26) form a coupled set which may be solved in principle to give the velocity field $\underline{u}(\underline{x},t)$ for all $\underline{x} \in R$, $t>0$ (provided the solution exists and is unique; that is, provided the problem is well posed). The pressure field $p(\underline{x},t)$ may also be obtained, up to an arbitrary constant, by substitution back into the Navier-Stokes equations. We thus have the basis of a method for solving the three-dimensional Navier-Stokes equations which we call the "vorticity/potential" method. In general, the method will involve numerical solution, and since equations (2.26) are coupled, an iterative scheme may be necessary. Such evidence as there is[59], which is for the special case where $\underline{u} \in \mathring{J}(R)$ i.e. $\underline{u} = \nabla_\wedge \underline{A}$, suggests that the apparent increase in complexity when using a potential representation for $\underline{u}$ is more than compensated for by the fact that the equation of continuity is satisfied automatically, and that pressure is eliminated as a dependent variable.

It was stated above that we can solve equations (2.26) given suitable boundary and initial conditions on h', $\underline{A}'$ and $\underline{\omega}$. Initial conditions present no problem. Given $\underline{u}(t=0) = \underline{u}_0$, we immediately know $\underline{\omega}(t=0) = \nabla_\wedge \underline{u}_0$, and hence $\underline{A}'(t=0)$ from equation (2.26b). We also know $h'(t=0)$ from the normal component boundary conditions on $\underline{u}$ and equation (2.26c). Boundary conditions, on the other hand, though straightforward, can cause difficulties. The correct boundary conditions, which follow directly from the conditions listed under equation (2.6), and the definition of vorticity, are:-

$$\left.\begin{aligned} \partial h'/\partial n &= \underline{u}.\underline{n} = U_n \\ (\nabla_\wedge \underline{A}')_{t_{1,2}} &= U_{t_{1,2}} - (\nabla h')_{t_{1,2}} \\ A'_n &= 0 \\ \underline{\omega} &= \nabla_\wedge \underline{u} \end{aligned}\right\} \text{ on } \delta R \qquad (2.26d)$$

where n, t_1 and t_2 refer to the normal and two tangential components of a vector, respectively. In the case of a simply-connected region, or a multiply-connected region with zero net flux across all of the surfaces S'_j characterising it (c.f. sub-section 2.2.2), we note that these boundary conditions can be simplified as follows:-

$$\left.\begin{aligned} \partial h'/\partial n &= U_n \\ A'_{t_1} &= 0 = A'_{t_2} \\ \partial(h_{t_1} h_{t_2} A'_n)/\partial n &= 0 \\ \underline{\omega} &= \nabla_\wedge \underline{u} \end{aligned}\right\} \text{ on } \delta R \qquad (2.26e)$$

where the scale factors h_{t_1} and h_{t_2} are defined as follows:-

if δR is sufficiently smooth (say, a Lyapunov surface, or regular surface in the sense of Kellogg[60]), then we can represent δR locally in terms of mutually orthogonal curvilinear coordinates ξ_n, ξ_{t_1}, and ξ_{t_2} as follows:-

$$\xi_n = \xi_n(\xi_{t_1}, \xi_{t_2}) \qquad (2.27)$$

If the position vector of a point on δR is $\underline{r} = \underline{r}(\xi_n, \xi_{t_1}, \xi_{t_2})$, then we define the scale factors h_i thus:-

$$h_i = \left| \partial \underline{r}/\partial \xi_i \right|$$

(It should be noted, incidentally, that the boundary conditions on $\underline{\omega}$ are not (all) known a priori, since we do not know $\underline{u}(t)$ in advance; numerical methods which involve the discretisation of time into steps of length Δt generally use $\underline{u}(t-\Delta t)$ to give boundary conditions on $\underline{\omega}$ at time t.)

The reason why the boundary conditions have been stressed is that, even quite recently, the boundary conditions on the vector potential in particular have not always been stated correctly. It has, for example, been asserted[61] that if the velocity field vanishes on δR, the vector potential can be required to vanish also, which is quite incorrect.

It might be argued that the use of a scalar plus a vector potential (c.f. equation (2.25)) is unnecessary, and that, by analogy with the streamfunction in two-dimensional and axi-symmetric three-dimensional applications (indeed, the vector potential is the full three-dimensional generalisation of the streamfunction), a single vector potential would be sufficient, as indeed it would, i.e.:-

$$\underline{u} = \nabla_\wedge \underline{A}^* \qquad (2.28)$$

There is, however, a difficulty then with the boundary conditions on $\underline{A}^*$, which are not at all straightforward[62]. Furthermore, if the boundary conditions on the velocity field are constant with respect to time, then so also is the complete scalar potential field h' (since it is defined by a Laplace equation with constant Neumann boundary conditions). So the field h' need only be determined once, at time t = 0. This, together with the simpler boundary conditions involved in the scalar plus vector potential representation (2.25), suggests that there is nothing to be gained from a single vector potential representation, a view supported elsewhere[63],[64]. (But note that in problems without flow-through, the scalar potential field h' vanishes, and a single vector potential representation for the velocity field is then appropriate.)

If the fluid is now assumed to be compressible, but thermal effects are still neglected (incorporating them in fact involves no essential difficulty), so that $\underline{u}$ is no longer solenoidal, we replace the potential representation (2.25) by:-

$$\underline{u} = \nabla h'' + \nabla_\wedge \underline{A}' \qquad (2.29)$$

where $\nabla h'' = \nabla \phi + \nabla h'$

The vector potential Poisson equation is still:-

$$\triangle \underline{A}' = -\nabla_\wedge \nabla_\wedge \underline{A}' = -\underline{\omega} \qquad (2.30a)$$

but the equation for h" is a Poisson equation:-

$$\nabla^2 h'' = \theta \qquad (2.30b)$$

where $\theta = \nabla . \underline{u}$ is the rate of expansion or dilatation of the flow field, the divergence being understood in the generalised sense. (This is in contrast to the Laplace equation for h' in the incompressible case above.) By taking the divergence and curl of the equations of motion (2.23), again in a generalised sense, we obtain:-

$$\frac{\partial \theta}{\partial t} + \tfrac{1}{2}\nabla^2 \left| (\nabla h'' + \nabla_\wedge \underline{A}') \right|^2 + (\nabla h'' + \nabla_\wedge \underline{A}').(\nabla_\wedge \underline{\omega}) - |\underline{\omega}|^2$$
$$= -\nabla(1/\rho).\nabla p - \frac{1}{\rho}\nabla^2 p + \frac{4}{3}\nu\nabla^2\theta + \nabla.\underline{F} \qquad (2.30c)$$

and

$$\frac{\partial \underline{\omega}}{\partial t} + ((\nabla h'' + \nabla_\wedge \underline{A}').\nabla)\underline{\omega} - (\underline{\omega}.\nabla)(\nabla h'' + \nabla_\wedge \underline{A}') + \underline{\omega}\theta$$
$$= -\nabla(1/\rho)_\wedge \nabla p + \nu\Delta\underline{\omega} + \nabla_\wedge \underline{F} \qquad (2.30d)$$

The boundary conditions are:-

$$\left.\begin{array}{l} \partial h''/\partial n = U_n \\ (\nabla_\wedge \underline{A}')_{t_{1,2}} = U_{t_{1,2}} - (\nabla h'')_{t_{1,2}} \\ A'_n = 0 \\ \left.\begin{array}{l} \underline{\omega} = \nabla_\wedge \underline{u} \\ \theta = \nabla.\underline{u} \end{array}\right\} \text{unknown a priori} \end{array}\right\} \text{on } \delta R \qquad (2.30e)$$

and the initial conditions are:-

$$\underline{\omega}(t=0) = \nabla_\wedge \underline{u}_0, \quad \theta(t=0) = \nabla.\underline{u}_0 \qquad (2.30f)$$

with $\underline{A}'(t=0)$ and h"(t=0) easily obtained from equations (2.30a) and (2.30b), respectively.

It is quite clear that determining the velocity and pressure fields for compressible flows from equations (2.30) is much more complex than it is for incompressible flows. Moreover, compressible flows can give rise to shock waves, and since θ will have to be defined in a generalised sense along these shock waves, additional equations (the Rankine-Hugoniot equations[65]) will be required to determine the discontinuities in velocity, pressure and density. Certainly, it is still possible to obtain solutions of the coupled system of equations, but it is readily apparent that a great deal more computational work is involved, compared with the incompressible case. When we also consider that the use of potentials neither satisfies continuity automatically, nor eliminates pressure as a dependent variable, the advantages of such an approach would seem to be minimal. Even if the flow was assumed to be irrotational, so that $\underline{\omega} \equiv \underline{0}$ in R, these disadvantages would remain, though the equations would be simpler, admittedly. This is in direct contrast to the incompressible case discussed above.

If the fluid is now assumed to be incompressible and non-Newtonian, then all of the advantages of a potential representation applied to the flow of an incompressible Newtonian fluid discussed above remain in force. Altering the form of the stress relationship certainly alters the complexity of the problem (for example, by the introduction of further non-linearity), but continuity will still be satisfied automatically, and pressure still be eliminated as a dependent variable. Analogously, a potential representation applied to the flow of a compressible non-Newtonian fluid will suffer from all the defects discussed in the preceding paragraph. We see, then, that non-Newtonian behaviour adds little formally to the method of solution using a potential representation.

2.3.2 The Orthogonal Projection Method

As in the preceding sub-section, we assume that the velocity field $\underline{u}$, which is the solution of equations (2.24), $\in L_2(R)$. We further assume that each of the terms in the equations of motion (the Navier-Stokes equations) also $\in L_2(R)$. This assumption implies that the total forces acting on the fluid are bounded. We now incorporate all body forces which are derivable from a single-valued scalar potential (for example, gravity forces[66]) into the $\frac{1}{\rho}\nabla p$ term; this involves no loss of generality. Then, using standard vector identities, it is easy to show that:-

$$\frac{\partial \underline{u}}{\partial t} \in \qquad U_1(R) \oplus U'(R) \oplus U_2(R) \oplus \mathring{J}(R)$$

$$(\underline{u}.\nabla)\underline{u} \in \mathring{G}(R) \oplus U_1(R) \oplus U'(R) \oplus U_2(R) \oplus \mathring{J}(R)$$

$$\frac{1}{\rho}\nabla p \in \mathring{G}(R) \oplus U_1(R) \oplus U'(R)$$

$$\nu\Delta\underline{u} \in \qquad U_1(R) \oplus U'(R) \oplus U_2(R) \oplus \mathring{J}(R)$$

$$\underline{F} \in \qquad U_2(R) \oplus \mathring{J}(R)$$

We note immediately that if we project each of the terms in the Navier-Stokes equations onto the subspace $U_2(R) \oplus \mathring{J}(R)$, then the resulting equations will contain no pressure term. Moreover, projection onto this subspace of solenoidal vector fields ensures that continuity is satisfied automatically. We thus have the basis of a method for solving the Navier-Stokes equations which, for obvious reasons, we call a "projection method". Before we proceed to develop this method, we note that such a projection method is very similar to the vorticity/potential method discussed in the previous sub-section; the essential difference is that here we use the primitive variables $\underline{u}$ and p, as opposed to the derived variables h', $\underline{A}'$ and $\underline{w}$ used there. A final point to note is that there is a vibrational component of the convection term $(\underline{u}.\nabla)\underline{u}$ which has to be balanced by the vibrational component of the pressure term $\frac{1}{\rho}\nabla p$. It might seem surprising that such a vibrational (or compressible) component is involved in the equations of motion of a solenoidal velocity field $\underline{u}$. It is, however, typical of the effects of the convection term.

There is, in fact, a variety of ways in which a projection method may be applied. In all the methods, however, the translational components $\underline{u}_{U_1}$ and $\underline{u}_{U'}$ have to be determined first (because they are not orthogonal to the $\frac{1}{\rho}\nabla p$ term), either by solution of equations (2.14) and (2.15) or, using a mixed potential/projection approach, by obtaining $\underline{u}_{U_1} + \underline{u}_{U'}$ as the solution of the Laplace equation:-

$$\left.\begin{aligned} \nabla^2 h' &\equiv 0 \\ \text{with Neumann boundary conditions} \quad \partial h'/\partial n &= U_n \end{aligned}\right\} \quad (2.31)$$

where $\underline{u}_{U_1} + \underline{u}_{U'} = \nabla h'$ almost everywhere. We now have to obtain $\underline{u}_{U_2}$ and $\underline{u}^o_J$. There are two basic ways in which we can do this. We can solve the vorticity transport equation (2.26a) for $\underline{\omega}$, and then obtain $\underline{u}_{U_2}$ and $\underline{u}^o_J$ from equations (2.16) and (2.20), respectively. Alternatively, if we denote the projection operator from $L_2(R)$ onto $U_2(R) \oplus \overset{o}{J}(R)$ by P^*, then operating on each of the terms in the Navier-Stokes equations with P^* gives:-

$$\frac{\partial P^*[\underline{u}]}{\partial t} + P^*\left[(\underline{u}.\nabla)\underline{u}\right] = \nu P^*\left[\Delta \underline{u}\right] + \underline{F} \qquad (2.32)$$

assuming that P^* and $\partial/\partial t$ commute; it is not hard to show that they do. Since we know $\underline{u}$ on δR, and have already determined $\underline{u}_{U_1} + \underline{u}_{U'}$, we can determine $\underline{u}_{U_2} + \underline{u}^o_J$ on δR. We may thus solve equation (2.32) for $P^*[\underline{u}] = \underline{u}_{U_2} + \underline{u}^o_J$ using an analytical expression for P^* derived from equations (2.16) and (2.20) above. Whichever method is adopted, the velocity field $\underline{u}$ can then obtained by straight addition of the various components, and the pressure field determined up to an arbitrary constant by substitution back into the Navier-Stokes equations.

In general, of course, the various equations involved in the particular projection method adopted will have to be solved numerically. This presents no great difficulty as far as $\underline{u}_{U_1}$ and $\underline{u}_{U'}$ are concerned, whatever method is used to obtain them. A difficulty does arise, however, in obtaining $\underline{u}_{U_2}$ and $\underline{u}^o_J$. Although the integral relations (2.16) and (2.20) can be and have been used, they are computationally inefficient for general problems[67] (recall the comments made on the vorticity/integral relation method in sub-section 1.2.2 of Chapter 1). We will, therefore, eliminate the integral relations method from further consideration. Use of the operator P^* for projecting the Navier-Stokes equations onto the subspace $U_2(R) \oplus \overset{o}{J}(R)$, on the other hand, implies a priori knowledge of P^*, and the analytical determination of it (from equations (2.16) and (2.20)) may well be very difficult, so that a numerical analogue must be sought. One way of obtaining a numerical analogue is to use an iterative scheme to project vector fields onto $U_2(R) \oplus \mathring{J}(R)$. This approach has been applied to the Navier-Stokes equations for the special case where $\underline{u} \in \mathring{J}(R)$[68], and it seems that the apparent increase in complexity when using this projection method, as opposed to straight solution of the Navier-Stokes equations, is more than compensated for by the fact that continuity is automatically satisfied, and by the elimination of pressure as an immediate dependent variable. (Pressure is in fact retained as part of the iterative scheme, which can be shown not to rely on orthogonal projection at all, although it does give the projection of the Navier-Stokes equations onto $\mathring{J}(R)$. The reason why the iterative scheme used does not involve orthogonal projection follows directly from the fact that the scheme will also give the translational velocity components $\underline{u}_{U_1}$ and $\underline{u}_{U'}$, in flow-through and moving-boundary problems, as well as $\underline{u}_{U_2}$ and $\underline{u}_{\mathring{J}}$, even though $\underline{u}_{U_1}$ and $\underline{u}_{U'}$, are not orthogonal to the $\frac{1}{\rho}\nabla p$ term.) As we would expect, this is precisely analogous to the results we obtained using the vorticity/potential method in the previous sub-section.

Again, as with the vorticity/potential method, if the fluid is now assumed to be compressible, but thermal effects are still neglected (although they may be incorporated with no essential difficulty), complications arise. Analysing the equations of motion (2.23), we can show that:-

$$\frac{\partial \underline{u}}{\partial t} \in \mathring{G}(R) \oplus U_1(R) \oplus U'(R) \oplus U_2(R) \oplus \mathring{J}(R)$$

$$(\underline{u}.\nabla)\underline{u} \in \mathring{G}(R) \oplus U_1(R) \oplus U'(R) \oplus U_2(R) \oplus \mathring{J}(R)$$

$$\frac{1}{\rho}\nabla p \in \mathring{G}(R) \oplus U_1(R) \oplus U'(R) \oplus U_2(R) \oplus \mathring{J}(R)$$

$$\nu \triangle \underline{u} \in \mathring{G}(R) \oplus U_1(R) \oplus U'(R) \oplus U_2(R) \oplus \mathring{J}(R)$$

$$\frac{\nu}{3}\nabla(\nabla.\underline{u}) \in \mathring{G}(R) \oplus U_1(R) \oplus U'(R)$$

$$\underline{F} \in U_2(R) \oplus \mathring{J}(R)$$

where we have again incorporated all body forces derivable from a single-valued scalar potential in the $\frac{1}{\rho}\nabla p$ term. We can easily see that all the projections of $\underline{u}$ are coupled via the boundary conditions as well as the equations of motion, so we have to solve for all the projections simultaneously. Furthermore, continuity is not automatically satisfied, and pressure cannot usefully be eliminated, since we need it to determine $\underline{u}_G^s$. The advantages of applying the projection method to compressible flows would thus seem to be minimal. It is hardly surprising that this is precisely what we found when we considered applying the vorticity/potential method to compressible flows in the previous sub-section.

Again, if the fluid is assumed to be non-Newtonian, then precisely the same comments may be made as were made in the previous sub-section when discussing the application of the vorticity/potential method to non-Newtonian fluids. The fact that the fluid is non-Newtonian will only alter the subspaces onto which the viscous term in the equations of motion has non-zero projections.

2.3.3 Conclusions

In the absence of direct theoretical or numerical comparisons, it is difficult to say with certainty whether the projection method is superior to the vorticity/potential method, or vice versa. One possible indicator is that the projection approach uses the primitive variables velocity and pressure (and density, if the flow is compressible), whereas the potential approach uses the derived variable vorticity (and rate of dilatation, if the flow is compressible), together with a scalar and a vector potential. Thus the latter method requires the existence of one more space derivative of velocity than does the former. So it may well be that, in flows with high velocity gradients (turbulent flows, for example), the projection method may prove superior. Moreover, the projection method involves only four dependent variables (or five, if the flow is compressible), whereas the vorticity/potential method involves seven (or ten, if the flow is compressible). So, in principle at least, more computational work is involved in the vorticity/potential method, compared with the projection method.

On the other hand, the vorticity/potential method is much more straightforward. The exact form of the equations, together with their boundary and initial conditions, is known, whereas the projection method involves a projection operator whose explicit form is unknown. And it is this factor which seems far and away the most important drawback to the projection method. Clearly, it is difficult to say that one method has definite advantages over the other. But the complete lack of ambiguity with which the vorticity/potential method can be used does seem to weigh heavily and, in the final analysis, decisively in its favour.

What we can be definite about, however, is that the use of either a potential or a projection approach in incompressible flow calculations is almost certain to reduce the amount of computational work required to solve them. Involving, as they do, only minor assumptions additional to those inherent in the equations of motion, continuity and state themselves, their use can be confidently recommended.

Chapter 3 – Numerical Solution of the Navier-Stokes Equations for Flow past a Sphere

3.1 Introduction

In this chapter, we will show how the vorticity/potential method developed in the previous chapter can be applied to a particular problem, namely flow past a sphere. In particular, we shall consider the flow of an incompressible Newtonian fluid of constant kinematic viscosity (a so-called "Navier-Stokes fluid"[(69)]) moving either uniformly, or with a constant unidirectional rate of shear, at infinity past a stationary, neutrally buoyant, non-rotating sphere. This does, of course, represent an idealised flow situation: for example, no fluid is completely incompressible, and the sphere would be unlikely to be stationary and non-rotating. On the other hand, the problem is sufficiently general to enable us to predict certain important features of the flow past the sphere and, in any case, it would be a relatively straightforward matter to extend the problem to more general flow situations.

The principal reason for studying this problem is that it has important physical applications – for example, fluid-particle flow past solid boundaries, such as fluid-borne reactants in chemical reactors, blood corpuscles in veins and arteries, and silt on river beds – where, provided the particles are approximately spherical in shape and are not too close to one another, the flow can be approximated by our idealised problem.

Three important points must be borne in mind when we set about formulating and solving this problem. First, we are primarily interested in obtaining its steady-state solution (which we interpret in a sufficiently wide sense to cover, for example, periodic vortex shedding). Secondly, because many physical situations involve low Reynolds number flows, we will aim to solve the equations primarily for flows with Reynolds numbers less than (say) 10^4, although it will obviously be an advantage if we can obtain results for Reynolds numbers higher than this. (This also means that we will not have to deal with highly turbulent flows, and that we can use a variety of (very) low Reynolds number analytical solutions to the problem, making Stokes flow and other similar approximations, for comparison purposes.) The third and final point which we must bear in mind is that the problem, once formulated, will be solved numerically on a computer, since there is no way of solving it analytically. It is, therefore, important to realise that this solution is not necessarily the same, even qualitatively, as the analytical solution. (For example, turbulence, which affects the point of separation of a flow past a body such as a sphere, may be on too small a scale to be predicted computationally, so that the numerical solution may differ from the analytical solution.) These three points will constantly recur as guiding principles in the formulation and solution of our problem.

In the next section, 3.2, we will formulate the complete problem of flow past the sphere analytically, and then numerically, using finite-difference methods. Then, in section 3.3, we will show how the problem may be solved computationally and finally, in section 3.4, we will present the results we have obtained.

3.2 Formulation of the Problem

3.2.1 Analytical Formulation

Our aim, as discussed in section 3.1 above, is to determine the motion of a Navier-Stokes fluid moving either uniformly, or with a constant unidirectional shear rate, at infinity past a neutrally buoyant, stationary, non-rotating sphere. We will assume that the velocity and pressure fields $\underline{u}$ and p, respectively, of the fluid are sufficiently differentiable for the flow to be determined by the Navier-Stokes equations:-

$$\frac{\partial \underline{u}}{\partial t} + (\underline{u}.\nabla)\underline{u} = - \frac{1}{\rho}\nabla p + \nu\Delta\underline{u} \qquad (3.1)$$

together with the constraint of continuity:-

$$\nabla.\underline{u} \equiv 0 \qquad (3.2)$$

where ρ and ν denote, respectively, the density and kinematic viscosity of the fluid, and t denotes time. Whether the system of equations (3.1) - (3.2) does in fact possess a solution, and whether this solution is unique, is, as we saw in sub-section 1.2.1 of Chapter 1, still unknown. In the absence of evidence to the contrary, we will assume that it does.

The boundary conditions to be imposed on the solution are clearly:-

$\underline{u} \equiv \underline{0}$ on the surface of the sphere,

and $\underline{u} \rightarrow \underline{U}_{\infty}$ as the space coordinate tends to infinity; $\underline{U}_{\infty}$ is the velocity field at infinity.

(The boundary condition $\underline{u} \equiv \underline{0}$ on the surface of the sphere implies that there is no flow-through and no slip. Other boundary conditions can be considered, for example permeable surfaces(70) (where there is flow-through), and the flow of rarefied gases(71) (where there is slip).)

The initial conditions to be imposed are, on the other hand, not quite so obvious. As we mentioned in section 3.1, we are primarily interested in the steady-state solution of the system (3.1) - (3.2). (It could be argued, therefore, that we need only solve the steady-state equivalent of equation (3.1) - i.e. put $\partial\underline{u}/\partial t \equiv \underline{0}$. This presupposes, however, that such a steady-state solution actually exists - the occurrence of periodic vortex shedding behind bluff bodies, turbulence at high Reynolds numbers, etc, suggests that it may not.) We can, therefore, choose suitable initial conditions for the unsteady system by acknowledging this, and assuming (not unreasonably(72)) that the asymptotic, or "late-time", solution of the unsteady equations, obtained as time t tends to infinity, is the steady-state solution, and that it is effectively independent of the choice of initial conditions. On this assumption, therefore, we can choose (virtually) any initial conditions that satisfy the boundary conditions and be reasonably certain that the late-time solution so obtained is the steady-state solution we desire. The exact choice of initial conditions will be discussed in sub-section 3.3.2 below.

In order to solve the system of equations (3.1) - (3.2), we will use, for the reasons discussed in sub-section 2.3.3 of Chapter 2, the vorticity/potential method, as opposed to the projection method. Following the argument of sub-section 2.3.1 of Chapter 2, and in the same notation, we might put:-

$$\underline{u} = \nabla h' + \nabla_\wedge \underline{A}'$$

We can, however, simplify this representation. Because the surface of the sphere is in effect a solid wall, there can be no flow through it. Thus the potential representation of $\underline{u}$ can contain no $\sum_{i=1}^{m} \alpha_i \nabla h_i$ term. Moreover, because the flow region is not multiply-connected, the potential representation of $\underline{u}$ can contain no $\sum_{j=1}^{n} \beta_j \nabla h^j$ term. Thus we can use the following more specific representation of $\underline{u}$:-

$$\underline{u} = \nabla h + \nabla_\wedge \underline{A} \qquad (3.3)$$

(which is equivalent to saying that $\underline{u} \in U'(R) \oplus \mathring{J}(R)$), where:-

$$\left.\begin{array}{l} \left.\begin{array}{l} \nabla^2 h = 0 \\ \Delta \underline{A} = -\nabla_\wedge \nabla_\wedge \underline{A} = -\nabla_\wedge \underline{u} = -\underline{w} \\ \nabla \cdot \underline{A} \equiv 0 \end{array}\right\} \text{in } R \\ \left.\begin{array}{l} \partial h / \partial n = U_n \\ \underline{A}_\wedge \underline{n} \equiv \underline{0} \\ \partial (h_{t_1} h_{t_2} A_n) / \partial n \equiv 0 \end{array}\right\} \text{on } \delta R \end{array}\right\} (3.4)$$

Again, following the argument of sub-section 2.3.1, we eliminate pressure by taking the curl of the terms in the Navier-Stokes equations, and obtain the vorticity transport equation:-

$$\frac{\partial \underline{\omega}}{\partial t} + (\underline{u}.\nabla)\underline{\omega} - (\underline{\omega}.\nabla)\underline{u} = \nu\Delta\underline{\omega} = -\nu\nabla_\wedge\nabla_\wedge\underline{\omega}$$

or, substituting from equation (3.3):-

$$\frac{\partial \underline{\omega}}{\partial t} + ((\nabla h + \nabla_\wedge \underline{A}).\nabla)\underline{\omega} - (\underline{\omega}.\nabla)(\nabla h + \nabla_\wedge \underline{A}) = -\nu\nabla_\wedge\nabla_\wedge\underline{\omega} \quad (3.5)$$

with boundary conditions:-

$$\left.\begin{aligned} \underline{\omega} &= \nabla_\wedge\nabla_\wedge\underline{A} \quad \text{on the sphere} \\ \underline{\omega} &= \nabla_\wedge\underline{U}_\infty \quad \text{at infinity} \end{aligned}\right\} \quad (3.6)$$

The system of equations (3.4) - (3.6) forms a complete set from which, if the problem is well-posed, the vorticity ($\underline{\omega}$), scalar potential (h), and vector potential ($\underline{A}$), fields may be determined at all points in the flow field, and for all times greater than zero. (We note, however, that h is arbitrary up to a constant.) The velocity field may then be obtained from equation (3.3), and the pressure field up to an arbitrary constant by substitution back into equation (3.1).

We note, incidentally, that the potential representation (3.3) for $\underline{u}$ is invalid in the unbounded flow field round the sphere, since the motion at infinity means that $\underline{u}$ is not Lebesgue square-integrable. This difficulty can be overcome in one (or both) of two ways. We can put $\underline{u} = \underline{u}' + \underline{U}_\infty$, where $\underline{U}_\infty$ is the velocity field at infinity; clearly $\underline{u}'$ is then Lebesgue square-integrable, so we can decompose $\underline{u}'$ instead of $\underline{u}$. Alternatively, since we are eventually going to solve equations (3.4) - (3.6) numerically, we will generally use a finite representation of the unbounded flow field. This will usually mean that we enclose the sphere, radius r_0, in an envelope of radius r^* ($r^* \gg r_0$) and ignore the rest of the flow field outside the envelope. Inside the envelope, the velocity field is then Lebesgue square-integrable. The question then arises: what boundary conditions do we impose on the envelope? However large the envelope is, if we merely impose the boundary conditions which apply at infinity, we will eventually run into difficulties with respect to wake lengths etc. Other boundary conditions which might approximate the flow at this envelope, such as parallel flow downstream etc, have been investigated for two-dimensional flows using a streamfunction/vorticity formulation, the conclusion being that, while there is no entirely satisfactory answer, we should be as precise as possible when specifying the boundary conditions on the envelope[(73),(74)] - which generally means imposing the boundary conditions which apply at infinity. Of course, this means that, for example, wakes will be curtailed in length compared with an unbounded flow. On the other hand, infinite regions are not (generally) encountered in physical problems, so we might argue that phenomena such as wake curtailment by outer boundary interaction are physically realistic. (There is, indeed, no very good reason for considering unbounded regions, other than ease of analytical formulation and, in some cases, solution.)

Since our aim here is to solve the equations (3.4) - (3.6) numerically, we will in fact adopt the latter course, and enclose the sphere (radius r_o) in a concentric spherical envelope (radius $r^* \gg r_o$). We will henceforth, therefore, understand all boundary conditions "at infinity" to mean conditions on this spherical envelope, unless otherwise stated. This does not, in fact, alter the form of the conditions (3.4) and (3.6); it merely alters where they are to be applied.

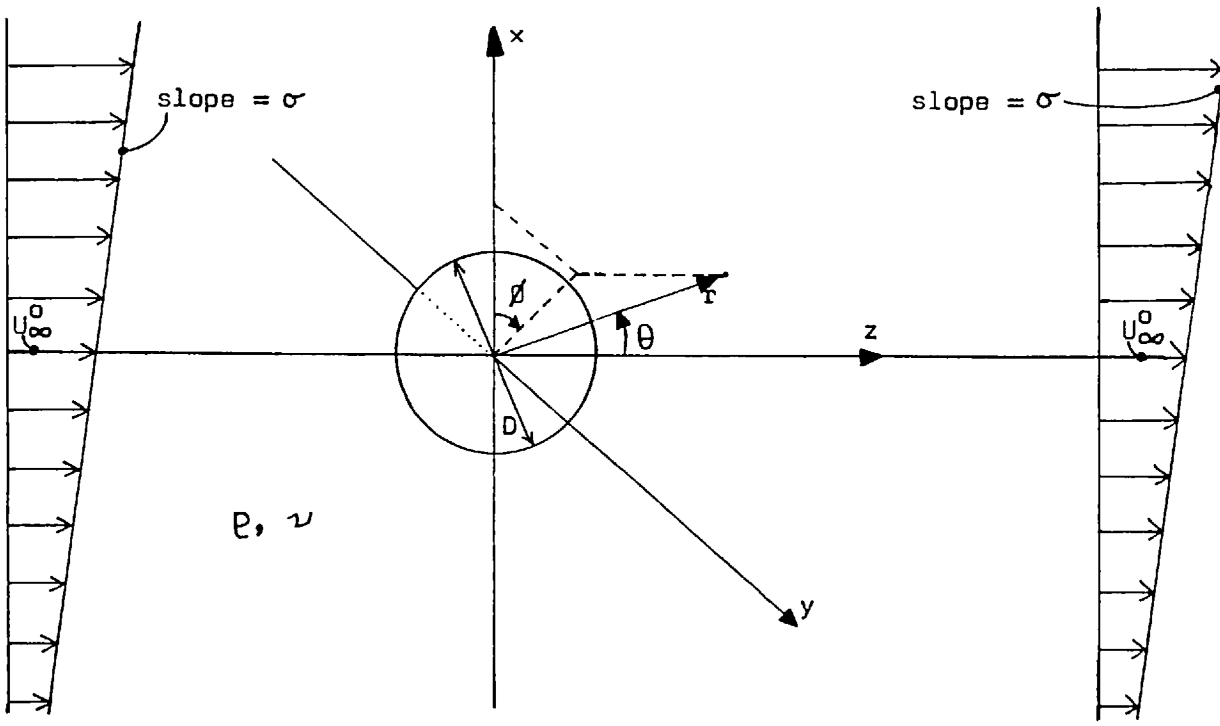
3.2.2 Dimensionless Form of the Equations in Spherical Polar Coordinates

The obvious and natural system of coordinates to use for flow in an unbounded region past a sphere is (right-handed) spherical polar coordinates (r, θ, ϕ). It will also be useful, however, to have a system of (right-handed) rectangular Cartesian coordinates (x, y, z) with origin at the centre of the sphere, and aligned so that:-

(i) the flow at infinity is in the positive z ($\theta = 0$) direction;

(ii) the flow varies at infinity in the x ($\theta = \pi/2$; $\phi = 0, \pi$) direction only, if at all.

Thus we have, diagramatically:-



The parameters characterising the system are:-

- the magnitude of the velocity at $x = 0$, $y = 0$, $z \rightarrow \pm\infty$ ($r \rightarrow \infty$, $\theta = 0, \pi$), denoted by U^0_∞;
- the rate of shear at infinity, $\partial u_z / \partial x$, denoted by σ;
- the diameter of the sphere, denoted by D;
- the fluid density and kinematic viscosity, denoted by ρ and ν, respectively.

Using the methods of dimensional analysis, we can reduce the effective number of parameters from five (D, U^0_∞, σ, ρ, and ν) to two (a Reynolds number and a ratio of velocities). There is a variety of ways in which this reduction to dimensionless form can be performed depending on the choice of the characteristic velocity of the system. The choice rests naturally between the centre-line velocity at infinity, U^0_∞, and the product of the shear rate at infinity and the diameter of the sphere, σD.

We will adopt the convention that we will always use U_∞^o as the characteristic velocity unless it is identically zero, in which case we will use σD. Then, with D as the characteristic length of the system, we can easily show that:-

$$\frac{\partial \underline{\omega}}{\partial t} + ((\nabla h + \nabla_\wedge \underline{A}).\nabla)\underline{\omega} - (\underline{\omega}.\nabla)(\nabla h + \nabla_\wedge \underline{A}) = -\frac{1}{Re}\nabla_\wedge\nabla_\wedge \underline{\omega} \quad \text{in } R \qquad (3.7)$$

$$\left.\begin{aligned} \nabla_\wedge\nabla_\wedge \underline{A} &= \underline{\omega} \\ \nabla.\underline{A} &\equiv 0 \\ \nabla^2 h &\equiv 0 \end{aligned}\right\} \text{in } R \qquad (3.8)$$

$$\left.\begin{aligned} \partial h/\partial n &\equiv 0 \\ \partial(r^2 \sin\theta\, A_n)/\partial n &\equiv 0 \\ (\text{i.e. } \partial A_r/\partial r &= -\ 2A_r/r) \\ \underline{A}_\wedge \underline{n} \equiv \underline{0} \ (\text{i.e. } A_\theta &\equiv 0 \equiv A_\phi) \\ \omega_n &\equiv 0 \\ \underline{\omega}_\wedge \underline{n} &= (\nabla_\wedge\nabla_\wedge \underline{A})_\wedge \underline{n} \end{aligned}\right\} \text{at } r = r_o \qquad (3.9)$$

$$\left.\begin{aligned} \partial h/\partial n &= \underline{U}_\infty.\underline{n} \\ \partial A_r/\partial r &= -\ 2A_r/r \\ A_\theta \equiv 0 &\equiv A_\phi \\ \underline{\omega} &= -\ \underline{\sigma} \end{aligned}\right\} \text{at } r = r^* \qquad (3.10)$$

where Re is the Reynolds number of the system; its definition depends on the choice of the characteristic velocity. Thus:-

$$Re = \begin{cases} U_\infty^o D/\nu & \text{if } U_\infty^o \neq 0 \\ \sigma D^2/\nu & \text{if } U_\infty^o = 0 \end{cases};$$

$\underline{n}$ is the unit outer normal to δR; clearly, this is in the positive radial direction at $r = r^*$, and the negative radial direction at $r = r_o$;

the condition $\omega_n \equiv 0$ at $r = r_o$ follows directly from the no-slip condition on velocity;

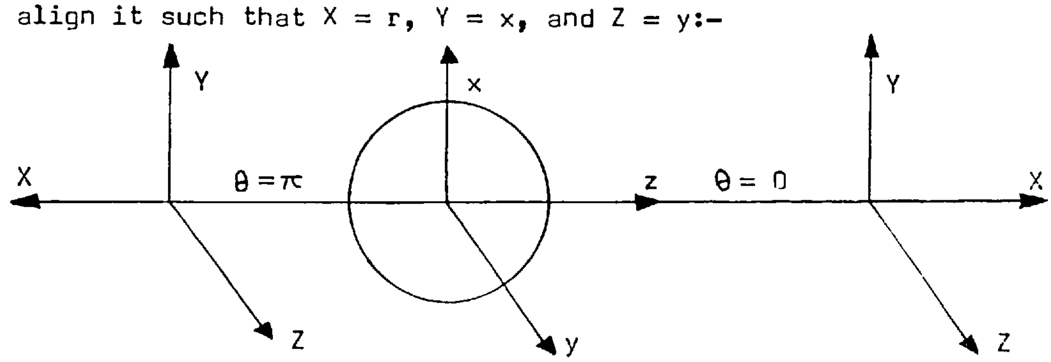
and $\underline{\sigma}$ is the vector $(\sigma_r, \sigma_\theta, \sigma_\phi) = (\sigma\sin\theta\sin\phi, \sigma\cos\theta\sin\phi, \sigma\cos\phi)$, with σ made dimensionless with respect to (U_∞^o/D) if $U_\infty^o \neq 0$, and itself if $U^0_\infty = 0$ (i.e. $\sigma = 1$).

Note that we use the same symbol to denote both dimensional and dimensionless variables; henceforth, we will always understand variables to be dimensionless, unless otherwise stated. Thus $r_0 = \frac{1}{2}$, and:-

$$\underline{U}_\infty = \begin{cases} ((1 + \sigma r^*\sin\theta\cos\phi)\cos\theta,\ -(1 + \sigma r^*\sin\theta\cos\phi)\sin\theta,\ 0) & \text{if } U^0_\infty \neq 0 \\ (r^*\sin\theta\cos\phi\cos\theta,\ -r^*\sin\theta\cos\phi\sin\theta,\ 0) & \text{if } U^0_\infty = 0 \end{cases}$$

A final point to note is that on the polar axis ($\theta = 0, \pi$), the θ- and ϕ-directions, and hence the θ- and ϕ-components of vectors, are undefined. Also, terms such as $\frac{1}{\sin\theta}$ tend to become unbounded. We can overcome these difficulties as follows:-

(i) to overcome the problem of undefined θ- and ϕ-components, we define, on the polar axis only, a local Cartesian coordinate system (X, Y, Z) and align it such that X = r, Y = x, and Z = y:-



Note that this implies a right-handed coordinate system on the $\theta = 0$ polar axis, and a left-handed system on the $\theta = \pi$ polar axis. Note also that because the boundary conditions on the flow are symmetrical about the x-z plane, the scalar potential field h is also symmetrical about this plane; this fact is used to simplify terms involving h on the polar axis. Such symmetry is not necessarily carried over to the vector potential or vorticity fields, however, as the possible existence of oscillating wakes etc makes clear, so no symmetry arguments can be used to simplify terms involving $\underline{A}$ or $\underline{\omega}$ on the polar axis.

(ii) the terms which tend to become unbounded on the polar axis are of the form $f/\sin\theta$, $f/\sin^2\theta$, $f/\tan\theta$, and $f/\tan\theta\sin\theta$. Now although $f(\theta^*)/g(\theta^*)$ may be undefined, where f and g are functions of θ, the limit of

$f(\theta)/g(\theta)$ as $\theta \longrightarrow \theta^*$ may exist. L'Hospital's rule[75] gives:-

$$\lim_{\theta \to \theta^*} \frac{f(\theta)}{g(\theta)} = \frac{f'(\theta^*)}{g'(\theta^*)}$$

the primes denoting differentiation with respect to θ. (The rule may be applied again if $f'(\theta^*)/g'(\theta^*)$ is undefined, and so on.) We may easily show that:-

$$\left.\begin{aligned}
&\lim_{\substack{\theta \to 0 \\ \theta \to \pi}} \frac{f}{\sin\theta} = \begin{cases} f' \\ -f' \end{cases} \\
&\lim_{\theta \to 0,\pi} \frac{f}{\sin^2\theta} = \tfrac{1}{2}f'' \\
&\lim_{\theta \to 0,\pi} \frac{f}{\tan\theta} = f' \\
&\lim_{\substack{\theta \to 0 \\ \theta \to \pi}} \frac{f}{\tan\theta\sin\theta} = \begin{cases} \tfrac{1}{2}f'' \\ -\tfrac{1}{2}f'' \end{cases}
\end{aligned}\right\} \quad (3.11)$$

which solves our second difficulty.

We are now in a position to write down the dimensionless form of the vorticity transport, vector potential Poisson, and scalar potential Laplace, equations, together with their boundary conditions, in spherical polar coordinates, and with suitable modifications on the polar axis. Because of their relative length, however, they are written elsewhere (c.f. Appendix 1).

3.2.3 Finite-Difference Formulation

The numerical solution of partial differential equations essentially involves the discretisation of the domain of definition of the dependent variables. Thus numerical solution of the unsteady equations of motion of a fluid involves discretisation of the three-dimensional region R in which the fluid is flowing, and the half-open time interval $[0,T)$, where T is sufficiently large for us to obtain all our desired results. (In practice, this generally means until steady-state has effectively been reached.)

The region R is the closed subspace of E_3 defined by:-

$$\left.\begin{array}{ll} & r_0 \leqslant r \leqslant r^* \\ \text{and hence} & 0 \leqslant \theta \leqslant \pi \\ \text{and} & 0 \leqslant \phi < 2\pi \end{array}\right\} \quad (3.12)$$

There is a variety of ways in which we may discretise R. The way in which we shall do it is probably the simplest: we discretise the radial coordinate r into n_r sections of equal length Δr, the polar angle θ into n_θ equiangular sections $\Delta\theta$, and the azimuthal angle ϕ into n_ϕ equiangular sections $\Delta\phi$. Thus if (i, j, k) denotes a node in the discretised region R', with coordinates (r, θ, ϕ) in the continuous region R,

$$\left.\begin{array}{lll} r = (i-1)\Delta r + r_0; & 1 \leqslant i \leqslant n_r + 1 \\ \theta = (j-1)\Delta\theta & ; \quad 1 \leqslant j \leqslant n_\theta + 1 \\ \phi = (k-1)\Delta\phi & ; \quad 1 \leqslant k \leqslant n_\phi \end{array}\right\} \quad (3.13)$$

where $n_r = (r^* - r_0)/\Delta r$, $n_\theta = \pi/\Delta\theta$, and $n_\phi = 2\pi/\Delta\phi$.

Note that this discretisation does not involve scaling of the space variables (for example, exponential scaling of the radial coordinate, which is often used to give increased spatial resolution near solid boundaries where velocity gradients tend to be greatest). Also, it uses constant intervals Δr, $\Delta\theta$, $\Delta\phi$. The reason for this is that the question of the superiority of any other approach over the straightforward one given above has not yet been resolved. (The increase in resolution near solid boundaries quoted above, for example, must be balanced against a loss of resolution in the shear layers at the edge of wakes.) Indeed, in certain circumstances, scaling and/or variable intervals can modify the problem adversely[76].

The discretisation of [0,T) is straightforward: we split it into n_t intervals of equal length Δt, so that:-

$$t = n\Delta t; \quad 0 \leqslant n \leqslant n_t = T/\Delta t \quad (3.14)$$

The maximum size of the time-step length Δt is generally governed by considerations of stability of the numerical scheme, and will be discussed in sub-section 3.4.1.

With this straightforward discretisation of the space $R \times [0,T)$, the product being understood in the sense of a Cartesian product[77], we can proceed to obtain finite-difference analogues of the partial derivatives in the vorticity transport, vector potential Poisson, and scalar potential Laplace, equations, and their boundary conditions, given in Appendix 1. (Finite-element analogues might be used instead of finite-difference analogues; our choice of the latter is necessarily arbitrary. We note, however, that the use of finite-element and finite-difference analogues can involve us in solving a similar, if not the same, set of matrix equations. Thus our choice between the two approaches, though arbitrary, is not necessarily of crucial importance.) Consider a function ψ of a variable x. Assuming that ψ is sufficiently differentiable, a Taylor series expansion of ψ about the point $x = x_o$ gives[78]:-

$$\psi(x_o+h) = \psi(x_o) + \frac{h}{1!}\psi'(x_o) + \frac{h^2}{2!}\psi''(x_o) + \dots$$

where a prime signifies differentiation with respect to x. Thus we have:-

$$\psi(x_o+h) - \psi(x_o-h) = 2h\psi'(x_o) + \frac{h^3}{3}\psi'''(x_o) + \dots$$

$$\text{or} \quad \psi'(x_o) = \frac{\psi(x_o+h) - \psi(x_o-h)}{2h} + O(h^2) \qquad (3.15)$$

where we say that $a(z)$ is $O(b(z))$ if $\lim_{z \to 0} \frac{a(z)}{b(z)} < \infty$. We see that equation (3.15) gives us an approximate expression for the first derivative of ψ with respect to x at the point x_o, in terms of values of ψ at the points (x_o+h) and (x_o-h), together with an idea of the size of the error involved in the approximation. Because $\psi'(x_o)$ is given in terms of ψ at points on either side of x_o, and because the error is $O(h^2)$, we say that equation (3.15) is a second-order centred-difference approximation to $\psi'(x_0)$. Similarly, we can show that:-

$$\psi'(x_0) = \frac{\psi(x_0+h) - \psi(x_0)}{h} + O(h) \tag{3.16}$$

which is a first-order forward-difference approximation to $\psi'(x_0)$. Backward-difference approximations are defined in a similar manner.

When we come to choose the form of spatial differencing that we will use, we might argue that points in the flow field receive diffused information from all directions, but convected information only from points upstream, and that this should be reflected in the spatial differencing of the diffusion and convection terms. This means that we should use centred-space differencing for the diffusion term, and so-called "upwind-differencing" for the convection term. Upwind-differencing is, however, generally of first-order accuracy only[79]. On the other hand, provided we do not use "downwind-differencing" for the convection term, whose meaning is obvious, and which can be shown to be unconditionally unstable[80], we can and will use centred-space differencing to maintain second-order accuracy both for the diffusion and for the convection terms wherever possible, i.e. at all interior nodes.

We now decide what form of time differencing we will use. Simple first-order time differencing of the equation:-

$$\frac{\partial\psi}{\partial t} = f(\psi) \tag{3.17a}$$

would give:-

$$\frac{\psi(t+\Delta t) - \psi(t)}{\Delta t} = F(\psi) \tag{3.17b}$$

where $F(\psi)$ denotes the finite-difference analogue of $f(\psi)$. If we evaluate $F(\psi)$ using values of ψ at time t, we can immediately determine $\psi(t+\Delta t)$ from equation (3.17b). Such a scheme is termed explicit, for obvious reasons. If, on the other hand, we evaluate $F(\psi)$ using values of ψ at time $t+\Delta t$, an implicit scheme results. Clearly, the latter scheme is more complex inasmuch as it involves implicit solution of equations of the form (3.17b). On the other hand, it can in general be shown to be stable for much larger time-step lengths Δt than the explicit scheme. Indeed, for certain linear equations, such as the diffusion equation:-

$$\frac{\partial \psi}{\partial t} = \nabla^2 \psi \qquad (3.18)$$

implicit schemes can generally be shown to be unconditionally stable, whereas explicit schemes have limitations on the size of Δt for stability[81]. (Various other semi-implicit schemes can be devised[82],[83], but we can neglect them for our purposes, because they offer no additional advantages.) As a result of the inherently superior stability of implicit schemes for large Δt, we will use such a scheme for the numerical solution of the vorticity transport equation. In dimensionless vector form, this is (c.f. equation (3.7)):-

$$\frac{\partial \underline{w}}{\partial t} + ((\nabla h + \nabla_\wedge \underline{A}).\nabla)\underline{w} - (\underline{w}.\nabla)(\nabla h + \nabla_\wedge \underline{A}) = -\frac{1}{Re}\nabla_\wedge \nabla_\wedge \underline{w} \qquad (3.19)$$

Assuming that ∇h is known – and it can be easily determined – at a given time-step n, we know $\underline{A}^n$ and $\underline{w}^n$, and require $\underline{A}^{n+1}$ and $\underline{w}^{n+1}$. To determine them implicitly, and yet retain a linear finite-difference scheme, so that solution of the implicit equations is straightforward, we solve the spatial finite-difference analogue of:-

$$\frac{\underline{w}^{n+1} - \underline{w}^n}{\Delta t} + ((\nabla h + \nabla_\wedge \underline{A}^n).\nabla)\underline{w}^{n+1} - (\underline{w}^{n+1}.\nabla)(\nabla h + \nabla_\wedge \underline{A}^n) = -\frac{1}{Re}\nabla_\wedge \nabla_\wedge \underline{w}^{n+1} \qquad (3.20)$$

In other words, the finite-difference form of the vorticity transport equation uses a vector potential field in the convection terms evaluated at the previous time-step. To update $\underline{A}$, all we need then do is solve the spatial finite-difference analogue of:-

$$\nabla_\wedge \nabla_\wedge \underline{A}^{n+1} = \underline{w}^{n+1} \qquad (3.21)$$

the right-hand side of which is known. (We could then use this estimate of $\underline{A}^{n+1}$ to get a better estimate of $\underline{w}^{n+1}$ in equation (3.20), and so on. In fact, provided Δt is not too large, such iteration is unnecessary, especially since we are primarily interested in the late-time solution of the equations, when changes in the field $\underline{A}$ between time-steps will be small. This point will be further discussed in the penultimate paragraph of this sub-section on enforcing the no-slip condition on the surface of the sphere.)

We now define $C_r^{m,n}$ to be the space of continuous r-dimensional vectors with continuous derivatives up to and including order m in space, and n in time. If we assume that $\underline{A} \in C_3^{8,2}$ (which implies that $\underline{w} \in C_3^{6,2}$) and also that $h \in C_1^{4,0}$ in R then, by suitable Taylor series expansions, we can obtain implicit (in time) centred-space finite-difference approximations to the various terms in the equations in Appendix 1 at all interior nodes in the discretised region R'. These approximations will be second-order correct in space, and first-order correct in time.

Boundary conditions, however, introduce complications, for two reasons. First of all, we cannot use centred-space differences at boundaries (schemes that do generally involve reflection principles, and are very suspect); we must use one-sided differences instead. Provided we assume that $h \in C_1^{4,0}$ and $\underline{A} \in C_3^{4,0}$ (so that $\underline{w} \in C_3^{2,0}$) on δR, the boundary of R, then use of one-sided differences presents no difficulties, though truncation errors do tend to be larger. For example, we can easily show that:-

$$\psi'(x_o) = \frac{-3\psi(x_o) + 4\psi(x_o+h) - \psi(x_o+2h)}{2h} + O(h^2) \qquad (3.22)$$

with the error of order h^2 in equation (3.22) being precisely twice that in equation (3.15).

The second complication arises from the use of a finite-difference, as opposed to an analytical, method of solution, and centres on enforcing the no-slip condition on the surface of the sphere. Our approach has been to set up equations for h, $\underline{A}$, and $\underline{\omega}$. Since the boundary conditions are time-invariant, we can solve for h at once, and regard it as a known field thereafter. Thus we have to solve two coupled second-order equations in $\underline{A}$ and $\underline{\omega}$. (We do not solve a single fourth-order equation in $\underline{A}$, since experience with a fourth-order streamfunction approach in two dimensions suggests that this can lead, as a result of imposing first-order boundary conditions, to computational inefficiency(84),(85), and instability(86).) On the surface of the sphere ($r = r_o$), we have the following boundary conditions:-

$$A_\theta \equiv 0 \equiv A_\phi$$

$$\partial A_r/\partial r = - 2A_r/r$$

$$\omega_r \equiv 0$$

$$\underline{\omega}\wedge\underline{n} = (\nabla\wedge\underline{u})\wedge\underline{n} = (\nabla\wedge\nabla\wedge\underline{A})\wedge\underline{n}$$

together with the no-slip condition $\underline{u} \equiv \underline{0}$. The first three conditions (on $\underline{A}$) are sufficient to determine $\underline{A}$. And the no-slip condition is necessary for us to specify $\underline{\omega}$ on the surface of the sphere (indeed, the no-slip condition provides the mechanism by which vorticity is generated at the surface of the sphere(87)). This is the only correct way in which the boundary conditions can be applied. A difficulty arises, however, in numerically enforcing the no-slip condition, so that vorticity is in fact correctly generated at the surface of the sphere. Out of various possible schemes, numerical results obtained both in the present work and elsewhere(88) suggest that the following scheme is best:-

$$\omega_r(r_o) \equiv 0$$

$$\omega_\theta(r_o) = \frac{-1}{2\Delta r}\Big(- 3v_\phi(r_o) + 4v_\phi(r_o+\Delta r) - v_\phi(r_o+2\Delta r)\Big) + \frac{V_\phi(r_o)}{r_o}$$

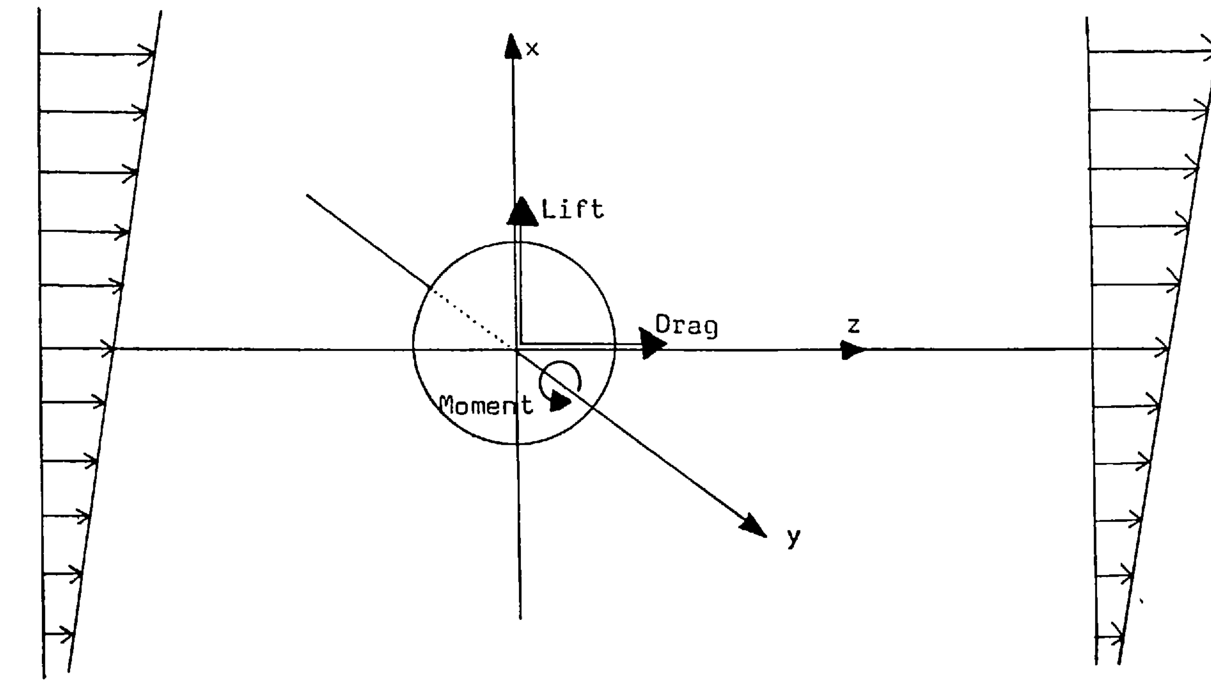
$$\omega_\phi(r_o) = \frac{1}{2\Delta r}\Big(- 3v_\theta(r_o) + 4v_\theta(r_o+\Delta r) - v_\theta(r_o+2\Delta r)\Big) - \frac{V_\theta(r_o)}{r_o} \qquad (3.23)$$

where $\underline{v} = \nabla_\wedge \underline{A}$ and $\underline{V} = \nabla h$. (Clearly, the no-slip condition implies that $\underline{v} + \underline{V} \equiv \underline{0}$ at $r = r_0$, which can be used to simplify these expressions.) Comparison of equations (3.23) with equation (3.22) suggests that this scheme is of second-order accuracy; since, however, $\underline{v}$ and $\underline{V}$ are obtained from $\underline{A}$ and h, respectively, by numerical differentiation, the scheme is, in fact, only of first-order accuracy. This is not necessarily a disadvantage, since first-order schemes for boundary conditions can give more accurate results than higher-order schemes[89], and also present fewer numerical stability problems. We note, incidentally, that the field $\underline{v}$ is not known a priori, so we use the field $\underline{v}$ evaluated at time t to give the boundary conditions on $\underline{\omega}$ at time $t+\Delta t$. We could then, if we wished, obtain an estimate of $\underline{v}(t+\Delta t)$, and hence a better estimate of the boundary conditions on $\underline{\omega}(t+\Delta t)$, and so on. Such iteration seems unnecessary, however, provided (as here) the $\nabla_\wedge \underline{A}$ terms in the vorticity transport equation are also evaluated at time t[90].

We are now able to write down the finite-difference form of the equations and boundary conditions of Appendix 1, together with suitable modifications on the polar axis. Because of their relative length, however, they are written elsewhere (c.f. Appendix 3; the notation used is that of this sub-section).

3.2.4 Drag, Lift and Moment Coefficients

This sub-section completes our formulation of the problem of flow past the neutrally buoyant, non-rotating, stationary sphere. We show how, given the vorticity field $\underline{\omega}$, we may calculate the drag in the positive z-direction, the lift in the positive x-direction, and the moment or torque (interpreted in a right-handed sense) about the y-axis, on the sphere. These are not the only net forces and moments that can act on the sphere - periodic vortex shedding, for example, would give



rise to periodic forces. We may reasonably expect, however, that, for low Reynolds numbers, the symmetry of the problem will be such that these are the predominant effects. We note, incidentally, that, because the sphere is assumed to be neutrally buoyant, there will be no buoyancy contribution to the lift or drag. Similarly, because the sphere is assumed to be stationary, it mainteins a constant position with respect to the flow at infinity. Finally, we note that, until otherwise stated, all variables are dimensional.

The drag force on the sphere may be thought of as comprising a form component (due to normal or pressure forces) and a friction component (due to tangential or viscous forces). The pressure force acting in the z-direction on an element of area dA is given by $(-p\cos\theta)dA$. Now the element of area dA is given, on the surface of the sphere (radius r_o), by $dA = r_o^2 \sin\theta \, d\theta \, d\phi$. Thus the form drag D_p on the sphere is given by:-

$$D_p = -r_o^2 \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} p \cos\theta \sin\theta \, d\theta \, d\phi \qquad (3.24)$$

If τ_{ab} denotes the flux of b-momentum in the a-direction, then the viscous force acting in the z-direction on an element of area dA is $(-\tau_{r\theta})(-\sin\theta)dA$. Thus the friction drag D_v on the sphere is given by:-

$$D_V = \int_{\phi=0}^{2\pi}\int_{\theta=0}^{\pi} \tau_{r\theta} \sin\theta\, r_o^2 \sin\theta \, d\theta \, d\phi$$

We can easily show, using the no-slip condition, that, on the surface of the sphere, $\tau_{r\theta} = -\mu\, \partial u_\theta/\partial r$ [91] $= -\mu w_\phi$, where μ is the viscosity of the fluid ($\mu = \rho\nu$). Thus:-

$$D_V = -\mu r_o^2 \int_{\phi=0}^{2\pi}\int_{\theta=0}^{\pi} w_\phi \sin^2\theta \, d\theta \, d\phi \qquad (3.25)$$

The total drag D is given by the sum of D_p and D_V.

The lift force on the sphere may, by analogy with the drag force, be thought of as comprising a form component and a friction component. The pressure force acting in the x-direction on an element of area dA is $(-p \sin\theta\cos\phi)dA$, so that the form lift L_p on the sphere is given by:-

$$L_p = - r_o^2 \int_{\phi=0}^{2\pi}\int_{\theta=0}^{\pi} p \sin^2\theta \cos\phi \, d\theta \, d\phi \qquad (3.26)$$

The viscous force acting in the x-direction on an element of area dA is $(-\tau_{r\theta})(\cos\theta\cos\phi)dA + (-\tau_{r\phi})(-\sin\phi)dA$. We can easily show that, on the surface of the sphere, $\tau_{r\phi} = -\mu\, \partial u_\phi/\partial r = \mu w_\theta$, in the same way that we showed that $\tau_{r\theta} = -\mu w_\phi$. Thus the friction lift L_V on the sphere is given by:-

$$L_V = \mu r_o^2 \int_{\phi=0}^{2\pi}\int_{\theta=0}^{\pi} (w_\theta \sin\phi + w_\phi \cos\theta\cos\phi) \sin\theta \, d\theta \, d\phi \qquad (3.27)$$

The total lift L is given by the sum of L_p and L_V.

The moment, or torque, M on the sphere arises as a result of viscous effects alone. Since, by assumption, the sphere is not rotating, we can easily show that the moment on an element of area dA on the surface of the sphere is $(-\tau_{r\theta}\cos\phi + \tau_{r\phi}\sin\phi\cos\theta)r_o dA$, so that:-

$$M = \mu r_o^3 \int_{\phi=0}^{2\pi}\int_{\theta=0}^{\pi} (w_\phi \cos\phi + w_\theta \sin\phi \cos\theta) \sin\theta \, d\theta \, d\phi \qquad (3.28)$$

Given $\underline{w}$ and p, we see that we can immediately obtain the drag, lift and moment on the sphere. Our formulation does not, however, give p explicitly; nevertheless, it is a simple matter to determine it on the surface of the sphere: we apply the no-slip condition on the velocity field to the Navier-Stokes equations, and can easily show that:-

$$\frac{1}{\nu r_o}\frac{\partial p}{\partial \theta} = \frac{2}{r_o}\frac{\partial u_\theta}{\partial r} + \frac{\partial^2 u_\theta}{\partial r^2} = \frac{w_\phi}{r_o} + \frac{\partial w_\phi}{\partial r}$$

and

$$\frac{1}{\nu r_o \sin\theta}\frac{\partial p}{\partial \phi} = \frac{2}{r_o}\frac{\partial u_\phi}{\partial r} + \frac{\partial^2 u_\phi}{\partial r^2} = -\frac{w_\theta}{r_o} - \frac{\partial w_\theta}{\partial r}$$

and hence:-

$$p = \begin{cases} \nu r_o \displaystyle\int_0^\theta \left(\frac{w_\phi}{r_o} + \frac{\partial w_\phi}{\partial r}\right) d\theta + p(r=r_o, \theta=0) \\ -\nu r_o \sin\theta \displaystyle\int_0^\phi \left(\frac{w_\theta}{r_o} + \frac{\partial w_\theta}{\partial r}\right) d\phi + p(r=r_o, \theta=\theta, \phi=0) \end{cases} \tag{3.29}$$

which gives p directly from a knowledge of $\underline{w}$.

As in the formulation of the rest of the problem, we use the methods of dimensional analysis to decrease the effective number of parameters which we have to deal with. We choose the diameter D of the sphere as a characteristic length, and the projected area of the sphere, $\pi r_o^2 = \frac{1}{4}\pi D^2$, as a characteristic area. Finally, as a characteristic kinetic energy per unit volume, we choose $\frac{1}{2}\rho U_{char}^2$, where $U_{char} = U_\infty^o$ if this is non-zero; otherwise, $U_{char} = \sigma D$. (Note that, by convention, the pressure is made dimensionless with respect to twice the characteristic kinetic energy per unit volume.) We can now define the dimensionless drag, lift and moment on the sphere as follows (note: these are not standard definitions, necessarily; indeed, there seem to be no universally accepted definitions):-

$$C_{DP} = \frac{8\, D_p}{\pi \rho D^2 U_{char}^2} = -\frac{1}{\pi}\int_{\phi=0}^{2\pi}\int_{\theta=0}^{\pi} p \sin 2\theta \, d\theta \, d\phi \tag{3.30}$$

$$C_{DV} = \frac{8\, D_V}{\pi \rho D^2 U_{char}^2} = -\frac{2}{\pi Re}\int_{\phi=0}^{2\pi}\int_{\theta=0}^{\pi} w_\phi \sin^2\theta \, d\theta \, d\phi \tag{3.31}$$

$$C_D = C_{DP} + C_{DV} \tag{3.32}$$

$$C_{LP} = \frac{8\, L_p}{\pi \rho D^2 U_{char}^2} = -\frac{2}{\pi} \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} p \sin^2\theta \cos\phi \, d\theta \, d\phi \tag{3.33}$$

$$C_{LV} = \frac{8\, L_v}{\pi \rho D^2 U_{char}^2} = \frac{2}{\pi Re} \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} (w_\theta \sin\phi + w_\phi \cos\theta \cos\phi) \sin\theta \, d\theta \, d\phi \tag{3.34}$$

$$C_L = C_{LP} + C_{LV} \tag{3.35}$$

$$C_M = \frac{-\,8\, M}{\pi \rho D^3 U_{char}^2} = -\frac{1}{\pi Re} \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} (w_\phi \cos\phi + w_\theta \sin\phi \cos\theta) \sin\theta \, d\theta \, d\phi \tag{3.36}$$

and:-

$$p = \begin{cases} \dfrac{1}{2Re} \displaystyle\int_0^{\theta} \left(2w_\phi + \frac{\partial w_\phi}{\partial r}\right) d\theta + p(r=\tfrac{1}{2}, \theta=0) \\[2ex] \dfrac{-1}{2Re \sin\theta} \displaystyle\int_0^{\phi} \left(2w_\theta + \frac{\partial w_\theta}{\partial r}\right) d\phi + p(r=\tfrac{1}{2}, \theta=\theta, \phi=0) \end{cases} \tag{3.37}$$

where all variables are now dimensionless again, and $Re = U_{char}D/\nu$. We note, incidentally, that, for positive values of the shear rate σ, we anticipate (correctly) that the moment M will be negative; this explains the minus sign in the expression for the moment coefficient C_M. We also note that, because pressure is arbitrary up to a constant, we can (and will) set $p(r=\frac{1}{2}, \theta=0) \equiv 0$ in equation (3.37).

Equations (3.30) - (3.37) give the drag, lift and moment coefficients of the sphere, together with the pressure field on the sphere, analytically in terms of the vorticity field $\underline{w}$. In order that we may determine them numerically, we require discrete analogues of line integrals (to obtain the pressure field on the surface of the sphere) and surface integrals (to obtain the various coefficients). The obvious way to obtain such discrete analogues is to replace the integrals by summations. Thus if some coordinate x is discretised into constant intervals Δx, the line integral $\int_{x_1}^{x_2} \psi(x)\, dx$ (where $x_2 = x_1 + n\Delta x$, say) might be replaced by the sum:-

$$\psi(x_1)\frac{\Delta x}{2} + \psi(x_1+\Delta x)\Delta x + \dots + \psi(x_1+(n-1)\Delta x)\Delta x + \psi(x_2)\frac{\Delta x}{2}$$

i.e. $$\int_{x_1}^{x_2} \psi(x)\, dx \simeq \Delta x \left(\frac{\psi(x_1) + \psi(x_2)}{2} + \sum_{i=1}^{n-1} \psi(x_1 + i\Delta x) \right) \tag{3.38}$$

which we may apply directly to equations (3.37). (Alternative, more accurate, methods such as Simpson's rule[92] cannot be used because the number of intervals n is not always suitable. Also, obtaining the line integral between points other than x_1 and x_2 (for example, x_1 and $x_2+\Delta x$) is particularly straightforward with expression (3.38).) Note that we can determine the pressure at any node by integration along a variety of paths; the results should be (and will be, analytically) identical. Any discrepancies that occur in the numerical implementation are due to errors in the solution for $\underline{w}$, and errors in the numerical line integration. The discrepancies thus give a guide to the errors involved in the whole numerical scheme.

A discrete analogue of a surface integral may be obtained in a similar manner to the discrete analogue of the line integral in equation (3.38). Thus it is easy to show that:-

$$\int_{\phi=0}^{2\pi}\int_{\theta=0}^{\pi} \psi \sin\theta \, d\theta \, d\phi \simeq 2\pi(\psi(\theta=0) + \psi(\theta=\pi))(1 - \cos\frac{\Delta\theta}{2}) + \sum_{\phi=0}^{2\pi-\Delta\phi} \sum_{\theta=\Delta\theta}^{\pi-\Delta\theta} 2\psi \sin\theta \sin\frac{\Delta\theta}{2} \Delta\phi \tag{3.39}$$

which we may apply directly to equations (3.30), (3.31), (3.33), (3.34), and (3.36).

With this, we complete the formulation of our problem.

3.3 Method of Solution

In this section, we will first discuss the methods we shall adopt to solve the finite-difference analogues of the vorticity transport, vector potential Poisson, and scalar potential Laplace, equations. We will then show how the complete solution of the problem may be obtained.

3.3.1 Solution of the Finite-Difference Equations

When we come to choose between various methods for solving the finite-difference equations developed in the previous section, we must distinguish between the parabolic (vorticity transport) equation and the two elliptic (vector potential Poisson and scalar potential Laplace) equations. In general, parabolic equations are based on a half-open integration domain, whereas elliptic equations are based on a closed integration domain. (Thus the vorticity transport equation has to be solved in the half-open region $R \times [0,T)$, whereas the vector potential Poisson and scalar potential Laplace equations have to be solved in the closed region R.) The result of this is that parabolic equations can be "marched" through the half-open domain from a (given) initial state, being modified by the boundary conditions as they go. Elliptic equations, on the other hand, have to be solved everywhere in the closed domain simultaneously. Consequently, the numerical schemes which we can use to solve parabolic equations are, in general, different from those we can use to solve elliptic equations.

Various methods exist for solving parabolic equations, depending on how we march in the half-open coordinate direction $d \in D$ in the integration domain $C \times D$ (in our problem, the coordinate d is time). As we discussed in sub-section 3.2.3 above, an explicit method of solution means that if we are at some point $d \in D$ in the integration domain $C \times D$ we may proceed explicitly to the point $d+\Delta d$ which is further along the integration domain, but instabilities may result. An implicit method, on the other hand, although very much more stable, will involve more computational work in proceeding from the point d to the point $d+\Delta d$. It is possible, however, to reduce the amount of computational work we must do in an implicit method by splitting each interval Δd into n equal sub-intervals, where n is the dimensionality of the space C in the half-open region CxD (in our problem, C is the region $R \subset E_3$, so n = 3). In the first sub-interval, between d and $d+\frac{\Delta d}{n}$, we solve equations which are implicit in one dimension of C only. In the second sub-interval, between $d+\frac{\Delta d}{n}$ and $d+\frac{2\Delta d}{n}$, we solve equations which are implicit in another one dimension of C only, and so on, until we have reached the point $d+\Delta d$, when we will have solved equations implicit in each of the n dimensions of C in turn. Such so-called "locally one-dimensional" or "alternating-direction implicit" methods are not new[93], although extensions to $n>2$ dimensions are more recent[94],[95]. Their basic value lies in the assumption that it will be easier to solve several small problems than one big one; provided this is the case (and it often is), then the advantages of such methods are obvious, combining relative simplicity of solution with high stability. The principal difficulty with such methods, on the other hand, is that they are relatively complex to formulate. Also, mixed derivatives require special treatment (though they can be handled[96]), and the presence of derivatives of higher than second-order means that it is not possible for the matrix equivalent of the finite-difference equations to be manipulated into a tridiagonal form (which is easy to invert[97]).

Because the parabolic equation of interest to us, the vorticity transport equation, possesses mixed spatial derivatives, as well as spatial derivatives of higher than second-order, difficulties arising in locally one-dimensional and alternating-direction implicit methods become particularly acute. We shall, therefore, use a fully implicit method for the vorticity transport equation, as already mentioned in sub-section 3.2.3. More particularly, we shall use an iterative method which, as it happens, can be applied both to the parabolic vorticity transport equation, and to the elliptic vector potential Poisson and scalar potential Laplace equations.

Iterative schemes, which are a common feature of many methods for solving elliptic equations, involve guessing a solution, and then determining by how much the guess does not satisfy the equations and the boundary conditions. On the basis of this information, a (better) estimate of the solution is obtained; the amount by which this does not satisfy the equations and the boundary conditions leads to a further estimate, and so on, until, if the process converges, the estimates tend to the solution of the equations. So that we can select the particular iteration scheme that we will use, it will be helpful if we express the various finite-difference equations in matrix form. The vorticity transport equation may be written:-

$$\underline{\underline{R}}\ \underline{w}^{*n+1} = \underline{w}^{*n} \quad \text{at interior nodes} \tag{3.40}$$

with Dirichlet boundary conditions on $\underline{w}^*$,

the vector potential Poisson equation may be written:-

$$\underline{\underline{S}}\ \underline{A}^{*n+1} = \underline{w}^{*n+1} \quad \text{at interior nodes} \tag{3.41}$$

with mixed boundary conditions on $\underline{A}^*$,

and the scalar potential Laplace equation may be written:-

$$\underline{\underline{T}}\ \underline{h}^{*} = \underline{0} \quad \text{at interior nodes} \tag{3.42}$$

with Neumann boundary conditions on $\underline{h}^*$,

where, if N is the number of interior nodes in the discretised region R', $\underline{w}^*$ is the 3N-element column vector whose elements are components of the field $\underline{w}$, $\underline{A}^*$ is the 3N-element column vector whose elements are components of the field $\underline{A}$, and $\underline{h}^*$ is the N-element column vector whose elements are from the field h. $\underline{\underline{R}}$, $\underline{\underline{S}}$, and $\underline{\underline{T}}$ are (sparse) square matrices. We may regard the right-hand sides of each of the above equations, together with the coefficients of the matrices $\underline{\underline{R}}$, $\underline{\underline{S}}$, and $\underline{\underline{T}}$, as known. We may, therefore, represent any of the above equations thus:-

$$\underline{\underline{A}}\ \underline{x} = \underline{y} \tag{3.43}$$

with $\underline{\underline{A}}$ and $\underline{y}$ known. Now we can decompose the matrix $\underline{\underline{A}}$ as follows:-

$$\underline{\underline{A}} = \underline{\underline{L}} + \underline{\underline{D}} + \underline{\underline{U}} \quad (3.44)$$

where $\underline{\underline{L}}$ is a strictly lower triangular matrix,

$\underline{\underline{D}}$ is a strictly diagonal matrix,

and $\underline{\underline{U}}$ is a strictly upper triangular matrix.

Using this decomposition, we can construct certain iterative schemes for solving equation (3.43)[98]. Thus the so-called "Jacobi" method may be written:-

$$\underline{x}^{(k)} = - \underline{\underline{D}}^{-1}(\underline{\underline{L}} + \underline{\underline{U}})\underline{x}^{(k-1)} + \underline{\underline{D}}^{-1}\underline{y} \quad (3.45)$$

the superscript k denoting the k-th approximation to $\underline{x}$. Similarly, the so-called "Gauss-Seidel" method may be written:-

$$\underline{x}^{(k)} = - (\underline{\underline{L}} + \underline{\underline{D}})^{-1}\underline{\underline{U}}\,\underline{x}^{(k-1)} + (\underline{\underline{L}} + \underline{\underline{D}})^{-1}\underline{y} \quad (3.46)$$

and the so-called "point successive over-relaxation" method may be written:-

$$\underline{x}^{(k)} = (\underline{\underline{D}} + \omega\underline{\underline{L}})^{-1}\left[((1 - \omega)\underline{\underline{D}} - \omega\underline{\underline{U}})\underline{x}^{(k-1)} + \omega\underline{y}\right] \quad (3.47)$$

where ω is a relaxation parameter, chosen so as to optimise the convergence of the iterative scheme (in general, $0 < \omega < 2$; usually, $1 \leqslant \omega < 2$). Any of these three well-known schemes may be initiated merely by guessing a suitable vector $\underline{x}^{(0)}$; this is generally quite straightforward[99]. Then, apart from the choice of ω in the point successive over-relaxation scheme, we can proceed from iteration to iteration until convergence is effectively achieved. It is possible to show, both analytically (for certain forms of the matrix $\underline{\underline{A}}$) and numerically, that of the three schemes mentioned the point successive over-relaxation scheme is generally the most efficient, and converges to the solution of equation (3.43) fastest, provided a suitable (preferably, an optimum) value of ω is used; the optimum value of ω can be determined either numerically or, in certain cases, analytically[100]. We will, therefore, use a point successive over-relaxation scheme for the solution of each of the finite-difference equations. Since, however, the matrices involved in these equations are not of the form for which analytical prediction of the optimum relaxation parameter is possible, a numerical trial-and-error procedure must be adopted instead.

To conclude, we should note that the schemes (3.45) - (3.47) are not the only possible ones nor, indeed, necessarily the most efficient ones that we might use. For example, "block" schemes can be devised[101], in which several elements of $\underline{x}$ are updated simultaneously; this can sometimes be desirable. These more complex schemes have not, however, been conclusively shown to be more efficient or faster than the simple schemes above.

3.3.2 Overall Solution

We are now in a position where we can solve the complete flow problem, given the necessary parameters which describe it (such as the Reynolds number) and suitable initial conditions. As we noted in sub-section 3.2.1, we are primarily interested in the late-time (steady-state) solution of the problem. It would be useful, therefore, from a computational point of view, if we could use an estimate of the solution as an initial condition, whether or not this estimate is likely to be physically realisable. For the case of uniform flow past the sphere, a good initial estimate might be Stokes flow[102]:-

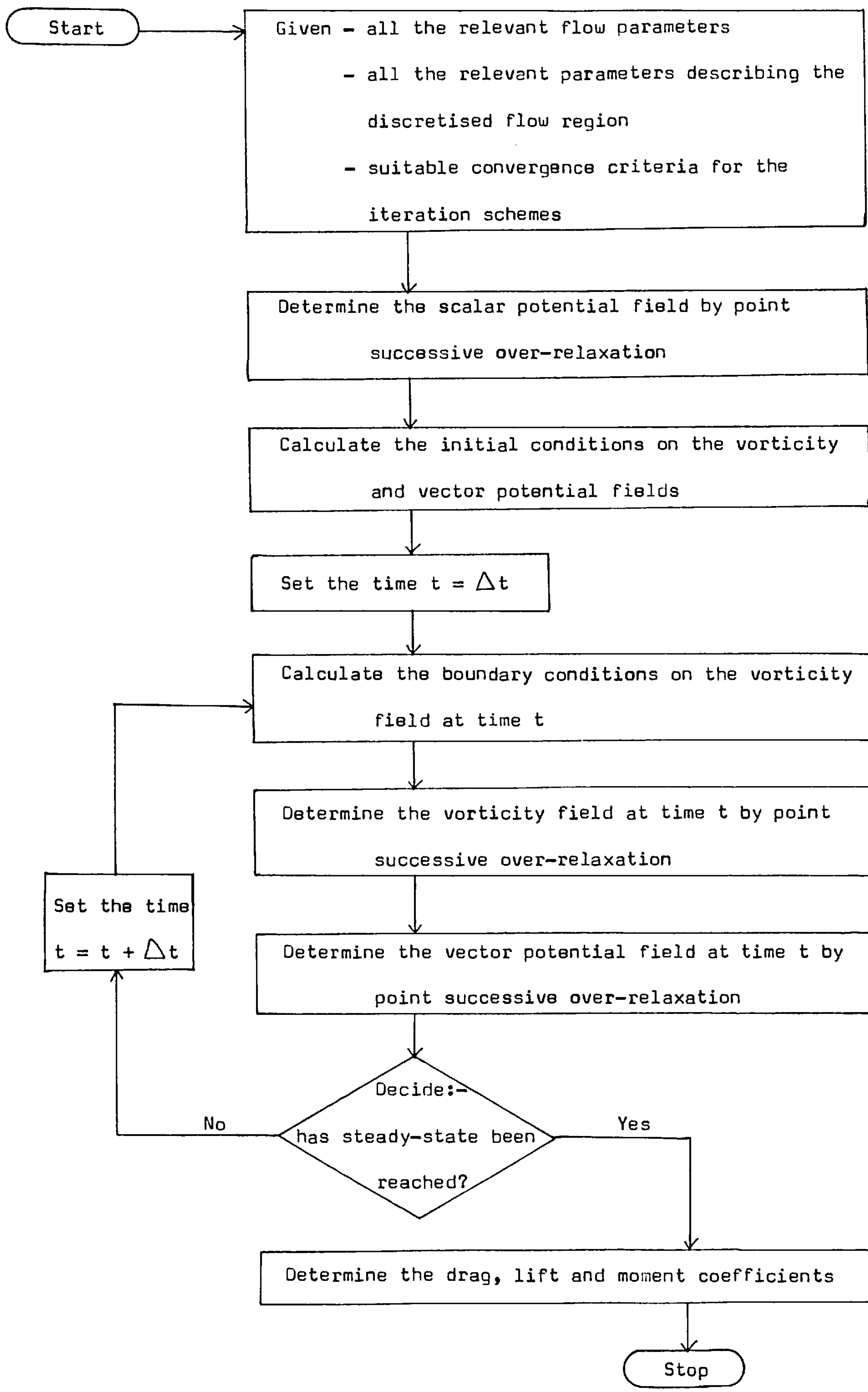
$$\left.\begin{aligned} u_r &= (1 - \tfrac{3}{4}(1/r) + \tfrac{1}{16}(1/r^3))\cos\theta \\ u_\theta &= -(1 - \tfrac{3}{8}(1/r) - \tfrac{1}{32}(1/r^3))\sin\theta \\ u_\phi &= 0 \end{aligned}\right\} \quad (3.48)$$

the variables being dimensionless. For such flows, this is in fact the initial condition we will use. For the case of shear flow past the sphere, however, the situation is not quite so straightforward. Although the analytical solution for centred linear-shear (i.e. $U^0_\infty \equiv 0$) Stokes flow exists[103], it is relatively complex and its use is, in fact, unnecessary, because it is not very critical what initial condition we use (provided it is reasonably smooth): the convergence time to reach steady-state will not vary much[104]. For shear flows, therefore, we might – and will – use a linear radial variation between the boundary conditions which hold on the sphere ($r = r_o$) and those which hold on the outer envelope ($r = r^*$) as an initial condition on the velocity field. Thus, since there is no slip on the surface of the sphere, and the velocity field at the outer envelope is $\underline{U}_\infty$, the initial condition is:–

$$\left.\begin{aligned} \underline{u} &= ((r - r_o)/(r^* - r_o))\ \underline{U}_\infty \\ \text{i.e.}\quad \underline{u} &= ((r - \tfrac{1}{2})/(r^* - \tfrac{1}{2}))\ \underline{U}_\infty \end{aligned}\right\} \quad (3.49)$$

Given these initial conditions on $\underline{u}$, it is a straightforward matter to obtain (analytical) initial conditions on the vorticity field $\underline{w}$, and hence (numerical) initial conditions on the vector potential field $\underline{A}$ by iterative solution of the vector potential Poisson equation. Note, incidentally, that when there is uniform flow past the sphere, we could either impose the analytical boundary conditions obtained by putting $r = r^*$, or those obtained by letting $r \rightarrow \infty$, at the outer envelope. Numerical tests indicate that it does not matter much which course we adopt; accordingly, to be consistent with the rest of the formulation, we impose the $r \rightarrow \infty$ boundary conditions at the outer envelope.

We can now proceed to solve the complete flow problem. A computer program has been written, in what is essentially FORTRAN IV, which will solve the problem, given all the relevant parameters defining the discretised flow system. A listing of the program and its associated subroutines is given in Appendix 4 (q.v.), while details of the storage requirements of the program are included with the results presented in section 3.4 below. We present here, on the other hand, a flowchart showing in outline the complete computational procedure:–


Start
Given - all the relevant flow parameters
- all the relevant parameters describing the discretised flow region
- suitable convergence criteria for the iteration schemes
Determine the scalar potential field by point successive over-relaxation
Calculate the initial conditions on the vorticity and vector potential fields
Set the time t = Δt
Calculate the boundary conditions on the vorticity field at time t
Determine the vorticity field at time t by point successive over-relaxation
Determine the vector potential field at time t by point successive over-relaxation
Decide:- has steady-state been reached?
No
Yes
Set the time t = t + Δt
Determine the drag, lift and moment coefficients
Stop

The implementation of this procedure raises three main questions, the first two of which are related:-

(i) how do we decide when the over-relaxation schemes have converged?

(ii) how do we decide when steady-state has been reached?

(iii) for any reasonable three-dimensional problem, the procedure involves a large amount of computer storage - what can we do to minimise it?

There is, in fact, no entirely satisfactory answer to the first two questions. Various criteria can be applied[105]; we will use the following:-

(i) the over-relaxation schemes will be said to have converged if:-

$$\max_{\text{all } i} \left| x_i^{(k)} - x_i^{(k-1)} \right| \leqslant \delta \qquad (3.50)$$

where $x_i^{(k)}$ represents the estimate of the element x_i of the vector $\underline{x}$ representing the unknown vorticity, vector potential, or scalar potential fields obtained in the k-th iteration, and δ is very small.

(ii) steady-state will be said to have been reached when a criterion analogous to inequality (3.50) has been satisfied, with $x_i^{(k)}$ now representing the value of x_i at the k-th time-step.

The answer to the third question is that we can do a great deal to minimise the computer storage required, with possibly the greatest savings being made in the storage of the matrices $\underline{R}$, $\underline{S}$, and $\underline{T}$ (c.f. equations (3.40) - (3.42)). As we have already noted, these matrices are (very) sparse. Accordingly, we need only store non-zero coefficients of these matrices, together with indicators of the positions of these coefficients in the matrices.

A variety of minor difficulties occur in the actual implementation of the computational scheme. The only one of any importance is the following. Because we are solving the problem numerically as opposed to analytically, the vector potential field $\underline{A}$ is not exactly solenoidal in the flow field. This can, and in certain circumstances does, manifest itself in the addition of a small constant field of the form $\nabla\phi$ to $\underline{A}$ at each iteration of the vector potential Poisson equation. Since the field $\nabla\phi$ does not affect the representation of the velocity field (recall:- $\nabla_\wedge\nabla\phi \equiv \underline{0}$), this is not a serious problem. All that we need to do is incorporate an additional (but alternative) convergence criterion into the relaxation scheme:-

$$\max_{\text{all } i} \left| (A_i^{*(k)} - A_i^{*(k-1)}) - (A_i^{*(k-1)} - A_i^{*(k-2)}) \right| \leqslant \varepsilon$$

i.e.
$$\max_{\text{all } i} \left| A_i^{*(k)} - 2A_i^{*(k-1)} + A_i^{*(k-2)} \right| \leqslant \varepsilon \qquad (3.51)$$

where $A_i^{*(k)}$ is the k-th approximation to the element A_i^* of the vector $\underline{A}^*$ representing the unknown elements of the vector potential field $\underline{A}$, and ε is very small.

3.4 Numerical Results

In this section, we will present the numerical results obtained using the computer program mentioned earlier. We will first present some basic results, concerning the number of mesh-points in the computational flow field, the position of the outer envelope, the choice of optimum relaxation parameters, and the maximum size of time-steps we can use. We will then present some typical late-time velocity fields predicted by the program. Finally, we will present computer predictions of the late-time drag, lift and moment coefficients of the sphere for a range of Reynolds numbers, shear rates at infinity, and centre-line velocities at infinity. The results will be discussed in section 4.1 of Chapter 4.

3.4.1 Preliminary Results

The computer on which all numerical results to be presented were obtained is the CDC 6400 / CYBER 7314 system at Imperial College, London. The maximum central (i.e. fast-access) memory available on this system is, after the requirements of the compiler etc have been met, 50K sixty-bit words (1K = 1024). This fact is important when we consider the number of mesh-points in our spatial discretisation of the flow field. The smallest problem that can give useful results for shear flow past a sphere involves a discretised region R' with two radial, two polar angle, and four azimuthal angle, spacings. This means that the flow region has six interior and twelve boundary mesh-points. Now the total storage requirements of the computer program simulating the flow for a problem of this size would be about 28K words on the 6400 / 7314 system, of which about 22K words would be needed to store the program and associated subroutines, which are independent of the size of the problem. The remaining 6K words are required for the storage of matrix coefficients, vector components, etc. The reason why such a large amount of storage is required for this small eighteen-node problem is that the analytical equations in spherical polar coordinates are relatively lengthy (c.f. Appendix 1), and hence the matrix band-widths in the discretised equations are relatively large. If we were to increase the size of the problem to one with, say, three radial, four polar angle, and four azimuthal angle, spacings, giving a flow region with fifty-six mesh-points, then the total storage required would be about 45K words. Because the maximum central memory available is 50K words, as we noted above, this (three radial) × (four polar angle) × (four azimuthal angle) spacing problem is in fact the largest that we can efficiently handle on the 6400 / 7314 system. Accordingly, in all computer runs, a fifty-six mesh-point flow field was used. Although this implies a very coarse mesh, quite acceptable results could be obtained using it, as we will see in sub-section 3.4.3.

Given this discretised region, we may now determine the optimum relaxation parameters for use in the point successive over-relaxation schemes. Numerical trial-and-error tests with various relaxation parameters showed that, for the scalar potential Laplace equation, the optimum value was 1.7, one hundred iterations being typically necessary for convergence with $\delta = 10^{-4}$ (c.f. inequality (3.50)). For both of the other equations, the optimum relaxation parameter was 1.0, corresponding to the Gauss-Seidel scheme, as opposed to point successive over-relaxation. This is not too surprising, in fact, because relatively few iterations are needed for convergence for these equations (typically, five for the vorticity transport equation and twenty for the vector potential Poisson equation, with $\delta = 10^{-5}$ and $\varepsilon = 10^{-3}$ (c.f. inequalities (3.50) and (3.51))). Although over-relaxation with the optimum relaxation parameter is generally the best strategy when a relatively large number of iterations is involved, using values of the relaxation parameter less than the optimum can lead to better results when only a small number of iterations is involved; indeed, if just one iteration is to be performed, it can be shown that the Gauss-Seidel scheme always gives the greatest reduction in error[106].

Having decided on suitable relaxation parameters, we may now determine the radius r* of the outer envelope. This is not, however, a straightforward problem. If we position it close to the sphere, the blockage effect of the sphere will be very high. Positioning it far away from the sphere, on the other hand, will lead to inaccuracy, since we can only have three radial spacings between the sphere and the envelope, and truncation errors are a function of this spacing. Some sort of compromise is clearly required. No single answer is satisfactory, however, for all flow situations. Thus where viscous effects are important (in the moment coefficient of the sphere, for example), we are interested in detail close to the sphere, and would aim to make r* relatively small. Where pressure effects are important, on the other hand (in the drag and lift coefficients of the sphere, for example), far field pressure data is important, and this is sensitive to blockage effects; consequently, we would aim to make r* relatively large. Numerical tests indicate that:-

(i) where pressure effects are important, the balance between blockage effects and loss of accuracy occurs at about $r = 1.25$;

(ii) where viscous effects are important, the balance occurs at about $r = 0.875$.

Consequently, we will choose $r^* = 1.25$ (corresponding to $\Delta r = 0.25$) when determining pressure dominated effects, and $r^* = 0.875$ ($\Delta r = 0.125$) when determining viscosity dominated effects. (So that the fineness of the balance can be seen, certain results obtained using both values of r* will be presented in the following sub-sections.)

The final numerical parameter which we have to fix is the size of the time-step, Δt. So that we can reach steady-state as rapidly as possible, we clearly want Δt to be as large as possible. Analytical stability analyses(107) for various model equations give indications of the maximum Δt that we can use but, in the final analysis, a trial-and-error procedure must be adopted. Numerical tests showed that the maximum Δt we can use varies approximately as the square-root of the Reynolds number of the problem, being about 10^{-3} at a Reynolds number of unity. The time taken to reach effective steady-state at this Reynolds number is about 10^{-1}, and varies approximately linearly with the Reynolds number. (The computer time required to set up and solve a typical problem, incidentally, is about five minutes at a Reynolds number of 10^{-2}, and ten minutes at a Reynolds number of 10^{2}, on the 6400 / 7314 system.)

To conclude this sub-section, we recall our assumptions that the late-time (or steady-state) solution and the computer time taken to reach steady-state are virtually independent of the choice of initial conditions (c.f. sub-sections 3.2.1 and 3.3.2). A series of numerical tests showed that these assumptions are in fact justified, provided the initial conditions are reasonably smooth.

3.4.2 Velocity Fields

Our purpose in presenting (late-time) velocity fields is to enable us to visualise the flow past the sphere noting, for example, recirculatory regions and stagnation points. Because of the relatively coarse nature of the mesh used in the computer program, however, localised or small-scale phenomena, such as points of flow separation, are unlikely to be accurately predicted, or even predicted at all, in the velocity fields. (This is in contrast with overall or integral properties of the flow, such as drag, lift and moment coefficients, which we might - correctly - expect to be more accurately predicted.) A direct result of this is that the predicted velocity fields at various Reynolds numbers are very similar qualitatively (and even quantitatively, at low Reynolds numbers, because the equations of motion are then effectively independent of Reynolds number). For this reason, therefore, we shall only present velocity fields obtained for a Reynolds number of unity. We shall, moreover, only present velocity fields for which:-

(i) the centre-line velocity at infinity $U^{o}_{\infty} = 1$ and the shear rate at infinity $\sigma \equiv 0$ i.e. uniform flow past the sphere;

(ii) $U^{o}_{\infty} \equiv 0$ and $\sigma = 1$ i.e. centred linear-shear flow past the sphere;

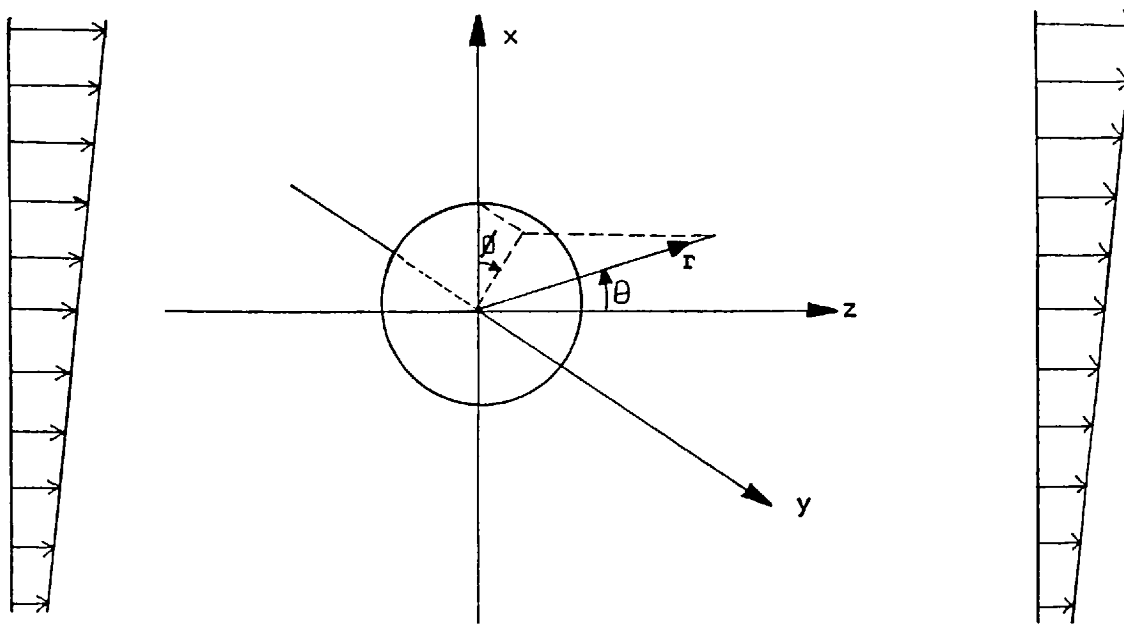
(iii) $U^{o}_{\infty} = 1$ and $\sigma = 1$ i.e. uniform plus centred linear-shear flow past the sphere.

This is because other combinations of the parameters U^{o}_{∞} and σ give qualitatively similar results.

When it comes to presenting the velocity fields in a useful way, we encounter the problem of depicting a three-dimensional field on a two-dimensional surface. Possibly the most useful information can be gained from projections of the velocity field onto:-

(i) the x-z plane $y = 0$ ($\phi = 0, \pi$), and

(ii) the y-z plane $x = 0$ ($\phi = \frac{\pi}{2}, \frac{3\pi}{2}$).



(There is nothing to be gained from a projection onto the x-y plane $z = 0$ ($\theta = \frac{\pi}{2}$) - i.e. the meridional plane - since the flow will be approximately parallel to the z-axis for all U^0_∞, σ and Re.) These projections of the velocity fields are given, for ease of reference, in Appendix 2.1.

3.4.3 Drag, Lift and Moment Coefficients

Our principal numerical results are the predicted (late-time) drag, lift and moment coefficients of the sphere obtained for various Reynolds numbers Re, centre-line velocities at infinity U^0_∞, and shear rates at infinity σ. As in the preceding sub-section, we will present results obtained for which:- $U^0_\infty = 1$, $\sigma \equiv 0$; $U^0_\infty \equiv 0$, $\sigma = 1$; and $U^0_\infty = 1$, $\sigma = 1$. For each of these cases, we will present results obtained for $Re = 10^{-2}$, 10^{-1}, 10^{0}, 10^{1}, and 10^{2}. There is little point in presenting results for Reynolds numbers less than about 10^{-2}, since the equations of motion are then effectively independent of Reynolds number, while at Reynolds numbers greater than about 300, the computational scheme becomes unstable, and no useful results can be obtained. (The reason for this instability is almost certainly a growth in the magnitude of the truncation errors involved in the finite-difference approximation of the equations of motion. The larger truncation errors are due to the increased importance of higher derivatives of the flow variables at higher Reynolds numbers. The instability is not, therefore, a numerical simulation of a physical flow instability leading, for example, to turbulence, though the two effects are clearly related.) The predicted drag, lift and moment coefficients are presented both in tabular and graphical form in Appendix 2.2.

Note, incidentally, that no lift or moment coefficients are presented for the case of uniform flow, and no drag or lift coefficients for the case of centred linear-shear flow; this is because, as we would expect physically, a uniform flow exerts no net lift or torque, and a centred linear-shear flow exerts no drag or lift, at least at low Reynolds numbers. We also note that results are presented for the case of uniform plus centred linear-shear flow for two positions of the outer envelope; this is for comparison purposes. (In fact, as we noted in sub-section 3.4.1, we use the inner position ($r^* = 0.875$) for prediction of the moment coefficient, and the outer position ($r^* = 1.25$) for predicition of the drag and lift coefficients, of the sphere.) Finally, we note that (again for the purposes of comparison):-

(i) an experimental drag coefficient curve(108) is plotted together with the computed drag coefficients for uniform flow past the sphere;

(ii) Stokes flow predictions(109) are plotted together with the computed drag coefficients for uniform flow (with or without centred linear-shear flow) past the sphere;

(iii) Stokes flow predictions(110) are plotted together with the computed moment coefficients for centred linear-shear flow (with or without uniform flow) past the sphere;

and

(iv) predictions from a (low Reynolds number) matched asymptotic expansions analysis[111] are plotted together with the computed lift coefficient for uniform plus centred linear-shear flow past the sphere.

Chapter 4 — Results and Conclusions

In this chapter, we first discuss the numerical results we have obtained in Chapter 3 for flow past a sphere, and then go on to discuss the feasibility of solving the three-dimensional Navier-Stokes equations for general problems using, in particular, the vorticity/potential method developed in Chapter 2.

4.1 Flow past a Sphere

4.1.1 Velocity and Pressure Fields

Inspection of the velocity fields presented in Appendix 2.1 (q.v.) indicates that the main features of the predicted flows are much as we might expect physically. All of the flows show maximum velocity gradients at the surface of the sphere (as they should), and all show a physically reasonable variation between the surface of the sphere and the outer envelope. In the case of uniform flow, there is an obvious symmetry about the z-axis, while in the case of centred linear-shear flow, there is an obvious anti-symmetry about both the x-y and y-z planes. In the case of uniform plus centred linear-shear flow, the effect of adding a uniform flow to a centred linear-shear flow is clear, the non-linearity of the addition being hardly noticeable at a Reynolds number of unity, as we would expect.

Certain features of the flows, in particular the slip that occurs on the surface of the sphere, are not correct, however. The most serious case of slip is shown by the uniform plus centred linear-shear flow when the radius of the outer envelope $r^* = 1.25$; it is rather less serious when r^* is reduced to 0.875. The reason for this is that, compensating errors apart, unless the iteration schemes involved in the numerical solution converge to a high degree, and the truncation errors involved in the finite-difference analogues of the various derivatives are very small, slip is bound to occur. Although it is relatively easy to ensure that the iteration schemes converge satisfactorily, truncation errors are only going to be small if a very fine mesh is used to define the discretised flow field (and we in fact use only fifty-six mesh-points to define the complete flow field; c.f. sub-section 3.4.1 of Chapter 3). This difficulty over slip is in fact inherent in the use of derived, rather than the primitive, variables in the solution of the Navier-Stokes equations; clearly, if we use the primitive variables, the no-slip condition can be imposed directly on the velocity field. On the other hand, the presence of these relatively large errors in the form of slip does not necessarily mean that there will be correspondingly large errors in the predicted drag, lift and moment coefficients. This is because these coefficients are obtained directly from the vorticity field by numerical integration, and integration is essentially a smoothing operation, in contrast to differentiation, where errors tend to be accentuated.

Errors do, of course, still arise in numerical integration, for a variety of reasons. We can, however, obtain an estimate of the magnitude of these errors from the line integrals used to obtain the pressure on the surface of the sphere from the vorticity. The contours used in these integrals start at the $\theta = 0$ node, and proceed along constant-ϕ paths to the $\theta = \pi$ node, so that several estimates of the pressure at this node are available (an average is in fact used). Analytically, these estimates will be equal, but numerically they are not, because of errors arising not only in the line integration, but also in the determination of the vorticity field (c.f. sub-section 3.2.4 of Chapter 3). The overall error, which we may define as the maximum percentage difference between these estimates and their average value relative to the maximum pressure on the surface of the sphere, averages ±30%, and is typically in the range

±10 to 50%. Thus we may expect that the average error in our predicted drag, lift and moment coefficients is also of the order of ±30% in general, and this is in fact what we will find in the next sub-section.

4.1.2 Drag, Lift and Moment Coefficients

Inspection of the predicted late-time drag, lift and moment coefficients presented in Appendix 2.2 (q.v.) indicates good qualitative agreement with experimental and theoretical results in general. In the case of uniform flow, both experimental and theoretical drag coefficients are available for comparison purposes. The Stokes (creeping) flow result (c.f. sub-section 3.4.3 of Chapter 3):-

$$C_D = 24/Re \qquad (4.1)$$

and experimental results agree for Reynolds numbers less than about 10^{-1}. At higher Reynolds numbers, inertial effects become relatively more important, and viscous effects become important only in a (thin) boundary layer near the sphere, so that the pressure (form) component of the drag coefficient increases in size relative to the viscous (friction) component. Assuming that the boundary layer remains laminar, the net effect is for the total drag coefficient to increase relative to the Stokes flow result (4.1) as the Reynolds number increases. For the limited range of Reynolds numbers investigated numerically, this is precisely how the predicted drag coefficient behaves (c.f. Graph 1 of Appendix 2.2). At low Reynolds numbers, the predicted drag coefficient varies as the reciprocal of the Reynolds number, being within 45% of both experimental and theoretical results. Then, at a Reynolds number of about 10^1, the drag coefficient shows an increase relative to reciprocal Reynolds number variation, with a corresponding increase in the importance of the pressure component with respect to the viscous component. The drag coefficient does not show as strong an increase as happens in practice, however, being about 35% low at a Reynolds number of 10^2. This is probably because of numerical errors in the predicted pressure component of the drag coefficient, which is relatively too small, even at low Reynolds numbers (where it comprises only 25% of the total predicted drag coefficient, and not 33%, as it ought to according to the Stokes flow result).

In the case of centred linear-shear flow, the Stokes flow result (c.f. sub-section 3.4.3 of Chapter 3):-

$$C_M = 4/Re \tag{4.2}$$

can be compared with our predicted moment coefficient (c.f. Graph 2 of Appendix 2.2). For Reynolds numbers less than about 10^1, the predicted moment coefficient does in fact vary as the reciprocal of the Reynolds number, though it is about 30% lower than the Stokes flow result (4.2). For higher Reynolds numbers, however, the moment coefficient shows a decrease relative to reciprooal Reynolds number variation. The reason for this is probably that viscous effects, which alone determine the moment coefficient (recall that it is independent of pressure; c.f. sub-section 3.2.4 of Chapter 3) are less important at higher Reynolds numbers.

In the case of uniform plus centred linear-shear flow, the drag and moment coefficients (c.f. Graphs 3 and 4 of Appendix 2.2) show the same trends, even quantitatively, as the drag coefficient for uniform flow and the moment coefficient for centred linear-shear flow, respectively (c.f. Graphs 1 and 2 of Appendix 2.2). (Note that a direct comparison is possible because the two alternative definitions of Reynolds number given in sub-section 3.2.2 of Chapter 3 are equivalent for this particular case of uniform plus centred linear-shear flow.) Although this agreement is hardly surprising at low Reynolds numbers, because the equations of motion are then effectively linear, it is a little more surprising at higher Reynolds numbers, where we might expect the non-linearity in the equations of motion to be more evident. Since predicted departures from linearity only seem to occur at Reynolds numbers greater than 10^1, however, as we noted when comparing the predicted drag and moment coefficients with Stokes flow results in the two preceding paragraphs, it may well be that non-linearity will become more evident at rather higher Reynolds numbers than those investigated.

The predicted lift coefficient for uniform plus centred linear-shear flow (c.f. Graph 5 of Appendix 2.2) varies as the reciprocal of the Reynolds number over the whole range of Reynolds numbers investigated (and between 90 and 95% of its magnitude is due to the pressure component, so that the viscous component is more or less negligible, certainly at low Reynolds numbers, and probably at high Reynolds numbers too, where pressure effects tend to be more important than viscous effects anyway). Now the Stokes flow assumption in fact implies zero lift, because there is no lift in either uniform or centred linear-shear flow, so there cannot be any when the two are combined, because the Stokes flow equations are linear. The presence of lift is, therefore, essentially a non-linear effect resulting from inertial forces (or, possibly, from non-Newtonian viscous forces). Incorporating the inertial effects in an Oseen-like outer expansion, and with a Stokes-like inner expansion, a matched asymptotic expansions analysis (c.f. sub-section 3.4.3 of Chapter 3) gives:-

$$C_L = 4.1124/\sqrt{Re} \qquad (4.3)$$

for the centre-line velocity and magnitude of shear at infinity used in our calculations. In the absence of experimental data, it is difficult to decide whether the predicted reciprocal Reynolds number variation or the theoretical reciprocal square-root of the Reynolds number variation (if either) is correct. One might argue that, at low Reynolds numbers, flow properties such as pressure and vorticity should be independent of, and hence that the lift coefficient should vary as the reciprocal of, the Reynolds number (c.f. equations (3.33) - (3.35) of Chapter 3). On the other hand, because lift is essentially a non-linear effect, one might argue that this will be reflected in the behaviour of the lift coefficient with respect to the Reynolds number. Experimental results are clearly of crucial importance in deciding the matter.

We conclude this sub-section by noting that (with the obvious exception of the lift coefficient for uniform plus centred linear-shear flow) the predicted coefficients agree with experimental and theoretical results to within a maximum of ±45% and an average of ±30%. As we would expect, this agrees with the average error of ±30% predicted in sub-section 4.1.1 above, and shows that the errors involved in the integration for pressure give a good guide to overall errors.

4.2 Solution of the Three-Dimensional Navier-Stokes Equations

We begin this section by presenting a brief summary of our principal theoretical and numerical results. We continue by discussing the main advantages and disadvantages of the vorticity/potential method, and finish by stating the overall conclusions which we are able to draw from our results.

From the various methods currently available for solving the three-dimensional Navier-Stokes equations, we were able in Chapter 1 to distinguish two, the vorticity/potential and projection methods, which are suitable for general flow problems. Examination of the common theoretical basis of these two methods, and demonstration of how each can be applied to a general flow problem, then enabled us in Chapter 2 to conclude that the vorticity/potential method is likely to be superior because of the complete lack of ambiguity with which it can be applied. We found in Chapter 3 that application of the vorticity/potential method to a test problem is very straightforward, and that it gives quite reasonable results, showing fair agreement (by and large) with experimental and theoretical results. The immediate conclusion which we can draw about the vorticity/potential method is, therefore, that it clearly works, even if it involves a relatively large computational effort. The method does, however, suffer from two practical drawbacks.

The first drawback results from the use of derived, rather than the primitive, variables. As we saw in sub-section 4.1.1, this can lead to various problems, such as slip on solid surfaces. In a sense, however, this is only an apparent drawback, because most flow properties, and certainly those of primary interest, are obtained directly from the vorticity field, which is in fact determined more accurately than the velocity field (as comparison of predicted vorticity and velocity fields with theoretical results shows). In any case, the drawback can effectively be eliminated by the use of a sufficiently fine mesh to define the discretised flow field, so that the effects of the drawback can be made as small as we please, computer resources permitting.

The second drawback with the vorticity/potential method results from the complexity of the equations of motion for the derived variables. Compared with the equations of motion for the primitive variables, those for the derived variables involve many more terms, especially if a non-Cartesian coordinate system is used. This large number of terms is important, because the computer storage required to deal with them is really the only factor that limits the size of flow problem that can be handled. Thus, until computers with much larger central (i.e. fast-access) memories become available, the range of three-dimensional problems that can be solved efficiently is rather restricted. (Extensive program manipulation can increase the maximum size of problem that can be handled, but this is very inefficient from the point of view of computer time.)

On the other hand, against these drawbacks, we note that for a relatively simple three-dimensional problem, the results presented in section 4.1 indicate that reasonably accurate predictions of overall properties (such as drag coefficients) can be made, even using a relatively coarse mesh to define the discretised flow field. We may, therefore, conclude that the vorticity/potential method is suitable for obtaining solutions of the three-dimensional Navier-Stokes equations numerically. Of course, in the absence of direct comparisons (and perhaps even with them), we clearly cannot assert that the vorticity/potential method is in any sense the "best" method for solving these equations. But we can be certain that the method will work for flows of incompressible Newtonian fluids in arbitrary flow fields. If, moreover, the flow field has a relatively simple geometry (i.e. one that is readily defined in an orthogonal, preferably a Cartesian, coordinate system), and the flow is dynamically simple (for example, it is not turbulent), then we can solve the three-dimensional Navier-Stokes equations numerically, using the vorticity/potential method, and obtain reasonably accurate results with quite modest computer resources. We can, therefore, confidently recommend its use.

Appendix 1 - Analytical Equations of Motion

In this appendix, we give the equations of motion, and associated boundary conditions, in dimensionless vorticity/potential form for uniform and linear-shear flow past a sphere, in the notation of Chapter 3. We note that certain symmetry properties of the flow have been used to simplify the equations, in particular for terms involving the scalar potential on the polar axis (c.f. sub-section 3.2.2 of Chapter 3). We also note that, on the polar axis, the θ and ϕ axes of the spherical polar coordinate system (r, θ, ϕ) defining the flow field refer by convention to the x and y axes, respectively, of the Cartesian coordinate system (x, y, z) also defining the flow field (again, c.f. sub-section 3.2.2 of Chapter 3).

(i) The scalar potential Laplace equation is:-

$$\frac{\partial^2 h}{\partial r^2} + \frac{2}{r}\frac{\partial h}{\partial r} + \frac{1}{r^2}\frac{\partial^2 h}{\partial \theta^2} + \frac{\cot\theta}{r^2}\frac{\partial h}{\partial \theta} + \frac{1}{r^2 \sin^2\theta}\frac{\partial^2 h}{\partial \phi^2} = 0$$

while, on the polar axis, we have:-

$$\frac{\partial^2 h}{\partial r^2} + \frac{2}{r}\frac{\partial h}{\partial r} + \frac{2}{r^2}\frac{\partial^2 h}{\partial \theta^2} = 0$$

The boundary conditions are:-

$$\frac{\partial h}{\partial r} \equiv 0 \qquad \text{at } r = r_o$$

$$\frac{\partial h}{\partial r} = (U_\infty^o + \sigma r^* \sin\theta \cos\phi)\cos\theta \qquad \text{at } r = r^*$$

(ii) The vector potential Poisson equation is:-

$$\omega_r = \frac{\cot\theta}{r}\frac{\partial A_\theta}{\partial r} + \frac{1}{r}\frac{\partial^2 A_\theta}{\partial r \partial \theta} + \frac{1}{r^2}\frac{\partial A_\theta}{\partial \theta} + \frac{\cot\theta}{r^2} A_\theta - \frac{1}{r^2}\frac{\partial^2 A_r}{\partial \theta^2}$$

$$- \frac{\cot\theta}{r^2}\frac{\partial A_r}{\partial \theta} - \frac{1}{r^2 \sin^2\theta}\frac{\partial^2 A_r}{\partial \phi^2} + \frac{1}{r \sin\theta}\frac{\partial^2 A_\phi}{\partial r \partial \phi} + \frac{1}{r^2 \sin\theta}\frac{\partial A_\phi}{\partial \phi}$$

$$w_\theta = \frac{1}{r^2\sin\theta}\frac{\partial^2 A_\phi}{\partial\theta\partial\phi} + \frac{\cot\theta}{r^2\sin\theta}\frac{\partial A_\phi}{\partial\phi} - \frac{1}{r^2\sin^2\theta}\frac{\partial^2 A_\theta}{\partial\phi^2} - \frac{2}{r}\frac{\partial A_\theta}{\partial r}$$

$$- \frac{\partial^2 A_\theta}{\partial r^2} + \frac{1}{r}\frac{\partial^2 A_r}{\partial r\partial\theta}$$

$$w_\phi = \frac{1}{r\sin\theta}\frac{\partial^2 A_r}{\partial r\partial\phi} - \frac{2}{r}\frac{\partial A_\phi}{\partial r} - \frac{\partial^2 A_\phi}{\partial r^2} - \frac{1}{r^2}\frac{\partial^2 A_\phi}{\partial\theta^2} - \frac{\cot\theta}{r^2}\frac{\partial A_\phi}{\partial\theta}$$

$$+ \frac{1}{r^2\sin^2\theta}A_\phi + \frac{1}{r^2\sin\theta}\frac{\partial^2 A_\theta}{\partial\theta\partial\phi} - \frac{\cot\theta}{r^2\sin\theta}\frac{\partial A_\theta}{\partial\phi}$$

while, on the polar axis, we have:-

$$w_r = \frac{2}{r}\frac{\partial^2 A_\theta}{\partial r\partial\theta} + \frac{2}{r^2}\frac{\partial A_\theta}{\partial\theta} - \frac{2}{r^2}\frac{\partial^2 A_r}{\partial\theta^2} - \frac{1}{2r^2}\frac{\partial^4 A_r}{\partial\theta^2\partial\phi^2}$$

$$\pm \frac{1}{r}\frac{\partial^3 A_\phi}{\partial r\partial\theta\partial\phi} \pm \frac{1}{r^2}\frac{\partial^2 A_\phi}{\partial\theta\partial\phi}$$

$$w_\theta = \pm\frac{3}{2r^2}\frac{\partial^3 A_\phi}{\partial\theta^2\partial\phi} - \frac{1}{2r^2}\frac{\partial^4 A_\theta}{\partial\theta^2\partial\phi^2} - \frac{2}{r}\frac{\partial A_\theta}{\partial r} - \frac{\partial^2 A_\theta}{\partial r^2} + \frac{1}{r}\frac{\partial^2 A_r}{\partial r\partial\theta}$$

$$w_\phi = \pm\frac{1}{r}\frac{\partial^3 A_r}{\partial r\partial\theta\partial\phi} - \frac{2}{r}\frac{\partial A_\phi}{\partial r} - \frac{\partial^2 A_\phi}{\partial r^2} - \frac{3}{2r^2}\frac{\partial^2 A_\phi}{\partial\theta^2} \pm \frac{1}{2r^2}\frac{\partial^3 A_\theta}{\partial\theta^2\partial\phi}$$

where, when two signs are indicated, we use the upper sign when $\theta = 0$, and the lower sign when $\theta = \pi$.

The boundary conditions are:-

$$\left.\begin{aligned} A_\theta \equiv 0 \equiv A_\phi \\ \frac{\partial A_r}{\partial r} = -2\frac{A_r}{r} \end{aligned}\right\} \text{ at } r = r_o,\ r^*$$

(iii) The vorticity transport equation is:-

$$\frac{\partial w_r}{\partial t} + \left(\frac{1}{r}\frac{\partial A_\phi}{\partial\theta} + \frac{\cot\theta}{r}A_\phi - \frac{1}{r\sin\theta}\frac{\partial A_\theta}{\partial\phi} + V_r\right)\frac{\partial w_r}{\partial r}$$

$$+ \left(\frac{1}{r^2\sin\theta}\frac{\partial A_r}{\partial\phi} - \frac{1}{r}\frac{\partial A_\phi}{\partial r} - \frac{A_\phi}{r^2} + \frac{1}{r}V_\theta\right)\frac{\partial w_r}{\partial\theta}$$

$$+ \left(\frac{1}{r\sin\theta}\frac{\partial A_\theta}{\partial r} + \frac{A_\theta}{r^2\sin\theta} - \frac{1}{r^2\sin\theta}\frac{\partial A_r}{\partial\theta} + \frac{1}{r\sin\theta}V_\phi\right)\frac{\partial w_r}{\partial\phi}$$

→

$$+ \left(\frac{1}{r^2} \frac{\partial A_\phi}{\partial \theta} + \frac{\cot\theta}{r^2} A_\phi - \frac{1}{r^2 \sin\theta} \frac{\partial A_\theta}{\partial \phi} + \frac{2}{r} V_r - \frac{\cot\theta}{r} \frac{\partial A_\phi}{\partial r} \right.$$

$$+ \frac{\cot\theta}{r} V_\theta - \frac{1}{r} \frac{\partial^2 A_\phi}{\partial r \partial \theta} + \frac{1}{r^2} \frac{\partial^2 h}{\partial \theta^2} + \frac{1}{r \sin\theta} \frac{\partial^2 A_\theta}{\partial r \partial \phi}$$

$$\left. + \frac{1}{r^2 \sin^2\theta} \frac{\partial^2 h}{\partial \phi^2} \right) w_r$$

$$- \left(\frac{1}{r^2 \sin\theta} \frac{\partial^2 A_\phi}{\partial \theta \partial \phi} + \frac{\cot\theta}{r^2 \sin\theta} \frac{\partial A_\phi}{\partial \phi} - \frac{1}{r^2 \sin^2\theta} \frac{\partial^2 A_\theta}{\partial \phi^2} + \frac{1}{r \sin\theta} \frac{\partial^2 h}{\partial r \partial \phi} \right) w_\phi$$

$$- \left(\frac{1}{r^2} \frac{\partial^2 A_\phi}{\partial \theta^2} - \frac{\mathrm{cosec}^2\theta}{r^2} A_\phi + \frac{\cot\theta}{r^2} \frac{\partial A_\phi}{\partial \theta} + \frac{\mathrm{cosec}\,\theta \cot\theta}{r^2} \frac{\partial A_\theta}{\partial \phi} \right.$$

$$\left. - \frac{1}{r^2 \sin\theta} \frac{\partial^2 A_\theta}{\partial \theta \partial \phi} + \frac{1}{r} \frac{\partial^2 h}{\partial r \partial \theta} \right) w_\theta$$

$$= \frac{-1}{Re} \left(\frac{\cot\theta}{r} \frac{\partial w_\theta}{\partial r} + \frac{1}{r} \frac{\partial^2 w_\theta}{\partial r \partial \theta} + \frac{1}{r^2} \frac{\partial w_\theta}{\partial \theta} + \frac{\cot\theta}{r^2} w_\theta - \frac{1}{r^2} \frac{\partial^2 w_r}{\partial \theta^2} \right.$$

$$\left. - \frac{\cot\theta}{r^2} \frac{\partial w_r}{\partial \theta} - \frac{1}{r^2 \sin^2\theta} \frac{\partial^2 w_r}{\partial \phi^2} + \frac{1}{r \sin\theta} \frac{\partial^2 w_\phi}{\partial r \partial \phi} + \frac{1}{r^2 \sin\theta} \frac{\partial w_\phi}{\partial \phi} \right)$$

$$\frac{\partial w_\theta}{\partial t} + \left(\frac{1}{r} \frac{\partial A_\phi}{\partial \theta} + \frac{\cot\theta}{r} A_\phi - \frac{1}{r \sin\theta} \frac{\partial A_\theta}{\partial \phi} + V_r \right) \frac{\partial w_\theta}{\partial r}$$

$$+ \left(\frac{1}{r^2 \sin\theta} \frac{\partial A_r}{\partial \phi} - \frac{1}{r} \frac{\partial A_\phi}{\partial r} - \frac{A_\phi}{r^2} + \frac{1}{r} V_\theta \right) \frac{\partial w_\theta}{\partial \theta}$$

$$+ \left(\frac{1}{r \sin\theta} \frac{\partial A_\theta}{\partial r} + \frac{1}{r^2 \sin\theta} A_\theta - \frac{1}{r^2 \sin\theta} \frac{\partial A_r}{\partial \theta} + \frac{1}{r \sin\theta} V_\phi \right) \frac{\partial w_\theta}{\partial \phi}$$

$$+ \left(\frac{1}{r} V_r + \frac{\cot\theta}{r^2 \sin\theta} \frac{\partial A_r}{\partial \phi} + \frac{\cot\theta}{r} V_\theta + \frac{1}{r} \frac{\partial^2 A_\phi}{\partial r \partial \phi} - \frac{\cot\theta}{r^2} A_\phi + \frac{\partial^2 h}{\partial r^2} \right.$$

$$\left. + \frac{1}{r^2 \sin\theta} \frac{\partial A_\theta}{\partial \phi} - \frac{1}{r^2 \sin\theta} \frac{\partial^2 A_r}{\partial \theta \partial \phi} + \frac{1}{r^2 \sin^2\theta} \frac{\partial^2 h}{\partial \phi^2} \right) w_\theta$$

$$+ \left(\frac{2}{r^2 \sin\theta} \frac{\partial A_r}{\partial \phi} - \frac{2}{r^2} A_\phi + \frac{1}{r} V_\theta - \frac{1}{r \sin\theta} \frac{\partial^2 A_r}{\partial r \partial \phi} + \frac{\partial^2 A_\phi}{\partial r^2} \right.$$

$$\left. - \frac{1}{r} \frac{\partial^2 h}{\partial r \partial \theta} + \frac{1}{r^2} \frac{\partial h}{\partial \theta} \right) w_r$$

⟶

$$- \left(\frac{1}{r^2 \sin^2\theta} \frac{\partial^2 A_r}{\partial \phi^2} - \frac{1}{r \sin\theta} \frac{\partial^2 A_\phi}{\partial r \partial \phi} - \frac{1}{r^2 \sin\theta} \frac{\partial A_\phi}{\partial \phi} + \frac{1}{r^2 \sin\theta} \frac{\partial^2 h}{\partial \theta \partial \phi} \right) w_\phi$$

$$= \frac{-1}{Re} \left(\frac{1}{r^2 \sin\theta} \frac{\partial^2 w_\phi}{\partial \theta \partial \phi} + \frac{\cot\theta}{r^2 \sin\theta} \frac{\partial w_\phi}{\partial \phi} - \frac{1}{r^2 \sin^2\theta} \frac{\partial^2 w_\theta}{\partial \phi^2} - \frac{2}{r} \frac{\partial w_\theta}{\partial r} \right.$$

$$\left. - \frac{\partial^2 w_\theta}{\partial r^2} + \frac{1}{r} \frac{\partial^2 w_r}{\partial r \partial \theta} \right)$$

$$\frac{\partial w_\phi}{\partial t} + \left(\frac{1}{r} \frac{\partial A_\phi}{\partial \theta} + \frac{\cot\theta}{r} A_\phi - \frac{1}{r \sin\theta} \frac{\partial A_\theta}{\partial \phi} + V_r \right) \frac{\partial w_\phi}{\partial r}$$

$$+ \left(\frac{1}{r^2 \sin\theta} \frac{\partial A_r}{\partial \phi} - \frac{1}{r} \frac{\partial A_\phi}{\partial r} - \frac{A_\phi}{r^2} + \frac{1}{r} V_\theta \right) \frac{\partial w_\phi}{\partial \theta}$$

$$+ \left(\frac{1}{r \sin\theta} \frac{\partial A_\theta}{\partial r} + \frac{1}{r^2 \sin\theta} A_\theta - \frac{1}{r^2 \sin\theta} \frac{\partial A_r}{\partial \theta} + \frac{1}{r \sin\theta} V_\phi \right) \frac{\partial w_\phi}{\partial \phi}$$

$$+ \left(\frac{1}{r} V_r + \frac{\cot\theta}{r} \frac{\partial A_\phi}{\partial r} - \frac{1}{r \sin\theta} \frac{\partial^2 A_\theta}{\partial r \partial \phi} + \frac{\partial^2 h}{\partial r^2} - \frac{\operatorname{cosec}\theta \cot\theta}{r^2} \frac{\partial A_r}{\partial \phi} \right.$$

$$\left. + \frac{1}{r^2 \sin\theta} \frac{\partial^2 A_r}{\partial \theta \partial \phi} - \frac{1}{r^2} \frac{\partial A_\phi}{\partial \theta} + \frac{1}{r^2} \frac{\partial^2 h}{\partial \theta^2} \right) w_\phi$$

$$+ \left(\frac{2}{r^2} A_\theta - \frac{2}{r^2} \frac{\partial A_r}{\partial \theta} + \frac{1}{r} V_\phi - \frac{\partial^2 A_\theta}{\partial r^2} + \frac{1}{r} \frac{\partial^2 A_r}{\partial r \partial \theta} + \frac{1}{r^2 \sin\theta} \frac{\partial h}{\partial \phi} \right.$$

$$\left. - \frac{1}{r \sin\theta} \frac{\partial^2 h}{\partial r \partial \phi} \right) w_r$$

$$+ \left(\frac{\cot\theta}{r} \frac{\partial A_\theta}{\partial r} + \frac{\cot\theta}{r^2} A_\theta - \frac{\cot\theta}{r^2} \frac{\partial A_r}{\partial \theta} + \frac{\cot\theta}{r} V_\phi - \frac{1}{r} \frac{\partial^2 A_\theta}{\partial r \partial \theta} \right.$$

$$\left. - \frac{1}{r^2} \frac{\partial A_\theta}{\partial \theta} + \frac{1}{r^2} \frac{\partial^2 A_r}{\partial \theta^2} - \frac{1}{r^2 \sin\theta} \frac{\partial^2 h}{\partial \theta \partial \phi} + \frac{1}{r^2 \sin\theta \tan\theta} \frac{\partial h}{\partial \phi} \right) w_\theta$$

$$= \frac{-1}{Re} \left(\frac{1}{r \sin\theta} \frac{\partial^2 w_r}{\partial r \partial \phi} - \frac{2}{r} \frac{\partial w_\phi}{\partial r} - \frac{\partial^2 w_\phi}{\partial r^2} - \frac{1}{r^2} \frac{\partial^2 w_\phi}{\partial \theta^2} - \frac{\cot\theta}{r^2} \frac{\partial w_\phi}{\partial \theta} \right.$$

$$\left. + \frac{1}{r^2 \sin^2\theta} w_\phi + \frac{1}{r^2 \sin\theta} \frac{\partial^2 w_\theta}{\partial \theta \partial \phi} - \frac{\cot\theta}{r^2 \sin\theta} \frac{\partial w_\theta}{\partial \phi} \right)$$

while, on the polar axis, we have:-

$$\frac{\partial w_r}{\partial t} + \left(\frac{2}{r}\frac{\partial A_\phi}{\partial \theta} \mp \frac{1}{r}\frac{\partial^2 A_\theta}{\partial\theta\partial\phi} + V_r\right)\frac{\partial w_r}{\partial r}$$

$$+ \left(\pm \frac{1}{r^2}\frac{\partial^2 A_r}{\partial\theta\partial\phi} - \frac{2}{r}\frac{\partial A_\phi}{\partial r} + \frac{2}{r}V_\theta \mp \frac{1}{r^2}\frac{\partial A_\theta}{\partial\phi} \pm \frac{1}{r}\frac{\partial^2 A_\theta}{\partial r\partial\phi}\right)\frac{\partial w_r}{\partial\theta}$$

$$+ \left(\pm \frac{1}{r}\frac{\partial^2 A_\theta}{\partial r\partial\theta} \pm \frac{1}{r^2}\frac{\partial A_\theta}{\partial\theta} \mp \frac{1}{r^2}\frac{\partial^2 A_r}{\partial\theta^2}\right)\frac{\partial w_r}{\partial\phi}$$

$$+ \left(\frac{2}{r^2}\frac{\partial A_\phi}{\partial\theta} \mp \frac{1}{r^2}\frac{\partial^2 A_\theta}{\partial\theta\partial\phi} + \frac{2}{r}V_r - \frac{2}{r}\frac{\partial^2 A_\phi}{\partial r\partial\theta} + \frac{2}{r^2}\frac{\partial^2 h}{\partial\theta^2}\right.$$

$$\left.\pm \frac{1}{r}\frac{\partial^3 A_\theta}{\partial r\partial\theta\partial\phi}\right) w_r$$

$$- \left(\pm \frac{3}{2r^2}\frac{\partial^3 A_\phi}{\partial\theta^2\partial\phi} - \frac{1}{2r^2}\frac{\partial^4 A_\theta}{\partial\theta^2\partial\phi^2}\right) w_\phi$$

$$- \left(\frac{3}{2r^2}\frac{\partial^2 A_\phi}{\partial\theta^2} \mp \frac{1}{2r^2}\frac{\partial^3 A_\theta}{\partial\theta^2\partial\phi} + \frac{1}{r}\frac{\partial^2 h}{\partial r\partial\theta}\right) w_\theta$$

$$+ \left(\frac{A_\phi}{r} \mp \frac{1}{r}\frac{\partial A_\theta}{\partial\phi}\right)\frac{\partial^2 w_r}{\partial r\partial\theta} + \left(\pm \frac{1}{r^2}\frac{\partial A_r}{\partial\phi}\right)\frac{\partial^2 w_r}{\partial\theta^2}$$

$$+ \left(\pm \frac{1}{r}\frac{\partial A_\theta}{\partial r} \pm \frac{A_\theta}{r^2} \mp \frac{1}{r^2}\frac{\partial A_r}{\partial\theta}\right)\frac{\partial^2 w_r}{\partial\theta\partial\phi} - \left(\pm \frac{1}{r^2}\frac{\partial^2 A_\phi}{\partial\theta\partial\phi}\right)\frac{\partial w_\phi}{\partial\theta}$$

$$- \left(\pm \frac{1}{r^2}\frac{\partial A_\phi}{\partial\phi} - \frac{1}{r^2}\frac{\partial^2 A_\theta}{\partial\phi^2}\right)\frac{1}{2}\frac{\partial^2 w_\phi}{\partial\theta^2}$$

$$- \left(\frac{1}{r^2}\frac{\partial A_\phi}{\partial\theta} \mp \frac{1}{r^2}\frac{\partial^2 A_\theta}{\partial\theta\partial\phi}\right)\frac{\partial w_\theta}{\partial\theta} - \left(- \frac{A_\phi}{r^2} \pm \frac{1}{r^2}\frac{\partial A_\theta}{\partial\phi}\right)\frac{1}{2}\frac{\partial^2 w_\theta}{\partial\theta^2}$$

$$= \frac{-1}{Re}\left(\frac{2}{r}\frac{\partial^2 w_\theta}{\partial r\partial\theta} + \frac{2}{r^2}\frac{\partial w_\theta}{\partial\theta} - \frac{2}{r^2}\frac{\partial^2 w_r}{\partial\theta^2} - \frac{1}{2r^2}\frac{\partial^4 w_r}{\partial\theta^2\partial\phi^2} \pm \frac{1}{r}\frac{\partial^3 w_\phi}{\partial r\partial\theta\partial\phi}\right.$$

$$\left.\pm \frac{1}{r^2}\frac{\partial^2 w_\phi}{\partial\theta\partial\phi}\right)$$

$$\frac{\partial w_\theta}{\partial t} + \left(\frac{2}{r}\frac{\partial A_\phi}{\partial\theta} \mp \frac{1}{r}\frac{\partial^2 A_\theta}{\partial\theta\partial\phi} + V_r\right)\frac{\partial w_\theta}{\partial r}$$

$$+ \left(- \frac{1}{r}\frac{\partial A_\phi}{\partial r} - \frac{2}{r^2}A_\phi + \frac{2}{r}V_\theta \pm \frac{1}{r^2}\frac{\partial A_\theta}{\partial\phi}\right)\frac{\partial w_\theta}{\partial\theta}$$

→

$$+\left(\pm\frac{1}{r}\frac{\partial^2 A_\theta}{\partial r\partial\theta}\pm\frac{1}{r^2}\frac{\partial A_\theta}{\partial\theta}\mp\frac{1}{r^2}\frac{\partial^2 A_r}{\partial\theta^2}\right)\frac{\partial w_\theta}{\partial\phi}$$

$$+\left(\frac{1}{r}V_r\mp\frac{1}{2r^2}\frac{\partial^3 A_r}{\partial\theta^2\partial\phi}+\frac{1}{r^2}\frac{\partial^2 h}{\partial\theta^2}+\frac{1}{r}\frac{\partial^2 A_\phi}{\partial r\partial\theta}-\frac{1}{r^2}\frac{\partial A_\phi}{\partial\theta}\right.$$

$$\left.+\frac{\partial^2 h}{\partial r^2}\pm\frac{1}{r^2}\frac{\partial^2 A_\theta}{\partial\theta\partial\phi}\right)w_\theta$$

$$+\left(\pm\frac{2}{r^2}\frac{\partial^2 A_r}{\partial\theta\partial\phi}-\frac{2}{r^2}A_\phi+\frac{1}{r}V_\theta\mp\frac{1}{r}\frac{\partial^3 A_r}{\partial r\partial\theta\partial\phi}+\frac{\partial^2 A_\phi}{\partial r^2}\right.$$

$$\left.-\frac{1}{r}\frac{\partial^2 h}{\partial r\partial\theta}+\frac{1}{r^2}\frac{\partial h}{\partial\theta}\right)w_r$$

$$-\left(\frac{1}{2r^2}\frac{\partial^4 A_r}{\partial\theta^2\partial\phi^2}\mp\frac{1}{r}\frac{\partial^3 A_\phi}{\partial r\partial\theta\partial\phi}\mp\frac{1}{r^2}\frac{\partial^2 A_\phi}{\partial\theta\partial\phi}\right)w_\phi$$

$$+\left(\frac{A_\phi}{r}\mp\frac{1}{r}\frac{\partial A_\theta}{\partial\phi}\right)\frac{\partial^2 w_\theta}{\partial r\partial\theta}+\left(\pm\frac{3}{2r^2}\frac{\partial A_r}{\partial\phi}\right)\frac{\partial^2 w_\theta}{\partial\theta^2}$$

$$+\left(\pm\frac{1}{r}\frac{\partial A_\theta}{\partial r}\pm\frac{A_\theta}{r^2}\mp\frac{1}{r^2}\frac{\partial A_r}{\partial\theta}\right)\frac{\partial^2 w_\theta}{\partial\theta\partial\phi}-\left(\frac{1}{2r^2}\frac{\partial^2 A_r}{\partial\phi^2}\right)\frac{\partial^2 w_\phi}{\partial\theta^2}$$

$$+\left(\pm\frac{2}{r^2}\frac{\partial A_r}{\partial\phi}\mp\frac{1}{r}\frac{\partial^2 A_r}{\partial r\partial\phi}\right)\frac{\partial w_r}{\partial\theta}-\left(\mp\frac{1}{r}\frac{\partial^2 A_\phi}{\partial r\partial\phi}\mp\frac{1}{r^2}\frac{\partial A_\phi}{\partial\phi}\right)\frac{\partial w_\phi}{\partial\theta}$$

$$=\frac{-1}{Re}\left(\pm\frac{3}{2r^2}\frac{\partial^3 w_\phi}{\partial\theta^2\partial\phi}-\frac{1}{2r^2}\frac{\partial^4 w_\theta}{\partial\theta^2\partial\phi^2}-\frac{2}{r}\frac{\partial w_\theta}{\partial r}-\frac{\partial^2 w_\theta}{\partial r^2}+\frac{1}{r}\frac{\partial^2 w_r}{\partial r\partial\theta}\right)$$

$$\frac{\partial w_\phi}{\partial t}+\left(\frac{2}{r}\frac{\partial A_\phi}{\partial\theta}\mp\frac{1}{r}\frac{\partial^2 A_\theta}{\partial\theta\partial\phi}+V_r\right)\frac{\partial w_\phi}{\partial r}$$

$$+\left(\pm\frac{2}{r^2}\frac{\partial^2 A_r}{\partial\theta\partial\phi}-\frac{A_\phi}{r^2}+\frac{1}{r}V_\theta\mp\frac{1}{r}\frac{\partial^2 A_\theta}{\partial r\partial\phi}\right)\frac{\partial w_\phi}{\partial\theta}$$

$$+\left(\pm\frac{1}{r}\frac{\partial^2 A_\theta}{\partial r\partial\theta}\pm\frac{1}{r^2}\frac{\partial A_\theta}{\partial\theta}\mp\frac{1}{r^2}\frac{\partial^2 A_r}{\partial\theta^2}\right)\frac{\partial w_\phi}{\partial\phi}$$

$$+\left(\frac{1}{r}V_r+\frac{1}{r}\frac{\partial^2 A_\phi}{\partial r\partial\theta}\mp\frac{1}{r}\frac{\partial^3 A_\theta}{\partial r\partial\theta\partial\phi}+\frac{\partial^2 h}{\partial r^2}\pm\frac{1}{2r^2}\frac{\partial^3 A_r}{\partial\theta^2\partial\phi}\right.$$

$$\left.-\frac{1}{r^2}\frac{\partial A_\phi}{\partial\theta}+\frac{1}{r^2}\frac{\partial^2 h}{\partial\theta^2}\right)w_\phi$$

$$+ \left(\frac{2}{r^2} A_\theta - \frac{2}{r^2} \frac{\partial A_r}{\partial \theta} - \frac{\partial^2 A_\theta}{\partial r^2} + \frac{1}{r} \frac{\partial^2 h}{\partial r \partial \theta} \right) w_r$$

$$+ \left(\frac{A_\phi}{r} \mp \frac{1}{r} \frac{\partial A_\theta}{\partial \phi} \right) \frac{\partial^2 w_\phi}{\partial r \partial \theta} + \left(\pm \frac{1}{2r^2} \frac{\partial A_r}{\partial \phi} \right) \frac{\partial^2 w_\phi}{\partial \theta^2}$$

$$+ \left(\pm \frac{1}{r} \frac{\partial A_\theta}{\partial r} \pm \frac{A_\theta}{r^2} \mp \frac{1}{r^2} \frac{\partial A_r}{\partial \theta} \right) \frac{\partial^2 w_\phi}{\partial \theta \, \partial \phi}$$

$$+ \left(\frac{1}{r} \frac{\partial A_\theta}{\partial r} + \frac{A_\theta}{r^2} - \frac{1}{r^2} \frac{\partial A_r}{\partial \theta} \right) \frac{\partial w_\theta}{\partial \theta}$$

$$= \frac{-1}{Re} \left(\pm \frac{1}{r} \frac{\partial^3 w_r}{\partial r \partial \theta \partial \phi} - \frac{2}{r} \frac{\partial w_\phi}{\partial r} - \frac{\partial^2 w_\phi}{\partial r^2} - \frac{3}{2r^2} \frac{\partial^2 w_\phi}{\partial \theta^2} \pm \frac{1}{2r^2} \frac{\partial^3 w_\theta}{\partial \theta^2 \partial \phi} \right)$$

The boundary conditions are:-

$$\left. \begin{aligned} w_r &\equiv 0 \\ w_\theta &= -\frac{\partial v_\phi}{\partial r} + \frac{v_\phi}{r} \\ w_\phi &= \frac{\partial v_\theta}{\partial r} - \frac{v_\theta}{r} \end{aligned} \right\} \text{ at } r = r_o$$

$$\left. \begin{aligned} w_r &= -\sigma \sin\theta \sin\phi \\ w_\theta &= -\sigma \cos\theta \sin\phi \\ w_\phi &= -\sigma \cos\phi \end{aligned} \right\} \text{ at } r = r^*$$

where $\underline{v} = (v_r, v_\theta, v_\phi) = \nabla_\wedge \underline{A} = \Bigg(\frac{1}{r} \frac{\partial A_\phi}{\partial \theta} + A_\phi \frac{\cot\theta}{r} - \frac{1}{r \sin\theta} \frac{\partial A_\theta}{\partial \phi},$

$$\frac{1}{r \sin\theta} \frac{\partial A_r}{\partial \phi} - \frac{\partial A_\phi}{\partial r} - \frac{A_\phi}{r},$$

$$\frac{\partial A_\theta}{\partial r} + \frac{A_\theta}{r} - \frac{1}{r} \frac{\partial A_r}{\partial \theta} \Bigg)$$

and $\underline{V} = (V_r, V_\theta, V_\phi) = \nabla h = \left(\frac{\partial h}{\partial r}, \frac{1}{r} \frac{\partial h}{\partial \theta}, \frac{1}{r \sin\theta} \frac{\partial h}{\partial \phi} \right)$, while the boundary conditions on the polar axis at $r = r^*$ are:-

$$w_r \equiv 0, \quad w_\theta \equiv 0, \quad w_\phi = -\sigma.$$

Appendix 2 - Computational Results

In this appendix, we present certain computational results obtained for uniform and linear-shear flow past a sphere. The notation and flow configuration used are those of section 3.2 of Chapter 3.

Appendix 2.1 Velocity Fields

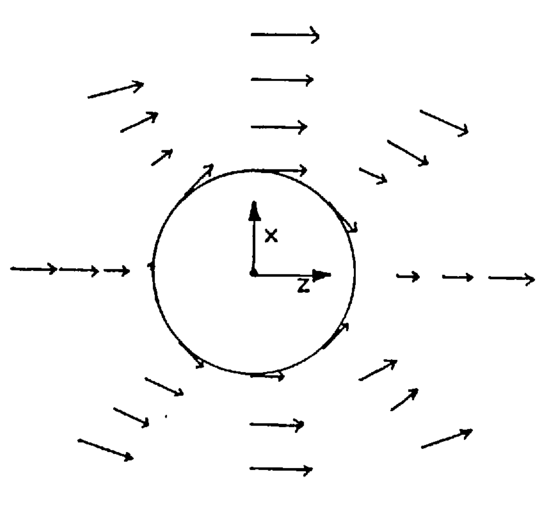
We present here typical late-time velocity fields for various combinations of the flow parameters U_∞^0 and σ, with Re = 1 throughout. The scale of the fields is as follows:-

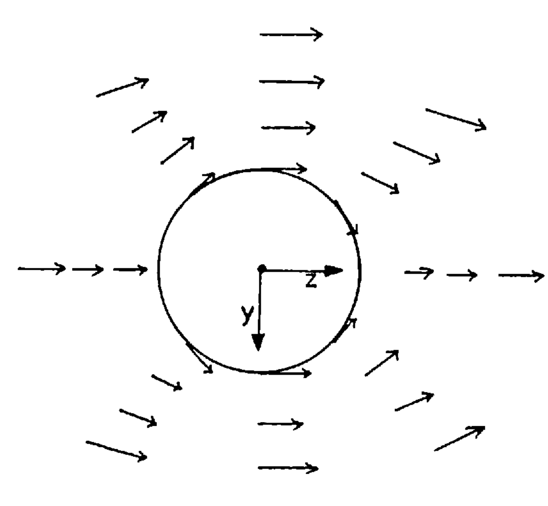
unit dimensionless length = 1.00 inches

and unit dimensionless velocity = 0.25 inches.

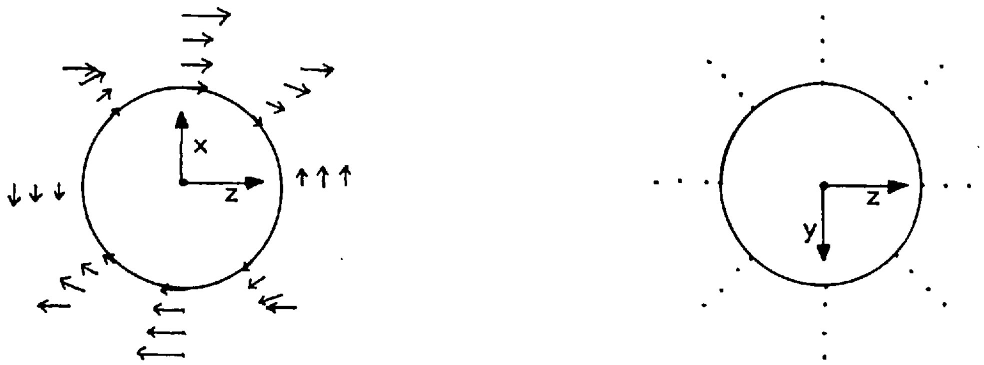
Note that two sets of fields are given for the case of uniform plus centred linear-shear flow; this is for comparison purposes.

(i) $U_\infty^0 = 1$ and $\sigma \equiv 0$ (i.e. uniform flow) with $r^* = 1.25$ (i.e. $\Delta r = 0.25$)



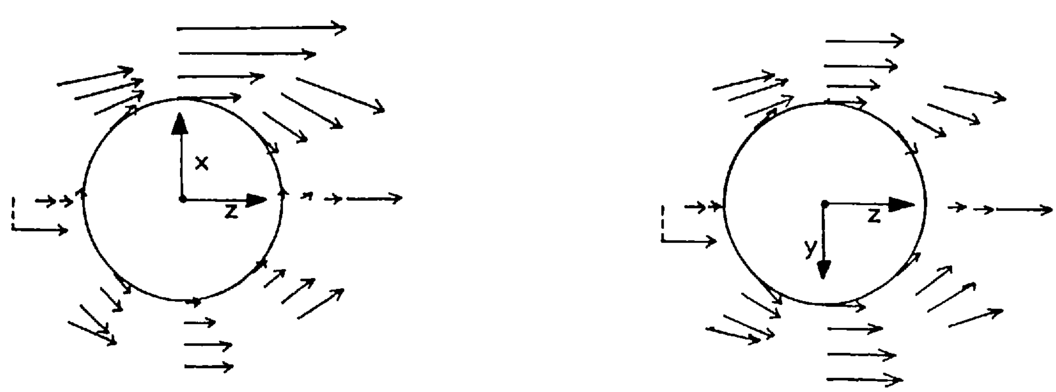


(ii) $U^0_\infty \equiv 0$ and $\sigma = 1$ (i.e. centred linear-shear flow) with $r^* = 0.875$ (i.e. $\Delta r = 0.125$)

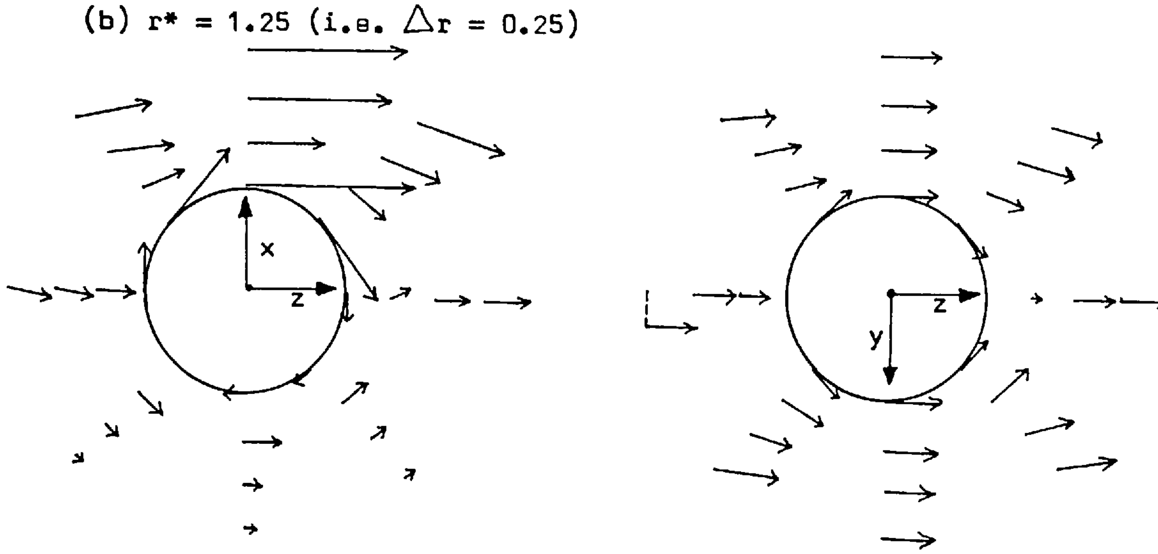


(iii) $U^0_\infty = 1$ and $\sigma = 1$ (i.e. uniform plus centred linear-shear flow) with

(a) $r^* = 0.875$ (i.e. $\Delta r = 0.125$)



(b) $r^* = 1.25$ (i.e. $\Delta r = 0.25$)



Appendix 2.2 Drag, Lift and Moment Coefficients

We present here computed late-time drag, lift and moment coefficients for various combinations of the flow parameters U^0_∞, σ and Re, together with certain experimental measurements and analytical predictions of these coefficients, for comparison purposes (c.f. sub-section 3.4.3 of Chapter 3). Note that two sets of results are given for the case of uniform plus centred linear-shear flow; again, this is for comparison purposes: results in parentheses will generally not be used (c.f. sub-section 3.4.1 of Chapter 3).

(i) $U^0_\infty = 1$ and $\sigma \equiv 0$ (i.e. uniform flow) with $r^* = 1.25$ (i.e. $\Delta r = 0.25$)

Re	C_{DP}	C_{DV}	C_D
10^{-2}	8.17×10^{2}	2.66×10^{3}	3.48×10^{3}
10^{-1}	8.17×10^{1}	2.66×10^{2}	3.48×10^{2}
10^{0}	8.77×10^{0}	2.69×10^{1}	3.56×10^{1}
10^{1}	1.26×10^{0}	2.83×10^{0}	4.08×10^{0}
10^{2}	3.39×10^{-1}	3.26×10^{-1}	6.65×10^{-1}

(ii) $U^0_\infty \equiv 0$ and $\sigma = 1$ (i.e.centred linear-shear flow) with $r^* = 0.875$ (i.e. $\Delta r = 0.125$)

Re	C_M
10^{-2}	2.76×10^{2}
10^{-1}	2.76×10^{1}
10^{0}	2.74×10^{0}
10^{1}	2.72×10^{-1}
10^{2}	1.50×10^{-2}

(iii) $U^0_\infty = 1$ and $\sigma = 1$ (i.e. uniform plus centred linear-shear flow) with

(a) $r^* = 0.875$ (i.e. $\Delta r = 0.125$)

Re	(C_{DP})	(C_{DV})	(C_D)	(C_{LP})	(C_{LV})	(C_L)	C_M
10^{-2}	4.57×10^{3}	6.54×10^{3}	1.11×10^{4}	7.47×10^{2}	8.29×10^{1}	8.30×10^{2}	3.27×10^{2}
10^{-1}	4.57×10^{2}	6.54×10^{2}	1.11×10^{3}	7.50×10^{1}	8.36×10^{0}	8.34×10^{1}	3.27×10^{1}
10^{0}	4.58×10^{1}	6.54×10^{1}	1.11×10^{2}	8.18×10^{0}	8.87×10^{-1}	9.07×10^{0}	2.92×10^{0}
10^{1}	5.03×10^{0}	6.64×10^{0}	1.17×10^{1}	8.42×10^{-1}	1.59×10^{-2}	1.00×10^{0}	2.71×10^{-1}
10^{2}	2.20×10^{0}	8.25×10^{-1}	3.03×10^{0}	-2.52×10^{-1}	-1.40×10^{-2}	-2.66×10^{-1}	1.10×10^{-2}

(b) $r^* = 1.25$ (i.e. $\Delta r = 0.25$)

Re	C_{DP}	C_{DV}	C_D	C_{LP}	C_{LV}	C_L	(C_M)
10^{-2}	8.17×10^{2}	2.66×10^{3}	3.48×10^{3}	5.91×10^{2}	6.87×10^{1}	6.60×10^{2}	4.92×10^{1}
10^{-1}	8.17×10^{1}	2.66×10^{2}	3.48×10^{2}	5.92×10^{1}	7.06×10^{0}	6.62×10^{1}	4.89×10^{0}
10^{0}	8.21×10^{0}	2.65×10^{1}	3.47×10^{1}	8.40×10^{0}	7.41×10^{-1}	9.14×10^{0}	-3.36×10^{-1}
10^{1}	1.04×10^{0}	2.72×10^{0}	3.76×10^{0}	9.85×10^{-1}	8.80×10^{-2}	1.07×10^{0}	-1.04×10^{-1}
10^{2}	3.41×10^{-1}	3.24×10^{-1}	6.65×10^{-1}	2.11×10^{-1}	-9.00×10^{-3}	2.02×10^{-1}	-3.40×10^{-2}

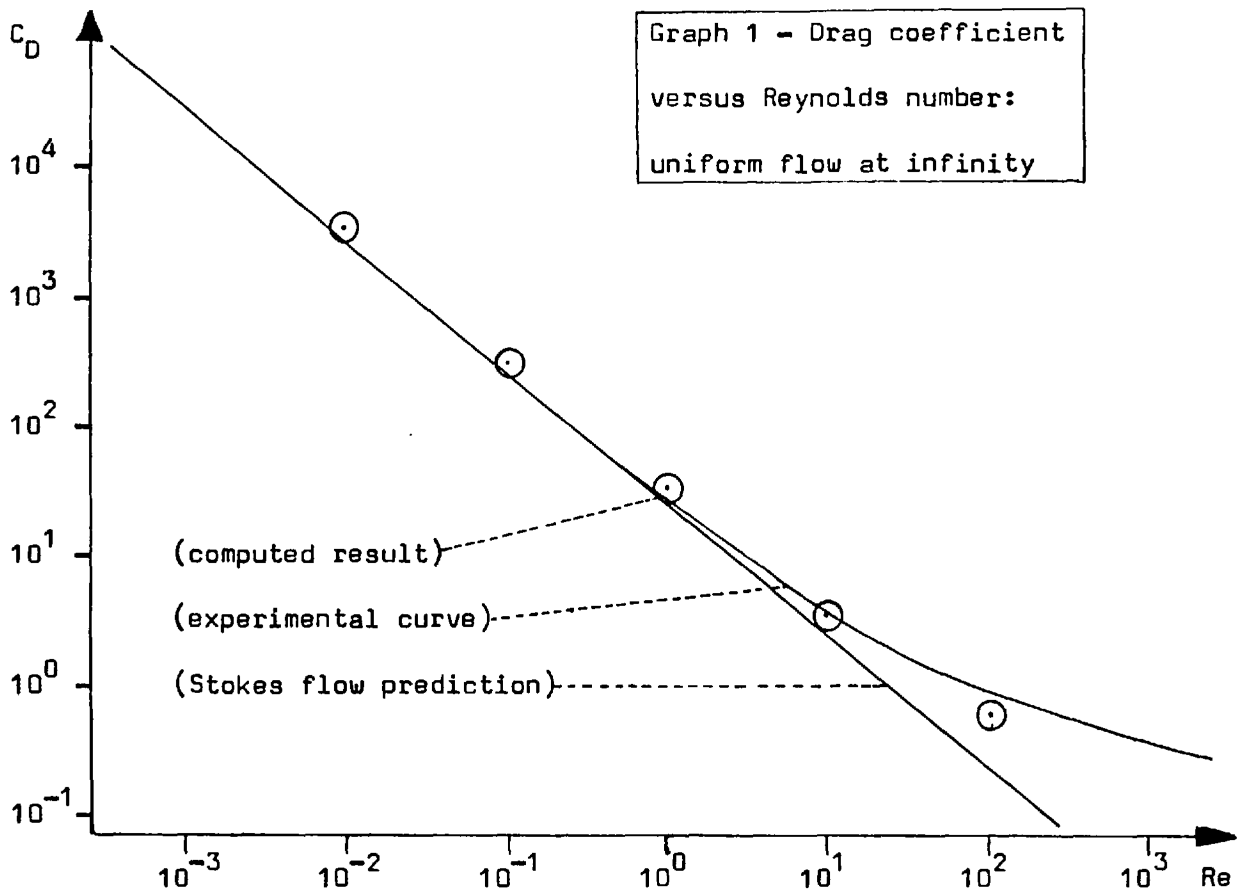
Graphs of computed overall coefficients versus Reynolds number are given as follows:-

Graph 1 - Drag coefficient for uniform flow at infinity;

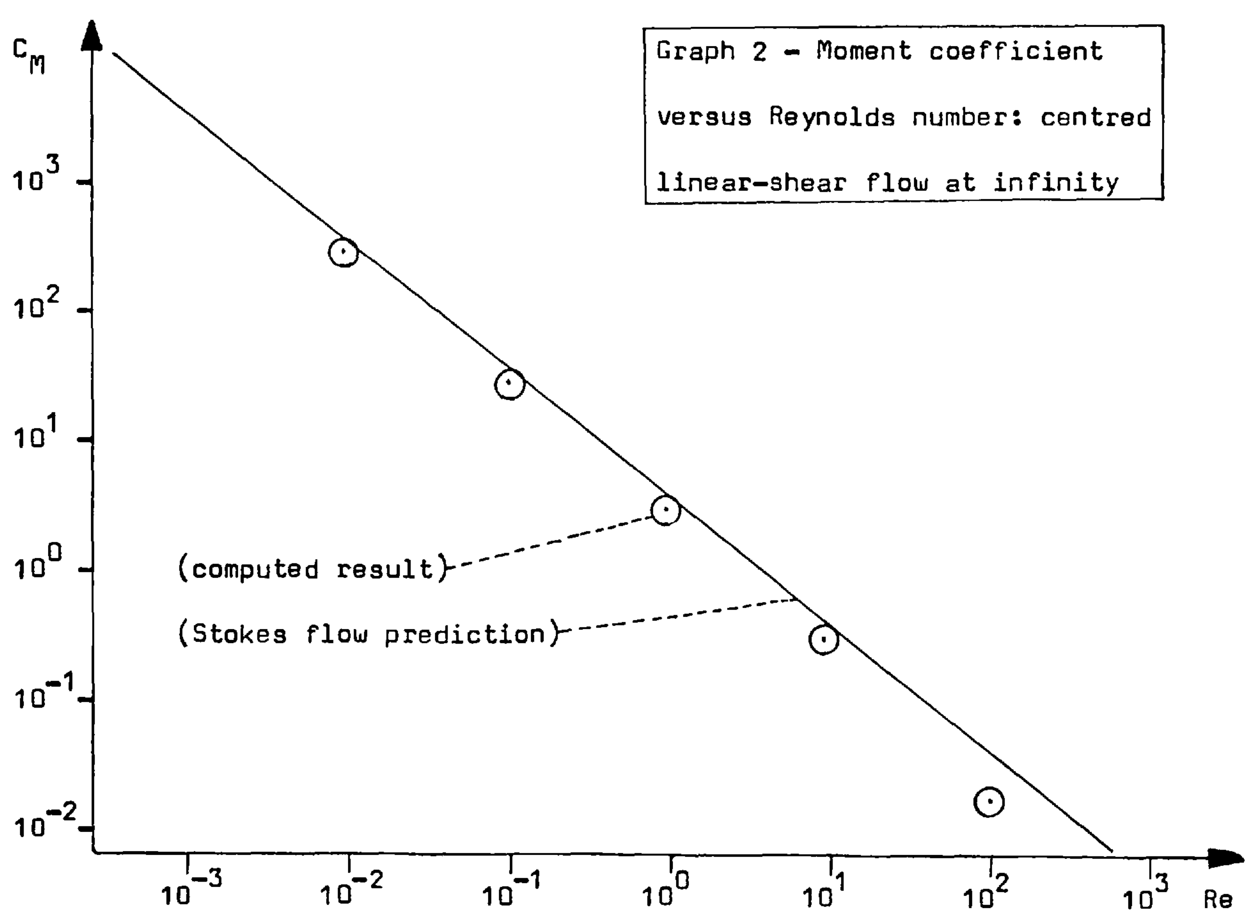
Graph 2 - Moment coefficient for centred linear-shear flow at infinity;

Graphs 3 to 5 - Drag, moment and lift coefficients, respectively, for uniform plus centred linear-shear flow at infinity.

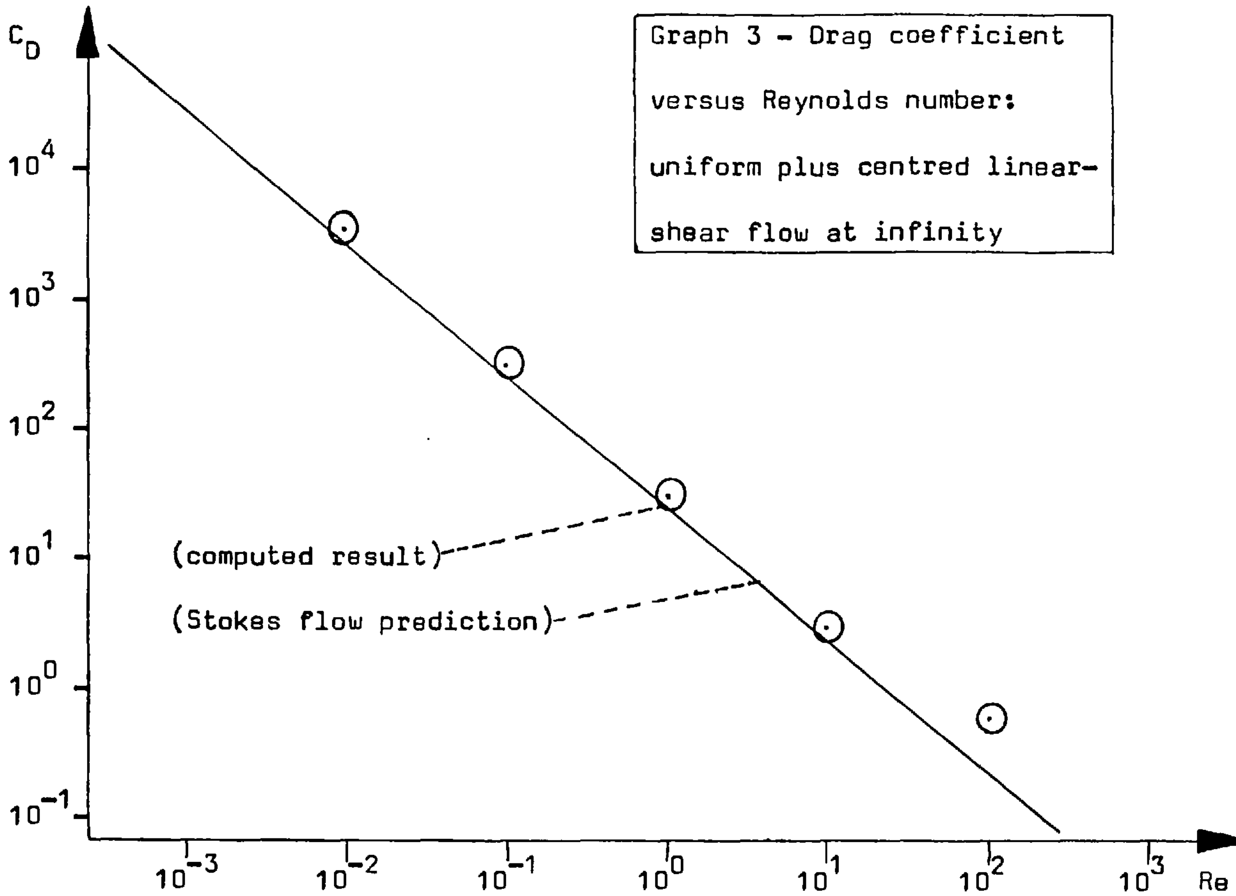
The curves of experimental and theoretical coefficients included for comparison purposes are referenced in sub-section 3.4.3 of Chapter 3.



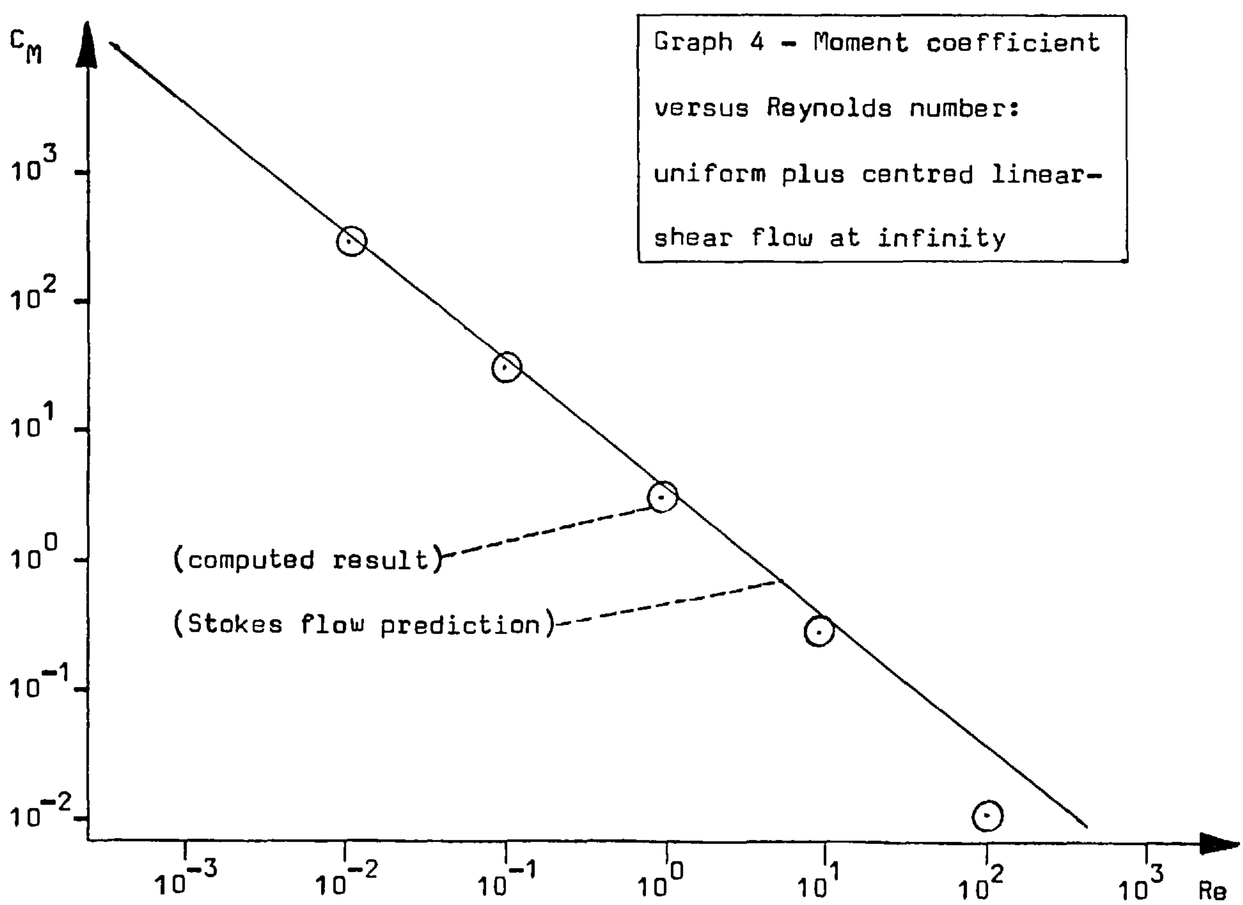
Graph 1 - Drag coefficient versus Reynolds number: uniform flow at infinity
C_D
10^4
10^3
10^2
10^1
10^0
10^{-1}
(computed result)
(experimental curve)
(Stokes flow prediction)
10^{-3}
10^{-2}
10^{-1}
10^0
10^1
10^2
10^3
Re



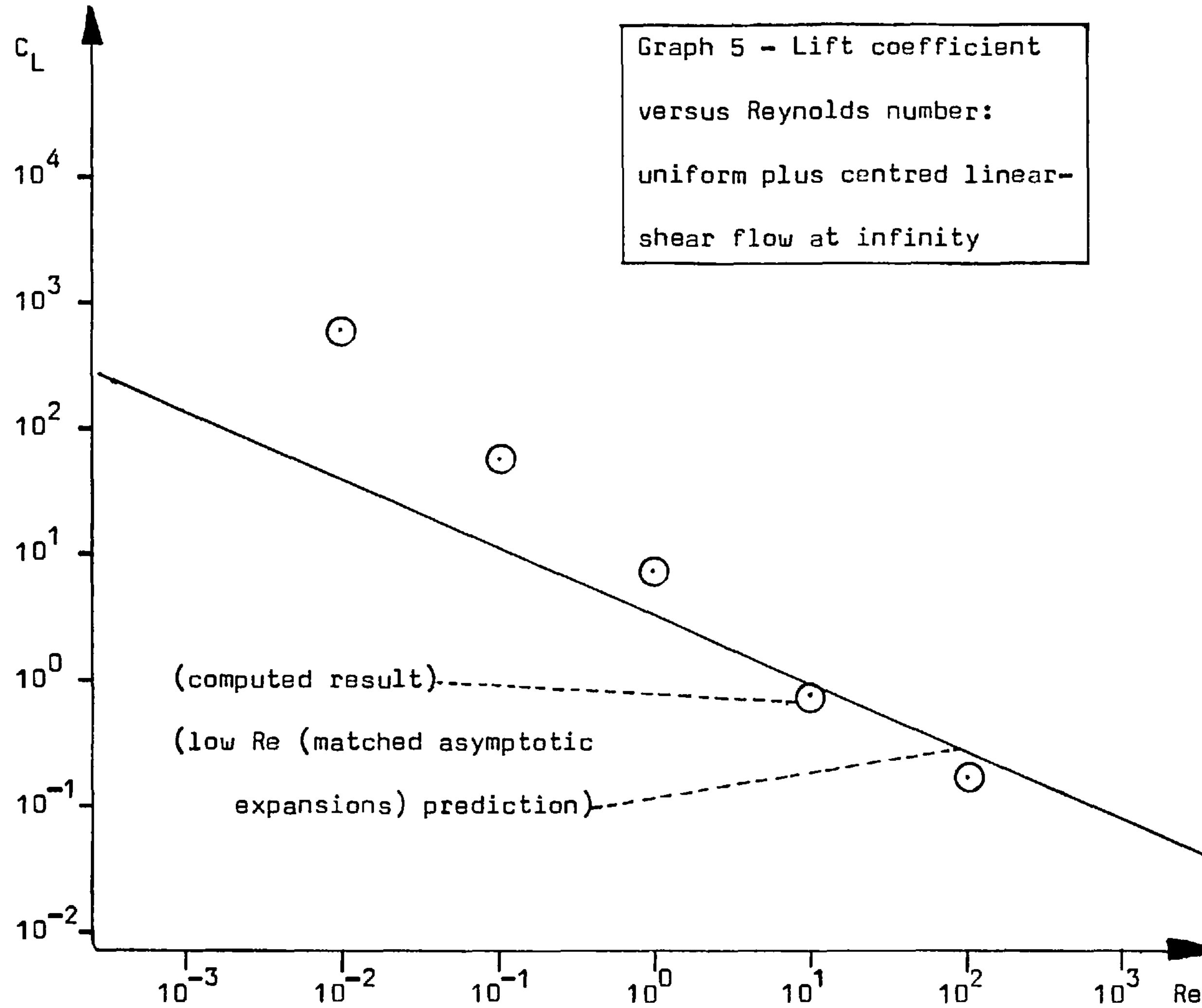
Graph 2 - Moment coefficient versus Reynolds number: centred linear-shear flow at infinity
C_M
10^3
10^2
10^1
10^0
10^{-1}
10^{-2}
(computed result)
(Stokes flow prediction)
10^{-3}
10^{-2}
10^{-1}
10^0
10^1
10^2
10^3
Re



Graph 3 - Drag coefficient versus Reynolds number: uniform plus centred linear-shear flow at infinity



Graph 4 - Moment coefficient versus Reynolds number: uniform plus centred linear-shear flow at infinity


Graph 5 - Lift coefficient versus Reynolds number: uniform plus centred linear-shear flow at infinity
C_L
10^4
10^3
10^2
10^1
10^0
10^{-1}
10^{-2}
(computed result)
(low Re (matched asymptotic expansions) prediction)
10^{-3}
10^{-2}
10^{-1}
10^0
10^1
10^2
10^3
Re

Appendix 3 – <u>Finite-Difference Equations of Motion</u>

(i) The scalar potential Laplace equation is:-

$$h_{i+1}\left(\frac{1}{\Delta r^2}+\frac{1}{r\Delta r}\right)+h_{i-1}\left(\frac{1}{\Delta r^2}-\frac{1}{r\Delta r}\right)+h_{j+1}\left(\frac{1}{r^2\Delta\theta^2}+\frac{\cot\theta}{2r^2\Delta\theta}\right)$$

$$+h_{j-1}\left(\frac{1}{r^2\Delta\theta^2}-\frac{\cot\theta}{2r^2\Delta\theta}\right)+h_{k+1}\left(\frac{1}{r^2\sin^2\theta\,\Delta\phi^2}\right)+h_{k-1}\left(\frac{1}{r^2\sin^2\theta\,\Delta\phi^2}\right)$$

$$+h\left(-\frac{2}{\Delta r^2}-\frac{2}{r^2\Delta\theta^2}-\frac{2}{r^2\sin^2\theta\,\Delta\phi^2}\right)=0$$

while, on the polar axis, we have:-

$$h_{i+1}\left(\frac{1}{\Delta r^2}+\frac{1}{r\Delta r}\right)+h_{i-1}\left(\frac{1}{\Delta r^2}-\frac{1}{r\Delta r}\right)+h_{j+1}\left(\frac{2}{r^2\Delta\theta^2}\right)$$

$$+h_{j-1}\left(\frac{2}{r^2\Delta\theta^2}\right)+h\left(-\frac{2}{\Delta r^2}-\frac{4}{r^2\Delta\theta^2}\right)=0$$

The boundary conditions are:-

$$\frac{1}{2\Delta r}\left(-3\,h+4\,h_{i+1}-h_{i+2}\right)=0 \quad \text{at } i=1$$

$$\frac{-1}{2\Delta r}\left(-3\,h+4\,h_{i-1}-h_{i-2}\right)=$$

$$\left(U^0_\infty+\sigma r^*\sin\theta\cos\phi\right)\cos\theta \quad \text{at } i=n_r+1$$

(ii) The vector potential Poisson equation is:-

$$\omega_r = A_r\left(\frac{2}{r^2\Delta\theta^2}+\frac{2}{r^2\sin^2\theta\,\Delta\phi^2}\right)+A_\theta\left(\frac{1}{r^2\tan\theta}\right)+A_{\theta\,i+1}\left(\frac{1}{2r\tan\theta\,\Delta r}\right)$$

$$-A_{\theta\,i-1}\left(\frac{1}{2r\tan\theta\,\Delta r}\right)+A_{r\,j+1}\left(\frac{-1}{r^2\Delta\theta^2}-\frac{1}{2r^2\tan\theta\,\Delta\theta}\right)$$

$$-A_{r\,j-1}\left(\frac{1}{r^2\Delta\theta^2}-\frac{1}{2r^2\tan\theta\,\Delta\theta}\right)+A_{\theta\,j+1}\left(\frac{1}{2r^2\Delta\theta}\right)-A_{\theta\,j-1}\left(\frac{1}{2r^2\Delta\theta}\right)$$

$$+A_{r\,k+1}\left(\frac{-1}{r^2\sin^2\theta\,\Delta\phi^2}\right)-A_{r\,k-1}\left(\frac{1}{r^2\sin^2\theta\,\Delta\phi^2}\right)+A_{\phi\,k+1}\left(\frac{1}{2r^2\sin\theta\,\Delta\phi}\right)$$

$$-A_{\phi\,k-1}\left(\frac{1}{2r^2\sin\theta\,\Delta\phi}\right)+A_{\theta\,i+1\,j+1}\left(\frac{1}{4r\Delta r\Delta\theta}\right)-A_{\theta\,i-1\,j+1}\left(\frac{1}{4r\Delta r\Delta\theta}\right)$$

→

$$- A_{\theta i+1\,j-1}\left(\frac{1}{4r\Delta r\Delta\theta}\right) + A_{\theta i-1\,j-1}\left(\frac{1}{4r\Delta r\Delta\theta}\right)$$

$$+ A_{\phi i+1k+1}\left(\frac{1}{4r\sin\theta\,\Delta r\,\Delta\phi}\right) - A_{\phi i-1k+1}\left(\frac{1}{4r\sin\theta\,\Delta r\,\Delta\phi}\right)$$

$$- A_{\phi i+1k-1}\left(\frac{1}{4r\sin\theta\,\Delta r\,\Delta\phi}\right) + A_{\phi i-1k-1}\left(\frac{1}{4r\sin\theta\,\Delta r\,\Delta\phi}\right)$$

$$w_\theta = A_\theta\left(\frac{2}{r^2\sin^2\theta\,\Delta\phi^2} + \frac{2}{\Delta r^2}\right) + A_{\theta i+1}\left(\frac{-1}{r\Delta r} - \frac{1}{\Delta r^2}\right)$$

$$- A_{\theta i-1}\left(\frac{-1}{r\Delta r} + \frac{1}{\Delta r^2}\right) + A_{\theta k+1}\left(\frac{-1}{r^2\sin^2\theta\,\Delta\phi^2}\right) - A_{\theta k-1}\left(\frac{1}{r^2\sin^2\theta\,\Delta\phi^2}\right)$$

$$+ A_{\phi k+1}\left(\frac{1}{2r^2\sin\theta\tan\theta\,\Delta\phi}\right) - A_{\phi k-1}\left(\frac{1}{2r^2\sin\theta\tan\theta\,\Delta\phi}\right)$$

$$+ A_{r i+1\,j+1}\left(\frac{1}{4r\Delta r\Delta\theta}\right) - A_{r i-1\,j+1}\left(\frac{1}{4r\Delta r\Delta\theta}\right) - A_{r i+1\,j-1}\left(\frac{1}{4r\Delta r\Delta\theta}\right)$$

$$+ A_{r i-1\,j-1}\left(\frac{1}{4r\Delta r\Delta\theta}\right) + A_{\phi j+1k+1}\left(\frac{1}{4r^2\sin\theta\,\Delta\theta\,\Delta\phi}\right)$$

$$- A_{\phi j-1k+1}\left(\frac{1}{4r^2\sin\theta\,\Delta\theta\Delta\phi}\right) - A_{\phi j+1k-1}\left(\frac{1}{4r^2\sin\theta\,\Delta\theta\Delta\phi}\right)$$

$$+ A_{\phi j-1k-1}\left(\frac{1}{4r^2\sin\theta\,\Delta\theta\Delta\phi}\right)$$

$$w_\phi = A_\phi\left(\frac{2}{\Delta r^2} + \frac{2}{r^2\Delta\theta^2} + \frac{1}{r^2\sin^2\theta}\right) + A_{\phi i+1}\left(\frac{-1}{r\Delta r} - \frac{1}{\Delta r^2}\right)$$

$$- A_{\phi i-1}\left(\frac{-1}{r\Delta r} + \frac{1}{\Delta r^2}\right) + A_{\phi j+1}\left(\frac{-1}{r^2\Delta\theta^2} - \frac{1}{2r^2\tan\theta\,\Delta\theta}\right)$$

$$- A_{\phi j-1}\left(\frac{1}{r^2\Delta\theta^2} - \frac{1}{2r^2\tan\theta\,\Delta\theta}\right) + A_{\theta k+1}\left(\frac{-1}{2r^2\sin\theta\tan\theta\,\Delta\phi}\right)$$

$$- A_{\theta k-1}\left(\frac{-1}{2r^2\sin\theta\tan\theta\,\Delta\phi}\right) + A_{r i+1k+1}\left(\frac{1}{4r\sin\theta\,\Delta r\,\Delta\phi}\right)$$

$$- A_{r i-1k+1}\left(\frac{1}{4r\sin\theta\,\Delta r\,\Delta\phi}\right) - A_{r i+1k-1}\left(\frac{1}{4r\sin\theta\,\Delta r\,\Delta\phi}\right)$$

→

$$+ A_{r\,i-1\,k-1}\left(\frac{1}{4r\sin\theta\Delta r\Delta\phi}\right) + A_{\theta\,j+1\,k+1}\left(\frac{1}{4r^2\sin\theta\Delta\theta\Delta\phi}\right)$$

$$- A_{\theta\,j-1\,k+1}\left(\frac{1}{4r^2\sin\theta\Delta\theta\Delta\phi}\right) - A_{\theta\,j+1\,k-1}\left(\frac{1}{4r^2\sin\theta\Delta\theta\Delta\phi}\right)$$

$$+ A_{\theta\,j-1\,k-1}\left(\frac{1}{4r^2\sin\theta\Delta\theta\Delta\phi}\right)$$

while, on the polar axis, we have:-

$$w_r = A_r\left(\frac{-2}{r^2\Delta\theta^2\Delta\phi^2} + \frac{4}{r^2\Delta\theta^2}\right) + A_{r\,j+1}\left(\frac{-2}{r^2\Delta\theta^2} + \frac{1}{r^2\Delta\theta^2\Delta\phi^2}\right)$$

$$- A_{r\,j-1}\left(\frac{2}{r^2\Delta\theta^2} - \frac{1}{r^2\Delta\theta^2\Delta\phi^2}\right) + A_{\theta\,j+1}\left(\frac{1}{r^2\Delta\theta}\right) - A_{\theta\,j-1}\left(\frac{1}{r^2\Delta\theta}\right)$$

$$+ A_{r\,k+1}\left(\frac{1}{r^2\Delta\theta^2\Delta\phi^2}\right) - A_{r\,k-1}\left(\frac{-1}{r^2\Delta\theta^2\Delta\phi^2}\right) + A_{\theta\,i+1\,j+1}\left(\frac{1}{2r\Delta r\Delta\theta}\right)$$

$$- A_{\theta\,i-1\,j+1}\left(\frac{1}{2r\Delta r\Delta\theta}\right) - A_{\theta\,i+1\,j-1}\left(\frac{1}{2r\Delta r\Delta\theta}\right)$$

$$+ A_{\theta\,i-1\,j-1}\left(\frac{1}{2r\Delta r\Delta\theta}\right) + A_{r\,j+1\,k+1}\left(\frac{-1}{2r^2\Delta\theta^2\Delta\phi^2}\right)$$

$$- A_{r\,j-1\,k+1}\left(\frac{1}{2r^2\Delta\theta^2\Delta\phi^2}\right) - A_{r\,j+1\,k-1}\left(\frac{1}{2r^2\Delta\theta^2\Delta\phi^2}\right)$$

$$+ A_{r\,j-1\,k-1}\left(\frac{-1}{2r^2\Delta\theta^2\Delta\phi^2}\right) + A_{\phi\,j+1\,k+1}\left(\frac{\pm 1}{4r^2\Delta\theta\Delta\phi}\right)$$

$$- A_{\phi\,j-1\,k+1}\left(\frac{\pm 1}{4r^2\Delta\theta\Delta\phi}\right) - A_{\phi\,j+1\,k-1}\left(\frac{\pm 1}{4r^2\Delta\theta\Delta\phi}\right)$$

$$+ A_{\phi\,j-1\,k-1}\left(\frac{\pm 1}{4r^2\Delta\theta\Delta\phi}\right) + A_{\phi\,i+1\,j+1\,k+1}\left(\frac{\pm 1}{8r\Delta r\Delta\theta\Delta\phi}\right)$$

$$- A_{\phi\,i-1\,j+1\,k+1}\left(\frac{\pm 1}{8r\Delta r\Delta\theta\Delta\phi}\right) - A_{\phi\,i+1\,j-1\,k+1}\left(\frac{\pm 1}{8r\Delta r\Delta\theta\Delta\phi}\right)$$

$$+ A_{\phi\,i-1\,j-1\,k+1}\left(\frac{\pm 1}{8r\Delta r\Delta\theta\Delta\phi}\right) - A_{\phi\,i+1\,j+1\,k-1}\left(\frac{\pm 1}{8r\Delta r\Delta\theta\Delta\phi}\right)$$

$$+ A_{\phi\,i-1\,j+1\,k-1}\left(\frac{\pm 1}{8r\Delta r\Delta\theta\Delta\phi}\right) + A_{\phi\,i+1\,j-1\,k-1}\left(\frac{\pm 1}{8r\Delta r\Delta\theta\Delta\phi}\right)$$

→

$$- A_{\phi i-1\,j-1\,k-1}\left(\frac{\pm 1}{8r\Delta r\Delta\theta\Delta\phi}\right)$$

$$\omega_\theta = A_\theta\left(\frac{-2}{r^2\Delta\theta^2\Delta\phi^2} + \frac{2}{\Delta r^2}\right) + A_{\theta i+1}\left(\frac{-1}{r\Delta r} - \frac{1}{\Delta r^2}\right)$$

$$- A_{\theta i-1}\left(\frac{-1}{r\Delta r} + \frac{1}{\Delta r^2}\right) + A_{\theta j+1}\left(\frac{1}{r^2\Delta\theta^2\Delta\phi^2}\right) - A_{\theta j-1}\left(\frac{-1}{r^2\Delta\theta^2\Delta\phi^2}\right)$$

$$+ A_{\theta k+1}\left(\frac{1}{r^2\Delta\theta^2\Delta\phi^2}\right) - A_{\theta k-1}\left(\frac{-1}{r^2\Delta\theta^2\Delta\phi^2}\right) + A_{\phi k+1}\left(\frac{\mp 3}{2r^2\Delta\theta^2\Delta\phi}\right)$$

$$- A_{\phi k-1}\left(\frac{\mp 3}{2r^2\Delta\theta^2\Delta\phi}\right) + A_{r i+1\,j+1}\left(\frac{1}{4r\Delta r\Delta\theta}\right) - A_{r i-1\,j+1}\left(\frac{1}{4r\Delta r\Delta\theta}\right)$$

$$- A_{r i+1\,j-1}\left(\frac{1}{4r\Delta r\Delta\theta}\right) + A_{r i-1\,j-1}\left(\frac{1}{4r\Delta r\Delta\theta}\right)$$

$$+ A_{\theta j+1\,k+1}\left(\frac{-1}{2r^2\Delta\theta^2\Delta\phi^2}\right) - A_{\theta j-1\,k+1}\left(\frac{1}{2r^2\Delta\theta^2\Delta\phi^2}\right)$$

$$- A_{\theta j+1\,k-1}\left(\frac{1}{2r^2\Delta\theta^2\Delta\phi^2}\right) + A_{\theta j-1\,k-1}\left(\frac{-1}{2r^2\Delta\theta^2\Delta\phi^2}\right)$$

$$+ A_{\phi j+1\,k+1}\left(\frac{\pm 3}{4r^2\Delta\theta^2\Delta\phi}\right) - A_{\phi j-1\,k+1}\left(\frac{\mp 3}{4r^2\Delta\theta^2\Delta\phi}\right)$$

$$- A_{\phi j+1\,k-1}\left(\frac{\pm 3}{4r^2\Delta\theta^2\Delta\phi}\right) + A_{\phi j-1\,k-1}\left(\frac{\mp 3}{4r^2\Delta\theta^2\Delta\phi}\right)$$

$$\omega_\phi = A_\phi\left(\frac{2}{\Delta r^2} + \frac{3}{r^2\Delta\theta^2}\right) + A_{\phi i+1}\left(\frac{-1}{r\Delta r} - \frac{1}{\Delta r^2}\right) - A_{\phi i-1}\left(\frac{-1}{r\Delta r} + \frac{1}{\Delta r^2}\right)$$

$$+ A_{\phi j+1}\left(\frac{-3}{2r^2\Delta\theta^2}\right) - A_{\phi j-1}\left(\frac{3}{2r^2\Delta\theta^2}\right) + A_{\theta k+1}\left(\frac{\mp 1}{2r^2\Delta\theta^2\Delta\phi}\right)$$

$$- A_{\theta k-1}\left(\frac{\mp 1}{2r^2\Delta\theta^2\Delta\phi}\right) + A_{\theta j+1\,k+1}\left(\frac{\pm 1}{4r^2\Delta\theta^2\Delta\phi}\right)$$

$$- A_{\theta j-1\,k+1}\left(\frac{\mp 1}{4r^2\Delta\theta^2\Delta\phi}\right) - A_{\theta j+1\,k-1}\left(\frac{\pm 1}{4r^2\Delta\theta^2\Delta\phi}\right)$$

$$+ A_{\theta j-1\,k-1}\left(\frac{\mp 1}{4r^2\Delta\theta^2\Delta\phi}\right) + A_{r i+1\,j+1\,k+1}\left(\frac{\pm 1}{8r\Delta r\Delta\theta\Delta\phi}\right)$$

⟶

$$- A_{ri-1\,j+1\,k+1}\left(\frac{\pm 1}{8r\Delta r\Delta\theta\Delta\phi}\right) - A_{ri+1\,j-1\,k+1}\left(\frac{\pm 1}{8r\Delta r\Delta\theta\Delta\phi}\right)$$

$$+ A_{ri-1\,j-1\,k+1}\left(\frac{\pm 1}{8r\Delta r\Delta\theta\Delta\phi}\right) - A_{ri+1\,j+1\,k-1}\left(\frac{\pm 1}{8r\Delta r\Delta\theta\Delta\phi}\right)$$

$$+ A_{ri-1\,j+1\,k-1}\left(\frac{\pm 1}{8r\Delta r\Delta\theta\Delta\phi}\right) + A_{ri+1\,j-1\,k-1}\left(\frac{\pm 1}{8r\Delta r\Delta\theta\Delta\phi}\right)$$

$$- A_{ri-1\,j-1\,k-1}\left(\frac{\pm 1}{8r\Delta r\Delta\theta\Delta\phi}\right)$$

The boundary conditions are:-

$$\left.\begin{array}{l} A_\theta \equiv 0 \equiv A_\phi \\ A_r\left(\frac{2}{r_o} - \frac{3}{2\Delta r}\right) + A_{ri+1}\left(\frac{2}{\Delta r}\right) + A_{ri+2}\left(\frac{-1}{2\Delta r}\right) = 0 \end{array}\right\} \text{ at } i = 1$$

$$\left.\begin{array}{l} A_\theta \equiv 0 \equiv A_\phi \\ A_r\left(\frac{-2}{r^*} - \frac{3}{2\Delta r}\right) + A_{ri-1}\left(\frac{2}{\Delta r}\right) + A_{ri-2}\left(\frac{-1}{2\Delta r}\right) = 0 \end{array}\right\} \text{ at } i = n_r+1$$

(iii) The vorticity transport equation is:-

$$\frac{1}{\Delta t}\,w_r^n = w_r\left(\frac{1}{\Delta t} + \frac{A_{\phi j+1} - A_{\phi j-1}}{2r^2\Delta\theta} + \frac{A_\phi}{r^2\tan\theta} - \frac{A_{\theta k+1} - A_{\theta k-1}}{2r^2\sin\theta\Delta\phi}\right.$$

$$+ \frac{2}{r}V_r - \frac{A_{\phi i+1} - A_{\phi i-1}}{2r\tan\theta\Delta r} + \frac{h_{j+1} - 2h + h_{j-1}}{r^2\Delta\theta^2} + \frac{V_\theta}{r\tan\theta}$$

$$- \frac{A_{\phi i+1\,j+1} - A_{\phi i-1\,j+1} - A_{\phi i+1\,j-1} + A_{\phi i-1\,j-1}}{4r\Delta r\Delta\theta}$$

$$+ \frac{A_{\theta i+1\,k+1} - A_{\theta i-1\,k+1} - A_{\theta i+1\,k-1} + A_{\theta i-1\,k-1}}{4r\sin\theta\Delta r\Delta\phi}$$

$$\left.+ \frac{h_{k+1} - 2h + h_{k-1}}{r^2\sin^2\theta\Delta\phi^2} + \frac{1}{Re}\left\{\frac{2}{r^2\Delta\theta^2} + \frac{2}{r^2\sin^2\theta\Delta\phi^2}\right\}\right)$$

$$+ w_\theta\left(- \frac{A_{\phi j+1} - 2A_\phi + A_{\phi j-1}}{r^2\Delta\theta^2} + \frac{1}{r^2\sin^2\theta}A_\phi - \frac{A_{\phi j+1} - A_{\phi j-1}}{2r^2\tan\theta\Delta\theta}\right.$$

$$- \frac{A_{\theta k+1} - A_{\theta k-1}}{2r^2\sin\theta\tan\theta\Delta\phi} + \frac{A_{\theta j+1\,k+1} - A_{\theta j-1\,k+1} - A_{\theta j+1\,k-1} + A_{\theta j-1\,k-1}}{4r^2\sin\theta\Delta\theta\Delta\phi}$$

→

$$- \frac{h_{i+1\,j+1} - h_{i-1\,j+1} - h_{i+1\,j-1} + h_{i-1\,j-1}}{4r\Delta r\Delta\theta} + \frac{1}{Re\, r^2 \tan\theta} \Bigg)$$

$$+ w_{\phi} \Bigg(- \frac{A_{\phi\, i+1k+1} - A_{\phi\, i-1k+1} - A_{\phi\, i+1k-1} + A_{\phi\, i-1k-1}}{4r^2 \sin\theta \Delta\theta\, \Delta\phi}$$

$$- \frac{A_{\phi k+1} - A_{\phi k-1}}{2r^2 \sin\theta \tan\theta \Delta\phi} + \frac{A_{\theta k+1} - 2A_{\theta} + A_{\theta k-1}}{r^2 \sin^2\theta\, \Delta\phi^2}$$

$$- \frac{h_{i+1k+1} - h_{i-1k+1} - h_{i+1k-1} + h_{i-1k-1}}{4r \sin\theta\, \Delta r\, \Delta\phi} \Bigg)$$

$$+ w_{r\,i+1} \left(\frac{A_{\phi j+1} - A_{\phi j-1}}{4r\, \Delta r \Delta\theta} + \frac{A_{\phi}}{2r \tan\theta \Delta r} - \frac{A_{\theta k+1} - A_{\theta k-1}}{4r \sin\theta \Delta r \Delta\phi} + \frac{V_r}{2\Delta r} \right)$$

$$- w_{r\,i-1} \left(\frac{A_{\phi j+1} - A_{\phi j-1}}{4r\, \Delta r \Delta\theta} + \frac{A_{\phi}}{2r \tan\theta \Delta r} - \frac{A_{\theta k+1} - A_{\theta k-1}}{4r \sin\theta \Delta r \Delta\phi} + \frac{V_r}{2\Delta r} \right)$$

$$+ w_{\theta i+1} \left(\frac{1}{2\, Re\, r \tan\theta \Delta r} \right) - w_{\theta i-1} \left(\frac{1}{2\, Re\, r \tan\theta \Delta r} \right)$$

$$+ w_{r\,j+1} \Bigg(\frac{A_{r k+1} - A_{r k-1}}{4r^2 \sin\theta\, \Delta\theta\, \Delta\phi} - \frac{A_{\phi i+1} - A_{\phi i-1}}{4r\, \Delta r\, \Delta\theta} - \frac{A_{\phi}}{2r^2 \Delta\theta} + \frac{V_{\theta}}{2r\Delta\theta}$$

$$+ \frac{1}{Re} \left\{ \frac{-1}{r^2 \Delta\theta^2} - \frac{1}{2r^2 \tan\theta\, \Delta\theta} \right\} \Bigg)$$

$$- w_{r\,j-1} \Bigg(\frac{A_{r k+1} - A_{r k-1}}{4r^2 \sin\theta \Delta\theta \Delta\phi} - \frac{A_{\phi i+1} - A_{\phi i-1}}{4r \Delta r \Delta\theta} - \frac{A_{\phi}}{2r^2 \Delta\theta} + \frac{V_{\theta}}{2r\Delta\theta}$$

$$+ \frac{1}{Re} \left\{ \frac{1}{r^2 \Delta\theta^2} - \frac{1}{2r^2 \tan\theta\, \Delta\theta} \right\} \Bigg)$$

$$+ w_{\theta j+1} \left(\frac{1}{2\, Re\, r^2 \Delta\theta} \right) - w_{\theta j-1} \left(\frac{1}{2\, Re\, r^2 \Delta\theta} \right)$$

$$+ w_{r k+1} \Bigg(\frac{A_{\theta i+1} - A_{\theta i-1}}{4r \sin\theta\, \Delta r\, \Delta\phi} + \frac{A_{\theta}}{2r^2 \sin\theta \Delta\phi} - \frac{A_{r j+1} - A_{r j-1}}{4r^2 \sin\theta \Delta\theta \Delta\phi}$$

$$+ \frac{V_{\phi}}{2r \sin\theta \Delta\phi} - \frac{1}{Re\, r^2 \sin^2\theta\, \Delta\phi^2} \Bigg)$$

$$- w_{r k-1} \Bigg(\frac{A_{\theta i+1} - A_{\theta i-1}}{4\, r \sin\theta \Delta r \Delta\phi} + \frac{A_{\theta}}{2r^2 \sin\theta \Delta\phi} - \frac{A_{r j+1} - A_{r j-1}}{4r^2 \sin\theta \Delta\theta \Delta\phi}$$

→

$$+ \frac{V_\phi}{2r\sin\theta\,\Delta\phi} + \frac{1}{Re\,r^2\sin\theta^2\,\Delta\phi^2}\Bigg)$$

$$+ w_{\phi k+1}\left(\frac{1}{2\,Re\,r^2\sin\theta\,\Delta\phi}\right) - w_{\phi k-1}\left(\frac{1}{2\,Re\,r^2\sin\theta\,\Delta\phi}\right)$$

$$+ w_{\theta i+1\,j+1}\left(\frac{1}{4\,Re\,r\,\Delta r\,\Delta\theta}\right) - w_{\theta i-1\,j+1}\left(\frac{1}{4\,Re\,r\,\Delta r\,\Delta\theta}\right)$$

$$- w_{\theta i+1\,j-1}\left(\frac{1}{4\,Re\,r\,\Delta r\,\Delta\theta}\right) + w_{\theta i-1\,j-1}\left(\frac{1}{4\,Re\,r\,\Delta r\,\Delta\theta}\right)$$

$$+ w_{\phi i+1k+1}\left(\frac{1}{4\,Re\,r\sin\theta\,\Delta r\,\Delta\phi}\right) - w_{\phi i-1k+1}\left(\frac{1}{4\,Re\,r\sin\theta\,\Delta r\,\Delta\phi}\right)$$

$$- w_{\phi i+1k-1}\left(\frac{1}{4\,Re\,r\sin\theta\,\Delta r\,\Delta\phi}\right) + w_{\phi i-1k-1}\left(\frac{1}{4\,Re\,r\sin\theta\,\Delta r\,\Delta\phi}\right)$$

$$\frac{1}{\Delta t}\,w_\theta^n = w_r\Bigg(\frac{A_{rk+1} - A_{rk-1}}{r^2\sin\theta\,\Delta\phi} - \frac{2}{r^2}A_\phi + \frac{1}{r}V_\theta + \frac{A_{\phi i+1} - 2A_\phi + A_{\phi i-1}}{\Delta r^2}$$

$$- \frac{A_{ri+1k+1} - A_{ri-1k+1} - A_{ri+1k-1} + A_{ri-1k-1}}{4r\sin\theta\,\Delta r\,\Delta\phi}$$

$$- \frac{h_{i+1\,j+1} - h_{i-1\,j+1} - h_{i+1\,j-1} + h_{i-1\,j-1}}{4r\,\Delta r\,\Delta\theta} + \frac{h_{j+1} - h_{j-1}}{2r^2\,\Delta\theta}\Bigg)$$

$$+ w_\theta\Bigg(\frac{1}{\Delta t} + \frac{1}{r}V_r + \frac{A_{rk+1} - A_{rk-1}}{2r^2\sin\theta\tan\theta\,\Delta\phi} + \frac{V_\theta}{r\tan\theta} - \frac{A_\phi}{r^2\tan\theta}$$

$$+ \frac{A_{\phi i+1\,j+1} - A_{\phi i-1\,j+1} - A_{\phi i+1\,j-1} + A_{\phi i-1\,j-1}}{4r\,\Delta r\,\Delta\theta} + \frac{A_{\theta k+1} - A_{\theta k-1}}{2r^2\sin\theta\,\Delta\phi}$$

$$+ \frac{h_{i+1} - 2h + h_{i-1}}{\Delta r^2} + \frac{h_{k+1} - 2h + h_{k-1}}{r^2\sin^2\theta\,\Delta\phi^2}$$

$$- \frac{A_{rj+1k+1} - A_{rj-1k+1} - A_{rj+1k-1} + A_{rj-1k-1}}{4r^2\sin\theta\,\Delta\theta\,\Delta\phi}$$

$$+ \frac{1}{Re}\left\{\frac{2}{r^2\sin^2\theta\,\Delta\phi^2} + \frac{2}{\Delta r^2}\right\}\Bigg)$$

$$+ w_\phi\Bigg(- \frac{A_{rk+1} - 2A_r + A_{rk-1}}{r^2\sin^2\theta\,\Delta\phi^2} + \frac{A_{\phi k+1} - A_{\phi k-1}}{2r^2\sin\theta\,\Delta\phi}$$

→

$$+ \frac{A_{\phi i+1k+1} - A_{\phi i-1k+1} - A_{\phi i+1k-1} + A_{\phi i-1k-1}}{4r \sin\theta\, \Delta r\, \Delta\phi}$$

$$- \left. \frac{h_{j+1k+1} - h_{j-1k+1} - h_{j+1k-1} + h_{j-1k-1}}{4r^2 \sin\theta\, \Delta\theta\, \Delta\phi} \right)$$

$$+ \omega_{\theta\, i+1} \left(\frac{A_{\phi j+1} - A_{\phi j-1}}{4r\, \Delta r\, \Delta\theta} + \frac{A_{\phi}}{2r \tan\theta\, \Delta r} - \frac{A_{\theta k+1} - A_{\theta k-1}}{4r \sin\theta\, \Delta r\, \Delta\phi} \right.$$

$$\left. + \frac{V_r}{2\Delta r} + \frac{1}{Re} \left\{ \frac{-1}{r\Delta r} - \frac{1}{\Delta r^2} \right\} \right)$$

$$- \omega_{\theta\, i-1} \left(\frac{A_{\phi j+1} - A_{\phi j-1}}{4r\, \Delta r\, \Delta\theta} + \frac{A_{\phi}}{2r \tan\theta\, \Delta r} - \frac{A_{\theta k+1} - A_{\theta k-1}}{4r \sin\theta\, \Delta r\, \Delta\phi} \right.$$

$$\left. + \frac{V_r}{2\Delta r} + \frac{1}{Re} \left\{ \frac{-1}{r\Delta r} + \frac{1}{\Delta r^2} \right\} \right)$$

$$+ \omega_{\theta j+1} \left(\frac{A_{rk+1} - A_{rk-1}}{4r^2 \sin\theta\, \Delta\theta\, \Delta\phi} - \frac{A_{\phi i+1} - A_{\phi i-1}}{4r\, \Delta r \Delta\theta} - \frac{A_{\phi}}{2r^2 \Delta\theta} + \frac{V_{\theta}}{2r\Delta\theta} \right)$$

$$- \omega_{\theta\, j-1} \left(\frac{A_{rk+1} - A_{rk-1}}{4r^2 \sin\theta\, \Delta\theta\, \Delta\phi} - \frac{A_{\phi i+1} - A_{\phi i-1}}{4r\, \Delta r \Delta\theta} - \frac{A_{\phi}}{2r^2 \Delta\theta} + \frac{V_{\theta}}{2r\Delta\theta} \right)$$

$$+ \omega_{\theta k+1} \left(\frac{A_{\theta i+1} - A_{\theta i-1}}{4r \sin\theta\, \Delta r\, \Delta\phi} + \frac{A_{\theta}}{2r^2 \sin\theta\, \Delta\phi} - \frac{A_{r j+1} - A_{r j-1}}{4r^2 \sin\theta\, \Delta\theta\, \Delta\phi} \right.$$

$$\left. + \frac{V_{\phi}}{2r \sin\theta\, \Delta\phi} - \frac{1}{Re\, r^2 \sin^2\theta\, \Delta\phi^2} \right)$$

$$- \omega_{\theta k-1} \left(\frac{A_{\theta i+1} - A_{\theta i-1}}{4r \sin\theta\, \Delta r\, \Delta\phi} + \frac{A_{\theta}}{2r^2 \sin\theta\, \Delta\phi} - \frac{A_{r j+1} - A_{r j-1}}{4r^2 \sin\theta\, \Delta\theta\, \Delta\phi} \right.$$

$$\left. + \frac{V_{\phi}}{2r \sin\theta\, \Delta\phi} + \frac{1}{Re\, r^2 \sin^2\theta\, \Delta\phi^2} \right)$$

$$+ \omega_{\phi k+1} \left(\frac{1}{2\, Re\, r^2 \sin\theta \tan\theta\, \Delta\phi} \right) - \omega_{\phi k-1} \left(\frac{1}{2 Re\, r^2 \sin\theta \tan\theta\, \Delta\phi} \right)$$

$$+ \omega_{r i+1\, j+1} \left(\frac{1}{4\, Re\, r\, \Delta r \Delta\theta} \right) - \omega_{r i-1\, j+1} \left(\frac{1}{4\, Re\, r\, \Delta r \Delta\theta} \right)$$

$$- \omega_{r i+1\, j-1} \left(\frac{1}{4\, Re\, r\, \Delta r \Delta\theta} \right) + \omega_{r i-1\, j-1} \left(\frac{1}{4\, Re\, r\, \Delta r\, \Delta\theta} \right)$$

→

$$+ w_{\phi j+1k+1}\left(\frac{1}{4Re\, r^2 \sin\theta\, \Delta\theta\, \Delta\phi}\right) - w_{\phi j-1k+1}\left(\frac{1}{4Re\, r^2 \sin\theta\, \Delta\theta\Delta\phi}\right)$$

$$- w_{\phi j+1k-1}\left(\frac{1}{4Re\, r^2 \sin\theta\, \Delta\theta\, \Delta\phi}\right) + w_{\phi j-1k-1}\left(\frac{1}{4Re\, r^2 \sin\theta\Delta\theta\Delta\phi}\right)$$

$$\frac{1}{\Delta t} w_{\phi}^{n} = w_r\left(\frac{2}{r^2} A_\theta - \frac{A_{rj+1} - A_{rj-1}}{r^2\Delta\theta} + \frac{1}{r} V_\phi - \frac{A_{\theta i+1} - 2A_\theta + A_{\theta i-1}}{\Delta r^2}\right.$$

$$+ \frac{A_{ri+1j+1} - A_{ri-1j+1} - A_{ri+1j-1} + A_{ri-1j-1}}{4r\Delta r\Delta\theta}$$

$$\left. + \frac{h_{k+1} - h_{k-1}}{2r^2\sin\theta\Delta\phi} - \frac{h_{i+1k+1} - h_{i-1k+1} - h_{i+1k-1} + h_{i-1k-1}}{4r\sin\theta\Delta r\Delta\phi}\right)$$

$$+ w_\theta\left(\frac{A_{\theta i+1} - A_{\theta i-1}}{2r\tan\theta\,\Delta r} + \frac{A_\theta}{r^2\tan\theta} - \frac{A_{rj+1} - A_{rj-1}}{2r^2\tan\theta\,\Delta\theta} + \frac{V_\phi}{r\tan\theta}\right.$$

$$- \frac{A_{\theta j+1} - A_{\theta j-1}}{2r^2\Delta\theta} - \frac{A_{\theta i+1j+1} - A_{\theta i-1j+1} - A_{\theta i+1j-1} + A_{\theta i-1j-1}}{4r\Delta r\Delta\theta}$$

$$+ \frac{A_{rj+1} - 2A_r + A_{rj-1}}{r^2\Delta\theta^2} + \frac{h_{k+1} - h_{k-1}}{2r^2\sin\theta\tan\theta\Delta\phi}$$

$$\left. - \frac{h_{j+1k+1} - h_{j-1k+1} - h_{j+1k-1} + h_{j-1k-1}}{4r^2\sin\theta\,\Delta\theta\,\Delta\phi}\right)$$

$$+ w_\phi\left(\frac{1}{\Delta t} + \frac{1}{r}V_r + \frac{A_{\phi i+1} - A_{\phi i-1}}{2r\tan\theta\Delta r} + \frac{h_{i+1} - 2h + h_{i-1}}{\Delta r^2}\right.$$

$$- \frac{A_{\theta i+1k+1} - A_{\theta i-1k+1} - A_{\theta i+1k-1} + A_{\theta i-1k-1}}{4r\sin\theta\,\Delta r\,\Delta\phi}$$

$$- \frac{A_{rk+1} - A_{rk-1}}{2r^2\sin\theta\tan\theta\Delta\phi} - \frac{A_{\phi j+1} - A_{\phi j-1}}{2r^2\Delta\theta} + \frac{h_{j+1} - 2h + h_{j-1}}{r^2\Delta\theta^2}$$

$$+ \frac{A_{rj+1k+1} - A_{rj-1k+1} - A_{rj+1k-1} + A_{rj-1k-1}}{4r^2\sin\theta\,\Delta\theta\,\Delta\phi}$$

$$\left. + \frac{1}{Re}\left\{\frac{2}{\Delta r^2} + \frac{2}{r^2\Delta\theta^2} + \frac{1}{r^2\sin^2\theta}\right\}\right)$$

$$+ w_{\phi i+1}\left(\frac{A_{\phi j+1} - A_{\phi j-1}}{4r\Delta r\Delta\theta} + \frac{A_\phi}{2r\tan\theta\Delta r} - \frac{A_{\theta k+1} - A_{\theta k-1}}{4r\sin\theta\,\Delta r\Delta\phi} + \frac{V_r}{2\Delta r}\right.$$

⟶

$$+ \frac{1}{Re}\left\{\frac{-1}{r\Delta r} - \frac{1}{\Delta r^2}\right\}\Bigg)$$

$$- w_{\phi i-1}\Bigg(\frac{A_{\phi i+1} - A_{\phi i-1}}{4r\Delta r\Delta\theta} + \frac{A_\phi}{2r\tan\theta\Delta r} - \frac{A_{\theta k+1} - A_{\theta k-1}}{4r\sin\theta\Delta r\Delta\phi} + \frac{V_r}{2\Delta r}$$

$$+ \frac{1}{Re}\left\{\frac{-1}{r\Delta r} + \frac{1}{\Delta r^2}\right\}\Bigg)$$

$$+ w_{\phi j+1}\Bigg(\frac{A_{rk+1} - A_{rk-1}}{4r^2\sin\theta\Delta\theta\Delta\phi} - \frac{A_{\phi i+1} - A_{\phi i-1}}{4r\Delta r\Delta\theta} - \frac{A_\phi}{2r^2\Delta\theta} + \frac{V_\theta}{2r\Delta\theta}$$

$$+ \frac{1}{Re}\left\{\frac{-1}{r^2\Delta\theta^2} - \frac{1}{2r^2\tan\theta\Delta\theta}\right\}\Bigg)$$

$$- w_{\phi j-1}\Bigg(\frac{A_{rk+1} - A_{rk-1}}{4r^2\sin\theta\Delta\theta\Delta\phi} - \frac{A_{\phi i+1} - A_{\phi i-1}}{4r\Delta r\Delta\theta} - \frac{A_\phi}{2r^2\Delta\theta} + \frac{V_\theta}{2r\Delta\theta}$$

$$+ \frac{1}{Re}\left\{\frac{1}{r^2\Delta\theta^2} - \frac{1}{2r^2\tan\theta\Delta\theta}\right\}\Bigg)$$

$$+ w_{\theta k+1}\left(\frac{-1}{2\,Re\,r^2\sin\theta\tan\theta\Delta\phi}\right) - w_{\theta k-1}\left(\frac{-1}{2\,Re\,r^2\sin\theta\tan\theta\Delta\phi}\right)$$

$$+ w_{\phi k+1}\Bigg(\frac{A_{\theta i+1} - A_{\theta i-1}}{4r\sin\theta\Delta r\Delta\phi} + \frac{A_\theta}{2r^2\sin\theta\Delta\phi} - \frac{A_{rj+1} - A_{rj-1}}{4r^2\sin\theta\Delta\theta\Delta\phi}$$

$$+ \frac{V_\phi}{2r\sin\theta\Delta\phi}\Bigg)$$

$$- w_{\phi k-1}\Bigg(\frac{A_{\theta i+1} - A_{\theta i-1}}{4r\sin\theta\Delta r\Delta\phi} + \frac{A_\theta}{2r^2\sin\theta\Delta\phi} - \frac{A_{rj+1} - A_{rj-1}}{4r^2\sin\theta\Delta\theta\Delta\phi}$$

$$+ \frac{V_\phi}{2r\sin\theta\Delta\phi}\Bigg)$$

$$+ w_{ri+1k+1}\left(\frac{1}{4Re\,r\sin\theta\Delta r\Delta\phi}\right) - w_{ri-1k+1}\left(\frac{1}{4Re\,r\sin\theta\Delta r\Delta\phi}\right)$$

$$- w_{ri+1k-1}\left(\frac{1}{4Re\,r\sin\theta\Delta r\Delta\phi}\right) + w_{ri-1k-1}\left(\frac{1}{4Re\,r\sin\theta\Delta r\Delta\phi}\right)$$

$$+ w_{\theta j+1k+1}\left(\frac{1}{4\,Re\,r^2\sin\theta\Delta\theta\Delta\phi}\right) - w_{\theta j-1k+1}\left(\frac{1}{4\,Re\,r^2\sin\theta\Delta\theta\Delta\phi}\right)$$

→

$$- w_{\theta j+1k-1}\left(\frac{1}{4Re\,r^2\sin\theta\,\Delta\theta\Delta\phi}\right) + w_{\theta j-1k-1}\left(\frac{1}{4Re\,r^2\sin\theta\,\Delta\theta\Delta\phi}\right)$$

while, on the polar axis, we have:-

$$\frac{1}{\Delta t} w_r^n = w_r \left(\frac{1}{\Delta t} + \frac{A_{\phi j+1} - A_{\phi j-1}}{r^2\Delta\theta} + \frac{2}{r} V_r \right.$$

$$\mp \frac{A_{\theta j+1k+1} - A_{\theta j-1k+1} - A_{\theta j+1k-1} + A_{\theta j-1k-1}}{4r^2\Delta\theta\Delta\phi}$$

$$- \frac{A_{\phi i+1\,j+1} - A_{\phi i-1\,j+1} - A_{\phi i+1\,j-1} + A_{\phi i-1\,j-1}}{2r\Delta r\Delta\theta}$$

$$+ \frac{2(h_{j+1} - 2h + h_{j-1})}{r^2\Delta\theta^2} \mp \frac{A_{rk+1} - A_{rk-1}}{r^2\Delta\theta^2\Delta\phi}$$

$$\pm \frac{\begin{array}{l}(A_{\theta i+1\,j+1k+1} - A_{\theta i-1\,j+1k+1} - A_{\theta i+1\,j-1k+1} + A_{\theta i-1\,j-1k+1} \\ - A_{\theta i+1\,j+1k-1} + A_{\theta i-1\,j+1k-1} + A_{\theta i+1\,j-1k-1} - A_{\theta i-1\,j-1k-1})\end{array}}{8r\Delta r\Delta\theta\Delta\phi}$$

$$\left. + \frac{1}{Re}\left\{ \frac{4}{r^2\Delta\theta^2} - \frac{2}{r^2\Delta\theta^2\Delta\phi^2} \right\} \right)$$

$$+ w_\theta \left(-\frac{3}{2}\,\frac{A_{\phi j+1} - 2A_\phi + A_{\phi j-1}}{r^2\Delta\theta^2} \pm \frac{A_{\theta k+1} - A_{\theta k-1}}{2r^2\Delta\theta^2\Delta\phi} - \frac{A_\phi}{r^2\Delta\theta^2} \right.$$

$$\pm \frac{\begin{array}{l}(A_{\theta j+1k+1} - 2A_{\theta k+1} + A_{\theta j-1k+1} - A_{\theta j+1k-1} + 2A_{\theta k-1} \\ - A_{\theta j-1k-1})\end{array}}{4r^2\Delta\theta^2\Delta\phi}$$

$$\left. - \frac{h_{i+1\,j+1} - h_{i-1\,j+1} - h_{i+1\,j-1} + h_{i-1\,j-1}}{4r\Delta r\Delta\theta} \right)$$

$$+ w_\phi \left(\mp \frac{3}{4}\,\frac{\begin{array}{l}(A_{\phi j+1k+1} - 2A_{\phi k+1} + A_{\phi j-1k+1} - A_{\phi j+1k-1} + 2A_{\phi k-1} \\ - A_{\phi j-1k-1})\end{array}}{r^2\Delta\theta^2\Delta\phi} \right.$$

$$+ \frac{\begin{array}{l}(A_{\theta j+1k+1} - 2A_{\theta k+1} + A_{\theta j-1k+1} - 2A_{\theta j+1} + 4A_\theta - 2A_{\theta j-1} \\ + A_{\theta j+1k-1} - 2A_{\theta k-1} + A_{\theta j-1k-1})\end{array}}{2r^2\Delta\theta^2\Delta\phi^2}$$

$$\left. \pm \frac{A_{\phi k+1} - A_{\phi k-1}}{2r^2\Delta\theta^2\Delta\phi} - \frac{A_{\theta k+1} - 2A_\theta + A_{\theta k-1}}{r^2\Delta\theta^2\Delta\phi^2} \right) \longrightarrow$$

$$+ w_{ri+1}\left(\frac{A_{\phi j+1} - A_{\phi j-1}}{2r\Delta r\Delta\theta} + \frac{V_r}{2\Delta r} \right.$$

$$\left. \mp \frac{A_{\theta j+1k+1} - A_{\theta j-1k+1} - A_{\theta j+1k-1} + A_{\theta j-1k-1}}{8r\Delta r\Delta\theta\,\Delta\phi} \right)$$

$$- w_{ri-1}\left(\frac{A_{\phi j+1} - A_{\phi j-1}}{2r\Delta r\Delta\theta} + \frac{V_r}{2\Delta r} \right.$$

$$\left. \mp \frac{A_{\theta j+1k+1} - A_{\theta j-1k+1} - A_{\theta j+1k-1} + A_{\theta j-1k-1}}{8r\Delta r\Delta\theta\Delta\phi} \right)$$

$$+ w_{rj+1}\left(\pm \frac{A_{rj+1k+1} - A_{rj-1k+1} - A_{rj+1k-1} + A_{rj-1k-1}}{8r^2\Delta\theta^2\Delta\phi} + \frac{V_\theta}{r\Delta\theta} \right.$$

$$- \frac{A_{\phi i+1} - A_{\phi i-1}}{2r\Delta r\Delta\theta} \mp \frac{A_{\theta k+1} - A_{\theta k-1}}{4r^2\Delta\theta\Delta\phi} \pm \frac{A_{rk+1} - A_{rk-1}}{2r^2\Delta\theta^2\Delta\phi}$$

$$\pm \frac{A_{\theta i+1k+1} - A_{\theta i-1k+1} - A_{\theta i+1k-1} + A_{\theta i-1k-1}}{8r\Delta r\Delta\theta\,\Delta\phi}$$

$$\left. + \frac{1}{Re}\left\{ \frac{-2}{r^2\Delta\theta^2} + \frac{1}{r^2\Delta\theta^2\Delta\phi^2} \right\} \right)$$

$$- w_{rj-1}\left(\pm \frac{A_{rj+1k+1} - A_{rj-1k+1} - A_{rj+1k-1} + A_{rj-1k-1}}{8r^2\Delta\theta^2\Delta\phi} + \frac{V_\theta}{r\Delta\theta} \right.$$

$$- \frac{A_{\phi i+1} - A_{\phi i-1}}{2r\Delta r\Delta\theta} \mp \frac{A_{\theta k+1} - A_{\theta k-1}}{4r^2\Delta\theta\Delta\phi} \mp \frac{A_{rk+1} - A_{rk-1}}{2r^2\Delta\theta^2\,\Delta\phi}$$

$$\pm \frac{A_{\theta i+1k+1} - A_{\theta i-1k+1} - A_{\theta i+1k-1} + A_{\theta i-1k-1}}{8r\Delta r\Delta\theta\Delta\phi}$$

$$\left. + \frac{1}{Re}\left\{ \frac{2}{r^2\Delta\theta^2} - \frac{1}{r^2\Delta\theta^2\Delta\phi^2} \right\} \right)$$

$$+ w_{\theta j+1}\left(- \frac{A_{\phi j+1} - A_{\phi j-1}}{4r^2\Delta\theta^2} + \frac{A_\phi}{2r^2\Delta\theta^2} \mp \frac{A_{\theta k+1} - A_{\theta k-1}}{4r^2\Delta\theta^2\Delta\phi} \right.$$

$$\left. \pm \frac{A_{\theta j+1k+1} - A_{\theta j-1k+1} - A_{\theta j+1k-1} + A_{\theta j-1k-1}}{8r^2\Delta\theta^2\Delta\phi} + \frac{1}{Re\,r^2\Delta\theta} \right)$$

$$- w_{\theta j-1}\left(- \frac{A_{\phi j+1} - A_{\phi j-1}}{4r^2\Delta\theta^2} - \frac{A_\phi}{2r^2\Delta\theta^2} \pm \frac{A_{\theta k+1} - A_{\theta k-1}}{4r^2\Delta\theta^2\Delta\phi} \right.$$

→

$$\pm \frac{A_{\theta j+1k+1} - A_{\theta j-1k+1} - A_{\theta j+1k-1} + A_{\theta j-1k-1}}{8r^2\Delta\theta^2\Delta\phi} + \frac{1}{Re\, r^2\Delta\theta}\Bigg)$$

$$+ w_{\phi j+1}\Bigg(\mp \frac{A_{\phi j+1k+1} - A_{\phi j-1k+1} - A_{\phi j+1k-1} + A_{\phi j-1k-1}}{8r^2\Delta\theta^2\Delta\phi}$$

$$\mp \frac{A_{\phi k+1} - A_{\phi k-1}}{4r^2\Delta\theta^2\Delta\phi} + \frac{A_{\theta k+1} - 2A_{\theta} + A_{\theta k-1}}{2r^2\Delta\theta^2\Delta\phi^2}\Bigg)$$

$$- w_{\phi j-1}\Bigg(\mp \frac{A_{\phi j+1k+1} - A_{\phi j-1k+1} - A_{\phi j+1k-1} + A_{\phi j-1k-1}}{8r^2\Delta\theta^2\Delta\phi}$$

$$\pm \frac{A_{\phi k+1} - A_{\phi k-1}}{4r^2\Delta\theta^2\Delta\phi} - \frac{A_{\theta k+1} - 2A_{\theta} + A_{\theta k-1}}{2r^2\Delta\theta^2\Delta\phi^2}\Bigg)$$

$$+ w_{rk+1}\Bigg(\pm \frac{A_{\theta i+1j+1} - A_{\theta i-1j+1} - A_{\theta i+1j-1} + A_{\theta i-1j-1}}{8r\Delta r\Delta\theta\Delta\phi}$$

$$\pm \frac{A_{\theta j+1} - A_{\theta j-1}}{4r^2\Delta\theta\Delta\phi} \mp \frac{A_{rj+1} - 2A_{r} + A_{rj-1}}{2r^2\Delta\theta^2\Delta\phi} + \frac{1}{Re\, r^2\Delta\theta^2\Delta\phi^2}\Bigg)$$

$$- w_{rk-1}\Bigg(\pm \frac{A_{\theta i+1j+1} - A_{\theta i-1j+1} - A_{\theta i+1j-1} + A_{\theta i-1j-1}}{8r\Delta r\Delta\theta\Delta\phi}$$

$$\pm \frac{A_{\theta j+1} - A_{\theta j-1}}{4r^2\Delta\theta\Delta\phi} \mp \frac{A_{rj+1} - 2A_{r} + A_{rj-1}}{2r^2\Delta\theta^2\Delta\phi} - \frac{1}{Re\, r^2\Delta\theta^2\Delta\phi^2}\Bigg)$$

$$+ w_{ri+1j+1}\left(\frac{A_{\phi}}{4r\Delta r\Delta\theta} \mp \frac{A_{\theta k+1} - A_{\theta k-1}}{8r\Delta r\Delta\theta\Delta\phi}\right)$$

$$- w_{ri-1j+1}\left(\frac{A_{\phi}}{4r\Delta r\Delta\theta} \mp \frac{A_{\theta k+1} - A_{\theta k-1}}{8r\Delta r\Delta\theta\Delta\phi}\right)$$

$$- w_{ri+1j-1}\left(\frac{A_{\phi}}{4r\Delta r\Delta\theta} \mp \frac{A_{\theta k+1} - A_{\theta k-1}}{8r\Delta r\Delta\theta\Delta\phi}\right)$$

$$+ w_{ri-1j-1}\left(\frac{A_{\phi}}{4r\Delta r\Delta\theta} \mp \frac{A_{\theta k+1} - A_{\theta k-1}}{8r\Delta r\Delta\theta\Delta\phi}\right)$$

$$+ w_{\theta i+1j+1}\left(\frac{1}{2Re\, r\Delta r\Delta\theta}\right) \quad - w_{\theta i-1j+1}\left(\frac{1}{2Re\, r\Delta r\Delta\theta}\right)$$

$$- w_{\theta i+1j-1}\left(\frac{1}{2Re\, r\Delta r\Delta\theta}\right) \quad + w_{\theta i-1j-1}\left(\frac{1}{2Re\, r\Delta r\Delta\theta}\right)$$

⟶

$$+ \, w_{r\,j+1\,k+1} \left(\pm \frac{A_{\theta i+1} - A_{\theta i-1}}{8r\,\Delta r\,\Delta\theta\,\Delta\phi} \pm \frac{A_\theta}{4r^2\,\Delta\theta\,\Delta\phi} \mp \frac{A_{r\,j+1} - A_{r\,j-1}}{8r^2\,\Delta\theta^2\,\Delta\phi} - \frac{1}{2\,Re\,r^2\,\Delta\theta^2\,\Delta\phi^2} \right)$$

$$- \, w_{r\,j-1\,k+1} \left(\pm \frac{A_{\theta i+1} - A_{\theta i-1}}{8r\,\Delta r\,\Delta\theta\,\Delta\phi} \pm \frac{A_\theta}{4r^2\,\Delta\theta\,\Delta\phi} \mp \frac{A_{r\,j+1} - A_{r\,j-1}}{8r^2\,\Delta\theta^2\,\Delta\phi} + \frac{1}{2\,Re\,r^2\,\Delta\theta^2\,\Delta\phi^2} \right)$$

$$- \, w_{r\,j+1\,k-1} \left(\pm \frac{A_{\theta i+1} - A_{\theta i-1}}{8r\,\Delta r\,\Delta\theta\,\Delta\phi} \pm \frac{A_\theta}{4r^2\,\Delta\theta\,\Delta\phi} \mp \frac{A_{r\,j+1} - A_{r\,j-1}}{8r^2\,\Delta\theta^2\,\Delta\phi} + \frac{1}{2\,Re\,r^2\,\Delta\theta^2\,\Delta\phi^2} \right)$$

$$+ \, w_{r\,j-1\,k-1} \left(\pm \frac{A_{\theta i+1} - A_{\theta i-1}}{8r\,\Delta r\,\Delta\theta\,\Delta\phi} \pm \frac{A_\theta}{4r^2\,\Delta\theta\,\Delta\phi} \mp \frac{A_{r\,j+1} - A_{r\,j-1}}{8r^2\,\Delta\theta^2\,\Delta\phi} - \frac{1}{2\,Re\,r^2\,\Delta\theta^2\,\Delta\phi^2} \right)$$

$$+ \, w_{\phi\,j+1\,k+1} \left(\frac{\pm 1}{4\,Re\,r^2\,\Delta\theta\,\Delta\phi} \right) - w_{\phi\,j-1\,k+1} \left(\frac{\pm 1}{4\,Re\,r^2\,\Delta\theta\,\Delta\phi} \right)$$

$$- \, w_{\phi\,j+1\,k-1} \left(\frac{\pm 1}{4\,Re\,r^2\,\Delta\theta\,\Delta\phi} \right) + w_{\phi\,j-1\,k-1} \left(\frac{\pm 1}{4\,Re\,r^2\,\Delta\theta\,\Delta\phi} \right)$$

$$+ \, w_{\phi\,i+1\,j+1\,k+1} \left(\frac{\pm 1}{8\,Re\,r\,\Delta r\,\Delta\theta\,\Delta\phi} \right) - w_{\phi\,i-1\,j+1\,k+1} \left(\frac{\pm 1}{8\,Re\,r\,\Delta r\,\Delta\theta\,\Delta\phi} \right)$$

$$- \, w_{\phi\,i+1\,j-1\,k+1} \left(\frac{\pm 1}{8\,Re\,r\,\Delta r\,\Delta\theta\,\Delta\phi} \right) + w_{\phi\,i-1\,j-1\,k+1} \left(\frac{\pm 1}{8\,Re\,r\,\Delta r\,\Delta\theta\,\Delta\phi} \right)$$

$$- \, w_{\phi\,i+1\,j+1\,k-1} \left(\frac{\pm 1}{8\,Re\,r\,\Delta r\,\Delta\theta\,\Delta\phi} \right) + w_{\phi\,i-1\,j+1\,k-1} \left(\frac{\pm 1}{8\,Re\,r\,\Delta r\,\Delta\theta\,\Delta\phi} \right)$$

$$+ \, w_{\phi\,i+1\,j-1\,k-1} \left(\frac{\pm 1}{8\,Re\,r\,\Delta r\,\Delta\theta\,\Delta\phi} \right) - w_{\phi\,i-1\,j-1\,k-1} \left(\frac{\pm 1}{8\,Re\,r\,\Delta r\,\Delta\theta\,\Delta\phi} \right)$$

$$\frac{1}{\Delta t}\, w_\theta^n = w_r \left(\pm \frac{A_{r\,j+1\,k+1} - A_{r\,j-1\,k+1} - A_{r\,j+1\,k-1} + A_{r\,j-1\,k-1}}{2r^2\,\Delta\theta\,\Delta\phi} - \frac{2}{r^2}\, A_\phi \right.$$

⟶

$$+ \frac{1}{r} V_\theta + \frac{A_{\phi i+1} - 2A_\phi + A_{\phi i-1}}{\Delta r^2}$$

$$\mp \frac{(A_{r i+1\, j+1 k+1} - A_{r i-1\, j+1 k+1} - A_{r i+1\, j-1 k+1} + A_{r i-1\, j-1 k+1} - A_{r i+1\, j+1 k-1} + A_{r i-1\, j+1 k-1} + A_{r i+1\, j-1 k-1} - A_{r i-1\, j-1 k-1})}{8 r \Delta r \Delta\theta \Delta\phi}$$

$$\left. - \frac{h_{i+1\, j+1} - h_{i-1\, j+1} - h_{i+1\, j-1} + h_{i-1\, j-1}}{4 r \Delta r \Delta\theta} + \frac{h_{j+1} - h_{j-1}}{2 r^2 \Delta\theta} \right)$$

$$+ w_\theta \left(\frac{1}{\Delta t} \mp \frac{(A_{r j+1 k+1} - 2A_{r k+1} + A_{r j-1 k+1} - A_{r j+1 k-1} + 2A_{r k-1} - A_{r j-1 k-1})}{4 r^2 \Delta\theta^2 \Delta\phi} \right.$$

$$+ \frac{1}{r} V_r + \frac{h_{j+1} - 2h + h_{j-1}}{r^2 \Delta\theta^2} - \frac{A_{\phi j+1} - A_{\phi j-1}}{2 r^2 \Delta\theta}$$

$$+ \frac{A_{\phi i+1\, j+1} - A_{\phi i-1\, j+1} - A_{\phi i+1\, j-1} + A_{\phi i-1\, j-1}}{4 r \Delta r \Delta\theta}$$

$$+ \frac{h_{i+1} - 2h + h_{i-1}}{\Delta r^2} \mp \frac{3}{2} \frac{A_{r k+1} - A_{r k-1}}{r^2 \Delta\theta^2 \Delta\phi}$$

$$\pm \frac{A_{\theta j+1 k+1} - A_{\theta j-1 k+1} - A_{\theta j+1 k-1} + A_{\theta j-1 k-1}}{4 r^2 \Delta\theta \Delta\phi}$$

$$\left. + \frac{1}{Re} \left\{ \frac{-2}{r^2 \Delta\theta^2 \Delta\phi^2} + \frac{2}{\Delta r^2} \right\} \right)$$

$$+ w_\phi \left(- \frac{(A_{r j+1 k+1} - 2A_{r k+1} + A_{r j-1 k+1} - 2A_{r j+1} + 4A_r - 2A_{r j-1} + A_{r j+1 k-1} - 2A_{r k-1} + A_{r j-1 k-1})}{2 r^2 \Delta\theta^2 \Delta\phi^2} \right.$$

$$\pm \frac{(A_{\phi i+1\, j+1 k+1} - A_{\phi i-1\, j+1 k+1} - A_{\phi i+1\, j-1 k+1} + A_{\phi i-1\, j-1 k+1} - A_{\phi i+1\, j+1 k-1} + A_{\phi i-1\, j+1 k-1} + A_{\phi i+1\, j-1 k-1} - A_{\phi i-1\, j-1 k-1})}{8 r \Delta r \Delta\theta \Delta\phi}$$

$$+ \frac{A_{r k+1} - 2A_r + A_{r k-1}}{r^2 \Delta\theta^2 \Delta\phi^2}$$

$$\left. \pm \frac{A_{\phi j+1 k+1} - A_{\phi j-1 k+1} - A_{\phi j+1 k-1} + A_{\phi j-1 k-1}}{4 r^2 \Delta\theta \Delta\phi} \right)$$

⟶

$$+ \omega_{\theta i+1} \left(\frac{A_{\phi j+1} - A_{\phi j-1}}{2r\Delta r\Delta\theta} + \frac{V_r}{2\Delta r} \right.$$

$$\mp \frac{A_{\theta j+1k+1} - A_{\theta j-1k+1} - A_{\theta j+1k-1} + A_{\theta j-1k-1}}{8r\Delta r\Delta\theta\Delta\phi}$$

$$\left. + \frac{1}{Re}\left\{\frac{-1}{r\Delta r} - \frac{1}{\Delta r^2}\right\} \right)$$

$$- \omega_{\theta i-1} \left(\frac{A_{\phi j+1} - A_{\phi j-1}}{2r\Delta r\Delta\theta} + \frac{V_r}{2\Delta r} \right.$$

$$\mp \frac{A_{\theta j+1k+1} - A_{\theta j-1k+1} - A_{\theta j+1k-1} + A_{\theta j-1k-1}}{8r\Delta r\Delta\theta\Delta\phi}$$

$$\left. + \frac{1}{Re}\left\{\frac{-1}{r\Delta r} + \frac{1}{\Delta r^2}\right\} \right)$$

$$+ \omega_{r j+1} \left(\pm \frac{A_{rk+1} - A_{rk-1}}{2r^2\Delta\theta\Delta\phi} \right.$$

$$\left. \mp \frac{A_{ri+1k+1} - A_{ri-1k+1} - A_{ri+1k-1} + A_{ri-1k-1}}{8r\Delta r\Delta\theta\Delta\phi} \right)$$

$$- \omega_{r j-1} \left(\pm \frac{A_{rk+1} - A_{rk-1}}{2r^2\Delta\theta\Delta\phi} \right.$$

$$\left. \mp \frac{A_{ri+1k+1} - A_{ri-1k+1} - A_{ri+1k-1} + A_{ri-1k-1}}{8r\Delta r\Delta\theta\Delta\phi} \right)$$

$$+ \omega_{\theta j+1} \left(- \frac{A_{\phi i+1} - A_{\phi i-1}}{4r\Delta r\Delta\theta} - \frac{A_\phi}{r^2\Delta\theta} + \frac{V_\theta}{r\Delta\theta} \pm \frac{A_{\theta k+1} - A_{\theta k-1}}{4r^2\Delta\theta\Delta\phi} \right.$$

$$\left. \pm \frac{3}{4}\frac{A_{rk+1} - A_{rk-1}}{r^2\Delta\theta^2\Delta\phi} + \frac{1}{Re\, r^2\Delta\theta^2\Delta\phi^2} \right)$$

$$- \omega_{\theta j-1} \left(- \frac{A_{\phi i+1} - A_{\phi i-1}}{4r\Delta r\Delta\theta} - \frac{A_\phi}{r^2\Delta\theta} + \frac{V_\theta}{r\Delta\theta} \pm \frac{A_{\theta k+1} - A_{\theta k-1}}{4r^2\Delta\theta\Delta\phi} \right.$$

$$\left. \mp \frac{3}{4}\frac{A_{rk+1} - A_{rk-1}}{r^2\Delta\theta^2\Delta\phi} - \frac{1}{Re\, r^2\Delta\theta^2\Delta\phi^2} \right)$$

$$+ \omega_{\phi j+1} \left(\pm \frac{A_{\phi i+1k+1} - A_{\phi i-1k+1} - A_{\phi i+1k-1} + A_{\phi i-1k-1}}{8r\Delta r\Delta\theta\Delta\phi} \right.$$

→

$$\pm \frac{A_{\phi k+1} - A_{\phi k-1}}{4r^2\Delta\theta\Delta\phi} - \frac{A_{rk+1} - 2A_r + A_{rk-1}}{2r^2\Delta\theta^2\Delta\phi^2}\Big)$$

$$- \omega_{\phi j-1}\Big(\pm \frac{A_{\phi i+1k+1} - A_{\phi i-1k+1} - A_{\phi i+1k-1} + A_{\phi i-1k-1}}{8r\Delta r\Delta\theta\Delta\phi}$$

$$\pm \frac{A_{\phi k+1} - A_{\phi k-1}}{4r^2\Delta\theta\Delta\phi} + \frac{A_{rk+1} - 2A_r + A_{rk-1}}{2r^2\Delta\theta^2\Delta\phi^2}\Big)$$

$$+ \omega_{\theta k+1}\Big(\pm \frac{A_{\theta i+1 j+1} - A_{\theta i-1 j+1} - A_{\theta i+1 j-1} + A_{\theta i-1 j-1}}{8r\Delta r\Delta\theta\Delta\phi}$$

$$\pm \frac{A_{\theta j+1} - A_{\theta j-1}}{4r^2\Delta\theta\Delta\phi} \mp \frac{A_{rj+1} - 2A_r + A_{rj-1}}{2r^2\Delta\theta^2\Delta\phi} + \frac{1}{Re\, r^2\Delta\theta^2\Delta\phi^2}\Big)$$

$$- \omega_{\theta k-1}\Big(\pm \frac{A_{\theta i+1 j+1} - A_{\theta i-1 j+1} - A_{\theta i+1 j-1} + A_{\theta i-1 j-1}}{8r\Delta r\Delta\theta\Delta\phi}$$

$$\pm \frac{A_{\theta j+1} - A_{\theta j-1}}{4r^2\Delta\theta\Delta\phi} \mp \frac{A_{rj+1} - 2A_r + A_{rj-1}}{2r^2\Delta\theta^2\Delta\phi} - \frac{1}{Re\, r^2\Delta\theta^2\Delta\phi^2}\Big)$$

$$+ \omega_{\phi k+1}\Big(\mp \frac{3}{2Re\, r^2\Delta\theta^2\Delta\phi}\Big) - \omega_{\phi k-1}\Big(\mp \frac{3}{2Re\, r^2\Delta\theta^2\Delta\phi}\Big)$$

$$+ \omega_{ri+1\,j+1}\Big(\frac{1}{4Re\, r\Delta r\Delta\theta}\Big) - \omega_{ri-1\,j+1}\Big(\frac{1}{4Re\, r\Delta r\Delta\theta}\Big)$$

$$- \omega_{ri+1\,j-1}\Big(\frac{1}{4Re\, r\Delta r\Delta\theta}\Big) + \omega_{ri-1\,j-1}\Big(\frac{1}{4Re\, r\Delta r\Delta\theta}\Big)$$

$$+ \omega_{\theta i+1\,j+1}\Big(\frac{A_\phi}{4r\Delta r\Delta\theta} \mp \frac{A_{\theta k+1} - A_{\theta k-1}}{8r\Delta r\Delta\theta\Delta\phi}\Big)$$

$$- \omega_{\theta i-1\,j+1}\Big(\frac{A_\phi}{4r\Delta r\Delta\theta} \mp \frac{A_{\theta k+1} - A_{\theta k-1}}{8r\Delta r\Delta\theta\Delta\phi}\Big)$$

$$- \omega_{\theta i+1\,j-1}\Big(\frac{A_\phi}{4r\Delta r\Delta\theta} \mp \frac{A_{\theta k+1} - A_{\theta k-1}}{8r\Delta r\Delta\theta\Delta\phi}\Big)$$

$$+ \omega_{\theta i-1\,j-1}\Big(\frac{A_\phi}{4r\Delta r\Delta\theta} \mp \frac{A_{\theta k+1} - A_{\theta k-1}}{8r\Delta r\Delta\theta\Delta\phi}\Big)$$

$$+ \omega_{\theta j+1k+1}\Big(\pm \frac{A_{\theta i+1} - A_{\theta i-1}}{8r\Delta r\Delta\theta\Delta\phi} \pm \frac{A_\theta}{4r^2\Delta\theta\Delta\phi} \mp \frac{A_{rj+1} - A_{rj-1}}{8r^2\Delta\theta^2\Delta\phi}$$

→

$$- \frac{1}{2Re\,r^2 \Delta\theta^2 \Delta\phi} \Bigg)$$

$$- \omega_{\theta\, j-1k+1} \left(\pm \frac{A_{\theta i+1} - A_{\theta i-1}}{8r \Delta r \Delta\theta \Delta\phi} \pm \frac{A_\theta}{4r^2 \Delta\theta\Delta\phi} \mp \frac{A_{r j+1} - A_{r j-1}}{8r^2 \Delta\theta^2 \Delta\phi} \right.$$

$$\left. + \frac{1}{2Re\,r^2 \Delta\theta^2 \Delta\phi} \right)$$

$$- \omega_{\theta j+1k-1} \left(\pm \frac{A_{\theta i+1} - A_{\theta i-1}}{8r \Delta r \Delta\theta \Delta\phi} \pm \frac{A_\theta}{4r^2 \Delta\theta\Delta\phi} \mp \frac{A_{r j+1} - A_{r j-1}}{8r^2 \Delta\theta^2 \Delta\phi} \right.$$

$$\left. + \frac{1}{2Re\,r^2 \Delta\theta^2 \Delta\phi} \right)$$

$$+ \omega_{\theta j-1k-1} \left(\pm \frac{A_{\theta i+1} - A_{\theta i-1}}{8r \Delta r \Delta\theta \Delta\phi} \pm \frac{A_\theta}{4r^2 \Delta\theta\Delta\phi} \mp \frac{A_{r j+1} - A_{r j-1}}{8r^2 \Delta\theta^2 \Delta\phi} \right.$$

$$\left. - \frac{1}{2Re\,r^2 \Delta\theta^2 \Delta\phi} \right)$$

$$+ \omega_{\phi j+1k+1} \left(\pm \frac{3}{4Re\,r^2 \Delta\theta^2 \Delta\phi} \right) - \omega_{\phi j-1k+1} \left(\mp \frac{3}{4Re\,r^2 \Delta\theta^2 \Delta\phi} \right)$$

$$- \omega_{\phi j+1k-1} \left(\pm \frac{3}{4Re\,r^2 \Delta\theta^2 \Delta\phi} \right) + \omega_{\phi j-1k-1} \left(\mp \frac{3}{4Re\,r^2 \Delta\theta^2 \Delta\phi} \right)$$

$$\frac{1}{\Delta t} \omega_\phi^n = \omega_r \left(\frac{2}{r^2} A_\theta - \frac{A_{r j+1} - A_{r j-1}}{r^2 \Delta\theta} - \frac{A_{\theta i+1} - 2A_\theta + A_{\theta i-1}}{\Delta r^2} \right.$$

$$\left. + \frac{A_{r i+1\, j+1} - A_{r i-1\, j+1} - A_{r i+1\, j-1} + A_{r i-1\, j-1}}{4r \Delta r \Delta\theta} \right)$$

$$+ \omega_\phi \left(\frac{1}{\Delta t} + \frac{1}{r} V_r + \frac{h_{i+1} - 2h + h_{i-1}}{\Delta r^2} \right.$$

$$+ \frac{A_{\phi i+1\, j+1} - A_{\phi i-1\, j+1} - A_{\phi i+1\, j-1} + A_{\phi i-1\, j-1}}{4r \Delta r \Delta\theta}$$

$$\mp \frac{\begin{array}{l}(A_{\theta i+1\, j+1k+1} - A_{\theta i-1\, j+1k+1} - A_{\theta i+1\, j-1k+1} + A_{\theta i-1\, j-1k+1} \\ - A_{\theta i+1\, j+1k-1} + A_{\theta i-1\, j+1k-1} + A_{\theta i+1\, j-1k-1} - A_{\theta i-1\, j-1k-1})\end{array}}{8r \Delta r \Delta\theta \Delta\phi}$$

$$\pm \frac{A_{r j+1k+1} - 2A_{rk+1} + A_{r j-1k+1} - A_{r j+1k-1} + 2A_{rk-1} - A_{r j-1k-1}}{4r^2 \Delta\theta^2 \Delta\phi}$$

→

$$- \frac{A_{\phi j+1} - A_{\phi j-1}}{2r^2 \Delta\theta} + \frac{h_{j+1} - 2h + h_{j-1}}{r^2 \Delta\theta^2} \mp \frac{A_{rk+1} - A_{rk-1}}{2r^2 \Delta\theta^2 \Delta\phi}$$

$$+ \frac{1}{Re}\left\{\frac{2}{\Delta r^2} + \frac{3}{r^2 \Delta\theta^2}\right\}\Bigg)$$

$$+ w_{\phi i+1}\Bigg(\frac{A_{\phi j+1} - A_{\phi j-1}}{2r\Delta r \Delta\theta} + \frac{V_r}{2\Delta r}$$

$$\mp \frac{A_{\theta j+1k+1} - A_{\theta j-1k+1} - A_{\theta j+1k-1} + A_{\theta j-1k-1}}{8r\Delta r \Delta\theta \Delta\phi}$$

$$+ \frac{1}{Re}\left\{\frac{-1}{r\Delta r} - \frac{1}{\Delta r^2}\right\}\Bigg)$$

$$- w_{\phi i-1}\Bigg(\frac{A_{\phi j+1} - A_{\phi j-1}}{2r\Delta r \Delta\theta} + \frac{V_r}{2\Delta r}$$

$$\mp \frac{A_{\theta j+1k+1} - A_{\theta j-1k+1} - A_{\theta j+1k-1} + A_{\theta j-1k-1}}{8r\Delta r \Delta\theta \Delta\phi}$$

$$+ \frac{1}{Re}\left\{\frac{-1}{r\Delta r} + \frac{1}{\Delta r^2}\right\}\Bigg)$$

$$+ w_{\theta j+1}\left(\frac{A_{\theta i+1} - A_{\theta i-1}}{4r\Delta r \Delta\theta} + \frac{A_\theta}{2r^2\Delta\theta} - \frac{A_{rj+1} - A_{rj-1}}{4r^2\Delta\theta^2}\right)$$

$$- w_{\theta j-1}\left(\frac{A_{\theta i+1} - A_{\theta i-1}}{4r\Delta r \Delta\theta} + \frac{A_\theta}{2r^2\Delta\theta} - \frac{A_{rj+1} - A_{rj-1}}{4r^2\Delta\theta^2}\right)$$

$$+ w_{\phi j+1}\Bigg(\pm \frac{A_{rj+1k+1} - A_{rj-1k+1} - A_{rj+1k-1} + A_{rj-1k-1}}{4r^2 \Delta\theta^2 \Delta\phi} - \frac{A_\phi}{2r^2\Delta\theta}$$

$$+ \frac{V_\theta}{2r\Delta\theta} \mp \frac{A_{\theta i+1k+1} - A_{\theta i-1k+1} - A_{\theta i+1k-1} + A_{\theta i-1k-1}}{8r\Delta r \Delta\theta \Delta\phi}$$

$$\pm \frac{A_{rk+1} - A_{rk-1}}{4r^2\Delta\theta^2\Delta\phi} - \frac{3}{2Re\, r^2\Delta\theta^2}\Bigg)$$

$$- w_{\phi j-1}\Bigg(\pm \frac{A_{rj+1k+1} - A_{rj-1k+1} - A_{rj+1k-1} + A_{rj-1k-1}}{4r^2 \Delta\theta^2 \Delta\phi} - \frac{A_\phi}{2r^2\Delta\theta}$$

$$+ \frac{V_\theta}{2r\Delta\theta} \mp \frac{A_{\theta i+1k+1} - A_{\theta i-1k+1} - A_{\theta i+1k-1} + A_{\theta i-1k-1}}{8r\Delta r \Delta\theta \Delta\phi}$$

→

$$\mp \frac{A_{rk+1} - A_{rk-1}}{4r^2\Delta\theta^2\Delta\phi} + \frac{3}{2Re\, r^2\Delta\theta^2} \Bigg)$$

$$+ w_{\theta k+1}\left(\mp \frac{1}{2Re r^2\Delta\theta^2\Delta\phi}\right) - w_{\theta k-1}\left(\mp \frac{1}{2Re r^2\Delta\theta^2\Delta\phi}\right)$$

$$+ w_{\phi k+1}\Bigg(\pm \frac{A_{\theta i+1\, j+1} - A_{\theta i-1\, j+1} - A_{\theta i+1\, j-1} + A_{\theta i-1\, j-1}}{8r\Delta r\Delta\theta\Delta\phi}$$

$$\pm \frac{A_{\theta j+1} - A_{\theta j-1}}{4r^2\Delta\theta\Delta\phi} \mp \frac{A_{r j+1} - 2A_r + A_{r j-1}}{2r^2\Delta\theta^2\Delta\phi}\Bigg)$$

$$- w_{\phi k-1}\Bigg(\pm \frac{A_{\theta i+1\, j+1} - A_{\theta i-1\, j+1} - A_{\theta i+1\, j-1} + A_{\theta i-1\, j-1}}{8r\Delta r\Delta\theta\Delta\phi}$$

$$\pm \frac{A_{\theta j+1} - A_{\theta j-1}}{4r^2\Delta\theta\Delta\phi} \mp \frac{A_{r j+1} - 2A_r + A_{r j-1}}{2r^2\Delta\theta^2\Delta\phi}\Bigg)$$

$$+ w_{\phi i+1\, j+1}\left(\frac{A_\phi}{4r\Delta r\Delta\theta} \mp \frac{A_{\theta k+1} - A_{\theta k-1}}{8r\Delta r\Delta\theta\Delta\phi}\right)$$

$$- w_{\phi i-1\, j+1}\left(\frac{A_\phi}{4r\Delta r\Delta\theta} \mp \frac{A_{\theta k+1} - A_{\theta k-1}}{8r\Delta r\Delta\theta\Delta\phi}\right)$$

$$- w_{\phi i+1\, j-1}\left(\frac{A_\phi}{4r\Delta r\Delta\theta} \mp \frac{A_{\theta k+1} - A_{\theta k-1}}{8r\Delta r\Delta\theta\Delta\phi}\right)$$

$$+ w_{\phi i-1\, j-1}\left(\frac{A_\phi}{4r\Delta r\Delta\theta} \mp \frac{A_{\theta k+1} - A_{\theta k-1}}{8r\Delta r\Delta\theta\Delta\phi}\right)$$

$$+ w_{\theta j+1 k+1}\left(\pm \frac{1}{4Re r^2\Delta\theta^2\Delta\phi}\right) - w_{\theta j-1 k+1}\left(\mp \frac{1}{4Re r^2\Delta\theta^2\Delta\phi}\right)$$

$$- w_{\theta j+1 k-1}\left(\pm \frac{1}{4Re r^2\Delta\theta^2\Delta\phi}\right) + w_{\theta j-1 k-1}\left(\mp \frac{1}{4Re r^2\Delta\theta^2\Delta\phi}\right)$$

$$+ w_{\phi j+1 k+1}\left(\pm \frac{A_{\theta i+1} - A_{\theta i-1}}{8r\Delta r\Delta\theta\Delta\phi} \pm \frac{A_\theta}{4r^2\Delta\theta\Delta\phi} \mp \frac{A_{r j+1} - A_{r j-1}}{8r^2\Delta\theta^2\Delta\phi}\right)$$

$$- w_{\phi j-1 k+1}\left(\pm \frac{A_{\theta i+1} - A_{\theta i-1}}{8r\Delta r\Delta\theta\Delta\phi} \pm \frac{A_\theta}{4r^2\Delta\theta\Delta\phi} \mp \frac{A_{r j+1} - A_{r j-1}}{8r^2\Delta\theta^2\Delta\phi}\right)$$

$$- w_{\phi j+1 k-1}\left(\pm \frac{A_{\theta i+1} - A_{\theta i-1}}{8r\Delta r\Delta\theta\Delta\phi} \pm \frac{A_\theta}{4r^2\Delta\theta\Delta\phi} \mp \frac{A_{r j+1} - A_{r j-1}}{8r^2\Delta\theta^2\Delta\phi}\right)$$

→

$$+ w_{\phi j-1k-1}\left(\pm \frac{A_{\theta i+1} - A_{\theta i-1}}{8r\Delta r\Delta\theta\Delta\phi} \pm \frac{A_\theta}{4r^2\Delta\theta\Delta\phi} \mp \frac{A_{rj+1} - A_{rj-1}}{8r^2\Delta\theta^2\Delta\phi}\right)$$

$$+ w_{ri+1\,j+1k+1}\left(\pm \frac{1}{8Re\,r\Delta r\Delta\theta\Delta\phi}\right) - w_{ri-1\,j+1k+1}\left(\pm \frac{1}{8Re\,r\Delta r\Delta\theta\Delta\phi}\right)$$

$$- w_{ri+1\,j-1k+1}\left(\pm \frac{1}{8Re\,r\Delta r\Delta\theta\Delta\phi}\right) + w_{ri-1\,j-1k+1}\left(\pm \frac{1}{8Re\,r\Delta r\Delta\theta\Delta\phi}\right)$$

$$- w_{ri+1\,j+1k-1}\left(\pm \frac{1}{8Re\,r\Delta r\Delta\theta\Delta\phi}\right) + w_{ri-1\,j+1k-1}\left(\pm \frac{1}{8Re\,r\Delta r\Delta\theta\Delta\phi}\right)$$

$$+ w_{ri+1\,j-1k-1}\left(\pm \frac{1}{8Re\,r\Delta r\Delta\theta\Delta\phi}\right) - w_{ri-1\,j-1k-1}\left(\pm \frac{1}{8Re\,r\Delta r\Delta\theta\Delta\phi}\right)$$

The boundary conditions are:-

$$\left.\begin{aligned} w_r &\equiv 0 \\ w_\theta &= \frac{-1}{2\Delta r}\left(-3v_\phi + 4v_{\phi i+1} - v_{\phi i+2}\right) + \frac{V_\phi}{r_o} \\ w_\phi &= \frac{1}{2\Delta r}\left(-3v_\theta + 4v_{\theta i+1} - v_{\theta i+2}\right) - \frac{V_\theta}{r_o} \end{aligned}\right\}\text{at } i = 1$$

$$\left.\begin{aligned} w_r &= -\sigma\sin\theta\sin\phi \\ w_\theta &= -\sigma\cos\theta\sin\phi \\ w_\phi &= -\sigma\cos\phi \end{aligned}\right\}\text{at } i = n_r+1$$

(note that the no-slip condition at i = 1 means that $v_\theta = -V_\theta$ and $v_\phi = -V_\phi$, so we can simplify the above expressions), while the boundary conditions on the polar axis at $i = n_r+1$ are:-

$$w_r \equiv 0$$

$$w_\theta \equiv 0$$

$$w_\phi = -\sigma$$

Appendix 4 – Computer Program Listing

In this appendix, we present a listing of the program written to solve the finite-difference equations of Appendix 3. The main programs and associated subroutines which form the overall program are listed in alphabetical order as follows:-

Programs	UROT,
	UROTA,
and	UTRANS.
Subroutines	AUTHOR,
	COEFF,
	MATMA,
	MATMA1-8,
	MATRIX,
	MATRJ,
	MATRJ1-7,
	MATRY,
	MATRY1-9,
	PRESS,
	SCRIBE,
	SOR,
	SORJMA,
	SORY,
	SPEED,
	START1-2,
	TYPIST,
	VELOX,
	VORBC1-2,
and	WRITER.

```
      PROGRAM UROT(DATA,OUTPUT,DATB,TAPE5=DATA,TAPE6=OUTPUT,
     1TAPE7=DATB)
      DIMENSION VORTEX(168), A(168), ANEW(168), ASTAR(112)
      DIMENSION DIFF1(112), DIFF2(112)
      DIMENSION V(168), U(168), H(56), VORNEW(168)
      COMMON/COMAUT/N1,N2,N3,NODES,RI,RO,RE,SHEAR,W1,DELT1,W,
     1DELTY,DELT
      COMMON/COMWRI/IOPT1,IOPT2,IOPT3,M1,DIFFJM,REST1,M,DIFFY,RESTY,
     1DIFFCE,TIME,NSTEP,NODE3
      REAL JL, JD, JU, MAL, MAD, MAU
      COMMON YL(112,20), YD(112), YU(112,20), NYL(112,20), NYU(112,20)
      COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84,27)
      COMMON MAL(84,27), MAD(84), MAU(84,27)
C ------ *       *       *       *       *       *       *       *       *
C        THIS PROGRAM DETERMINES THE ROTATIONAL COMPONENT OF
C        THE VELOCITY FIELD, TOGETHER WITH THE VORTICITY AND
C        VECTOR POTENTIAL FIELDS, FOR UNIFORM AND LINEAR SHEAR
C        FLOW PAST A SPHERE.
C        THIS IS PERFORMED BY SOLVING THE VORTICITY TRANSPORT
C        AND VECTOR POTENTIAL POISSON EQUATIONS NUMERICALLY.
C
C ------ OVERALL PROBLEM PARAMETERS
C        SET NUMBER OF RADIAL SPACINGS              (GE 3)
      N1 = 3
C        SET NUMBER OF POLAR ANGLE SPACINGS         (GE 3)
      N2 = 4
C        SET NUMBER OF AZIMUTHAL ANGLE SPACINGS (GE 4)
C        THIS NUMBER MUST BE EXACTLY DIVISIBLE BY FOUR
      N3 = 4
C        SET INNER RADIUS
      RI = 0.5
C        SET OUTER RADIUS
      RO = 1.1
C        SET NUMBER OF NODES ( = (N1 + 1)*((N3*(N2 - 1)) + 2) )
      NODES = 56
      NODE3 = 3*NODES
      NALPHA = NODES/(N1+1)
C        SET REYNOLDS NUMBER
      RE = 1.0
C        SET MAGNITUDE OF SHEAR AT OUTER BOUNDARY
      SHEAR = 0.0
C        SET LENGTH OF TIME-STEP
      DELTAT = 0.002
C        SET NUMBER OF TIME-STEPS
      NSTEPS = 40
C        SET OUTPUT SELECTION PARAMETERS IOPT1-3 AS
C        FOLLOWS - THE VORTICITY FIELD IS WRITTEN OUT
C                    UNLESS IOPT1 = 0.
C                  - THE VECTOR POTENTIAL FIELD IS
C                    WRITTEN OUT UNLESS IOPT2 = 0.
C                  - THE ROTATIONAL COMPONENT OF THE
C                    VELOCITY FIELD IS WRITTEN OUT
C                    UNLESS IOPT3 = 0.
      IOPT1 = 0
      IOPT2 = 0
      IOPT3 = 1
C ------ VORTICITY TRANSPORT EQUATION PARAMETERS
C        SET NUMBER OF UNKNOWNS VORTEX
C        ( = 3*(N1 - 1)*((N3*(N2 - 1) + 2)) )
C        (REMEMBER TO SET UNLABELLED COMMONS ACCORDINGLY)
      NEQN = 84
C        SET RELAXATION FACTOR
      W1 = 1.0
```

```
C         SET MAXIMUM TOLERABLE NUMBER OF ITERATIONS
      MMAX1 = 20
C         SET MAXIMUM TOLERABLE SQUARE OF THE DIFFERENCE BETWEEN
C         ELEMENTS IN SUCCESSIVE ITERATIONS FOR CONVERGENCE
      DELT1 = 0.000000001
C ------ VECTOR POTENTIAL POISSON EQUATION PARAMETERS
C         SET NUMBER OF UNKNOWNS A ( = ((3*N1) - 1)*((N3*(N2 - 1)) + 2)
C         (REMEMBER TO SET UNLABELLED COMMONS ACCORDINGLY)
      NEQNY = 112
C         SET RELAXATION FACTOR
      W = 1.0
C         SET MAXIMUM TOLERABLE NUMBER OF ITERATIONS
      MMAX = 100
C         SET MAXIMUM TOLERABLE SQUARE OF THE DIFFERENCE BETWEEN
C         ELEMENTS IN SUCCESSIVE ITERATIONS FOR CONVERGENCE
      DELTY = 0.000000001
C         SET MAXIMUM TOLERABLE DIFFERENCE BETWEEN SUCCESSIVE
C         DIFFERENCES IN SUCCESSIVE ITERATIONS FOR CONVERGENCE
      DELT = 0.000001
C ------ *      *      *      *      *      *      *      *      *
C         READ FIELDS DETERMINED BY PROGRAM UTRANS
      READ(5)(H(K),K=1,NODES)
      READ(5)(U(L),L=1,NODE3)
      READ(5)(V(M),M=1,NODE3)
      READ(5)(VORTEX(N),N=1,NODE3)
      REWIND 5
C         SET UP THE TIME-INVARIANT MATRICES AND BOUNDARY CONDITIONS,
C         AND THE INITIAL CONDITIONS ON THE VECTOR POTENTIAL FIELD
      DO 900 I1=1,NODE3
      A(I1) = 0.0
  900 CONTINUE
      CALL VORBC1(VORTEX,NODE3,SHEAR,N2,N3,NALPHA)
      CALL MATRJ(NEQN,N1,N2,N3,RI,RO,U,H,NODES,NODE3,RE)
      CALL MATRY(N1,N2,N3,RI,RO,NEQNY)
      CALL SORY(VORTEX,A,ANEW,ASTAR,NALPHA,NEQNY,NODE3,W,DELTY,
     1M,MMAX,RESTY,DIFFY,DIFF1,DIFF2,DELT,DIFFCE)
      CALL AUTHOR
      NSTEP = 0
      TIME = 0.0
      CALL WRITER(VORTEX,A,V,NODE3)
C         PROCEED FROM TIME-STEP TO TIME-STEP UPDATING THE
C         VORTICITY, VECTOR POTENTIAL, AND VELOCITY FIELDS
      DO 901 NSTEP=1,NSTEPS
C         LENGTH OF TIME-STEP ALTERATION FACILITY
      IF(NSTEP.EQ.21) DELTAT = DELTAT*10.0
      TIME = TIME + DELTAT
      CALL VORBC2(VORTEX,NODE3,NALPHA,N1,RI,RO,V,U)
      CALL MATMA(A,NODE3,NEQN,N1,N2,N3,RI,RO)
      CALL SORJMA(NEQN,VORTEX,VORNEW,W1,DELT1,M1,MMAX1,
     1REST1,DIFFJM,NODE3,NALPHA,DELTAT)
      CALL SORY(VORTEX,A,ANEW,ASTAR,NALPHA,NEQNY,NODE3,W,DELTY,
     1M,MMAX,RESTY,DIFFY,DIFF1,DIFF2,DELT,DIFFCE)
      CALL SPEED(A,V,NODE3,N1,N2,N3,RI,RO)
      CALL WRITER(VORTEX,A,V,NODE3)
  901 CONTINUE
C         WRITE PARAMETERS REQUIRED IN PROGRAM UROTA
      WRITE(7)NODE3,NALPHA,N1,N2,N3,NSTEPS
      WRITE(7)RI,RO,RE,TIME
C         WRITE VORTICITY FIELD REQUIRED IN PROGRAM UROTA
      WRITE(7)(VORTEX(I),I=1,NODE3)
      STOP
      END
```

```
      PROGRAM UROTA(DATB,OUTPUT,TAPE5=DATB,TAPE6=OUTPUT)
      DIMENSION VORTEX(168), P(14), SUM(4), PTEST(4)
C ------ *       *       *       *       *       *       *       *       *
C          THIS PROGRAM USES THE RESULTS OF PROGRAM UROT TO
C          CALCULATE THE PRESSURE FIELD ON THE SURFACE OF THE
C          SPHERE. AND THE DRAG, LIFT AND MOMENT COEFFICIENTS
C          OF THE SPHERE.
C          ALL PARAMETERS ARE SET IN PROGRAM UROT, WITH THE
C          EXCEPTION OF IOPT4, WHICH IS SET BELOW.
C
C          READ PARAMETERS USED IN PROGRAM UROT
      READ(5)NODE3,NALPHA,N1,N2,N3,NSTEP
      READ(5)RI,RO,RE,TIME
C          READ VORTICITY FIELD DETERMINED BY PROGRAM UROT
      READ(5)(VORTEX(I),I=1,NODE3)
      REWIND 5
C          SET OUTPUT SELECTION PARAMETER IOPT4 AS
C          FOLLOWS - THE PRESSURE FIELD ON THE SURFACE
C                    OF THE SPHERE IS WRITTEN OUT UNLESS
C                    IOPT4 = 0.
      IOPT4 = 1
C ------ *       *       *       *       *       *       *       *       *
      CALL PRESS(VORTEX,NODE3,RI,RO,RE,N1,N2,N3,P,NALPHA,
     1SUM,PTEST,ERRMAX)
      CALL COEFF(VORTEX,NODE3,P,NALPHA,N2,N3,RE,CDP,CDV,CLP,CLV,CM)
      CALL TYPIST(P,NALPHA,CDP,CDV,CLP,CLV,CM,NSTEP,TIME,IOPT4,ERRMAX)
      STOP
      END
```

```
      PROGRAM UTRANS(INPUT,OUTPUT,DATA,TAPE5=INPUT,TAPE6=OUTPUT,
     1TAPE7=DATA)
      DIMENSION E(4,55), F(4,55), NE(4,55), NF(4,55)
      DIMENSION B(55), H(56), HGUESS(55)
      DIMENSION U(168), V(168), VORTEX(168)
C ------ *       *       *       *       *       *       *       *       *
C          THIS PROGRAM DETERMINES THE TRANSLATIONAL COMPONENT
C          OF THE VELOCITY FIELD, AND ASSOCIATED SCALAR POTENTIAL
C          FIELD, FOR UNIFORM OR LINEAR SHEAR FLOW PAST A SPHERE.
C          THIS IS PERFORMED BY SOLVING THE SCALAR POTENTIAL
C          LAPLACE EQUATION NUMERICALLY.
C          THIS PROGRAM ALSO SETS UP THE INITIAL CONDITIONS FOR
C          PROGRAM UROT AS FOLLOWS - FOR EFFECTIVELY UNIFORM FLOW
C                                    PAST THE SPHERE, STOKES FLOW IS
C                                    ASSUMED INITIALLY.
C                                  - FOR OTHER FLOWS, A LINEAR RADIAL
C                                    VARIATION BETWEEN THE INNER AND
C                                    OUTER BOUNDARY CONDITIONS ON
C                                    VELOCITY IS ASSUMED INITIALLY.
C
```

```
C         SET NUMBER OF RADIAL SPACINGS              (GE 3)
      N1 = 3
C         SET NUMBER OF POLAR ANGLE SPACINGS         (GE 3)
      N2 = 4
C         SET NUMBER OF AZIMUTHAL ANGLE SPACINGS (GE 4)
C         THIS NUMBER MUST BE EXACTLY DIVISIBLE BY FOUR
      N3 = 4
C         SET INNER RADIUS
      RI = 0.5
C         SET OUTER RADIUS
      RO = 1.1
C         SET CENTRE-LINE VELOCITY AT OUTER BOUNDARY
      U0 = 1.0
C         SET MAGNITUDE OF SHEAR AT OUTER BOUNDARY
      SHEAR = 0.0
C         SET NUMBER OF NODES ( = (N1 + 1)*((N3*(N2 - 1)) + 2 ) )
C         (REMEMBER TO SET DIMENSIONS ACCORDINGLY)
      NODES = 56
      NEQNS = NODES - 1
C         SET RELAXATION FACTOR
      W = 1.7
C         SET MAXIMUM TOLERABLE NUMBER OF ITERATIONS
      MMAX = 500
C         SET MAXIMUM TOLERABLE SQUARE OF THE DIFFERENCE BETWEEN
C         ELEMENTS IN SUCCESSIVE ITERATIONS FOR CONVERGENCE
      DELTA = 0.00000001
C ------ *     *     *     *     *     *     *     *     *
C         SET UP INITIAL GUESS TO SCALAR POTENTIAL FIELD
      DO 999 I = 1,NEQNS
      HGUESS(I) = 1.0
  999 CONTINUE
      NODE2 = NODES + NODES
      NODE3 = NODE2 + NODES
C         DETERMINE TRANSLATIONAL COMPONENT OF VELOCITY
      CALL MATRIX(N1,N2,N3,RI,RO,U0,SHEAR,E,F,NE,NF,B,NEQNS)
      CALL SOR(E,F,NE,NF,B,H,HGUESS,NEQNS,NODES,W,M,MMAX,DELTA,REST,
     1DIFFCE)
      CALL VELOX(H,B,NODES,NEQNS,U,U0,N1,N2,N3,RI,RO,NODE2,NODE3)
      CALL SCRIBE (N1,N2,N3,RI,RO,U0,SHEAR,NODES,W,M,DELTA,H,REST,
     1DIFFCE,NODE2,NODE3,U)
C         SET UP INITIAL CONDITIONS ON VELOCITY AND VORTICITY
      IF(ABS(SHEAR).GE.0.000001) GO TO 1000
      CALL START1(V,VORTEX,U,NODE3,N1,N2,N3,RI,RO)
      GO TO 1001
 1000 CALL START2(V,VORTEX,U,NODE3,N1,N2,N3,RI,RO,SHEAR,U0)
 1001 WRITE(7)(H(K),K=1,NODES)
      WRITE(7)(U(L),L=1,NODE3)
      WRITE(7)(V(M),M=1,NODE3)
      WRITE(7)(VORTEX(N),N=1,NODE3)
      STOP
      END
```

```
      SUBROUTINE AUTHOR
      COMMON/COMAUT/N1,N2,N3,NODES,PI,RO,RE,SHEAR,W1,DELT1,W,
     1DELTY,DELT
C
C
C
C           THIS SUBROUTINE WRITES OUT THE PRINCIPAL PARAMETERS
C           USED IN PROGRAM UROT.
C
C
C
      WRITE(6,950)
  950 FORMAT(1H1/////)
      WRITE(6,951)RI,RO,RE,SHEAR
  951 FORMAT(1X,18HRADIUS OF SPHERE =,F7.4,32X,
     126HRADIUS OF OUTER BOUNDARY =,F7.4/1X,
     217HREYNOLDS NUMBER =,F9.1,31X,
     338HMAGNITUDE OF SHEAR AT OUTER BOUNDARY =,F7.4)
      WRITE(6,952)N1,N2,N3,NODES
  952 FORMAT(1H0,27HNUMBER OF RADIAL SPACINGS =,I3,27X,
     132HNUMBER OF POLAR ANGLE SPACINGS =,I3/1X,
     236HNUMBER OF AZIMUTHAL ANGLE SPACINGS =,I3,18X,
     317HNUMBER OF NODES =,I5)
      WRITE(6,953)W1,DELT1
  953 FORMAT(1H0,39HVORTICITY TRANSPORT EQUATION PARAMETERS/1X,10X,
     126HRELAXATION FACTOR CHOSEN =,F7.4/1X,10X,
     257HMAXIMUM SQUARE DIFFERENCE BETWEEN ELEMENTS IN SUCCESSIVE ,
     328HITERATIONS FOR CONVERGENCE =,F15.13)
      WRITE(6,954)W,DELTY,DELT
  954 FORMAT(1H0,44HVECTOR POTENTIAL POISSON EQUATION PARAMETERS/1X,10X
     126HRELAXATION FACTOR CHOSEN =,F7.4/1X,10X,
     257HMAXIMUM SQUARE DIFFERENCE BETWEEN ELEMENTS IN SUCCESSIVE ,
     328HITERATIONS FOR CONVERGENCE =,F15.13/1X,10X,
     457HMAXIMUM SQUARE DIFFERENCE BETWEEN SUCCESSIVE DIFFERENCES ,
     542HIN SUCCESSIVE ITERATIONS FOR CONVERGENCE =,F15.13)
      RETURN
      END
```

```
      SUBROUTINE COEFF(VORTEX,NODE3,P,NALPHA,NN2,NN3,RRE,CDP,CDV,CLP,
     1CLV,CM)
      DIMENSION VORTEX(NODE3), P(NALPHA)
C
C
C
C           THIS SUBROUTINE CALCULATES - THE FORM AND FRICTION
C                                        DRAG COEFFICIENTS CDP
C                                        AND CDV, RESPECTIVELY.
C                                      - THE FORM AND FRICTION
C                                        LIFT COEFFICIENTS CLP
C                                        AND CLV, RESPECTIVELY.
C                                      - THE MOMENT COEFFICIENT
C                                        CM.
C
C
C
      N2 = NN2
      N3 = NN3
      RE = RRE
      PI = 3.14159265358898
      CA = 2.0/PI
      CB = CA/RE
      CC = -CB/2.0
      XN2 = N2
      XN3 = N3
      H2 = PI/XN2
      H3 = 2.0*PI/XN3
      S = SIN(H2/2.0)
      C = COS(H2/2.0)
      DA = 2.0*S*H3
      DB = 2.0*PI*(1.0 - C)
```

```
C           THETA = 0 POLAR NODE.
      L = 1
      LY = 2
      LZ = 3
      EA = -P(1)*DB
      EB = 0.0
      FA = EA*CA
      FB = EB*CB
      GA = 0.0
      GB = VORTEX(3)*DB
      HA = GA*CA
      HB = GB*CB
      BA = GB
      BB = BA*CC
C           NON-POLAR NODES.
      J = 2
      K = 1
      THETA = H2
      PHI = 0.0
  810 L = L + 1
      LY = LY + 3
      LZ = LY + 1
      ST = SIN(THETA)
      CT = COS(THETA)
      SP = SIN(PHI)
      CP = COS(PHI)
      EA = -P(L)*CT*ST*DA
      EB = -VORTEX(LZ)*ST*ST*DA
      FA = FA + (EA*CA)
      FB = FB + (EB*CB)
      GA = -P(L)*ST*ST*SP*DA
      GB = ((VORTEX(LY)*SP) + (VORTEX(LZ)*CT*CP))*ST*DA
      HA = HA + (GA*CA)
      HB = HB + (GB*CB)
      BA = ((VORTEX(LZ)*CP) + (VORTEX(LY)*SP*CT))*ST*DA
      BB = BB + (BA*CC)
      IF(K.EQ.N3) GO TO 811
      K = K + 1
      PHI = PHI + H3
      GO TO 810
  811 IF(J.EQ.N2) GO TO 812
      J = J + 1
      K = 1
      THETA = THETA + H2
      PHI = 0.0
      GO TO 810
C           THETA = PI POLAR NODE.
  812 L = L + 1
      LY = LY + 3
      LZ = LY + 1
      EA = P(L)*DB
      EB = 0.0
      FA = FA + (EA*CA)
      FB = FB + (EB*CB)
      GA = 0.0
      GB = VORTEX(LZ)*DB
      HA = HA + (GA*CA)
      HB = HB + (GB*CB)
      BA = GB
      BB = BB + (BA*CC)
      CDP = FA
      CDV = FB
      CLP = HA
      CLV = HB
      CM = BB
      RETURN
      END
```

```
      SUBROUTINE MATMA(A,NNODE3,NNEQNM,NN1,NN2,NN3,RPI,RPO)
      DIMENSION A(NNODE3)
      COMMON/COMMA1/LJX,LJY,LJZ,LX,LY,LZ
      COMMON/COMMA2/NALPH3,N33,N330N2,N330N4,N390N4,H2,H3
      COMMON/COMMA3/NC23,NCN23,CP,SP,CPP,SPP,CPM,SPM
      COMMON/COMMA4/LXPA,LXPB,LXPC,LXPD,LXPE,LXPF,LXPG,LXPH,LXPI,LXPJ,
     1LXPK,LXPL,LXPM,LXPN,LXPO,LXPP,LXPQ,LXPR,LXPS,LXPT,LXPU,LXPV,LXPW,
     2LXPX,LXPY,LXPAA,LXPBB,LXPCC,LXPDD,LXPEE,LXPFF,LXPGG,
     3              LXMA,LXMB,LXMC,LXMD,LXME,LXMF,LXMG,LXMH,LXMI,LXMJ,
     4LXMK,LXML,LXMM,LXMN,LXMO,LXMP,LXMQ,LXMR,LXMS,LXMT,LXMU,LXMV,LXMW,
     5LXMX,LXMY,LXMAA,LXMBB,LXMCC,LXMDD,LXMEE,LXMFF,LXMGG
      COMMON/COMMA5/LYPA,LYPB,LYPC,LYPD,LYPE,LYPF,LYPG,LYPH,LYPI,LYPJ,
     1LYPK,LYPL,LYPM,LYPN,LYPO,LYPP,LYPQ,LYPR,LYPS,LYPT,LYPU,LYPV,LYPW,
     2LYPX,LYPY,LYPAA,LYPBB,LYPCC,LYPDD,LYPEE,LYPFF,LYPGG,
     3              LYMA,LYMB,LYMC,LYMD,LYME,LYMF,LYMG,LYMH,LYMI,LYMJ,
     4LYMK,LYML,LYMM,LYMN,LYMO,LYMP,LYMQ,LYMR,LYMS,LYMT,LYMU,LYMV,LYMW,
     5LYMX,LYMY,LYMAA,LYMBB,LYMCC,LYMDD,LYMEE,LYMFF,LYMGG
      COMMON/COMMA6/LZPA,LZPB,LZPC,LZPD,LZPE,LZPF,LZPG,LZPH,LZPI,LZPJ,
     1LZPK,LZPL,LZPM,LZPN,LZPO,LZPP,LZPQ,LZPR,LZPS,LZPT,LZPU,LZPV,LZPW,
     2LZPX,LZPY,LZPAA,LZPBB,LZPCC,LZPDD,LZPEE,LZPFF,LZPGG,
     3              LZMA,LZMB,LZMC,LZMD,LZME,LZMF,LZMG,LZMH,LZMI,LZMJ,
     4LZMK,LZML,LZMM,LZMN,LZMO,LZMP,LZMQ,LZMR,LZMS,LZMT,LZMU,LZMV,LZMW,
     5LZMX,LZMY,LZMAA,LZMBB,LZMCC,LZMDD,LZMEE,LZMFF,LZMGG
      COMMON/COMMA7/D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11,D12,D13,D14,D15,
     1D16,D17,D18,D19,D20,D21,D22,D23,D24
      REAL JL, JD, JU, MAL, MAD, MAU
      COMMON YL(112,20), YD(112), YU(112,20), NYL(112,20), NYU(112,20)
      COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84,27)
      COMMON MAL(84,27), MAD(84), MAU(84,27)
C
C
C
C          THIS SUBROUTINE SETS UP THOSE TERMS IN THE MATRIX
C          EQUATION ANALOGUE OF THE VORTICITY TRANSPORT
C          EQUATION WHICH DEPEND ON THE VECTOR POTENTIAL FIELD A.
C          SUBROUTINES MATMA1-7 SET UP THE MATRIX COEFFICIENTS
C          MAL/MAD/MAU. THE NODE LOCATIONS NL/NU ARE SET UP BY
C          SUBROUTINES MATRJ AND MATRJ1-7 (Q.V.).
C          SUBROUTINE MATMA8 SETS UP THE ARGUMENTS OF THE
C          VECTOR POTENTIAL FIELD A.
C
C
C
      NEQNM = NNEQNM
      N1 = NN1
      N2 = NN2
      N3 = NN3
      RI = RPI
      RO = RRO
      NODE3 = NNODE3
      NALPH3 = NODE3/(N1 + 1)
      NALPHA = NALPH3/3
      N33 = N3*3
      N330N2 = N33/2
      N330N4 = N33/4
      N390N4 = N330N4*3
      PI = 3.14159265358998
      XN1 = N1
      XN2 = N2
      XN3 = N3
      H1 = (RO - RI)/XN1
      H2 = PI/XN2
      H3 = 2.0*PI/XN3
      LX = 1 + NALPH3
      LY = 2 + NALPH3
      LZ = 3 + NALPH3
      LJX = 1
      LJY = 2
      LJZ = 3
      I = 2
      R = RI + H1
  699 J = 1
      K = 1
      THETA = 0.0
      PHI = 0.0
      NC2 = (NALPHA*(I-1)) + 1
      NC23 = NC2*3
      NCN2 = NALPHA*I
      NCN23 = NCN2*3
      D1 = 1.0/2
      D2 = 1.0/H1
      D3 = D1*D2
```

```
      D4  = D2*D2
      D5  = D1/H2
      D6  = D1*D5
      D7  = D2*D5
      D8  = D5*D5
      D9  = D6/H3
      D10  = D9/H2
      D11  = D10/H3
      D12  = D7/H3
      D24  = D1*D1
      CALL MATMA8
C         INTERIOR NODES (THETA = 0 POLAR AXIS).
      CALL MATMA1(A,NODE3)
      LX = LX + 3
      LY = LX + 1
      LZ = LY + 1
      LJX = LJX + 3
      LJY = LJX + 1
      LJZ = LJY + 1
      J = 2
      THETA = H2
  700 ST = SIN(THETA)
      TT = TAN(THETA)
      PHIP = PHI + H3
      PHIM = PHI - H3
      CP = COS(PHI)
      SP = SIN(PHI)
      CPP = COS(PHIP)
      SPP = SIN(PHIP)
      CPM = COS(PHIM)
      SPM = SIN(PHIM)
      D13 = D1/TT
      D14 = D13*D1
      D15 = D13*D2
      D16 = D13*D5
      D17 = D24/(ST*ST)
      D18 = D1/(ST*H3)
      D19 = D13*D1
      D20 = D19/(ST*H3)
      D21 = D18*D2
      D22 = D19/H2
      D23 = D19/TT
  701 CALL MATMA8
      IF(LXME.GE.1) GO TO 7015
      LXME = 1
      LYME = 1
      LZME = 1
 7015 IF(LZPD.LE.NODE3) GO TO 7016
      LXPD = NODE3
      LYPD = NODE3
      LZPD = NODE3
C         INTERIOR NODES (GENERAL CASE).
 7016 CALL MATMA2(A,NODE3)
      IF(K.NE.1) GO TO 702
C         INTERIOR NODES (K = 1).
      CALL MATMA3(A,NODE3)
  702 IF(K.NE.N3) GO TO 703
C         INTERIOR NODES (K = N3).
      CALL MATMA4(A,NODE3)
  703 IF(J.NE.2) GO TO 704
C         INTERIOR NODES (J = 2).
      CALL MATMA5(A,NODE3,K,N3,N33)
  704 IF(J.NE.N2) GO TO 705
C         INTERIOR NODES (J = N2).
      CALL MATMA6(A,NODE3,K,N3,N33)
  705 LX = LX + 3
      LY = LX + 1
      LZ = LY + 1
      LJX = LJX + 3
      LJY = LJY + 1
      LJZ = LJY + 1
      IF(K.EQ.N3) GO TO 706
      K = K + 1
      Z = K - 1
      PHI = (2.0*Z*PI)/XN3
      PHIP = PHI + H3
      PHIM = PHI - H3
      CP = COS(PHI)
      SP = SIN(PHI)
      CPP = COS(PHIP)
      SPP = SIN(PHIP)
      CPM = COS(PHIM)
      SPM = SIN(PHIM)
```

```
      GO TO 701
  706 K = 1
      PHI = 0.0
      IF(J.EQ.N2) GO TO 707
      J = J + 1
      Y = J - 1
      THETA = Y*PI/XN2
      GO TO 700
  707 CALL MATMA8
C         INTERIOR NODES (THETA = PI POLAR AXIS).
      CALL MATMA7(A,NODE3)
      LX = LX + 3
      LY = LX + 1
      LZ = LY + 1
      LJX = LJX + 3
      LJY = LJX + 1
      LJZ = LJY + 1
      IF(I.EQ.N1) GO TO 708
      I = I + 1
      R = R + H1
      GO TO 699
  708 RETURN
      END


      SUBROUTINE MATMA1(A,NODE3)
      DIMENSION A(NODE3)
      COMMON/COMMA1/LJX,LJY,LJZ,LX,LY,LZ
      COMMON/COMMA2/NALPH3,N33,N330N2,N330N4,N390N4,H2,H3
      COMMON/COMMA4/LXPA,LXPB,LXPC,LXPD,LXPE,LXPF,LXPG,LXPH,LXPI,LXPJ,
     1LXPK,LXPL,LXPM,LXPN,LXPO,LXPP,LXPQ,LXPR,LXPS,LXPT,LXPU,LXPV,LXPW,
     2LXPX,LXPY,LXPAA,LXPBB,LXPCC,LXPDD,LXPEE,LXPFF,LXPGG,
     3              LXMA,LXMB,LXMC,LXMD,LXME,LXMF,LXMG,LXMH,LXMI,LXMJ,
     4LXMK,LXML,LXMM,LXMN,LXMO,LXMP,LXMQ,LXMR,LXMS,LXMT,LXMU,LXMV,LXMW,
     5LXMX,LXMY,LXMAA,LXMBB,LXMCC,LXMDD,LXMEE,LXMFF,LXMGG
      COMMON/COMMA5/LYPA,LYPB,LYPC,LYPD,LYPE,LYPF,LYPG,LYPH,LYPI,LYPJ,
     1LYPK,LYPL,LYPM,LYPN,LYPO,LYPP,LYPQ,LYPR,LYPS,LYPT,LYPU,LYPV,LYPW,
     2LYPX,LYPY,LYPAA,LYPBB,LYPCC,LYPDD,LYPEE,LYPFF,LYPGG,
     3              LYMA,LYMB,LYMC,LYMD,LYME,LYMF,LYMG,LYMH,LYMI,LYMJ,
     4LYMK,LYML,LYMM,LYMN,LYMO,LYMP,LYMQ,LYMR,LYMS,LYMT,LYMU,LYMV,LYMW,
     5LYMX,LYMY,LYMAA,LYMBB,LYMCC,LYMDD,LYMEE,LYMFF,LYMGG
      COMMON/COMMA6/LZPA,LZPB,LZPC,LZPD,LZPE,LZPF,LZPG,LZPH,LZPI,LZPJ,
     1LZPK,LZPL,LZPM,LZPN,LZPO,LZPP,LZPQ,LZPR,LZPS,LZPT,LZPU,LZPV,LZPW,
     2LZPX,LZPY,LZPAA,LZPBB,LZPCC,LZPDD,LZPEE,LZPFF,LZPGG,
     3              LZMA,LZMB,LZMC,LZMD,LZME,LZMF,LZMG,LZMH,LZMI,LZMJ,
     4LZMK,LZML,LZMM,LZMN,LZMO,LZMP,LZMQ,LZMR,LZMS,LZMT,LZMU,LZMV,LZMW,
     5LZMX,LZMY,LZMAA,LZMBB,LZMCC,LZMDD,LZMEE,LZMFF,LZMGG
      COMMON/COMMA7/D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11,D12,D13,D14,D15,
     1D16,D17,D18,D19,D20,D21,D22,D23,D24
      REAL JL, JD, JU, MAL, MAD, MAU
      COMMON YL(112,20), YD(112), YU(112,20), NYL(112,20), NYU(112,20)
      COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84,27)
      COMMON MAL(84,27), MAD(84), MAU(84,27)
C
C
C
C          THIS SUBROUTINE SETS UP THE MATRIX COEFFICIENTS FOR
C          NODES ON THE THETA = 0 POLAR AXIS.
C
C
C
      DA = D6*(A(LZPC)-A(LZPN))
      DB = D9*(A(LYPW)-A(LYPX)-A(LYPB)+A(LYPY))/4.0
      DC = D7*(A(LZPF)-A(LZMF)-A(LZPO)+A(LZMO))/2.0
      DD = D12*(A(LYPAA)-A(LYMAA)-A(LYPBB)+A(LYMBB)-A(LYPD)+A(LYMD)+
     1A(LYPCC)-A(LYMCC))/8.0
      DE = D10*(A(LXPD)-A(LXPB))
      MAD(LJX) = DA-DB-DC+DD-DE
      DF = D10*(A(LYPW)-(2.0*A(LXPO))+A(LXPX)-A(LXPB)+(2.0*A(LXPP))-
     1A(LXPY))/4.0
      DG = DC/2.0
      DH = DA/2.0
      DI = 3.0*DE/2.0
      MAD(LJY) = -DF+DG-DH+DB-DI
      DJ = DF/2.0
      MAD(LJZ) = DG-DD+DF-DH-DJ
      DK = D7*(A(LZPC)-A(LZPN))/2.0
      DL = D12*(A(LYPW)-A(LYPX)-A(LYPB)+A(LYPY))/8.0
      MAL(LJX,1) = -DK+DL
```

```
      DM = D7*A(LZ)/4.0
      DN = D12*(A(LYPO)-A(LYPP))/8.0
      MAL(LJX,2) = -DM+DN
      MAL(LJX,3) = -MAL(LJX,2)
      DO 710 ND1=4,27
      MAL(LJX,ND1) = 0.0
  710 CONTINUE
      DP = D9*(A(LXPW)-A(LXPX)-A(LXPB)+A(LXPY))/2.0
      DQ = 2.0*D24*A(LZ)
      DR = D4*(A(LZPA)-(2.0*A(LZ))+A(LZMA))
      DS = D12*(A(LXPAA)-A(LXMAA)-A(LXPBB)+A(LXMBB)-A(LXPD)+A(LXMD)+
     1A(LXPCC)-A(LXMCC))/8.0
      MAL(LJY,1) = DP-DQ+DR-DS
      MAL(LJY,2) = MAL(LJX,1)
      MAL(LJY,3) = 0.0
      MAL(LJY,4) = 0.0
      MAL(LJY,5) = MAL(LJX,2)
      MAL(LJY,6) = MAL(LJX,3)
      DO 711 ND2=7,27
      MAL(LJY,ND2) = 0.0
  711 CONTINUE
      DT = 2.0*D24*A(LY)
      DU = D6*(A(LXPC)-A(LXPN))
      DV = D4*(A(LYPA)-(2.0*A(LY))+A(LYMA))
      DW = D7*(A(LXPF)-A(LXMF)-A(LXPQ)+A(LXMQ))/4.0
      MAL(LJZ,1) = DT-DU-DV+DW
      MAL(LJZ,2) = MAL(LJX,1)
      MAL(LJZ,3) = MAL(LJX,2)
      MAL(LJZ,4) = MAL(LJX,3)
      DO 712 ND3=5,27
      MAL(LJZ,ND3) = 0.0
  712 CONTINUE
      DX = 3.0*D8*(A(LZPC)-(2.0*A(LZ))+A(LZPN))/2.0
      DY = D10*(A(LYPW)-(2.0*A(LYPO))+A(LYPX)-A(LYPB)+(2.0*A(LYPP))-
     1A(LYPY))/4.0
      DZ = D8*A(LZ)
      DDA = D10*(A(LYPO)-A(LYPP))/2.0
      MAU(LJX,1) = -DX+DY-DZ+DDA
      DDB = 3.0*D10*(A(LZPW)-(2.0*A(LZPO))+A(LZPX)-A(LZPB)+(2.0*A(LZPP)
     1-A(LZPY))/4.0
      DDC = D11*(A(LYPW)-(2.0*A(LYPO))+A(LYPX)-(2.0*A(LYPC))+(4.0*A(LY)
     1-(2.0*A(LYPN))+A(LYPB)-(2.0*A(LYPP))+A(LYPY))/2.0
      DDD = D10*(A(LZPO)-A(LZPP))/2.0
      DDE = D11*(A(LYPO)-(2.0*A(LY))+A(LYPP))
      MAU(LJX,2) = -DDB+DDC+DDD-DDE
      MAU(LJX,3) = -MAL(LJX,1)
      DDF = DP/(4.0*H2)
      DDG = D7*(A(LZPA)-A(LZMA))/2.0
      DDH = DDA*H2/2.0
      DDI = D12*(A(LYPS)-A(LYMS)-A(LYPT)+A(LYMT))/8.0
      MAU(LJX,4) = DDF-DDG-DDH+DDI+DJ
      DDJ = DA/(H2*4.0)
      DDK = DB/(H2*2.0)
      DDL = DZ/2.0
      DDM = DDA/2.0
      MAU(LJX,5) = -DDJ+DDK+DDL-DDM
      DDN = D10*(A(LZPW)-A(LZPX)-A(LZPB)+A(LZPY))/8.0
      DDO = DDD/2.0
      DDP = DDE/2.0
      MAU(LJX,6) = -DDN-DDO+DDP
      DDQ = D12*(A(LYPF)-A(LYMF)-A(LYPQ)+A(LYMQ))/8.0
      DDR = D9*(A(LYPC)-A(LYPN))/4.0
      DDS = D10*(A(LXPC)-(2.0*A(LX))+A(LXPN))/2.0
      MAU(LJX,7) = DDQ+DDR-DDS
      MAU(LJX,8) = MAL(LJX,3)
      MAU(LJX,9) = MAL(LJX,2)
      MAU(LJX,10) = 0.0
      MAU(LJX,11) = 0.0
      DDT = D12*(A(LYPA)-A(LYMA))/8.0
      DDU = D9*A(LY)/4.0
      DDV = D10*(A(LXPC)-A(LXPN))/8.0
      MAU(LJX,12) = DDT+DDU-DDV
      MAU(LJX,13) = -MAU(LJX,12)
      DO 713 ND4=14,19
      MAU(LJX,ND4) = 0.0
  713 CONTINUE
      MAU(LJX,20) = -DDF+DDG+DDH-DDI+DJ
      MAU(LJX,21) = DDJ-DDK+DDL-DDM
      MAU(LJX,22) = DDN-DDO+DDP
      MAU(LJX,23) = -MAU(LJX,7)
      MAU(LJX,24) = MAU(LJX,13)
      MAU(LJX,25) = MAU(LJX,12)
      MAU(LJX,26) = 0.0
```

```
      MAU(LJX,27) = 0.0
      DDW = D11*(A(LXPW)-(2.0*A(LXPO))+A(LXPX)-(2.0*A(LXPC))+(4.0*A(LX))
     1-(2.0*A(LXPN))+A(LXPB)-(2.0*A(LXPP))+A(LXPY))/2.0
      DDX = D12*(A(LZPAA)-A(LZMAA)-A(LZPBB)+A(LZMBB)-A(LZPD)+A(LZMD)+
     1A(LZPCC)-A(LZMCC))/8.0
      DDY = 2.0*H2*DDM
      DDZ = D11*(A(LXPO)-(2.0*A(LX))+A(LXPP))
      MAU(LJY,1) = -DDW+DDX+DDY+DDZ
      DEA = D7*(A(LZPC)-A(LZPN))/2.0
      MAU(LJY,2) = DEA-DL
      DEB = H2*DJ
      DEC = D12*(A(LXPS)-A(LXMS)-A(LXPT)+A(LXMT))/8.0
      MAU(LJY,3) = DEB-DEC
      DED = D7*(A(LZPA)-A(LZMA))/4.0
      DEE = D6*A(LZ)
      DEF = DI/2.0
      MAU(LJY,4) = -DED-DEE+DDH+DEF
      DEG = D12*(A(LZPS)-A(LZMS)-A(LZPT)+A(LZMT))/8.0
      DEH = D9*(A(LZPO)-A(LZPP))/4.0
      DEI = DDZ/2.0
      MAU(LJY,5) = DEG+DEH-DEI
      MAU(LJY,6) = MAU(LJX,7)
      MAU(LJY,7) = 0.0
      MAU(LJY,8) = 0.0
      MAU(LJY,9) = 0.0
      MAU(LJY,10) = MAL(LJY,6)
      MAU(LJY,11) = MAL(LJY,5)
      MAU(LJY,12) = MAU(LJX,12)
      MAU(LJY,13) = MAU(LJX,13)
      MAU(LJY,14) = 0.0
      MAU(LJY,15) = 0.0
      MAU(LJY,16) = -MAU(LJY,3)
      MAU(LJY,17) = DED+DEE-DDH+DEF
      MAU(LJY,18) = -DEG-DEH-DEI
      MAU(LJY,19) = -MAU(LJX,7)
      MAU(LJY,20) = 0.0
      MAU(LJY,21) = MAU(LJX,13)
      MAU(LJY,22) = MAU(LJX,12)
      DO 714 ND5=23,27
      MAU(LJY,ND5) = 0.0
  714 CONTINUE
      MAU(LJZ,1) = -MAL(LJX,1)
      MAU(LJZ,2) = 2.0*H3*MAU(LJX,12)
      DEJ = 2.0*DDF
      DEK = D6*A(LZ)/2.0
      DEL = DE/4.0
      MAU(LJZ,3) = DEJ-DEK-DDI+DEL
      MAU(LJZ,4) = 0.0
      MAU(LJZ,5) = MAU(LJX,7)
      MAU(LJZ,6) = MAL(LJY,6)
      MAU(LJZ,7) = MAL(LJY,5)
      MAU(LJZ,8) = 0.0
      MAU(LJZ,9) = 0.0
      MAU(LJZ,10) = MAU(LJX,12)
      MAU(LJZ,11) = MAU(LJX,13)
      MAU(LJZ,12) = 0.0
      MAU(LJZ,13) = 0.0
      MAU(LJZ,14) = 0.0
      MAU(LJZ,15) = 0.0
      MAU(LJZ,16) = -MAU(LJZ,2)
      MAU(LJZ,17) = -DEJ+DEK+DDI+DEL
      MAU(LJZ,18) = 0.0
      MAU(LJZ,19) = -MAU(LJX,7)
      MAU(LJZ,20) = 0.0
      MAU(LJZ,21) = 0.0
      MAU(LJZ,22) = MAU(LJX,13)
      MAU(LJZ,23) = MAU(LJX,12)
      DO 715 ND6=24,27
      MAU(LJZ,ND6) = 0.0
  715 CONTINUE
      RETURN
      END

      SUBROUTINE MATMA2(A,NODE3)
      DIMENSION A(NODE3)
      COMMON/COMMA1/LJX,LJY,LJZ,LX,LY,LZ
      COMMON/COMMA4/LXPA,LXPB,LXPC,LXPD,LXPE,LXPF,LXPG,LXPH,LXPI,LXPJ,
     1LXPK,LXPL,LXPM,LXPN,LXPO,LXPP,LXPQ,LXPR,LXPS,LXPT,LXPU,LXPV,LXPW,
     2LXPX,LXPY,LXPAA,LXPBB,LXPCC,LXPDD,LXPEE,LXPFF,LXPGG,
     3              LXMA,LXMB,LXMC,LXMD,LXME,LXMF,LXMG,LXMH,LXMI,LXMJ,
     4LXMK,LXML,LXMM,LXMN,LXMO,LXMP,LXMQ,LXMR,LXMS,LXMT,LXMU,LXMV,LXMW,
     5LXMX,LXMY,LXMAA,LXMBB,LXMCC,LXMDD,LXMEE,LXMFF,LXMGG
```

```
      COMMON/COMMA5/LYPA,LYPB,LYPC,LYPD,LYPE,LYPF,LYPG,LYPH,LYPI,LYPJ,
     1LYPK,LYPL,LYPM,LYPN,LYPO,LYPP,LYPQ,LYPR,LYPS,LYPT,LYPU,LYPV,LYPW,
     2LYPX,LYPY,LYPAA,LYPBB,LYPCC,LYPDD,LYPEE,LYPFF,LYPGG,
     3               LYMA,LYMB,LYMC,LYMD,LYME,LYMF,LYMG,LYMH,LYMI,LYMJ,
     4LYMK,LYML,LYMM,LYMN,LYMO,LYMP,LYMQ,LYMR,LYMS,LYMT,LYMU,LYMV,LYMW,
     5LYMX,LYMY,LYMAA,LYMBB,LYMCC,LYMDD,LYMEE,LYMFF,LYMGG
      COMMON/COMMA6/LZPA,LZPB,LZPC,LZPD,LZPE,LZPF,LZPG,LZPH,LZPI,LZPJ,
     1LZPK,LZPL,LZPM,LZPN,LZPO,LZPP,LZPQ,LZPR,LZPS,LZPT,LZPU,LZPV,LZPW,
     2LZPX,LZPY,LZPAA,LZPBB,LZPCC,LZPDD,LZPEE,LZPFF,LZPGG,
     3               LZMA,LZMB,LZMC,LZMD,LZME,LZMF,LZMG,LZMH,LZMI,LZMJ,
     4LZMK,LZML,LZMM,LZMN,LZMO,LZMP,LZMQ,LZMR,LZMS,LZMT,LZMU,LZMV,LZMW,
     5LZMX,LZMY,LZMAA,LZMBB,LZMCC,LZMDD,LZMEE,LZMFF,LZMGG
      COMMON/COMMA7/D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11,D12,D13,D14,D15,
     1D16,D17,D18,D19,D20,D21,D22,D23,D24
      REAL JL, JD, JU, MAL, MAD, MAU
      COMMON YL(112,20), YD(112), YU(112,20), NYL(112,20), NYU(112,20)
      COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84,27)
      COMMON MAL(84,27), MAD(84), MAU(84,27)
C
C
C
C        THIS SUBROUTINE SETS UP THE MATRIX COEFFICIENTS FOR
C        NODES NOT ON THE POLAR AXIS.
C
C
      DA = D6*(A(LZPB)-A(LZMB))/2.0
      DB = D14*A(LZ)
      DC = D19*(A(LYPC)-A(LYMC))/2.0
      DD = D15*(A(LZPA)-A(LZMA))/2.0
      DE = D7*(A(LZPD)-A(LZMD)-A(LZPE)+A(LZME))/4.0
      DF = D21*(A(LYPF)-A(LYMF)-A(LYPG)+A(LYMG))/4.0
      MAD(LJX) = DA+DB-DC-DD-DE+DF
      DG = D23*(A(LXPC)-A(LXMC))/2.0
      DH = D22*(A(LXPH)-A(LXMH)-A(LXPI)+A(LXMI))/4.0
      MAD(LJY) = DG+DE-DB+DC-DH
      MAD(LJZ) = DD-DF-DG+DH-DA
      DI = D7*(A(LZPB)-A(LZMB))/4.0
      DJ = D15*A(LZ)/2.0
      DK = D21*(A(LYPC)-A(LYMC))/4.0
      MAL(LJX,1) = -DI-DJ+DK
      MAL(LJX,2) = 0.0
      DL = D22*(A(LXPC)-A(LXMC))/4.0
      DM = D6*A(LZ)/2.0
      DN = D7*(A(LZPA)-A(LZMA))/4.0
      MAL(LJX,3) = -DL+DM+DN
      MAL(LJX,4) = 0.0
      DP = D21*(A(LYPA)-A(LYMA))/4.0
      DQ = D19*A(LY)/2.0
      DR = D22*(A(LXPB)-A(LXMB))/4.0
      MAL(LJX,5) = -DP-DQ+DR
      DO 720 ND1=6,27
      MAL(LJX,ND1) = 0.0
  720 CONTINUE
      DS = D19*(A(LXPC)-A(LXMC))
      DT = 2.0*D24*A(LZ)
      DU = D21*(A(LXPF)-A(LXMF)-A(LXPG)+A(LXMG))/4.0
      DV = D4*(A(LZPA)-(2.0*A(LZ))+A(LZMA))
      MAL(LJY,1) = DS-DT-DU+DV
      MAL(LJY,2) = -DM-DJ+DK
      MAL(LJY,3) = MAL(LJX,3)
      MAL(LJY,4) = MAL(LJX,5)
      DO 721 ND2=5,27
      MAL(LJY,ND2) = 0.0
  721 CONTINUE
      DX = 2.0*D24*A(LY)
      DY = D6*(A(LXPB)-A(LXMB))
      DZ = D4*(A(LYPA)-(2.0*A(LY))+A(LYMA))
      DDA = D7*(A(LXPD)-A(LXMD)-A(LXPE)+A(LXME))/4.0
      MAL(LJZ,1) = DX-DY-DZ+DDA
      DDB = D15*(A(LYPA)-A(LYMA))/2.0
      DDC = D14*A(LY)
      DDD = D16*(A(LXPB)-A(LXMB))/2.0
      DDE = D6*(A(LYPB)-A(LYMB))/2.0
      DDF = D7*(A(LYPD)-A(LYMD)-A(LYPE)+A(LYME))/4.0
      DDG = D8*(A(LXPB)-(2.0*A(LX))+A(LXMB))
      MAL(LJZ,2) = DDB+DDC-DDD-DDE-DDF+DDG
      MAL(LJZ,3) = MAL(LJX,1)
      MAL(LJZ,4) = MAL(LJX,3)
      MAL(LJZ,5) = 0.0
      MAL(LJZ,6) = MAL(LJX,5)
      DO 722 ND3=7,27
      MAL(LJZ,ND3) = 0.0
```

```
  722 CONTINUE
      DDH = D9*(A(LZPB)-(2.0*A(LZ))+A(LZMB))
      DDI = D17*A(LZ)
      DDJ = D16*(A(LZPB)-A(LZMB))/2.0
      DDK = D23*(A(LYPC)-A(LYMC))/2.0
      DDL = D22*(A(LYPH)-A(LYMH)-A(LYPI)+A(LYMI))/4.0
      MAU(LJX,1) = -DDH+DDI-DDJ-DDK+DDL
      DDM = D22*(A(LZPH)-A(LZMH)-A(LZPI)+A(LZMI))/4.0
      DDN = D23*(A(LZPC)-A(LZMC))/2.0
      DDO = D20*(A(LYPC)-(2.0*A(LY))+A(LYMC))
      MAU(LJX,2) = -DDM-DDN+DDO
      MAU(LJX,3) = -MAL(LJX,1)
      MAU(LJX,4) = 0.0
      MAU(LJX,5) = -MAL(LJX,3)
      MAU(LJX,6) = 0.0
      MAU(LJX,7) = -MAL(LJX,5)
      DO 723 ND4=8,27
      MAU(LJX,ND4) = 0.0
  723 CONTINUE
      DDP = D20*(A(LXPC)-(2.0*A(LX))+A(LXMC))
      DDQ = D21*(A(LZPF)-A(LZMF)-A(LZPG)+A(LZMG))/4.0
      DDR = D19*(A(LZPC)-A(LZMC))/2.0
      MAU(LJY,1) = -DDP+DDQ+DDR
      MAU(LJY,2) = -MAL(LJY,2)
      MAU(LJY,3) = -MAL(LJX,3)
      MAU(LJY,4) = -MAL(LJX,5)
      DO 724 ND5=5,27
      MAU(LJY,ND5) = 0.0
  724 CONTINUE
      MAU(LJZ,1) = -MAL(LJX,1)
      MAU(LJZ,2) = -MAL(LJX,3)
      MAU(LJZ,3) = 0.0
      MAU(LJZ,4) = -MAL(LJX,5)
      DO 725 ND6=5,27
      MAU(LJZ,ND6) = 0.0
  725 CONTINUE
      RETURN
      END

      SUBROUTINE MATMA3(A,NODE3)
      DIMENSION A(NODE3)
      COMMON/COMMA1/LJX,LJY,LJZ,LX,LY,LZ
      COMMON/COMMA2/NALPH3,N33,N330N2,N330N4,N390N4,H2,H3
      COMMON/COMMA4/LXPA,LXPB,LXPC,LXPD,LXPE,LXPF,LXPG,LXPH,LXPI,LXPJ,
     1LXPK,LXPL,LXPM,LXPN,LXPO,LXPP,LXPQ,LXPR,LXPS,LXPT,LXPU,LXPV,LXPW,
     2LXPX,LXPY,LXPAA,LXPBB,LXPCC,LXPDD,LXPEE,LXPFF,LXPGG,
     3              LXMA,LXMB,LXMC,LXMD,LXME,LXMF,LXMG,LXMH,LXMI,LXMJ,
     4LXMK,LXML,LXMM,LXMN,LXMO,LXMP,LXMQ,LXMR,LXMS,LXMT,LXMU,LXMV,LXMW,
     5LXMX,LXMY,LXMAA,LXMBB,LXMCC,LXMDD,LXMEE,LXMFF,LXMGG
      COMMON/COMMA5/LYPA,LYPB,LYPC,LYPD,LYPE,LYPF,LYPG,LYPH,LYPI,LYPJ,
     1LYPK,LYPL,LYPM,LYPN,LYPO,LYPP,LYPQ,LYPR,LYPS,LYPT,LYPU,LYPV,LYPW,
     2LYPX,LYPY,LYPAA,LYPBB,LYPCC,LYPDD,LYPEE,LYPFF,LYPGG,
     3              LYMA,LYMB,LYMC,LYMD,LYME,LYMF,LYMG,LYMH,LYMI,LYMJ,
     4LYMK,LYML,LYMM,LYMN,LYMO,LYMP,LYMQ,LYMR,LYMS,LYMT,LYMU,LYMV,LYMW,
     5LYMX,LYMY,LYMAA,LYMBB,LYMCC,LYMDD,LYMEE,LYMFF,LYMGG
      COMMON/COMMA6/LZPA,LZPB,LZPC,LZPD,LZPE,LZPF,LZPG,LZPH,LZPI,LZPJ,
     1LZPK,LZPL,LZPM,LZPN,LZPO,LZPP,LZPQ,LZPR,LZPS,LZPT,LZPU,LZPV,LZPW,
     2LZPX,LZPY,LZPAA,LZPBB,LZPCC,LZPDD,LZPEE,LZPFF,LZPGG,
     3              LZMA,LZMB,LZMC,LZMD,LZME,LZMF,LZMG,LZMH,LZMI,LZMJ,
     4LZMK,LZML,LZMM,LZMN,LZMO,LZMP,LZMQ,LZMR,LZMS,LZMT,LZMU,LZMV,LZMW,
     5LZMX,LZMY,LZMAA,LZMBB,LZMCC,LZMDD,LZMEE,LZMFF,LZMGG
      COMMON/COMMA7/D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11,D12,D13,D14,D15,
     1D16,D17,D18,D19,D20,D21,D22,D23,D24
      REAL JL, JD, JU, MAL, MAD, MAU
      COMMON YL(112,20), YD(112), YU(112,20), NYL(112,20), NYU(112,20)
      COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84,27)
      COMMON MAL(84,27), MAD(84), MAU(84,27)
C
C
C
C        THIS SUBROUTINE MAKES THE NECESSARY ALTERATIONS TO
C        THE MATRIX COEFFICIENTS FOR NODES WHERE PHI
C        VARIATIONS MOVE COEFFICIENTS FROM THE LOWER TO THE
C        UPPER TRIANGULAR MATRIX.
C
C
C
      DA = D6*(A(LZPB)-A(LZMB))/2.0
      DB = D14*A(LZ)
      DC = D13*(A(LYPC)-A(LYMC+N33))/2.0
      DD = D15*(A(LZPA)-A(LZMA))/2.0
      DE = D7*(A(LZPD)-A(LZMD)-A(LZPE)+A(LZME))/4.0
      DF = D21*(A(LYPF)-A(LYMF)-A(LYPG+N33)+A(LYMG+N33))/4.0
      MAD(LJX) = DA+DB-DC-DD-DE+DF
```

```
      DG = D23*(A(LXPC)-A(LXMC+N33))/2.0
      DH = D22*(A(LXPH)-A(LXMH)-A(LXPI+N33)+A(LXMI+N33))/4.0
      MAD(LJY) = DG+DE-DB+DC-DH
      MAD(LJZ) = DD-DF-DG+DH-DA
      DI = D7*(A(LZPB)-A(LZMB))/4.0
      DJ = D15*A(LZ)/2.0
      DK = D21*(A(LYPC)-A(LYMC+N33))/4.0
      MAL(LJX,1) = -DI-DJ+DK
      DL = D22*(A(LXPC)-A(LXMC+N33))/4.0
      DM = D6*A(LZ)/2.0
      DN = D7*(A(LZPA)-A(LZMA))/4.0
      MAL(LJX,3) = -DL+DM+DN
      DVA = MAL(LJX,5)
      MAL(LJX,5) = 0.0
      DS = D19*(A(LXPC)-A(LXMC+N33))
      DT = 2.0*D24*A(LZ)
      DU = D21*(A(LXPF)-A(LXMF)-A(LXPG+N33)+A(LXMG+N33))/4.0
      DV = D4*(A(LZPA)-(2.0*A(LZ))+A(LZMA))
      MAL(LJY,1) = DS-DT-DU+DV
      MAL(LJY,2) = -DN-DJ+DK
      MAL(LJY,3) = MAL(LJX,3)
      MAL(LJY,4) = 0.0
      MAL(LJZ,3) = MAL(LJX,1)
      MAL(LJZ,4) = MAL(LJX,3)
      MAL(LJZ,5) = 0.0
      DDH = D8*(A(LZPB)-(2.0*A(LZ))+A(LZMB))
      DDI = D17*A(LZ)
      DDJ = D16*(A(LZPB)-A(LZMB))/2.0
      DDK = D23*(A(LYPC)-A(LYMC+N33))/2.0
      DDL = D22*(A(LYPH)-A(LYMH)-A(LYPI+N33)+A(LYMI+N33))/4.0
      MAU(LJX,1) = -DDH+DDI-DDJ-DDK+DDL
      DDM = D22*(A(LZPH)-A(LZMH)-A(LZPI+N33)+A(LZMI+N33))/4.0
      DDN = D23*(A(LZPC)-A(LZMC+N33))/2.0
      DDO = D20*(A(LYPC)-(2.0*A(LY))+A(LYMC+N33))
      MAU(LJX,2) = -DDM-DDN+DDO
      MAU(LJX,3) = -MAL(LJX,1)
      MAU(LJX,5) = -MAL(LJX,3)
      MAU(LJX,13) = DVA
      DDP = D20*(A(LXPC)-(2.0*A(LX))+A(LXMC+N33))
      DDQ = D21*(A(LZPF)-A(LZMF)-A(LZPG+N33)+A(LZMG+N33))/4.0
      DDR = D19*(A(LZPC)-A(LZMC+N33))/2.0
      MAU(LJY,1) = -DDP+DDQ+DDR
      MAU(LJY,2) = -MAL(LJY,2)
      MAU(LJY,3) = -MAL(LJX,3)
      MAU(LJY,13) = DVA
      MAU(LJZ,1) = -MAL(LJX,1)
      MAU(LJZ,2) = -MAL(LJX,3)
      MAU(LJZ,13) = DVA
      RETURN
      END
      SUBROUTINE MATMA4(A,NODE3)
      DIMENSION A(NODE3)
      COMMON/COMMA1/LJX,LJY,LJZ,LX,LY,LZ
      COMMON/COMMA2/NALPH3,N33,N330N2,N330N4,N390N4,H2,H3
      COMMON/COMMA4/LXPA,LXPB,LXPC,LXPD,LXPE,LYPF,LXPG.LXPH,LXPI,LXPJ,
     1LXPK,LXPL,LXPM,LXPN,LXPO,LXPP,LXPQ,LXPR,LXPS,LXPT,LXPU,LXPV,LXPW,
     2LXPX,LXPY,LXPAA,LXPBB,LXPCC,LXPDD,LXPEE,LXPFF,LXPGG,
     3               LXMA,LXMB,LXMC,LXMD,LXME,LXMF,LXMG,LXMH,LXMI,LXMJ,
     4LXMK,LXML,LXMM,LXMN,LXMO,LXMP,LXMQ,LXMR,LXMS,LXMT,LXMU,LXMV,LXMW,
     5LXMX,LXMY,LXMAA,LXMBB,LXMCC,LXMDD,LXMEE,LXMFF,LXMGG
      COMMON/COMMA5/LYPA,LYPB,LYPC,LYPD,LYPE,LYPF,LYPG,LYPH,LYPI,LYPJ,
     1LYPK,LYPL,LYPM,LYPN,LYPO,LYPP,LYPQ,LYPR,LYPS,LYPT,LYPU,LYPV,LYPW,
     2LYPX,LYPY,LYPAA,LYPBB,LYPCC,LYPDD,LYPEE,LYPFF,LYPGG,
     3               LYMA,LYMB,LYMC,LYMD,LYME,LYMF,LYMG,LYMH,LYMI,LYMJ,
     4LYMK,LYML,LYMM,LYMN,LYMO,LYMP,LYMQ,LYMR,LYMS,LYMT,LYMU,LYMV,LYMW,
     5LYMX,LYMY,LYMAA,LYMBB,LYMCC,LYMDD,LYMEE,LYMFF,LYMGG
      COMMON/COMMA6/LZPA,LZPB,LZPC,LZPD,LZPE,LZPF,LZPG,LZPH,LZPI,LZPJ,
     1LZPK,LZPL,LZPM,LZPN,LZPO,LZPP,LZPQ,LZPR,LZPS,LZPT,LZPU,LZPV,LZPW,
     2LZPX,LZPY,LZPAA,LZPBB,LZPCC,LZPDD,LZPEE,LZPFF,LZPGG,
     3               LZMA,LZMB,LZMC,LZMD,LZME,LZMF,LZMG,LZMH,LZMI,LZMJ,
     4LZMK,LZML,LZMM,LZMN,LZMO,LZMP,LZMQ,LZMR,LZMS,LZMT,LZMU,LZMV,LZMW,
     5LZMX,LZMY,LZMAA,LZMBB,LZMCC,LZMDD,LZMEE,LZMFF,LZMGG
      COMMON/COMMA7/D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11,D12,D13,D14,D15,
     1D16,D17,D18,D19,D20,D21,D22,D23,D24
      REAL JL, JD, JU, MAL, MAD, MAU
      COMMON YL(112,20), YD(112), YU(112,20), NYL(112,20), NYU(112,20)
      COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84,27)
      COMMON MAL(84,27), MAD(84), MAU(84,27)
C
C
C
C        THIS SUBROUTINE MAKES THE NECESSARY ALTERATIONS TO
C        THE MATRIX COEFFICIENTS FOR NODES WHERE PHI
```

```
C        VARIATIONS MOVE COEFFICIENTS FROM THE UPPER TO THE
C        LOWER TRIANGULAR MATRIX.
C
C
C
      DA = D6*(A(LZPB)-A(LZMB))/2.0
      DB = D14*A(LZ)
      DC = D19*(A(LYPC-N33)-A(LYMC))/2.0
      DD = D15*(A(LZPA)-A(LZMA))/2.0
      DE = D7*(A(LZPD)-A(LZMD)-A(LZPE)+A(LZME))/4.0
      DF = D21*(A(LYPF-N33)-A(LYMF-N33)-A(LYPG)+A(LYMG))/4.0
      MAD(LJX) = DA+DB-DC-DD-DE+DF
      DG = D23*(A(LXPC-N33)-A(LXMC))/2.0
      DH = D22*(A(LXPH-N33)-A(LXMH-N33)-A(LXPI)+A(LXMI))/4.0
      MAD(LJY) = DG+DF-DB+DC-DH
      MAD(LJZ) = DD-DF-DG+DH-DA
      DI = D7*(A(LZPB)-A(LZMB))/4.0
      DJ = D15*A(LZ)/2.0
      DK = D21*(A(LYPC-N33)-A(LYMC))/4.0
      MAL(LJX,1) = -DI-DJ+DK
      DL = D22*(A(LXPC-N33)-A(LXMC))/4.0
      DM = D6*A(LZ)/2.0
      DN = D7*(A(LZPA)-A(LZMA))/4.0
      MAL(LJX,3) = -DL+DM+DN
      DVA = MAL(LJX,5)
      MAL(LJX,11) = -DVA
      DS = D19*(A(LXPC-N33)-A(LXMC))
      DT = 2.0*D24*A(LZ)
      DU = D21*(A(LXPF-N33)-A(LXMF-N33)-A(LXPG)+A(LXMG))/4.0
      DV = D4*(A(LZPA)-(2.0*A(LZ))+A(LZMA))
      MAL(LJY,1) = DS-DT-DU+DV
      MAL(LJY,2) = -DN-DJ+DK
      MAL(LJY,3) = MAL(LJX,3)
      MAL(LJY,10) = -DVA
      MAL(LJZ,3) = MAL(LJX,1)
      MAL(LJZ,4) = MAL(LJX,3)
      MAL(LJZ,12) = -DVA
      DDH = D8*(A(LZPB)-(2.0*A(LZ))+A(LZMB))
      DDI = D17*A(LZ)
      DDJ = D16*(A(LZPB)-A(LZMB))/2.0
      DDK = D23*(A(LYPC-N33)-A(LYMC))/2.0
      DDL = D22*(A(LYPH-N33)-A(LYMH-N33)-A(LYPI)+A(LYMI))/4.0
      MAU(LJX,1) = -DDH+DDI-DDJ-DDK+DDL
      DDM = D22*(A(LZPH-N33)-A(LZMH-N33)-A(LZPI)+A(LZMI))/4.0
      DDN = D23*(A(LZPC-N33)-A(LZMC))/2.0
      DDO = D20*(A(LYPC-N33)-(2.0*A(LY))+A(LYMC))
      MAU(LJX,2) = -DDM-DDN+DDO
      MAU(LJX,3) = -MAL(LJX,1)
      MAU(LJX,5) = -MAL(LJX,3)
      MAU(LJX,7) = 0.0
      DDP = D20*(A(LXPC-N33)-(2.0*A(LX))+A(LXMC))
      DDQ = D21*(A(LZPF-N33)-A(LZMF-N33)-A(LZPG)+A(LZMG))/4.0
      DDR = D19*(A(LZPC-N33)-A(LZMC))/2.0
      MAU(LJY,1) = -DDP+DDQ+DDR
      MAU(LJY,2) = -MAL(LJY,2)
      MAU(LJY,3) = -MAL(LJX,3)
      MAU(LJY,4) = 0.0
      MAU(LJZ,1) = -MAL(LJX,1)
      MAU(LJZ,2) = -MAL(LJX,3)
      MAU(LJZ,4) = 0.0
      RETURN
      END
      SUBROUTINE MATMA5(A,NODE3,KK,NN3,NN33)
      DIMENSION A(NODE3)
      COMMON/COMMA1/LJX,LJY,LJZ,LX,LY,LZ
      COMMON/COMMA3/NC23,NCN23,CP,SP,CPP,SPP,CPM,SPM
      COMMON/COMMA4/LXPA,LXPB,LXPC,LXPD,LXPE,LXPF,LXPG,LXPH,LXPI,LXPJ,
     1LXPK,LXPL,LXPM,LXPN,LXPO,LXPP,LXPQ,LXPR,LXPS,LXPT,LXPU,LXPV,LXPW,
     2LXPX,LXPY,LXPAA,LXPBB,LXPCC,LXPDD,LXPEE,LXPFF,LXPGG,
     3              LXMA,LXMB,LXMC,LXMD,LXME,LXMF,LXMG,LYMH,LXMI,LXMJ,
     4LXMK,LXML,LXMM,LXMN,LXMO,LXMP,LXMQ,LXMR,LXMS,LXMT,LXMU,LXMV,LXMW,
     5LXMX,LXMY,LXMAA,LXMBB,LXMCC,LYMDD,LXMEE,LXMFF,LXMGG
      COMMON/COMMA5/LYPA,LYPB,LYPC,LYPD,LYPE,LYPF,LYPG,LYPH,LYPI,LYPJ,
     1LYPK,LYPL,LYPM,LYPN,LYPO,LYPP,LYPQ,LYPR,LYPS,LYPT,LYPU,LYPV,LYPW,
     2LYPX,LYPY,LYPAA,LYPBB,LYPCC,LYPDD,LYPEE,LYPFF,LYPGG,
     3              LYMA,LYMB,LYMC,LYMD,LYME,LYMF,LYMG,LYMH,LYMI,LYMJ,
     4LYMK,LYML,LYMM,LYMN,LYMO,LYMP,LYMQ,LYMR,LYMS,LYMT,LYMU,LYMV,LYMW,
     5LYMX,LYMY,LYMAA,LYMBB,LYMCC,LYMDD,LYMEE,LYMFF,LYMGG
      COMMON/COMMA6/LZPA,LZPB,LZPC,LZPD,LZPE,LZPF,LZPG,LZPH,LZPI,LZPJ,
     1LZPK,LZPL,LZPM,LZPN,LZPO,LZPP,LZPQ,LZPR,LZPS,LZPT,LZPU,LZPV,LZPW,
     2LZPX,LZPY,LZPAA,LZPBB,LZPCC,LZPDD,LZPEE,LZPFF,LZPGG,
     3              LZMA,LZMB,LZMC,LZMD,LZME,LZMF,LZMG,LZMH,LZMI,LZMJ,
     4LZMK,LZML,LZMM,LZMN,LZMO,LZMP,LZMQ,LZMR,LZMS,LZMT,LZMU,LZMV,LZMW,
```

```
     5LZMX,LZMY,LZMAA,LZMBB,LZMCC,LZMDD,LZMEE,LZMFF,LZMGG
      COMMON/COMMA7/D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11,D12,D13,D14,D15,
     1D16,D17,D18,D19,D20,D21,D22,D23,D24
      REAL JL, JD, JU, MAL, MAD, MAU
      COMMON YL(112,20), YD(112), YU(112,20), NYL(112,20), NYU(112,20)
      COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84,27)
      COMMON MAL(84,27), MAD(84), MAU(84,27)
C
C
C     THIS SUBROUTINE MAKES THE NECESSARY ALTERATIONS TO
C     THE MATRIX COEFFICIENTS FOR NODES ADJACENT TO THE
C     THETA = 0 POLAR AXIS.
C
C
      K = KK
      N3 = NN3
      N33 = NN33
      AA = A(NC23-2)
      AB = A(NC23-1)
      AC = A(NC23)
      DA = D6*((AB*SP)-(AC*CP)+A(LZMB))/2.0
      DB = D7*((A(LYPFF)*SP)-(A(LZPFF)*CP)-(A(LYMFF)*SP)+(A(LZMFF)*CP)+
     1A(LZPE)-A(LZME))/4.0
      MAD(LJX) = MAD(LJX)+DA-DB
      DC = D22*(A(LXMH)-A(LXMI))/4.0
      MAD(LJY) = MAD(LJY)+DB-DC
      MAD(LJZ) = MAD(LJZ)+DC-DA
      DD = D7*((AB*SP)-(AC*CP)+A(LZMB))/4.0
      MAL(LJX,1) = MAL(LJX,1)-DD
      DVA = MAL(LJX,3)
      MAL(LJX,3) = 0.0
      DE = D21*(A(LYPA)-A(LYMA))/4.0
      DEA = D19*A(LY)/2.0
      DEB = D22*(A(LXPB)-AA)/4.0
      MAL(LJX,5) = -DE-DEA+DEB
      MAL(LJX,13) = DVA
      MAL(LJY,3) = 0.0
      MAL(LJY,4) = MAL(LJX,5)
      MAL(LJY,12) = DVA*CP
      MAL(LJY,13) = DVA*SP
      DF = D6*(-AA+A(LXMB))
      DG = D7*(-A(LXPFF)+A(LXMFF)+A(LXPE)-A(LXME))/4.0
      MAL(LJZ,1) = MAL(LJZ,1)-DF+DG
      DH = D16*(-AA+A(LXMB))/2.0
      DI = D6*(-(AB*CP)-(AC*SP)+A(LYMB))/2.0
      DJ = D7*(-(A(LYPFF)*CP)-(A(LZPFF)*SP)+(A(LYMFF)*CP)+(A(LZMFF)*SP)+
     1A(LYPE)-A(LYME))/4.0
      DK = D8*(AA-A(LXMB))
      MAL(LJZ,2) = MAL(LJZ,2)-DH-DI-DJ+DK
      MAL(LJZ,3) = MAL(LJX,1)
      MAL(LJZ,4) = 0.0
      MAL(LJZ,6) = MAL(LJX,5)
      MAL(LJZ,13) = -DVA*SP
      MAL(LJZ,14) = DVA*CP
      DL = D9*(-(AB*SP)+(AC*CP)-A(LZMB))
      DM = D16*((AB*SP)-(AC*CP)+A(LZMB))/2.0
      DN = D22*(-(AB*CPP)-(AC*SPP)+(AB*CPM)+(AC*SPM)+A(LYMH)-A(LYMI))/
     14.0
      MAU(LJX,1) = MAU(LJX,1)-DL-DM+DN
      DP = D22*((AB*SPP)-(AC*CPP)-(AB*SPM)+(AC*CPM)+A(LZMH)-A(LZMI))/4.0
      MAU(LJX,2) = MAU(LJX,2)-DP
      MAU(LJX,3) = -MAL(LJX,1)
      MAU(LJX,7) = -MAL(LJX,5)
      MAU(LJY,4) = -MAL(LJX,5)
      MAU(LJZ,1) = -MAL(LJX,1)
      MAU(LJZ,4) = -MAL(LJX,5)
      DVB = MAL(LJX,5)
      IF(K.NE.1) GO TO 753
      DQ = D22*(A(LXMI)-A(LXMI+N33))/4.0
      MAD(LJY) = MAD(LJY)-DQ
      MAD(LJZ) = MAD(LJZ)+DQ
      MAL(LJX,5) = 0.0
      MAL(LJY,4) = 0.0
      MAL(LJZ,6) = 0.0
      DR = D22*(A(LYMI)-A(LYMI+N33))/4.0
      MAU(LJX,1) = MAU(LJX,1)+DR
      DS = D22*(A(LZMI)-A(LZMI+N33))/4.0
      MAU(LJX,2) = MAU(LJX,2)-DS
      MAU(LJX,13) = DVB
      MAU(LJY,10) = DVB
      MAU(LJZ,10) = DVB
```

```
750 IF(K.NE.N3) GO TO 751
    DT = D22*(-A(LXMH)+A(LXMH-N33))/4.0
    MAD(LJY) = MAD(LJY)-DT
    MAD(LJZ) = MAD(LJZ)+DT
    MAL(LJX,11) = -DVB
    MAL(LJY,10) = -DVB
    MAL(LJZ,12) = -DVB
    DU = D22*(-A(LYMH)+A(LYMH-N33))/4.0
    MAU(LJX,1) = MAU(LJX,1)+DU
    DV = D22*(-A(LZMH)+A(LZMH-N33))/4.0
    MAU(LJX,2) = MAU(LJX,2)-DV
    MAU(LJX,7) = 0.0
    MAU(LJY,4) = 0.0
    MAU(LJZ,4) = 0.0
751 RETURN
    END
    SUBROUTINE MATMA6(A,NODE3,KK,NN3,NN33)
    DIMENSION A(NODE3)
    COMMON/COMMA1/LJX,LJY,LJZ,LX,LY,LZ
    COMMON/COMMA3/NC23,NCN23,CP,SP,CPP,SPP,CPM,SPM
    COMMON/COMMA4/LXPA,LXPB,LXPC,LXPD,LXPE,LXPF,LXPG,LXPH,LXPI,LXPJ,
   1LXPK,LXPL,LXPM,LXPN,LXPO,LXPP,LXPQ,LXPR,LXPS,LXPT,LXPU,LXPV,LXPW,
   2LXPX,LXPY,LXPAA,LXPBB,LXPCC,LXPDD,LXPEE,LXPFF,LXPGG,
   3             LXMA,LXMB,LXMC,LXMD,LXME,LXMF,LXMG,LXMH,LXMI,LXMJ,
   4LXMK,LXML,LXMM,LXMN,LXMO,LXMP,LXMQ,LXMR,LXMS,LXMT,LXMU,LXMV,LXMW,
   5LXMX,LXMY,LXMAA,LXMBB,LXMCC,LXMDD,LXMEE,LXMFF,LXMGG
    COMMON/COMMA5/LYPA,LYPB,LYPC,LYPD,LYPE,LYPF,LYPG,LYPH,LYPI,LYPJ,
   1LYPK,LYPL,LYPM,LYPN,LYPO,LYPP,LYPQ,LYPR,LYPS,LYPT,LYPU,LYPV,LYPW,
   2LYPX,LYPY,LYPAA,LYPBB,LYPCC,LYPDD,LYPEE,LYPFF,LYPGG,
   3             LYMA,LYMB,LYMC,LYMD,LYME,LYMF,LYMG,LYMH,LYMI,LYMJ,
   4LYMK,LYML,LYMM,LYMN,LYMO,LYMP,LYMQ,LYMR,LYMS,LYMT,LYMU,LYMV,LYMW,
   5LYMX,LYMY,LYMAA,LYMBB,LYMCC,LYMDD,LYMEE,LYMFF,LYMGG
    COMMON/COMMA6/LZPA,LZPB,LZPC,LZPD,LZPE,LZPF,LZPG,LZPH,LZPI,LZPJ,
   1LZPK,LZPL,LZPM,LZPN,LZPO,LZPP,LZPQ,LZPR,LZPS,LZPT,LZPU,LZPV,LZPW,
   2LZPX,LZPY,LZPAA,LZPBB,LZPCC,LZPDD,LZPEE,LZPFF,LZPGG,
   3             LZMA,LZMB,LZMC,LZMD,LZME,LZMF,LZMG,LZMH,LZMI,LZMJ,
   4LZMK,LZML,LZMM,LZMN,LZMO,LZMP,LZMQ,LZMR,LZMS,LZMT,LZMU,LZMV,LZMW,
   5LZMX,LZMY,LZMAA,LZMBB,LZMCC,LZMDD,LZMEE,LZMFF,LZMGG
    COMMON/COMMA7/D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11,D12,D13,D14,D15,
   1D16,D17,D18,D19,D20,D21,D22,D23,D24
    REAL JL, JD, JU, MAL, MAD, MAU
    COMMON YL(112,20), YD(112), YU(112,20), NYL(112,20), NYU(112,20)
    COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84,27)
    COMMON MAL(84,27), MAD(84), MAU(84,27)
C
C
C
C        THIS SUBROUTINE MAKES THE NECESSARY ALTERATIONS TO
C        THE MATRIX COEFFICIENTS FOR NODES ADJACENT TO THE
C        THETA = PI POLAR AXIS.
C
C
C
    K = KK
    N3 = NN3
    N33 = NN33
    AA = A(NCN23-2)
    AB = A(NCN23-1)
    AC = A(NCN23)
    DA = D6*(-(AB*SP)+(AC*CP)-A(LZPB))/2.0
    DB = D7*(-(A(LYPGG)*SP)+(A(LZPGG)*CP)+(A(LYMGG)*SP)-(A(LZMGG)*CP)-
   1A(LZPD)+A(LZMD))/4.0
    MAD(LJX) = MAD(LJX)+DA-DB
    DC = D22*(-A(LXPH)+A(LXPI))/4.0
    MAD(LJY) = MAD(LJY)+DB-DC
    MAD(LJZ) = MAD(LJZ)+DC-DA
    DD = D7*(-(AB*SP)+(AC*CP)-A(LZPB))/4.0
    MAL(LJX,1) = MAL(LJX,1)-DD
    DE = D21*(A(LYPA)-A(LYMA))/4.0
    DEA = D19*A(LY)/2.0
    DEB = D22*(AA-A(LXMB))/4.0
    MAL(LJX,5) = -DE-DEA+DEB
    MAL(LJY,4) = MAL(LJX,5)
    DF = D6*(AA-A(LXPB))
    DG = D7*(A(LXPGG)-A(LXMGG)-A(LXPD)+A(LXMD))/4.0
    MAL(LJZ,1) = MAL(LJZ,1)-DF+DG
    DH = D15*(AA-A(LXPB))/2.0
    DI = D6*(-(AB*CP)-(AC*SP)-A(LYPB))/2.0
    DJ = D7*(-(A(LYPGG)*CP)-(A(LZPGG)*SP)+(A(LYMGG)*CP)+(A(LZMGG)*SP)
   1-A(LYPD)+A(LYMD))/4.0
    DK = D8*(AA-A(LXPB))
    MAL(LJZ,2) = MAL(LJZ,2)-DH-DI-DJ+DK
    MAL(LJZ,3) = MAL(LJX,1)
```

```
      MAL(LJZ,6) = MAL(LJX,5)
      DL = D8*(-(AB*SP)+(AC*CP)-A(LZPB))
      DM = D16*(-(AB*SP)+(AC*CP)-A(LZPB))/2.0
      DN = D22*(-(AB*CPP)-(AC*SPP)+(AB*CPM)+(AC*SPM)-A(LYPH)+A(LYPI))/
     14.0
      MAU(LJX,1) = MAU(LJX,1)-DL-DM+DN
      DP = D22*(-(AB*SPP)+(AC*CPP)+(AB*SPM)-(AC*CPM)-A(LZPH)+A(LZPI))/
     14.0
      MAU(LJX,2) = MAU(LJX,2)-DP
      MAU(LJX,3) = -MAL(LJX,1)
      DVA = -MAU(LJX,5)
      MAU(LJX,5) = 0.0
      MAU(LJX,7) = -MAL(LJX,5)
      MAU(LJX,15) = -DVA
      MAU(LJY,3) = 0.0
      MAU(LJY,4) = -MAL(LJX,5)
      MAU(LJY,12) = DVA*CP
      MAU(LJY,13) = DVA*SP
      MAU(LJZ,1) = -MAL(LJX,1)
      MAU(LJZ,2) = 0.0
      MAU(LJZ,4) = -MAL(LJX,5)
      MAU(LJZ,11) = DVA*SP
      MAU(LJZ,12) = -DVA*CP
      DVB = MAL(LJX,5)
      IF(K.NE.1) GO TO 760
      DQ = D22*(-A(LXPI)+A(LXPI+N33))/4.0
      MAD(LJY) = MAD(LJY)-DQ
      MAD(LJZ) = MAD(LJZ)+DQ
      MAL(LJX,5) = 0.0
      MAL(LJY,4) = 0.0
      MAL(LJZ,6) = 0.0
      DR = D22*(-A(LYPI)+A(LYPI+N33))/4.0
      MAU(LJX,1) = MAU(LJX,1)+DR
      DS = D22*(-A(LZPI)+A(LZPI+N33))/4.0
      MAU(LJX,2) = MAU(LJX,2)-DS
      MAU(LJX,13) = DVB
      MAU(LJY,10) = DVB
      MAU(LJZ,10) = DVB
  760 IF(K.NE.N3) GO TO 761
      DT = D22*(A(LXPH)-A(LXPH-N33))/4.0
      MAD(LJY) = MAD(LJY)-DT
      MAD(LJZ) = MAD(LJZ)+DT
      MAL(LJX,11) = -DVB
      MAL(LJY,10) = -DVB
      MAL(LJZ,12) = -DVB
      DU = D22*(A(LYPH)-A(LYPH-N33))/4.0
      MAU(LJX,1) = MAU(LJX,1)+DU
      DV = D22*(A(LZPH)-A(LZPH-N33))/4.0
      MAU(LJX,2) = MAU(LJX,2)-DV
      MAU(LJX,7) = 0.0
      MAU(LJY,4) = 0.0
      MAU(LJZ,4) = 0.0
  761 RETURN
      END
      SUBROUTINE MATMA7(A,NODE3)
      DIMENSION A(NODE3)
      COMMON/COMMA1/LJX,LJY,LJZ,LX,LY,LZ
      COMMON/COMMA2/MALPH3,N33,N33ON2,N33ON4,N39ON4,H2,H3
      COMMON/COMMA4/LXPA,LXPB,LXPC,LXPD,LXPE,LXPF,LXPG,LXPH,LXPI,LXPJ,
     1LXPK,LXPL,LXPM,LXPN,LXPO,LXPP,LXPQ,LXPR,LXPS,LXPT,LXPU,LXPV,LXPW,
     2LXPX,LXPY,LXPAA,LXPBB,LXPCC,LXPDD,LXPEE,LXPFF,LXPGG,
     3                LXMA,LXMB,LXMC,LXMD,LXME,LXMF,LXMG,LXMH,LXMI,LXMJ,
     4LXMK,LXML,LXMM,LXMN,LXMO,LXMP,LXMQ,LXMR,LXMS,LXMT,LXMU,LXMV,LXMW,
     5LXMX,LXMY,LXMAA,LXMBB,LXMCC,LXMDD,LXMEE,LXMFF,LXMGG
      COMMON/COMMA5/LYPA,LYPB,LYPC,LYPD,LYPE,LYPF,LYPG,LYPH,LYPI,LYPJ,
     1LYPK,LYPL,LYPM,LYPN,LYPO,LYPP,LYPQ,LYPR,LYPS,LYPT,LYPU,LYPV,LYPW,
     2LYPX,LYPY,LYPAA,LYPBB,LYPCC,LYPDD,LYPEE,LYPFF,LYPGG,
     3                LYMA,LYMB,LYMC,LYMD,LYME,LYMF,LYMG,LYMH,LYMI,LYMJ,
     4LYMK,LYML,LYMM,LYMN,LYMO,LYMP,LYMQ,LYMR,LYMS,LYMT,LYMU,LYMV,LYMW,
     5LYMX,LYMY,LYMAA,LYMBB,LYMCC,LYMDD,LYMEE,LYMFF,LYMGG
      COMMON/COMMA6/LZPA,LZPB,LZPC,LZPD,LZPE,LZPF,LZPG,LZPH,LZPI,LZPJ,
     1LZPK,LZPL,LZPM,LZPN,LZPO,LZPP,LZPQ,LZPR,LZPS,LZPT,LZPU,LZPV,LZPW,
     2LZPX,LZPY,LZPAA,LZPBB,LZPCC,LZPDD,LZPEE,LZPFF,LZPGG,
     3                LZMA,LZMB,LZMC,LZMD,LZME,LZMF,LZMG,LZMH,LZMI,LZMJ,
     4LZMK,LZML,LZMM,LZMN,LZMO,LZMP,LZMQ,LZMR,LZMS,LZMT,LZMU,LZMV,LZMW,
     5LZMX,LZMY,LZMAA,LZMBB,LZMCC,LZMDD,LZMEE,LZMFF,LZMGG
      COMMON/COMMA7/D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11,D12,D13,D14,D15,
     1D16,D17,D18,D19,D20,D21,D22,D23,D24
      REAL JL, JD, JU, MAL, MAD, MAU
      COMMON YL(112,20), YD(112), YU(112,20), NYL(112,20), NYU(112,20)
      COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84,27)
      COMMON MAL(84,27), MAD(84), MAU(84,27)
```

```
C
C
C
C       THIS SUBROUTINE SETS UP THE MATRIX COEFFICIENTS FOR
C       NODES ON THE THETA = PI POLAR AXIS.
C
C
C
      DA = D6*(A(LZMB)-A(LZMN))
      DB = D9*(A(LYMW)-A(LYMX)-A(LYMC)+A(LYMY))/4.0
      DC = D7*(A(LZPE)-A(LZME)-A(LZPR)+A(LZMR))/2.0
      DD = D12*(A(LYPK)-A(LYMK)-A(LYPDD)+A(LYMDD)-A(LYPG)+A(LYMG)+
     1A(LYPEE)-A(LYMEE))/8.0
      DE = D10*(A(LXMO)-A(LXMP))
      MAD(LJX) = DA+DB-DC-DD+DE
      DF = D10*(A(LXMW)-(2.0*A(LXMO))+A(LXMX)-A(LXMC)+(2.0*A(LXMP))-
     1A(LXMY))/4.0
      DG = DC/2.0
      DH = DA/2.0
      DI = 3.0*DE/2.0
      MAD(LJY) = DF+DG-DH-DB+DI
      DJ = DE/2.0
      MAD(LJZ) = DG+DD-DF-DH+DJ
      DK = D7*(A(LZMB)-A(LZMN))/2.0
      DL = D12*(A(LYMW)-A(LYMX)-A(LYMC)+A(LYMY))/8.0
      MAL(LJX,1) = -DK-DL
      DM = D10*(A(LXMW)-A(LXMX)-A(LYMC)+A(LXMY))/8.0
      DN = D7*(A(LZPA)-A(LZMA))/2.0
      DP = D9*(A(LYMO)-A(LYMP))/4.0
      DQ = D12*(A(LYPU)-A(LYMU)-A(LYPV)+A(LYMV))/8.0
      MAL(LJX,2) = DM+DN-DP+DQ-DJ
      DR = D8*(A(LZMO)-A(LZMN))/4.0
      DS = DB/(2.0*H2)
      DT = D8*A(LZ)/2.0
      DU = DF/H2
      MAL(LJX,3) = DR+DS+DT+DU
      DV = D10*(A(LZMW)-A(LZMX)-A(LZMC)+A(LZMY))/8.0
      DW = D10*(A(LZMO)-A(LZMP))/4.0
      DX = D11*(A(LYMO)-(2.0*A(LY))+A(LYMP))/2.0
      MAL(LJX,4) = -DV+DW+DX
      DY = D12*(A(LYPE)-A(LYME)-A(LYPR)+A(LYMR))/8.0
      DZ = D9*(A(LYMB)-A(LYMN))/4.0
      DDA = D10*(A(LXMB)-(2.0*A(LX))+A(LXMN))/2.0
      MAL(LJX,5) = DY+DZ-DDA
      DDB = D7*A(LZ)/4.0
      DDC = D12*(A(LYMO)-A(LYMP))/8.0
      MAL(LJX,6) = -DDB-DDC
      MAL(LJX,7) = -MAL(LJX,6)
      MAL(LJX,8) = 0.0
      MAL(LJX,9) = 0.0
      DDD = D12*(A(LYPA)-A(LYMA))/8.0
      DDE = D9*A(LY)/4.0
      DDF = D10*(A(LXMB)-A(LXMN))/8.0
      MAL(LJX,10) = DDD+DDE-DDF
      MAL(LJX,11) = -MAL(LJY,10)
      DO 770 NO1=12,17
      MAL(LJX,NO1) = 0.0
  770 CONTINUE
      MAL(LJX,18) = -DM-DN+DP-DQ-DJ
      MAL(LJX,19) = -DR-DS+DT+DU
      MAL(LJX,20) = DV+DW+DX
      MAL(LJX,21) = -MAL(LJX,5)
      MAL(LJX,22) = MAL(LJY,11)
      MAL(LJX,23) = MAL(LJX,10)
      DO 771 NO2=24,27
      MAL(LJX,NO2) = 0.0
  771 CONTINUE
      DDG = 4.0*H2*DM
      DDH = 2.0*D24*A(LZ)
      DDI = D4*(A(LZPA)-(2.0*A(LZ))+A(LZMA))
      DDJ = D12*(A(LXPK)-A(LXMK)-A(LXPDD)+A(LXMDD)-A(LXPG)+A(LXMG)+
     1A(LXPEE)-A(LXMEE))/8.0
      MAL(LJY,1) = -DDG-DDH+DDI+DDJ
      MAL(LJY,2) = MAL(LJX,1)
      DDK = DJ*H2
      DDL = D12*(A(LXPU)-A(LXMU)-A(LXPV)+A(LXMV))/8.0
      MAL(LJY,3) = DDK-DDL
      DDM = D7*(A(LZPA)-A(LZMA))/4.0
      DDN = D6*A(LZ)
      DDO = DI/2.0
      MAL(LJY,4) = DDM+DDN+DR-DDO
      DDP = D12*(A(LZPU)-A(LZMU)-A(LZPV)+A(LZMV))/8.0
      DDQ = D9*(A(LZMO)-A(LZMP))/4.0
```

```
      DDQ = D11*(A(LXMO)-(2.0*A(LX))+A(LXMP))/2.0
      MAL(LJY,5) = DDP+DDPQ-DDQ
      MAL(LJY,6) = MAL(LJX,5)
      MAL(LJY,7) = 0.0
      MAL(LJY,8) = 0.0
      MAL(LJY,9) = 0.0
      MAL(LJY,10) = MAL(LJX,6)
      MAL(LJY,11) = MAL(LJX,7)
      MAL(LJY,12) = MAL(LJX,10)
      MAL(LJY,13) = MAL(LJX,11)
      MAL(LJY,14) = 0.0
      MAL(LJY,15) = 0.0
      MAL(LJY,16) = -MAL(LJY,3)
      MAL(LJY,17) = -DDM-DDN-DP-DDO
      MAL(LJY,18) = -DDP-DDPQ-DDQ
      MAL(LJY,19) = -MAL(LJX,5)
      MAL(LJY,20) = 0.0
      MAL(LJY,21) = MAL(LJX,11)
      MAL(LJY,22) = MAL(LJX,10)
      DO 772 ND3=23,27
      MAL(LJY,ND3) = 0.0
  772 CONTINUE
      DDR = 2.0*D24*A(LY)
      DDS = D6*(A(LXMB)-A(LXMN))
      DDT = D4*(A(LYPA)-(2.0*A(LY))+A(LYMA))
      DDU = D7*(A(LXPE)-A(LXME)-A(LXPR)+A(LXMR))/4.0
      MAL(LJZ,1) = DDR-DDS-DDT+DDU
      MAL(LJZ,2) = MAL(LJX,1)
      MAL(LJZ,3) = -2.0*H3*MAL(LJX,10)
      DDV = 2.0*DM
      DDW = DDN/2.0
      DDX = DE/4.0
      MAL(LJZ,4) = DDV+DDW-DQ-DDX
      MAL(LJZ,5) = 0.0
      MAL(LJZ,6) = MAL(LJX,5)
      MAL(LJZ,7) = MAL(LJX,6)
      MAL(LJZ,8) = MAL(LJX,7)
      MAL(LJZ,9) = 0.0
      MAL(LJZ,10) = 0.0
      MAL(LJZ,11) = MAL(LJX,10)
      MAL(LJZ,12) = MAL(LJX,11)
      MAL(LJZ,13) = 0.0
      MAL(LJZ,14) = 0.0
      MAL(LJZ,15) = 0.0
      MAL(LJZ,16) = 0.0
      MAL(LJZ,17) = -MAL(LJZ,7)
      MAL(LJZ,18) = -DDV-DDW+DQ-DDX
      MAL(LJZ,19) = 0.0
      MAL(LJZ,20) = -MAL(LJX,5)
      MAL(LJZ,21) = 0.0
      MAL(LJZ,22) = 0.0
      MAL(LJZ,23) = MAL(LJX,11)
      MAL(LJZ,24) = MAL(LJX,10)
      MAL(LJZ,25) = 0.0
      MAL(LJZ,26) = 0.0
      MAL(LJZ,27) = 0.0
      DDY = 3.0*D3*(A(LZMB)-(2.0*A(LZ))+A(LZMN))/2.0
      DDZ = D10*(A(LYMW)-(2.0*A(LYMO))+A(LYMX)-A(LYMC)+(2.0*A(LYMP))-
     1A(LYMY))/4.0
      DEA = 2.0*DT
      DEB = 2.0*DU
      MAU(LJX,1) = -DDY-DDZ-DEA-DEB
      DEC = 3.0*D10*(A(LZMW)-(2.0*A(LZMO))+A(LZMX)-A(LZMC)+(2.0*A(LZMP))
     1-A(LZMY))/4.0
      DED = D11*(A(LYMW)-(2.0*A(LYMO))+A(LYMX)-(2.0*A(LYMB))+
     1(4.0*A(LY))-(2.0*A(LYMN))+A(LYMC)-(2.0*A(LYMP))+A(LYMY))/2.0
      DEE = 2.0*DW
      DEF = 2.0*DX
      MAU(LJX,2) = DEC+DED-DEE-DEF
      MAU(LJX,3) = -MAL(LJX,1)
      MAU(LJX,4) = MAL(LJX,7)
      MAU(LJX,5) = MAL(LJX,6)
      DO 773 ND4=6,27
      MAU(LJX,ND4) = 0.0
  773 CONTINUE
      DEG = D11*(A(LXMW)-(2.0*A(LXMO))+A(LYMX)-(2.0*A(LXMB))+
     1(4.0*A(LX))-(2.0*A(LXMN))+A(LXMC)-(2.0*A(LXMP))+A(LXMY))/2.0
      DEH = D12*(A(LZPK)-A(LZMK)-A(LZPDD)+A(LZMDD)-A(LZPG)+A(LZMG)+
     1A(LZPEE)-A(LZMEE))/8.0
      DEI = 2.0*H2*DV
      DEJ = 2.0*DDO
      MAU(LJY,1) = -DEG-DEH-DEI+DEJ
      MAU(LJY,2) = -MAL(LJX,1)
```

```
      MAU(LJY,3) = 0.0
      MAU(LJY,4) = 0.0
      MAU(LJY,5) = MAL(LJX,7)
      MAU(LJY,6) = MAL(LJX,6)
      DO 774 ND5=7,27
      MAU(LJY,ND5) = 0.0
  774 CONTINUE
      MAU(LJZ,1) = -MAL(LJX,1)
      MAU(LJZ,2) = MAL(LJX,7)
      MAU(LJZ,3) = MAL(LJX,6)
      DO 775 ND6=4,27
      MAU(LJZ,ND6) = 0.0
  775 CONTINUE
      RETURN
      END
      SUBROUTINE MATMA8
      COMMON/COMMA1/LJX,LJY,LJZ,LX,LY,LZ
      COMMON/COMMA2/NALPH3,N33,N330N2,N330N4,N390N4,H2,H3
      COMMON/COMMA3/NC23,NCN23,CP,SP,CPP,SPP,CPM,SPM
      COMMON/COMMA4/LXPA,LXPB,LXPC,LXPD,LXPE,LXPF,LXPG,LXPH,LXPI,LXPJ,
     1LXPK,LXPL,LXPM,LXPN,LXPO,LXPP,LXPQ,LXPR,LXPS,LXPT,LXPU,LXPV,LXPW,
     2LXPX,LXPY,LXPAA,LXPBB,LXPCC,LXPDD,LXPEE,LXPFF,LXPGG,
     3                LXMA,LXMB,LXMC,LXMD,LXME,LXMF,LXMG,LXMH,LXMI,LXMJ,
     4LXMK,LXML,LXMM,LXMN,LXMO,LXMP,LXMQ,LXMR,LXMS,LXMT,LXMU,LXMV,LXMW,
     5LXMX,LXMY,LXMAA,LXMBB,LXMCC,LXMDD,LXMEE,LXMFF,LXMGG
      COMMON/COMMA5/LYPA,LYPB,LYPC,LYPD,LYPE,LYPF,LYPG,LYPH,LYPI,LYPJ,
     1LYPK,LYPL,LYPM,LYPN,LYPO,LYPP,LYPQ,LYPR,LYPS,LYPT,LYPU,LYPV,LYPW,
     2LYPX,LYPY,LYPAA,LYPBB,LYPCC,LYPDD,LYPEE,LYPFF,LYPGG,
     3                LYMA,LYMB,LYMC,LYMD,LYME,LYMF,LYMG,LYMH,LYMI,LYMJ,
     4LYMK,LYML,LYMM,LYMN,LYMO,LYMP,LYMQ,LYMR,LYMS,LYMT,LYMU,LYMV,LYMW,
     5LYMX,LYMY,LYMAA,LYMBB,LYMCC,LYMDD,LYMEE,LYMFF,LYMGG
      COMMON/COMMA6/LZPA,LZPB,LZPC,LZPD,LZPE,LZPF,LZPG,LZPH,LZPI,LZPJ,
     1LZPK,LZPL,LZPM,LZPN,LZPO,LZPP,LZPQ,LZPR,LZPS,LZPT,LZPU,LZPV,LZPW,
     2LZPX,LZPY,LZPAA,LZPBB,LZPCC,LZPDD,LZPEE,LZPFF,LZPGG,
     3                LZMA,LZMB,LZMC,LZMD,LZME,LZMF,LZMG,LZMH,LZMI,LZMJ,
     4LZMK,LZML,LZMM,LZMN,LZMO,LZMP,LZMQ,LZMR,LZMS,LZMT,LZMU,LZMV,LZMW,
     5LZMX,LZMY,LZMAA,LZMBB,LZMCC,LZMDD,LZMEE,LZMFF,LZMGG
C
C
C
C           THIS SUBROUTINE SETS UP THE ARGUMENTS OF THE
C           VECTOR POTENTIAL FIELD A.
C
C
C
      LXPA = LX + NALPH3
      LYPA = LY + NALPH3
      LZPA = LZ + NALPH3
      LXMA = LX - NALPH3
      LYMA = LY - NALPH3
      LZMA = LZ - NALPH3
      LXPB = LX + N33
      LYPB = LY + N33
      LZPB = LZ + N33
      LXMB = LX - N33
      LYMB = LY - N33
      LZMB = LZ - N33
      LXPC = LX + 3
      LYPC = LY + 3
      LZPC = LZ + 3
      LXMC = LX - 3
      LYMC = LY - 3
      LZMC = LZ - 3
      LXPD = LXPB + NALPH3
      LYPD = LYPB + NALPH3
      LZPD = LZPB + NALPH3
      LXMD = LXPB - NALPH3
      LYMD = LYPB - NALPH3
      LZMD = LZPB - NALPH3
      LXPE = LXMB + NALPH3
      LYPE = LYMB + NALPH3
      LZPE = LZMB + NALPH3
      LXME = LXMB - NALPH3
      LYME = LYMB - NALPH3
      LZME = LZMB - NALPH3
      LXPF = LXPC + NALPH3
      LYPF = LYPC + NALPH3
      LZPF = LZPC + NALPH3

      LXMF = LXPC - NALPH3
      LYMF = LYPC - NALPH3
      LZMF = LZPC - NALPH3
      LXPG = LXMC + NALPH3
      LYPG = LYMC + NALPH3
```

```
LZPG = LZMC + NALPH3
LXMG = LXMC - NALPH3
LYMG = LYMC - NALPH3
LZMG = LZMC - NALPH3
LXPH = LXPC + N33
LYPH = LYPC + N33
LZPH = LZPC + N33
LXMH = LXPC - N33
LYMH = LYPC - N33
LZMH = LZPC - N33
LXPI = LXMC + N33
LYPI = LYMC + N33
LZPI = LZMC + N33
LXMI = LXMC - N33
LYMI = LYMC - N33
LZMI = LZMC - N33
LXPJ = LXPH + NALPH3
LYPJ = LYPH + NALPH3
LZPJ = LZPH + NALPH3
LXMJ = LXPH - NALPH3
LYMJ = LYPH - NALPH3
LZMJ = LZPH - NALPH3
LXPK = LXMH + NALPH3
LYPK = LYMH + NALPH3
LZPK = LZMH + NALPH3
LXMK = LXMH - NALPH3
LYMK = LYMH - NALPH3
LZMK = LZMH - NALPH3
LXPL = LXPI + NALPH3
LYPL = LYPI + NALPH3
LZPL = LZPI + NALPH3
LXML = LXPI - NALPH3
LYML = LYPI - NALPH3
LZML = LZPI - NALPH3
LXPM = LXMI + NALPH3
LYPM = LYMI + NALPH3
LZPM = LZMI + NALPH3
LXMM = LXMI - NALPH3
LYMM = LYMI - NALPH3
LZMM = LZMI - NALPH3
LXPN = LX + N33ON2 + 3
LYPN = LY + N33ON2 + 3
LZPN = LZ + N33ON2 + 3
LXMN = LX - N33ON2
LYMN = LY - N33ON2
LZMN = LZ - N33ON2
LXPO = LX + N33ON4 + 3
LYPO = LY + N33ON4 + 3
LZPO = LZ + N33ON4 + 3
LXMO = LX - N39ON4
LYMO = LY - N39ON4
LZMO = LZ - N39ON4
LXPP = LX + N39ON4 + 3
LYPP = LY + N39ON4 + 3
LZPP = LZ + N39ON4 + 3
LXMP = LX - N33ON4
LYMP = LY - N33ON4
LZMP = LZ - N33ON4
LXPQ = LXPN + NALPH3
LYPQ = LYPN + NALPH3
LZPQ = LZPN + NALPH3
LXMQ = LXPN - NALPH3
LYMQ = LYPN - NALPH3
LZMQ = LZPN - NALPH3
LXPR = LXMN + NALPH3
LYPR = LYMN + NALPH3
LZPR = LZMN + NALPH3
LXMR = LXMN - NALPH3
LYMR = LYMN - NALPH3
LZMR = LZMN - NALPH3
LXPS = LXPO + NALPH3
LYPS = LYPO + NALPH3
LZPS = LZPO + NALPH3
LXMS = LXPO - NALPH3
LYMS = LYPO - NALPH3
LZMS = LZPO - NALPH3
LXPT = LXPP + NALPH3
LYPT = LYPP + NALPH3
LZPT = LZPP + NALPH3
LXMT = LXPP - NALPH3
LYMT = LYPP - NALPH3
LZMT = LZPP - NALPH3
LXPU = LXMO + NALPH3
```

```
LYPU = LYMO + NALPH3
LZPU = LZMO + NALPH3
LXMU = LXMO - NALPH3
LYMU = LYMO - NALPH3
LZMU = LZMO - NALPH3
LXPV = LXMP + NALPH3
LYPV = LYMP + NALPH3
LZPV = LZMP + NALPH3
LXMV = LXMP - NALPH3
LYMV = LYMP - NALPH3
LZMV = LZMP - NALPH3
LXPW = LX + 6
LYPW = LY + 6
LZPW = LZ + 6
LXMW = LX - N33 + 3
LYMW = LY - N33 + 3
LZMW = LZ - N33 + 3
LXPX = LX + N330M2 + 6
LYPX = LY + N330M2 + 6
LZPX = LZ + N330M2 + 6
LXMX = LX - N330M2 + 3
LYMX = LY - N330M2 + 3
LZMX = LZ - N330M2 + 3
LXPY = LX + N330M2
LYPY = LY + N330M2
LZPY = LZ + N330M2
LXMY = LX - N330M2 - 3
LYMY = LY - N330M2 - 3
LZMY = LZ - N330M2 - 3
LXPAA = LXPW + NALPH3
LYPAA = LYPW + NALPH3
LZPAA = LZPW + NALPH3
LXMAA = LXPW - NALPH3
LYMAA = LYPW - NALPH3
LZMAA = LZPW - NALPH3
LXPBB = LXPX + NALPH3
LYPBB = LYPX + NALPH3
LZPBB = LZPX + NALPH3
LXMBB = LXPX - NALPH3
LYMBB = LYPX - NALPH3
LZMBB = LZPX - NALPH3
LXPCC = LXPQ - 3
LYPCC = LYPQ - 3
LZPCC = LZPQ - 3
LXMCC = LXMQ - 3
LYMCC = LYMQ - 3
LZMCC = LZMQ - 3
LXPDD = LXPR + 3
LYPDD = LYPR + 3
LZPDD = LZPR + 3
LXMDD = LXMR + 3
LYMDD = LYMR + 3
LZMDD = LZMR + 3
LXPEE = LXPR - 3
LYPEE = LYPR - 3
LZPEE = LZPR - 3
LXMEE = LXMR - 3
LYMEE = LYMR - 3
LZMEE = LZMR - 3
LXPFF = NC23 + NALPH3 - 2
LYPFF = LXPFF + 1
LZPFF = LXPFF + 2
LXMFF = NC23 - NALPH3 - 2
LYMFF = LXMFF + 1
LZMFF = LXMFF + 2
LXPGG = NCN23 + NALPH3 - 2
LYPGG = LXPGG + 1
LZPGG = LXPGG + 2
LXMGG = NCN23 - NALPH3 - 2
LYMGG = LXMGG + 1
LZMGG = LXMGG + 2
RETURN
END
```

```
      SUBROUTINE MATRIX (NN1,NN2,NN3,RRI,RRO,UU0,SSHEAR,E,F,NE,NF,B,
     1NNEQNS)
      DIMENSION E(4,NNEQNS),F(4,NNEQNS),NE(4,NNEQNS),NF(4,NNEQNS)
      DIMENSION B(NNEQNS)
C
C
C
C         THIS SUBROUTINE SETS UP THE MATRIX EQUATION ANALOGUE
C         OF LAPLACE'S EQUATION WITH NEUMANN BOUNDARY CONDITIONS
C         DESCRIBING THE POTENTIAL FLOW OF A LINEAR-SHEAR (AT
C         INFINITY) VELOCITY FIELD PAST A SPHERE OF RADIUS RI.
C         THE BOUNDARY CONDITIONS AT INFINITY ARE REPLACED BY
C         SIMILAR CONDITIONS ON A SPHERE OF RADIUS RO
C         CONCENTRIC WITH THE SPHERE OF RADIUS RI. SECOND-ORDER
C         FINITE-DIFFERENCE APPROXIMATIONS ARE USED BOTH FOR
C         THE LAPLACIAN AND FOR THE NEUMANN BOUNDARY CONDITIONS.
C
C
C
      N1 = NN1
      N2 = NN2
      N3 = NN3
      RI = RRI
      RO = RRO
      U0 = UU0
      SHEAR = SSHEAR
      NEQNS = NNEQNS
      PI = 3.14159265358898
      XN1 = N1
      XN2 = N2
      XN3 = N3
      H1 = (RO - RI)/XN1
      H2 = PI/XN2
      H3 = (2.0*PI)/XN3
      NALPHA = (N3*(N2 - 1))+2
C        INNER BOUNDARY CONDITIONS.
      E(1,1) = 0.0
      E(2,1) = 0.0
      E(3,1) = 0.0
      E(4,1) = 0.0
      F(1,1) = 0.0
      F(2,1) = 0.0
      F(3,1) = -4.0/3.0
      F(4,1) =  1.0/3.0
      NE(1,1) = NEQNS + 1
      NE(2,1) = NEQNS + 1
      NE(3,1) = NEQNS + 1
      NE(4,1) = NEQNS + 1
      NF(1,1) = NEQNS + 1
      NF(2,1) = NEQNS + 1
      NF(3,1) = 1 + NALPHA
      NF(4,1) = 1 + (2*NALPHA)
      B(1) = 0.0
      I = 1
      J = 2
      K = 1
      N = 2
    1 E(1,N) = 0.0
      E(2,N) = 0.0
      E(3,N) = 0.0
      E(4,N) = 0.0
      F(1,N) = 0.0
      F(2,N) = 0.0
      F(3,N) = -4.0/3.0
      F(4,N) =  1.0/3.0
      NE(1,N) = NEQNS + 1
      NE(2,N) = NEQNS + 1
      NE(3,N) = NEQNS + 1
      NE(4,N) = NEQNS + 1
      NF(1,N) = NEQNS + 1
      NF(2,N) = NEQNS + 1
      NF(3,N) = N + NALPHA
      NF(4,N) = N + (2*NALPHA)
      B(N) = 0.0
      K = K + 1
      N = N + 1
      IF(K.LT.N3+1) GO TO 1
      K = 1
      J = J + 1
      IF(J.LT.N2+1) GO TO 1
      E(1,N) = 0.0
      E(2,N) = 0.0
```

```
      E(3,N) = 0.0
      E(4,N) = 0.0
      F(1,N) = 0.0
      F(2,N) = 0.0
      F(3,N) = -4.0/3.0
      F(4,N) =  1.0/3.0
      NE(1,N) = NEQNS + 1
      NE(2,N) = NEQNS + 1
      NE(3,N) = NEQNS + 1
      NE(4,N) = NEQNS + 1
      NF(1,N) = NEQNS + 1
      NF(2,N) = NEQNS + 1
      NF(3,N) = N + NALPHA
      NF(4,N) = N + (2*NALPHA)
      B(N) = 0.0
      I = 2
      J = 1
      K = 1
      N = N + 1
   16 X = I - 1
      R = (X*H1) + RI
      P1 = 1.0/(H1*H1)
      P2 = 1.0/(R*H1)
      P3 = 2.0/(R*R*H2*H2)
      P = -2.0*(P1 + P3)
C        INTERIOR NODES (THETA = 0 POLAR AXIS).
      E(1,N) = 0.0
      E(2,N) = (P1 - P2)/P
      E(3,N) = 0.0
      E(4,N) = 0.0
      F(1,N) = 0.0
      F(2,N) = P3/P
      F(3,N) = (P1 + P2)/P
      F(4,N) = P3/P
      NE(1,N) = NEQNS + 1
      NE(2,N) = N - NALPHA
      NE(3,N) = NEQNS + 1
      NE(4,N) = NEQNS + 1
      NF(1,N) = NEQNS + 1
      NF(2,N) = N + 1
      NF(3,N) = N + NALPHA
      NF(4,N) = N + 1 + (N3/2)
      B(N) = 0.0
      J = 2
      Y = J - 1
      THETA = (Y*PI)/XN2
      S = SIN(THETA)
      C = COS(THETA)
      N = N + 1
    4 T1 = 1.0/(H1*H1)
      T2 = 1.0/(R*R*H2*H2)
      T3 = 1.0/(R*R*H3*H3*S*S)
      T4 = 1.0/(R*H1)
      T5 = C/(2.0*S*R*R*H2)
      T = -2.0*(T1+T2+T3)
C        INTERIOR NODES (ADJACENT TO THE THETA = 0 POLAR AXIS).
      E(1,N) = 0.0
      E(2,N) = (T1-T4)/T
      E(3,N) = (T2-T5)/T
      E(4,N) = T3/T
      F(1,N) = T3/T
      F(2,N) = (T2+T5)/T
      F(3,N) = (T1+T4)/T
      F(4,N) = 0.0
      NE(1,N) = NEQNS + 1
      NE(2,N) = N - NALPHA
      NE(3,N) = 1 + (NALPHA*(I - 1))
      NE(4,N) = N - 1
      NF(1,N) = N + 1
      NF(2,N) = N + N3
      NF(3,N) = N + NALPHA
      NF(4,N) = NEQNS + 1
      IF (K.EQ.1) GO TO 20
      IF(K.EQ.N3) GO TO 21
      GO TO 22
   20 E(4,N) = 0.0
      F(4,N) = T3/T
      NE(4,N) = NEQNS + 1
      NF(4,N) = N - 1 + N3
      GO TO 22
   21 E(1,N) = T3/T
      F(1,N) = 0.0
```

```
      NE(1,N) = N + 1 - N3
      NF(1,N) = NEQNS + 1
   22 B(N) = 0.0
      K = K + 1
      N = N + 1
      IF(K.LT.N3+1) GO TO 4
      K = 1
      J = J + 1
      Y = J - 1
      THETA = (PI*Y)/XN2
      S = SIN(THETA)
      C = COS(THETA)
      IF(J.EQ.N2) GO TO 5
    9 T1 = 1.0/(H1*H1)
      T2 = 1.0/(R*R*H2*H2)
      T3 = 1.0/(R*R*H3*H3*S*S)
      T4 = 1.0/(R*H1)
      T5 = C/(2.0*S*R*R*H2)
      T = -2.0*(T1+T2+T3)
C        INTERIOR NODES (GENERAL CASE).
      E(1,N) = 0.0
      E(2,N) = (T1-T4)/T
      E(3,N) = (T2-T5)/T
      E(4,N) = T3/T
      F(1,N) = T3/T
      F(2,N) = (T2+T5)/T
      F(3,N) = (T1+T4)/T
      F(4,N) = 0.0
      NE(1,N) = NEQNS + 1
      NE(2,N) = N - NALPHA
      NE(3,N) = N - N3
      NE(4,N) = N - 1
      NF(1,N) = N + 1
      NF(2,N) = N + N3
      NF(3,N) = N + NALPHA
      NF(4,N) = NEQNS + 1
      IF(K.EQ.1) GO TO 6
      IF(K.EQ.N3) GO TO 7
      GO TO 8
    6 E(4,N) = 0.0
      F(4,N) = T3/T
      NE(4,N) = NEQNS + 1
      NF(4,N) = N - 1 + N3
      GO TO 8
    7 E(1,N) = T3/T
      F(1,N) = 0.0
      NE(1,N) = N + 1 - N3
      NF(1,N) = NEQNS + 1
    8 B(N) = 0.0
      K = K + 1
      N = N + 1
      IF(K.LT.N3+1) GO TO 9
      K = 1
      J = J + 1
      Y = J - 1
      THETA = (PI*Y)/XN2
      S = SIN(THETA)
      C = COS(THETA)
      IF(J.EQ.N2) GO TO 5
      GO TO 9
    5 T1 = 1.0/(H1*H1)
      T2 = 1.0/(R*R*H2*H2)
      T3 = 1.0/(R*R*H3*H3*S*S)
      T4 = 1.0/(R*H1)
      T5 = C/(2.0*S*R*R*H2)
      T = -2.0*(T1+T2+T3)
C        INTERIOR NODES (ADJACENT TO THE THETA = PI POLAR AXIS).
      E(1,N) = 0.0
      E(2,N) = (T1-T4)/T
      E(3,N) = (T2-T5)/T
      E(4,N) = T3/T
      F(1,N) = T3/T
      F(2,N) = (T2+T5)/T
      F(3,N) = (T1+T4)/T
      F(4,N) = 0.0
      NE(1,N) = NEQNS + 1
      NE(2,N) = N - NALPHA
      NE(3,N) = N - N3
      NE(4,N) = N - 1
      NF(1,N) = N + 1
      NF(2,N) = NALPHA
      NF(3,N) = N + NALPHA
      NF(4,N) = NEQNS + 1
```

```
      IF(K.EQ.1) GO TO 10
      IF(K.EQ.N3) GO TO 11
      GO TO 12
   10 E(4,N) = 0.0
      F(4,N) = T3/T
      NE(4,N) = NEQNS + 1
      NF(4,N) = N - 1 + N3
      GO TO 12
   11 E(1,N) = T3/T
      F(1,N) = 0.0
      NE(1,N) = N + 1 - N3
      NF(1,N) = NEQNS + 1
   12 B(N) = 0.0
      K = K + 1
      N = N + 1
      IF(K.LT.N3+1) GO TO 5
      K = 1
      J = N2 + 1
      P1 = 1.0/(H1*H1)
      P2 = 1.0/(R*H1)
      P3 = 2.0/(R*R*H2*H2)
      P = -2.0*(P1 + P3)
C        INTERIOR NODES (THETA = PI POLAR AXIS).
      E(1,N) = P3/P
      E(2,N) = (P1 - P2)/P
      E(3,N) = P3/P
      E(4,N) = 0.0
      F(1,N) = 0.0
      F(2,N) = 0.0
      F(3,N) = (P1 + P2)/P
      F(4,N) = 0.0
      NE(1,N) = N - (N3/2)
      NE(2,N) = N - NALPHA
      NE(3,N) = N - N3
      NE(4,N) = NEQNS + 1
      NF(1,N) = NEQNS + 1
      NF(2,N) = NEQNS + 1
      NF(3,N) = N + NALPHA
      NF(4,N) = NEQNS + 1
      B(N) = 0.0
      I = I + 1
      J = 1
      K = 1
      N = N + 1
      IF(I.EQ.N1+1) GO TO 15
      GO TO 16
C        OUTER BOUNDARY CONDITIONS.
   15 E(1,N) =  1.0/3.0
      E(2,N) = -4.0/3.0
      E(3,N) = 0.0
      E(4,N) = 0.0
      F(1,N) = 0.0
      F(2,N) = 0.0
      F(3,N) = 0.0
      F(4,N) = 0.0
      NE(1,N) = N - (2*NALPHA)
      NE(2,N) = N - NALPHA
      NE(3,N) = NEQNS + 1
      NE(4,N) = NEQNS + 1
      NF(1,N) = NEQNS + 1
      NF(2,N) = NEQNS + 1
      NF(3,N) = NEQNS + 1
      NF(4,N) = NEQNS + 1
      B(N) = 2.0*H1*U0/3.0
      J = 2
      Y = J - 1
      THETA = (Y*PI)/XN2
      S = SIN(THETA)
      C = COS(THETA)
      N = N + 1
   17 E(1,N) =  1.0/3.0
      E(2,N) = -4.0/3.0
      E(3,N) = 0.0
      E(4,N) = 0.0
      F(1,N) = 0.0
      F(2,N) = 0.0
      F(3,N) = 0.0
      F(4,N) = 0.0
      NE(1,N) = N - (2*NALPHA)
      NE(2,N) = N - NALPHA
      NE(3,N) = NEQNS + 1
      NE(4,N) = NEQNS + 1
```

```
NF(1,N) = NEQNS + 1
NF(2,N) = NEQNS + 1
NF(3,N) = NEQNS + 1
NF(4,N) = NEQNS + 1
Z = K - 1
PHI = (2.0*PI*Z)/XN3
CP = COS(PHI)
B(N) = 2.0*H1*(U0 + (SHEAR*R0*S*CP))*C/3.0
K = K + 1
N = N + 1
IF(K.LT.N3+1) GO TO 17
J = J + 1
Y = J - 1
THETA = (PI*Y)/XN2
S = SIN(THETA)
C = COS(THETA)
K = 1
IF(J.LT.N2+1) GO TO 17
RETURN
END
```

```
      SUBROUTINE MATRJ(NNEQNJ,NN1,NN2,NN3,RRI,RRO,U,H,NNODES,NNODE3,RRE
      DIMENSION U(NNODE3), H(NNODES)
      COMMON/CCMJ1/REINV,NALPH3,N33,N330N2,N330N4,N390N4,LX,LY,LZ,
     1LJX,LJY,LJZ
      COMMON/CCMJ2/D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11,D12
      COMMON/CCMJ3/D13,D14,D15,D16,D17,D18,D19,D20,D21,D22,D23
      COMMON/CCMJ4/DA,DB,DC,DD,DE,DF,DG
      COMMON/CCMJ5/DH,DI,DJ,DK,DL
      COMMON/CCMJ6/DM,DN,DP,DQ,DR,DS
      COMMON/CCMJ7/DT,DU,DV,DW,DX,DY,DZ
      COMMON/CCMJ8/NC23,NCN23,CP,SP,CPP,SPP,CPM,SPM
      REAL JL, JD, JU, MAL, MAD, MAU
      COMMON YL(112,20), YD(112), YU(112,20), MYL(112,20), MYU(112,20)
      COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84,27)
      COMMON MAL(84,27), MAD(84), MAU(84,27)
C
C
C
C         THIS SUBROUTINE SETS UP THOSE TERMS IN THE MATRIX
C         EQUATION ANALOGUE OF THE VORTICITY TRANSPORT
C         EQUATION WHICH ARE INDEPENDENT OF THE VECTOR POTENTIAL
C         FIELD A. IN ORDER TO ALLOW THE SIZE OF THE TIME-STEP
C         TO BE VARIED EFFICIENTLY, THE DIAGONAL MATRIX
C         COEFFICIENTS JD ARE SET UP WITHOUT THE RECIPROCAL
C         TIME-STEP TERM INCLUDED.
C         SUBROUTINES MATRJ1-7 SET UP THE MATRIX COEFFICIENTS
C         AND NODE LOCATIONS JL/JD/JU AND NL/NU RESPECTIVELY.
C
C
C
      NEQNJ = NNEQNJ
      N1 = NN1
      N2 = NN2
      N3 = NN3
      NODES = NNODES
      NODE2 = 2*NODES
      NODE3 = NNODE3
      RE = RRE
      REINV = 1.0/RE
      RI = RRI
      RO = RRO
      NALPHA = NODES/(N1+1)
      NALPH3 = 3*NALPHA
      N3ON2 = N3/2
      N33 = 3*N3
      N330N2 = N33/2
      N330N4 = N33/4
      N390N4 = 3*N330N4
      PI = 3.14159265358 98
      XN1 = N1
      XN2 = N2
      XN3 = N3
      H1 = (RO - RI)/XN1
      H2 = PI/XN2
      H3 = 2.0*PI/XN3
      L = NALPH3 + 1
      LX = 1 + NALPH3
      LY = 2 + NALPH3
      LZ = 3 + NALPH3
      LJX = 1
      LJY = 2
      LJZ = 3
      I = 2
      R = RI + H1
  207 J = 1
      K = 1
      THETA = 0.0
      PHI = 0.0
      NC2 = (NALPHA*(I-1)) + 1
      NC23 = 3*NC2
      NCN2 = NALPHA*I
      NCN23 = 3*NCN2
      D1 = 1.0/R
      D2 = 1.0/H1
      D3 = D1*D2
      D4 = D2*D2
      D5 = D1/H2
      D6 = D1*D5
      D7 = D2*D5
      D8 = D5*D5
      D9 = D6/H3
      D10 = D9/H2
```

```
      D11 = D10/H3
      D12 = D7/H3
      DA = U(L)
      DB = U(L+NODES)
      DC = U(L+NODE2)
      DD = H(L+NALPHA+1)-H(L-NALPHA+1)-H(L+NALPHA+N3ON2+1)+
     1H(L-NALPHA+N3ON2+1)
      DE = H(L+1)-H(L+N3ON2+1)
      DT = H(L+1)-(2.0*H(L))+H(L+N3ON2+1)
      DU = H(L+NALPHA)-(2.0*H(L))+H(L-NALPHA)
C        INTERIOR NODES (THETA = 0 POLAR AXIS).
      CALL MATRI1
      L = L + 1
      LX = LX + 3
      LY = LX + 1
      LZ = LY + 1
      LJX = LJX + 3
      LJY = LJX + 1
      LJZ = LJY + 1
      J = 2
      THETA = PI/XN2
  288 ST = SIN(THETA)
      TT = TAN(THETA)
      PHIP = PHI + H3
      PHIM = PHI - H3
      CP = COS(PHI)
      SP = SIN(PHI)
      CPP = COS(PHIP)
      SPP = SIN(PHIP)
      CPM = COS(PHIM)
      SPM = SIN(PHIM)
      D13 = D1/TT
      D14 = D13*D1
      D15 = D13*D2
      D16 = D13*D5
      D17 = D1*D1/(ST*ST)
      D18 = D1/(ST*H3)
      D19 = D18*D1
      D20 = D19/(ST*H3)
      D21 = D18*D2
      D22 = D19/H2
      D23 = D19/TT
  289 DA = U(L)
      DB = U(L+NODES)
      DC = U(L+NODE2)
      LP = L + NALPHA + N3
      LM = L - NALPHA - N3
      IF(LP.GT.NODES) GO TO 2895
      IF(LM.LT.1) GO TO 2895
      DH = H(L+NALPHA+N3)-H(L-NALPHA+N3)-H(L+NALPHA-N3)+H(L-NALPHA-N3)
      GO TO 2896
 2895 DH = 0.0
 2896 DI = H(L+N3)-H(L-N3)
      DJ = H(L+1)-H(L-1)
      DK = H(L+NALPHA+1)-H(L-NALPHA+1)-H(L+NALPHA-1)+H(L-NALPHA-1)
      DL = H(L+N3+1)-H(L-N3+1)-H(L+N3-1)+H(L-N3-1)
      DV = H(L+NALPHA)-(2.0*H(L))+H(L-NALPHA)
      DW = H(L+N3)-(2.0*H(L))+H(L-N3)
      DX = H(L+1)-(2.0*H(L))+H(L-1)
      IF(K.NE.1) GO TO 290
      DJ = H(L+1)-H(L-1+N3)
      DK = H(L+NALPHA+1)-H(L-NALPHA+1)-H(L+NALPHA-1+N3)+H(L-NALPHA-1+N3
      DL = H(L+N3+1)-H(L-N3+1)-H(L+N3-1+N3)+H(L-1)
      DX = H(L+1)-(2.0*H(L))+H(L-1+N3)
  290 IF(K.NE.N3) GO TO 291
      DJ = H(L+1-N3)-H(L-1)
      DK = H(L+NALPHA+1-N3)-H(L-NALPHA+1-N3)-H(L+NALPHA-1)+H(L-NALPHA-1
      DL = H(L+1)-H(L-N3+1-N3)-H(L+N3-1)+H(L-N3-1)
      DX = H(L+1-N3)-(2.0*H(L))+H(L-1)
  291 IF(J.NE.2) GO TO 292
      DH = H(L+NALPHA+N3)-H(L-NALPHA+N3)-H(NC2+NALPHA)+H(NC2-NALPHA)
      DI = H(L+N3)-H(NC2)
      DP = H(L+N3+1)-H(L+N3-1)
      DY = H(L+N3)-(2.0*H(L))+H(NC2)
  292 IF(J.NE.N2) GO TO 293
      DQ = H(NCN2+NALPHA)-H(NCN2-NALPHA)-H(L+NALPHA-N3)+H(L-NALPHA-N3)
      DR = H(NCN2)-H(L-N3)
      DS = -H(L-N3+1)+H(L-N3-1)
      DZ = H(NCN2)-(2.0*H(L))+H(L-N3)
  293 IF((J.NE.2).OR.(K.NE.1)) GO TO 294
      DP = H(L+N3+1)-H(L+N3-1+N3)
  294 IF((J.NE.2).OR.(K.NE.N3)) GO TO 295
      DP = H(L+1)-H(L+N3-1)
```

```
  295 IF((J.NE.N2).OR.(K.NE.1)) GO TO 296
      DS = -H(L-N3+1)+H(L-1)
  296 IF((J.NE.N2).OR.(K.NE.N3)) GO TO 297
      DS = -H(L-N3+1-N3)+H(L-N3-1)
C         INTERIOR NODES (GENERAL CASE).
  297 CALL MATRJ2
      IF(K.NE.1) GO TO 298
C         INTERIOR NODES (K = 1).
      CALL MATRJ3
  298 IF(K.NE.N3) GO TO 299
C         INTERIOR NODES (K = N3).
      CALL MATRJ4
  299 IF(J.NE.2) GO TO 300
C         INTERIOR NODES (J = 2).
      CALL MATRJ5
  300 IF(J.NE.N2) GO TO 301
C         INTERIOR NODES (J = N2).
      CALL MATRJ6
  301 L = L + 1
      LX = LX + 3
      LY = LX + 1
      LZ = LY + 1
      LJX = LJX + 3
      LJY = LJX + 1
      LJZ = LJY + 1
      IF(K.EQ.N3) GO TO 302
      K = K + 1
      Z = K - 1
      PHI = (2.0*Z*PI)/XN3
      PHIP = PHI + H3
      PHIM = PHI - H3
      CP = COS(PHI)
      SP = SIN(PHI)
      CPP = COS(PHIP)
      SPP = SIN(PHIP)
      CPM = COS(PHIM)
      SPM = SIN(PHIM)
      GO TO 289
  302 K = 1
      PHI = 0.0
      IF(J.EQ.N2) GO TO 303
      J = J + 1
      Y = J - 1
      THETA = Y*PI/XN2
      GO TO 288
  303 DA = U(L)
      DB = U(L+NODES)
      DC = U(L+NODE2)
      DF = H(L+NALPHA-N3)-H(L-NALPHA-N3)-H(L+NALPHA-N3ON2)+
     1H(L-NALPHA-N3ON2)
      DS = H(L-N3)-H(L-N3ON2)
      DU = H(L+NALPHA)-(2.0*H(L))+H(L-NALPHA)
      DV = H(L-N3)-(2.0*H(L))+H(L-N3ON2)
C          INTERIOR NODES (THETA = PI POLAR AXIS).
      CALL MATRJ7
      L = L + 1
      LX = LX + 3
      LY = LX + 1
      LZ = LY + 1
      LJX = LJX + 3
      LJY = LJX + 1
      LJZ = LJY + 1
      IF(I.EQ.N1) GO TO 304
      I = I + 1
      R = R + H1
      GO TO 287
  304 RETURN
      END


       SUBROUTINE MATRJ1
       COMMON/COMJ1/REINV,NALPH3,N33,N33ON2,N33ON4,N39ON4,LX,LY,LZ,
      1LJX,LJY,LJZ
       COMMON/COMJ2/D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11,D12
       COMMON/COMJ4/DA,DB,DC,DD,DE,DF,DG
       COMMON/COMJ7/DT,DU,DV,DW,DX,DY,DZ
       REAL JL, JD, JU, MAL, MAD, MAU
       COMMON YL(112,20), YD(112), YU(112,20), NYL(112,20), NYU(112,20)
       COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84,27)
       COMMON MAL(84,27), MAD(84), MAU(84,27)
```

```
C
C
C
C     THIS SUBROUTINE SETS UP THE MATRIX COEFFICIENTS AND
C     NODE LOCATIONS FOR NODES ON THE THETA = 0 POLAR AXIS.
C
C
C
      JD(LJX) = (2.0*D1*DA)+(2.0*D8*DT)+(((4.0*D8)-(2.0*D11))*REINV)
      JD(LJY) = (D1*DA)+(D8*DT)+(D4*DU)+(((2.0*D4)-(2.0*D11))*REINV)
      JD(LJZ) = (D1*DA)+(D8*DT)+(D4*DU)+(((2.0*D4)+(3.0*D8))*REINV)
      JL(LJX,1) = -D2*DA/2.0
      NL(LJX,1) = LX - NALPH3
      JL(LJX,2) = 0.0
      NL(LJX,2) = LX - NALPH3 + 3
      JL(LJX,3) = 0.0
      NL(LJX,3) = LX - NALPH3 + N33ON2 + 3
      JL(LJX,4) = -REINV*D7/2.0
      NL(LJX,4) = LY - NALPH3 + 3
      JL(LJX,5) = -JL(LJX,4)
      NL(LJX,5) = LY - NALPH3 + N33ON2 + 3
      JL(LJX,6) = -REINV*D12/8.0
      NL(LJX,6) = LZ - NALPH3 + 6
      JL(LJX,7) = -JL(LJX,6)
      NL(LJX,7) = LZ - NALPH3 + N33ON2 + 6
      JL(LJX,8) = -JL(LJX,6)
      NL(LJX,8) = LZ - NALPH3 + N33
      JL(LJX,9) = JL(LJX,6)
      NL(LJX,9) = LZ - NALPH3 + N33ON2
      DO 310 ND1=10,27
      JL(LJX,ND1) = 0.0
      NL(LJX,ND1) = 1
  310 CONTINUE
      JL(LJY,1) = (D1*DA)-(D7*DD/4.0)+(D6*DE/2.0)
      NL(LJY,1) = LX
      JL(LJY,2) = -(D2*DA/2.0)+((D3-D4)*REINV)
      NL(LJY,2) = LY - NALPH3
      JL(LJY,3) = JL(LJX,4)/2.0
      NL(LJY,3) = LX - NALPH3 + 3
      JL(LJY,4) = -JL(LJY,3)
      NL(LJY,4) = LX - NALPH3 + N33ON2 + 3
      JL(LJY,5) = 0.0
      NL(LJY,5) = LY - NALPH3 + 3
      JL(LJY,6) = 0.0
      NL(LJY,6) = LY - NALPH3 + N33ON2 + 3
      DO 311 ND2=7,27
      JL(LJY,ND2) = 0.0
      NL(LJY,ND2) = 1
  311 CONTINUE
      JL(LJZ,1) = 0.0
      NL(LJZ,1) = LX
      JL(LJZ,2) = -(D2*DA/2.0)-((D4-D3)*REINV)
      NL(LJZ,2) = LZ - NALPH3
      JL(LJZ,3) = 0.0
      NL(LJZ,3) = LZ - NALPH3 + 3
      JL(LJZ,4) = 0.0
      NL(LJZ,4) = LZ - NALPH3 + N33ON2 + 3
      JL(LJZ,5) = JL(LJX,6)
      NL(LJZ,5) = LX - NALPH3 + 6
      JL(LJZ,6) = -JL(LJX,6)
      NL(LJZ,6) = LX - NALPH3 + N33ON2 + 6
      JL(LJZ,7) = -JL(LJX,6)
      NL(LJZ,7) = LX - NALPH3 + N33
      JL(LJZ,8) = JL(LJX,6)
      NL(LJZ,8) = LX - NALPH3 + N33ON2
      DO 312 ND3=9,27
      JL(LJZ,ND3) = 0.0
      NL(LJZ,ND3) = 1
  312 CONTINUE
      JU(LJX,1) = -D7*DD/4.0
      NU(LJX,1) = LY
      JU(LJX,2) = 0.0
      NU(LJX,2) = LZ
      JU(LJX,3) = D2*DA/2.0
      NU(LJX,3) = LX + NALPH3
      JU(LJX,4) = (D5*DD)+((D11-(2.0*D8))*REINV)
      NU(LJX,4) = LY + 3
      JU(LJX,5) = D6*REINV
      NU(LJX,5) = LY + 3
      JU(LJY,6) = 0.0
      NU(LJX,6) = LZ + 3
      JU(LJX,7) = D11*REINV
```

```
NU(LJX,7) = LX + N330N4 + 3
JU(LJX,8) = 0.0
NU(LJX,8) = LX + NALPH3 + 3
JU(LJX,9) = 0.0
NU(LJX,9) = LX + NALPH3 + N330N2 + 3
JU(LJX,10) = -JL(LJX,4)
NU(LJX,10) = LY + NALPH3 + 3
JU(LJX,11) = -JU(LJX,10)
NU(LJX,11) = LY + NALPH3 + N330N2 + 3
JU(LJX,12) = -JU(LJX,7)/2.0
NU(LJX,12) = LX + 6
JU(LJX,13) = JU(LJX,12)
NU(LJX,13) = LX + N33
JU(LJX,14) = D9*REINV/4.0
NU(LJX,14) = LZ + 6
JU(LJX,15) = -JU(LJX,14)
NU(LJX,15) = LZ + N33
JU(LJX,16) = -JL(LJX,6)
NU(LJX,16) = LZ + NALPH3 + 6
JU(LJX,17) = JL(LJX,6)
NU(LJX,17) = LZ + NALPH3 + N330N2 + 6
JU(LJX,18) = JL(LJX,6)
NU(LJX,18) = LZ + NALPH3 + N33
JU(LJX,19) = -JL(LJX,6)
NU(LJX,19) = LZ + NALPH3 + N330N2
JU(LJX,20) = -(D5*D8)-(((2.0*D8)-D11)*REINV)
NU(LJX,20) = LX + N330N2 + 3
JU(LJX,21) = -D6*REINV
NU(LJX,21) = LY + N330N2 + 3
JU(LJX,22) = 0.0
NU(LJX,22) = LZ + N330N2 + 3
JU(LJX,23) = JU(LJX,7)
NU(LJX,23) = LX + N390N4 + 3
JU(LJX,24) = JU(LJX,12)
NU(LJX,24) = LX + N330N2 + 6
JU(LJX,25) = JU(LJX,12)
NU(LJX,25) = LX + N330N2
JU(LJX,26) = -JU(LJX,14)
NU(LJX,26) = LZ + N330N2 + 6
JU(LJX,27) = JU(LJX,14)
NU(LJX,27) = LZ + N330N2
JU(LJY,1) = 0.0
NU(LJY,1) = LZ
JU(LJY,2) = (D2*DA/2.0)-((D3+D4)*REINV)
NU(LJY,2) = LY + NALPH3
JU(LJY,3) = 0.0
NU(LJY,3) = LX + 3
JU(LJY,4) = (D5*D8)+JU(LJX,7)
NU(LJY,4) = LY + 3
JU(LJY,5) = 0.0
NU(LJY,5) = LZ + 3
JU(LJY,6) = JU(LJX,7)
NU(LJY,6) = LY + N330N4 + 3
JU(LJY,7) = -3.0*D10*REINV/2.0
NU(LJY,7) = LZ + N330N4 + 3
JU(LJY,8) = -JL(LJY,3)
NU(LJY,8) = LX + NALPH3 + 3
JU(LJY,9) = JL(LJY,3)
NU(LJY,9) = LX + NALPH3 + N330N2 + 3
JU(LJY,10) = 0.0
NU(LJY,10) = LY + NALPH3 + 3
JU(LJY,11) = 0.0
NU(LJY,11) = LY + NALPH3 + N330N2 + 3
JU(LJY,12) = JU(LJX,12)
NU(LJY,12) = LY + 6
JU(LJY,13) = JU(LJX,12)
NU(LJY,13) = LY + N33
JU(LJY,14) = -JU(LJY,7)/2.0
NU(LJY,14) = LZ + 6
JU(LJY,15) = -JU(LJY,14)
NU(LJY,15) = LZ + N33
JU(LJY,16) = 0.0
NU(LJY,16) = LX + N330N2 + 3
JU(LJY,17) = JU(LJX,7) -(D5*D8)
NU(LJY,17) = LY + N330N2 + 3
JU(LJY,18) = 0.0
NU(LJY,18) = LZ + N330N2 + 3
JU(LJY,19) = JU(LJX,7)
NU(LJY,19) = LY + N390N4 + 3
JU(LJY,20) = -JU(LJY,7)
NU(LJY,20) = LZ + N390N4 + 3
JU(LJY,21) = JU(LJY,12)
```

```
      NU(LJY,21) = LY + N33ON2 + 6
      JU(LJY,22) = JU(LJX,12)
      NU(LJY,22) = LY + N33ON2
      JU(LJY,23) = JU(LJY,14)
      NU(LJY,23) = LZ + N33ON2 + 6
      JU(LJY,24) = JU(LJY,15)
      NU(LJY,24) = LZ + N33ON2
      DO 313 ND4=25,27
      JU(LJY,ND4) = 0.0
      NU(LJY,ND4) = 1
  313 CONTINUE
      JU(LJZ,1) = (D2*DA/2.0)+((-D3-D4)*REINV)
      NU(LJZ,1) = LZ + NALPH3
      JU(LJZ,2) = 0.0
      NU(LJZ,2) = LY + 7
      JU(LJZ,3) = (D5*D8/2.0)-(3.0*D8*REINV/2.0)
      NU(LJZ,3) = LZ + 7
      JU(LJZ,4) = JU(LJY,7)/3.0
      NU(LJZ,4) = LY + N33ON4 + 3
      JU(LJZ,5) = 0.0
      NU(LJZ,5) = LZ + N33ON4 + 3
      JU(LJZ,6) = 0.0
      NU(LJZ,6) = LZ + NALPH3 + 3
      JU(LJZ,7) = 0.0
      NU(LJZ,7) = LZ + NALPH3 + N33ON2 + 3
      JU(LJZ,8) = -JU(LJY,7)/6.0
      NU(LJZ,8) = LY + 6
      JU(LJZ,9) = -JU(LJZ,8)
      NU(LJZ,9) = LY + N33
      JU(LJZ,10) = 0.0
      NU(LJZ,10) = LZ + 6
      JU(LJZ,11) = 0.0
      NU(LJZ,11) = LZ + N33
      JU(LJZ,12) = -JL(LJX,6)
      NU(LJZ,12) = LX + NALPH3 + 6
      JU(LJZ,13) = JL(LJX,6)
      NU(LJZ,13) = LX + NALPH3 + N33ON2 + 6
      JU(LJZ,14) = JL(LJX,6)
      NU(LJZ,14) = LX + NALPH3 + N33
      JU(LJZ,15) = -JL(LJX,6)
      NU(LJZ,15) = LX + NALPH3 + N33ON2
      JU(LJZ,16) = 0.0
      NU(LJZ,16) = LY + N33ON2 + 7
      JU(LJZ,17) = -(D5*D8/2.0)-(3.0*D8*REINV/2.0)
      NU(LJZ,17) = LZ + N33ON2 + 3
      JU(LJZ,18) = -JU(LJZ,4)
      NU(LJZ,18) = LY + N39ON4 + 7
      JU(LJZ,19) = 0.0
      NU(LJZ,19) = LZ + N39ON4 + 7
      JU(LJZ,20) = JU(LJZ,8)
      NU(LJZ,20) = LY + N33ON2 + 6
      JU(LJZ,21) = JU(LJZ,9)
      NU(LJZ,21) = LY + N33ON2
      JU(LJZ,22) = 0.0
      NU(LJZ,22) = LZ + N33ON2 + 6
      JU(LJZ,23) = 0.0
      NU(LJZ,23) = LZ + N33ON2
      DO 314 ND5=24,27
      JU(LJZ,ND5) = 0.0
      NU(LJZ,ND5) = 1
  314 CONTINUE
      RETURN
      END
      SUBROUTINE MATRJ2
      COMMON/COMJ1/REINV,NALPH3,N33,N33ON2,N33ON4,N39ON4,LX,LY,LZ,
     1LJX,LJY,LJZ
      COMMON/COMJ2/D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11,D12
      COMMON/COMJ3/D13,D14,D15,D16,D17,D18,D19,D20,D21,D22,D23
      COMMON/COMJ4/DA,DB,DC,DD,DE,DF,DG
      COMMON/COMJ5/DH,DI,DJ,DK,DL
      COMMON/COMJ7/DT,D[illegible],DV,DW,DX,DY,DZ
      REAL JL, JD, JU, MAL, MAD, MAU
      COMMON YL(112,2[illegible]), YD(112), YU(112,20), MYL(112,20), MYU(112,20)
      COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84,27)
      COMMON MAL(84,27), MAD(84), MAU(84,27)
C
C
C
C         THIS SUBROUTINE SETS UP THE MATRIX COEFFICIENTS AND
C         NODE LOCATIONS FOR NODES NOT ON THE POLAR AXIS.
C
C
```

```
      JD(LJX) = (2.0*D1*DA)+(D13*DB)+(D8*DW)+(D20*DX)+
     1(((D8*2.0)+(D20*2.0))*REINV)
      JD(LJY) = (D1*DA)+(D13*DB)+(D4*DU)+(D20*DX)+
     1(((2.0*D20)+(2.0*D4))*REINV)
      JD(LJZ) = (D1*DA)+(D8*DW)+(D4*DU)+(((2.0*D4)+(2.0*D8)+D17)*REINV)
      JL(LJX,1) = -D2*DA/2.0
      NL(LJX,1) = LX - NALPH3
      JL(LJX,2) = -D15*REINV/2.0
      NL(LJX,2) = LY - NALPH3
      JL(LJX,3) = -(D5*DB/2.0)-((D8-(D16/2.0))*REINV)
      NL(LJX,3) = LX - N33
      JL(LJX,4) = -D6*REINV/2.0
      NL(LJX,4) = LY - N33
      JL(LJX,5) = -(D18*DC/2.0)-(D20*REINV)
      NL(LJX,5) = LX - 3
      JL(LJX,6) = -D19*REINV/2.0
      NL(LJX,6) = LZ - 3
      JL(LJX,7) = -D7*REINV/4.0
      NL(LJX,7) = LY - NALPH3 + N33
      JL(LJX,8) = -JL(LJY,7)
      NL(LJX,8) = LY - NALPH3 - N33
      JL(LJX,9) = -D21*REINV/4.0
      NL(LJX,9) = LZ - NALPH3 + 3
      JL(LJX,10) = -JL(LJX,9)
      NL(LJX,10) = LZ - NALPH3 - 3
      DO 320 ND1=11,27
      JL(LJX,ND1) = 0.0
      NL(LJX,ND1) = 1
  320 CONTINUE
      JL(LJY,1) = (D1*DG)-(D7*DH/4.0)+(D6*DT/2.0)
      NL(LJY,1) = LX
      JL(LJY,2) = -(D2*DA/2.0)-((D4-D3)*REINV)
      NL(LJY,2) = LY - NALPH3
      JL(LJY,3) = -D5*DB/2.0
      NL(LJY,3) = LY - N33
      JL(LJY,4) = -(D18*DC/2.0)-(D20*REINV)
      NL(LJY,4) = LY - 3
      JL(LJY,5) = -D23*REINV/2.0
      NL(LJY,5) = LZ - 3
      JL(LJY,6) = JL(LJX,7)
      NL(LJY,6) = LX - NALPH3 + N33
      JL(LJY,7) = JL(LJY,8)
      NL(LJY,7) = LX - NALPH3 - N33
      JL(LJY,8) = -D22*REINV/4.0
      NL(LJY,8) = LZ - N33 + 3
      JL(LJY,9) = -JL(LJY,8)
      NL(LJY,9) = LZ - N33 - 3
      DO 321 ND2=10,27
      JL(LJY,ND2) = 0.0
      NL(LJY,ND2) = 1
  321 CONTINUE
      JL(LJZ,1) = (D2*DC)+(D19*DJ/2.0)-(D21*DK/4.0)
      NL(LJZ,1) = LX
      JL(LJZ,2) = (D13*DC)+(D23*DJ/2.0)-(D22*DL/4.0)
      NL(LJZ,2) = LY
      JL(LJZ,3) = -(D2*DA/2.0)-((D4-D3)*REINV)
      NL(LJZ,3) = LZ - NALPH3
      JL(LJZ,4) = -(D5*DB/2.0)-((D8-(D16/2.0))*REINV)
      NL(LJZ,4) = LZ - N33
      JL(LJZ,5) = -JL(LJY,5)
      NL(LJZ,5) = LY - 3
      JL(LJZ,6) = -D18*DC/2.0
      NL(LJZ,6) = LZ - 3
      JL(LJZ,7) = JL(LJX,9)
      NL(LJZ,7) = LX - NALPH3 + 3
      JL(LJZ,8) = JL(LJX,10)
      NL(LJZ,8) = LX - NALPH3 - 3
      JL(LJZ,9) = JL(LJY,8)
      NL(LJZ,9) = LY - N33 + 3
      JL(LJZ,10) = JL(LJY,9)
      NL(LJZ,10) = LY - N33 - 3
      DO 322 ND3=11,27
      JL(LJZ,ND3) = 0.0
      NL(LJZ,ND3) = 1
  322 CONTINUE
      JU(LJX,1) = -(D7*DH/4.0)+(D14*REINV)
      NU(LJX,1) = LY
      JU(LJX,2) = -D21*DK/4.0
      NU(LJX,2) = LZ
      JU(LJX,3) = D2*DA/2.0
      NU(LJX,3) = LX + NALPH3
      JU(LJX,4) = -JL(LJX,2)
```

```
      NU(LJX,4)  =  LY + NALPH3
      JU(LJX,5)  =  (D5*DB/2.0)+((-D8-(D16/2.0))*REINV)
      NU(LJX,5)  =  LX + N33
      JU(LJX,6)  =  -JL(LJX,4)
      NU(LJX,6)  =  LY + N33
      JU(LJX,7)  =  (D18*DC/2.0)-(D20*REINV)
      NU(LJX,7)  =  LX + 3
      JU(LJX,8)  =  -JL(LJX,6)
      NU(LJX,8)  =  LZ + 3
      JU(LJX,9)  =  JL(LJX,9)
      NU(LJX,9)  =  LY + NALPH3 + N33
      JU(LJX,10)  =  JL(LJX,7)
      NU(LJX,10)  =  LY + NALPH3 - N33
      JU(LJX,11)  =  JL(LJX,10)
      NU(LJX,11)  =  LZ + NALPH3 + 3
      JU(LJX,12)  =  JL(LJX,9)
      NU(LJX,12)  =  LZ + NALPH3 - 3
      DO 323 ND4=13,27
      JU(LJX,ND4)  =  0.0
      NU(LJX,ND4)  =  1
  323 CONTINUE
      JU(LJY,1)  =  -D22*DL/4.0
      NU(LJY,1)  =  LZ
      JU(LJY,2)  =  (D2*DA/2.0)+((-D3-D4)*REINV)
      NU(LJY,2)  =  LY + NALPH3
      JU(LJY,3)  =  D5*DB/2.0
      NU(LJY,3)  =  LY + N33
      JU(LJY,4)  =  (D18*DC/2.0)-(D20*REINV)
      NU(LJY,4)  =  LY + 3
      JU(LJY,5)  =  -JL(LJY,5)
      NU(LJY,5)  =  LZ + 3
      JU(LJY,6)  =  JL(LJX,8)
      NU(LJY,6)  =  LX + NALPH3 + N33
      JU(LJY,7)  =  JL(LJX,7)
      NU(LJY,7)  =  LX + NALPH3 - N33
      JU(LJY,8)  =  JL(LJY,9)
      NU(LJY,8)  =  LZ + N33 + 3
      JU(LJY,9)  =  JL(LJY,8)
      NU(LJY,9)  =  LZ + N33 - 3
      DO 324 ND5=10,27
      JU(LJY,ND5)  =  0.0
      NU(LJY,ND5)  =  1
  324 CONTINUE
      JU(LJZ,1)  =  (D2*DA/2.0)+((-D3-D4)*REINV)
      NU(LJZ,1)  =  LZ + NALPH3
      JU(LJZ,2)  =  (D5*DB/2.0)+((-D8-(D16/2.0))*REINV)
      NU(LJZ,2)  =  LZ + N33
      JU(LJZ,3)  =  JL(LJY,5)
      NU(LJZ,3)  =  LY + 3
      JU(LJZ,4)  =  D18*DC/2.0
      NU(LJZ,4)  =  LZ + 3
      JU(LJZ,5)  =  JL(LJY,10)
      NU(LJZ,5)  =  LX + NALPH3 + 3
      JU(LJZ,6)  =  JL(LJX,9)
      NU(LJZ,6)  =  LX + NALPH3 - 3
      JU(LJZ,7)  =  JL(LJY,9)
      NU(LJZ,7)  =  LY + N33 + 3
      JU(LJZ,8)  =  JL(LJY,8)
      NU(LJZ,8)  =  LY + N33 - 3
      DO 325 ND6=9,27
      JU(LJZ,ND6)  =  0.0
      NU(LJZ,ND6)  =  1
  325 CONTINUE
      RETURN
      END

      SUBROUTINE MATRJ3
      COMMON/COMJ1/REINV,NALPH3,N33,N33ON2,N33ON4,N99ON4,LX,LY,LZ,
     1LJX,LJY,LJZ
      COMMON/COMJ2/D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11,D12
      COMMON/COMJ3/D13,D14,D15,D16,D17,D18,D19,D20,D21,D22,D23
      COMMON/COMJ4/DA,DB,DC,DD,DE,DF,DG
      REAL JL, JD, JU, MAL, MAD, MAU
      COMMON YL(112,20), YD(112), YU(112,20), NYL(112,20), NYU(112,20)
      COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84.27)
      COMMON MAL(84,27), MAD(84), MAU(84,27)
C
C
C
C          THIS SUBROUTINE MAKES THE NECESSARY ALTERATIONS TO
C          THE MATRIX COEFFICIENTS AND NODE LOCATIONS FOR
C          INTERIOR NODES WHERE PHI VARIATIONS MOVE COEFFICIENTS
C          FROM THE LOWER TO THE UPPER TRIANGULAR MATRIX.
```

```
C
C
C
      JL(LJX,5) = 0.0
      NL(LJX,5) = 1
      JL(LJX,6) = 0.0
      NL(LJX,6) = 1
      NL(LJX,10) = NL(LJX,10) + N33
      JL(LJY,4) = 0.0
      NL(LJY,4) = 1
      JL(LJY,5) = 0.0
      NL(LJY,5) = 1
      NL(LJY,9) = NL(LJY,9) + N33
      JL(LJZ,5) = 0.0
      NL(LJZ,5) = 1
      JL(LJZ,6) = 0.0
      NL(LJZ,6) = 1
      NL(LJZ,8) = NL(LJZ,8) + N33
      NL(LJZ,10) = NL(LJZ,10) + N33
      NU(LJX,12) = NU(LJX,12) + N33
      JU(LJX,13) = -(D18*DC/2.0)-(D20*REINV)
      NU(LJX,13) = LX - 3 + N33
      JU(LJX,14) = -D19*REINV/2.0
      NU(LJX,14) = LZ - 3 + N33
      NU(LJY,9) = NU(LJY,9) + N33
      JU(LJY,10) = JU(LJX,13)
      NU(LJY,10) = LY - 3 + N33
      JU(LJY,11) = -D23*REINV/2.0
      NU(LJY,11) = LZ - 3 + N33
      NU(LJZ,6) = NU(LJZ,6) + N33
      NU(LJZ,8) = NU(LJZ,8) + N33
      JU(LJZ,9) = -JU(LJY,11)
      NU(LJZ,9) = LY - 3 + N33
      JU(LJZ,10) = -D18*DC/2.0
      NU(LJZ,10) = LZ - 3 + N33
      RETURN
      END
      SUBROUTINE MATPJ4
      COMMON/COMJ1/REINV,NALPH3,N33,N33ON2,N33ON4,N39ON4,LX,LY,LZ,
     1LJX,LJY,LJZ
      COMMON/COMJ2/D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11,D12
      COMMON/COMJ3/D13,D14,D15,D16,D17,D18,D19,D20,D21,D22,D23
      COMMON/COMJ4/DA,DB,DC,DD,DE,DF,DG
      REAL JL, JD, JU, MAL, MAD, MAU
      COMMON YL(112,20), YD(112), YU(112,20), NYL(112,20), NYU(112,20)
      COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84,27)
      COMMON MAL(84,27), MAD(84), MAU(84,27)
C
C
C
C        THIS SUBROUTINE MAKES THE NECESSARY ALTERATIONS TO
C        THE MATRIX COEFFICIENTS AND NODE LOCATIONS FOR
C        INTERIOR NODES WHERE PHI VARIATIONS MOVE COEFFICIENTS
C        FROM THE UPPER TO THE LOWER TRIANGULAR MATRIX.
C
C
C
      NL(LJX,9) = NL(LJX,9) - N33
      JL(LJX,11) = (D18*DC/2.0)+(-D20*REINV)
      NL(LJX,11) = LX + 3 - N33
      JL(LJX,12) = D19*REINV/2.0
      NL(LJX,12) = LZ + 3 - N33
      NL(LJY,8) = NL(LJY,8) - N33
      JL(LJY,10) = JL(LJX,11)
      NL(LJY,10) = LY + 3 - N33
      JL(LJY,11) = D23*REINV/2.0
      NL(LJY,11) = LZ + 3 - N33
      NL(LJZ,7) = NL(LJZ,7) - N33
      NL(LJZ,9) = NL(LJZ,9) - N33
      JL(LJZ,11) = -JL(LJY,11)
      NL(LJZ,11) = LY + 3 - N33
      JL(LJZ,12) = D18*DC/2.0
      NL(LJZ,12) = LZ + 3 - N33
      JU(LJX,7) = 0.0
      NU(LJX,7) = 1
      JU(LJX,8) = 0.0
      NU(LJX,8) = 1
      NU(LJX,11) = NU(LJX,11) - N33
      JU(LJY,4) = 0.0
      NU(LJY,4) = 1
      JU(LJY,5) = 0.0
      NU(LJY,5) = 1
      NU(LJY,8) = NU(LJY,8) - N33
```

```
      JU(LJZ,3) = 0.0
      NU(LJZ,3) = 1
      JU(LJZ,4) = 0.0
      NU(LJZ,4) = 1
      NU(LJZ,5) = NU(LJZ,5) - N33
      NU(LJZ,7) = NU(LJZ,7) - N33
      RETURN
      END
      SUBROUTINE MATRJ5
      COMMON/COMJ1/REINV,NALPH3,N33,N330N2,N330N4,N390N4,LX,LY,LZ,
     1LJX,LJY,LJZ
      COMMON/COMJ2/D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11,D12
      COMMON/COMJ3/D13,D14,D15,D16,D17,D18,D19,D20,D21,D22,D23
      COMMON/COMJ4/DA,DB,DC,DD,DE,DF,DG
      COMMON/COMJ5/DH,DI,DJ,DK,DL
      COMMON/COMJ6/DM,DN,DP,DQ,DR,DS
      COMMON/COMJ7/DT,DU,DV,DW,DX,DY,DZ
      COMMON/COMJ8/NC23,NCN23,CP,SP,CPP,SPP,CPM,SPM
      REAL JL, JD, JU, MAL, MAD, MAU
      COMMON YL(112,20), YD(112), YU(112,20), MYL(112,20), MYU(112,20)
      COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84,27)
      COMMON MAL(84,27), MAD(84), MAU(84,27)
C
C
C
C        THIS SUBROUTINE MAKES THE NECESSARY ALTERATIONS TO
C        THE MATRIX COEFFICIENTS AND NODE LOCATIONS FOR
C        INTERIOR NODES ADJACENT TO THE THETA = 0 POLAR AXIS.
C
C
C
      JD(LJX) = (2.0*D1*DA)+(D17*D3)+(D8*DY)+(D20*DX)+
     1(((D8*2.0)+(D20*2.0))*REINV)
      JD(LJZ) = (D1*DA)+(D4*DU)+(D8*DY)+(((D4*2.0)+(D8*2.0)+D17)*REINV)
      JL(LJX,3) = 0.0
      NL(LJX,3) = 1
      JL(LJX,4) = 0.0
      NL(LJX,4) = 1
      JL(LJX,8) = 0.0
      NL(LJX,8) = 1
      JL(LJX,13) = -(D5*D8/2.0)-((D8-(D16/2.0))*REINV)
      NL(LJX,13) = NC23 - 2
      JL(LJX,14) = -(D5*REINV/2.0)*CP
      NL(LJX,14) = NC23 - 1
      JL(LJX,15) = -(D6*REINV/2.0)*SP
      NL(LJX,15) = NC23
      JL(LJY,16) = (D7*REINV/4.0)*CP
      NL(LJX,16) = NC23 - 1 - NALPH3
      JL(LJY,17) = (D7*REINV/4.0)*SP
      NL(LJX,17) = NC23 - NALPH3
      JL(LJY,1) = (D1*DP)-(D7*DM/4.0)+(D5*DM/2.0)
      JL(LJY,3) = 0.0
      NL(LJY,3) = 1
      JL(LJY,7) = 0.0
      NL(LJY,7) = 1
      JL(LJY,8) = 0.0
      NL(LJY,8) = 1
      JL(LJY,9) = 0.0
      NL(LJY,9) = 1
      JL(LJY,12) = -(D5*D8/2.0)*CP
      NL(LJY,12) = NC23 - 1
      JL(LJY,13) = -(D5*D8/2.0)*SP
      NL(LJY,13) = NC23
      JL(LJY,14) = D7*REINV/4.0
      NL(LJY,14) = NC23 - 2 - NALPH3
      JL(LJY,15) = (D22*REINV/4.0)*SPP
      NL(LJY,15) = NC23 - 1
      JL(LJY,16) = -(D22*REINV/4.0)*CPP
      NL(LJY,16) = NC23
      JL(LJY,17) = -(D22*REINV/4.0)*SPM
      NL(LJY,17) = NC23 - 1
      JL(LJY,18) = (D22*REINV/4.0)*CPM
      NL(LJY,18) = NC23
      JL(LJZ,2) = (D13*DD)+(D23*DJ/2.0)-(D22*DP/4.0)
      JL(LJZ,4) = 0.0
      NL(LJZ,4) = 1
      JL(LJZ,9) = 0.0
      NL(LJZ,9) = 1
      JL(LJZ,10) = 0.0
      NL(LJZ,10) = 1
      JL(LJZ,13) = -JL(LJY,13)*SP
      NL(LJZ,13) = NC23 - 1
```

```
      JL(LJZ,14)  =  JL(LJX,13)*CP
      NL(LJZ,14)  =  NC23
      JL(LJZ,15)  =  JL(LJY,16)
      NL(LJZ,15)  =  NC23 - 1
      JL(LJZ,16)  =  -JL(LJY,15)
      NL(LJZ,16)  =  NC23
      JL(LJZ,17)  =  JL(LJY,19)
      NL(LJZ,17)  =  NC23 - 1
      JL(LJZ,18)  =  -JL(LJY,17)
      NL(LJZ,18)  =  NC23
      JU(LJX,1)  = -(D7*DM/4.0)+(D14*REINV)
      JU(LJX,10)  =  0.0
      NU(LJX,10)  =  1
      JU(LJX,15)  =  -JL(LJX,16)
      NU(LJX,15)  =  NC23 - 1 + NALPH3
      JU(LJX,16)  =  -JL(LJX,17)
      NU(LJX,16)  =  NC23 + NALPH3
      JU(LJY,1)  =  -D22*DP/4.0
      JU(LJY,7)  =  0.0
      NU(LJY,7)  =  1
      JU(LJY,12)  =  -JL(LJY,14)
      NU(LJY,12)  =  NC23 - 2 + NALPH3
      RETURN
      END
      SUBROUTINE MATRJ6
      COMMON/COMJ1/REINV,NALPH3,N33,N330N2,N330N4,N390N4,LX,LY,LZ,
     1LJX,LJY,LJZ
      COMMON/COMJ2/D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11,D12
      COMMON/COMJ3/D13,D14,D15,D16,D17,D18,D19,D20,D21,D22,D23
      COMMON/COMJ4/DA,DB,DC,DD,DE,DF,DG
      COMMON/COMJ5/DH,DI,DJ,DK,DL
      COMMON/COMJ6/DM,DN,DP,DQ,DR,DS
      COMMON/COMJ7/DT,DU,DV,DW,DX,DY,DZ
      COMMON/COMJ8/NC23,NCN23,CP,SP,CPP,SPP,CPM,SPM
      REAL JL, JD, JU, MAL, MAD, MAU
      COMMON YL(112,20), YD(112), YU(112,20), MYL(112,20), MYU(112,20)
      COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84,27)
      COMMON MAL(84,27), MAD(84), MAU(84,27)
C
C
C
C        THIS SUBROUTINE MAKES THE NECESSARY ALTERATIONS TO
C        THE MATRIX COEFFICIENTS AND NODE LOCATIONS FOR
C        INTERIOR NODES ADJACENT TO THE THETA = PI POLAR AXIS.
C
C
C
      JD(LJX)  =  (2.0*D1*DA)+(D13*DB)+(D8*DZ)+(D20*DX)+
     1(((D8*2.0)+(D20*2.0))*REINV)
      JD(LJZ)  =  (D1*DA)+(D4*DU)+(D8*DZ)+(((D4*2.0)+(D8*2.0)+D17)*REINV)
      JL(LJX,7)  =  0.0
      NL(LJX,7)  =  1
      JL(LJX,13)  =  (D7*REINV/4.0)*CP
      NL(LJX,13)  =  NCN23 - 1 - NALPH3
      JL(LJX,14)  =  (D7*REINV/4.0)*SP
      NL(LJX,14)  =  NCN23 - NALPH3
      JL(LJY,1)  =  (D1*DB)-(D7*DQ/4.0)+(D6*DP/2.0)
      JL(LJY,6)  =  0.0
      NL(LJY,6)  =  1
      JL(LJY,12)  =  -D7*REINV/4.0
      NL(LJY,12)  =  NCN23 - 2 - NALPH3
      JL(LJZ,2)  =  (D13*DC)+(D23*DJ/2.0)-(D22*DS/4.0)
      JU(LJX,1)  =  -(D7*DQ/4.0)+(D14*REINV)
      JU(LJX,5)  =  0.0
      NU(LJX,5)  =  1
      JU(LJX,6)  =  0.0
      NU(LJX,6)  =  1
      JU(LJX,9)  =  0.0
      NU(LJX,9)  =  1
      JU(LJX,15)  =  (D5*D8/2.0)+((-D8-(D16/2.0))*REINV)
      NU(LJX,15)  =  NCN23 - 2
      JU(LJX,16)  =  -(D6*REINV/2.0)*CP
      NU(LJY,16)  =  NCN23 - 1
      JU(LJX,17)  =  -(D6*REINV/2.0)*SP
      NU(LJX,17)  =  NCN23
      JU(LJX,18)  =  JL(LJY,12)*CP
      NU(LJX,18)  =  NCN23 - 1 + NALPH3
      JU(LJX,19)  =  JL(LJY,12)*SP
      NU(LJX,19)  =  NCN23 + NALPH3
      JU(LJY,1)  =  -D22*DS/4.0
      JU(LJY,3)  =  0.0
      NU(LJY,3)  =  1
      JU(LJY,6)  =  0.0
```

```
      NU(LJY,6)  =  1
      JU(LJY,8)  =  0.0
      NU(LJY,8)  =  1
      JU(LJY,9)  =  0.0
      NU(LJY,9)  =  1
      JU(LJY,12)  =  -(D5*D8/2.0)*CP
      NU(LJY,12)  =  NCN23  -  1
      JU(LJY,13)  =  -(D5*D8/2.0)*SP
      NU(LJY,13)  =  NCN23
      JU(LJY,14)  =  -JL(LJY,12)
      NU(LJY,14)  =  NCN23  -  2  +  NALPH3
      JU(LJY,15)  =  -(D22*REINV/4.0)*SPP
      NU(LJY,15)  =  NCN23  -  1
      JU(LJY,16)  =  (D22*REINV/4.0)*CPP
      NU(LJY,16)  =  NCN23
      JU(LJY,17)  =  (D22*REINV/4.0)*SPM
      NU(LJY,17)  =  NCN23  -  1
      JU(LJY,18)  =  -(D22*REINV/4.0)*CPM
      NU(LJY,18)  =  NCN23
      JU(LJZ,2)  =  0.0
      NU(LJZ,2)  =  1
      JU(LJZ,7)  =  0.0
      NU(LJZ,7)  =  1
      JU(LJZ,8)  =  0.0
      NU(LJZ,8)  =  1
      JU(LJZ,11)  =  -JU(LJX,15)*SP
      NU(LJZ,11)  =  NCN23  -  1
      JU(LJZ,12)  =  JU(LJX,15)*CP
      NU(LJZ,12)  =  NCN23
      JU(LJZ,13)  =  -JU(LJY,16)
      NU(LJZ,13)  =  NCN23  -  1
      JU(LJZ,14)  =  JU(LJY,15)
      NU(LJZ,14)  =  NCN23
      JU(LJZ,15)  =  -JU(LJY,18)
      NU(LJZ,15)  =  NCN23  -  1
      JU(LJZ,16)  =  JU(LJY,17)
      NU(LJZ,16)  =  NCN23
      RETURN
      END
      SUBROUTINE MATRJZ
      COMMON/COMJ1/REINV,NALPH3,N33,N330N2,N330N4,N390N4,LX,LY,LZ,
     1LJX,LJY,LJZ
      COMMON/COMJ2/D1,D2,D3,D4,D5,D6,D7,D8,D9,D10,D11,D12
      COMMON/COMJ4/DA,DB,DC,DD,DE,DF,DG
      COMMON/COMJ7/DT,DU,DV,DW,DX,DY,DZ
      REAL JL, JO, JU, MAL, MAO, MAU
      COMMON YL(112,20), YO(112), YU(112,20), NYL(112,20), NYU(112,20)
      COMMON JL(84,27), JO(84), JU(84,27), NL(84,27), NU(84,27)
      COMMON MAL(84,27), MAO(84), MAU(84,27)
C
C
C
C        THIS SUBROUTINE SETS UP THE MATRIX COEFFICIENTS AND
C        NODE LOCATIONS FOR NODES ON THE THETA = PI POLAR AXIS.
C
C
C
      JO(LJX)  =  (2.0*D1*DA)+(2.0*D8*DV)+(((4.0*D3)-(2.0*D11))*REINV)
      JO(LJY)  =  (D1*DA)+(D8*DV)+(D4*DU)+(((2.0*D4)-(2.0*D11))*REINV)
      JO(LJZ)  =  (D1*DA)+(D8*DV)+(D4*DU)+(((2.0*D4)+(3.0*D8))*REINV)
      JL(LJX,1)  =  -DA*D2/2.0
      NL(LJX,1)  =  LX  -  NALPH3
      JL(LJX,2)  =  -(D8*D5)-(((2.0*D3)-D11)*REINV)
      NL(LJX,2)  =  LX  -  N330N2
      JL(LJX,3)  =  -D6*REINV
      NL(LJX,3)  =  LY  -  N330N2
      JL(LJX,4)  =  0.0
      NL(LJX,4)  =  LZ  -  N330N2
      JL(LJX,5)  =  D11*REINV
      NL(LJX,5)  =  LX  -  N330N4
      JL(LJX,6)  =  0.0
      NL(LJX,6)  =  LY  -  NALPH3  -N33
      JL(LJX,7)  =  0.0
      NL(LJX,7)  =  LX  -  NALPH3  -  N330N2
      JL(LJX,8)  =  -D7*REINV/2.0
      NL(LJX,8)  =  LY  -  NALPH3  -  N33
      JL(LJX,9)  =  -JL(LJX,8)
      NL(LJX,9)  =  LY  -  NALPH3  -  N330N2
      JL(LJX,10)  =  -JL(LJX,5)/2.0
      NL(LJX,10)  =  LX  -  N330N2  +  3
      JL(LJX,11)  =  JL(LJX,10)
      NL(LJX,11)  =  LX  -  N330N2  -  3
      JL(LJX,12)  =  D9*REINV/4.0
```

```
      NL(LJX,12) = LZ - N33ON2 + 3
      JL(LJX,13) = -JL(LJX,12)
      NL(LJX,13) = LZ - N33ON2 - 3
      JL(LJX,14) = D12*REINV/8.0
      NL(LJX,14) = LZ - NALPH3 - N33 + 3
      JL(LJX,15) = -JL(LJX,14)
      NL(LJX,15) = LZ - NALPH3 - N33ON2 + 3
      JL(LJX,16) = -JL(LJX,14)
      NL(LJX,16) = LZ - NALPH3 - 3
      JL(LJX,17) = JL(LJX,14)
      NL(LJX,17) = LZ - NALPH3 - N33ON2 - 3
      JL(LJX,18) = (D8*D5)+(((-2.0*D8)+D11)*REINV)
      NL(LJX,18) = LX - N33
      JL(LJX,19) = D6*REINV
      NL(LJX,19) = LY - N33
      JL(LJX,20) = 0.0
      NL(LJX,20) = LZ - N33
      JL(LJX,21) = JL(LJX,5)
      NL(LJX,21) = LX - N39ON4
      JL(LJX,22) = JL(LJX,10)
      NL(LJX,22) = LX - N33 + 3
      JL(LJX,23) = JL(LJX,10)
      NL(LJX,23) = LX - 3
      JL(LJX,24) = JL(LJY,13)
      NL(LJX,24) = LZ - N33 + 3
      JL(LJX,25) = JL(LJX,12)
      NL(LJX,25) = LZ - 3
      DO 370 ND1=26,27
      JL(LJX,ND1) = 0.0
      NL(LJX,ND1) = 1
  370 CONTINUE
      JL(LJY,1) = (D1*D3)-(D7*DF/4.0)+(D6*DG/2.0)
      NL(LJY,1) = LX
      JL(LJY,2) = -(D2*DA/2.0)+((D3-D4)*REINV)
      NL(LJY,2) = LY - NALPH3
      JL(LJY,3) = 0.0
      NL(LJY,3) = LX - N33ON2
      JL(LJY,4) = -(D5*D8)+JL(LJX,5)
      NL(LJY,4) = LY - N33ON2
      JL(LJY,5) = 0.0
      NL(LJY,5) = LZ - N33ON2
      JL(LJY,6) = JL(LJX,5)
      NL(LJY,6) = LY - N39ON4
      JL(LJY,7) = -3.0*D10*REINV/2.0
      NL(LJY,7) = LZ - N39ON4
      JL(LJY,8) = JL(LJX,8)/2.0
      NL(LJY,8) = LX - NALPH3 - N33
      JL(LJY,9) = -JL(LJY,8)
      NL(LJY,9) = LX - NALPH3 - N33ON2
      JL(LJY,10) = 0.0
      NL(LJY,10) = LY - NALPH3 - N33
      JL(LJY,11) = 0.0
      NL(LJY,11) = LY - NALPH3 - N33ON2
      JL(LJY,12) = JL(LJX,10)
      NL(LJY,12) = LY - N33ON2 + 3
      JL(LJY,13) = JL(LJX,10)
      NL(LJY,13) = LY - N33ON2 - 3
      JL(LJY,14) = JL(LJY,7)/2.0
      NL(LJY,14) = LZ - N33ON2 + 3
      JL(LJY,15) = -JL(LJY,14)
      NL(LJY,15) = LZ - N33ON2 - 3
      JL(LJY,16) = 0.0
      NL(LJY,16) = LX - N33
      JL(LJY,17) = (D5*D8)+JL(LJX,5)
      NL(LJY,17) = LY - N33
      JL(LJY,18) = 0.0
      NL(LJY,18) = LZ - N33
      JL(LJY,19) = JL(LJX,5)
      NL(LJY,19) = LY - N39ON4
      JL(LJY,20) = -JL(LJY,7)
      NL(LJY,20) = LZ - N39ON4
      JL(LJY,21) = JL(LJX,10)
      NL(LJY,21) = LY - N33 + 3
      JL(LJY,22) = JL(LJX,10)
      NL(LJY,22) = LY - 3
      JL(LJY,23) = JL(LJY,14)
      NL(LJY,23) = LZ - N33 + 3
      JL(LJY,24) = JL(LJY,15)
      NL(LJY,24) = LZ - 3
      DO 371 ND2=25,27
      JL(LJY,ND2) = 0.0
      NL(LJY,ND2) = 1
  371 CONTINUE
```

```
      JL(LJZ,1)  = 0.0
      NL(LJZ,1)  = LX
      JL(LJZ,2)  = -(D2*DA/2.0)-((D4-D3)*REINV)
      NL(LJZ,2)  = LZ - NALPH3
      JL(LJZ,3)  = 0.0
      NL(LJZ,3)  = LY - N33ON2
      JL(LJZ,4)  = -(D5*DB/2.0)-(3.0*D8*REINV/2.0)
      NL(LJZ,4)  = LZ - N33ON2
      JL(LJZ,5)  = JL(LJY,7)/3.0
      NL(LJZ,5)  = LY - N33ON4
      JL(LJZ,6)  = 0.0
      NL(LJZ,6)  = LZ - N33ON4
      JL(LJZ,7)  = 0.0
      NL(LJZ,7)  = LZ - NALPH3 - N33
      JL(LJZ,8)  = 0.0
      NL(LJZ,8)  = LZ - NALPH3 - N33ON2
      JL(LJZ,9)  = JL(LJY,7)/6.0
      NL(LJZ,9)  = LY - N33ON2 + 3
      JL(LJZ,10)  = -JL(LJZ,9)
      NL(LJZ,10)  = LY - N33ON2 - 3
      JL(LJZ,11)  = 0.0
      NL(LJZ,11)  = LZ - N33ON2 + 3
      JL(LJZ,12)  = 0.0
      NL(LJZ,12)  = LZ - N33ON2 - 3
      JL(LJZ,13)  = JL(LJX,14)
      NL(LJZ,13)  = LX - NALPH3 - N33 + 3
      JL(LJZ,14)  = -JL(LJX,14)
      NL(LJZ,14)  = LX - NALPH3 - N33ON2 + 3
      JL(LJZ,15)  = -JL(LJY,14)
      NL(LJZ,15)  = LX - NALPH3 - 3
      JL(LJZ,16)  = JL(LJY,14)
      NL(LJZ,16)  = LX - NALPH3 - N33ON2 - 3
      JL(LJZ,17)  = 0.0
      NL(LJZ,17)  = LY - N33
      JL(LJZ,18)  = (D5*DB/2.0)-(3.0*D8*REINV/2.0)
      NL(LJZ,18)  = LZ - N33
      JL(LJZ,19)  = -JL(LJZ,5)
      NL(LJZ,19)  = LY - N39ON4
      JL(LJZ,20)  = 0.0
      NL(LJZ,20)  = LZ - N39ON4
      JL(LJZ,21)  = JL(LJZ,9)
      NL(LJZ,21)  = LY - N33 + 3
      JL(LJZ,22)  = JL(LJZ,10)
      NL(LJZ,22)  = LY - 3
      JL(LJZ,23)  = 0.0
      NL(LJZ,23)  = LZ - N33 + 3
      JL(LJZ,24)  = 0.0
      NL(LJZ,24)  = LZ - 3
      DO 372 ND3=25,27
      JL(LJZ,ND3)  = 0.0
      NL(LJZ,ND3)  = 1
372   CONTINUE
      JU(LJX,1)  = -D7*DF/4.0
      NU(LJX,1)  = LY
      JU(LJX,2)  = 0.0
      NU(LJX,2)  = LZ
      JU(LJX,3)  = D2*DA/2.0
      NU(LJX,3)  = LX + NALPH3
      JU(LJX,4)  = 0.0
      NU(LJX,4)  = LX + NALPH3 - N33
      JU(LJX,5)  = 0.0
      NU(LJX,5)  = LX + NALPH3 - N33ON2
      JU(LJX,6)  = JL(LJY,9)
      NU(LJX,6)  = LY + NALPH3 - N33
      JU(LJX,7)  = JL(LJY,8)
      NU(LJX,7)  = LY + NALPH3 - N33ON2
      JU(LJX,8)  = -JL(LJY,14)
      NU(LJX,8)  = LZ + NALPH3 - N33 + 3
      JU(LJX,9)  = JL(LJY,14)
      NU(LJX,9)  = LZ + NALPH3 - N33ON2 + 3
      JU(LJX,10)  = JL(LJY,14)
      NU(LJX,10)  = LZ + NALPH3 - 3
      JU(LJX,11)  = -JL(LJY,14)
      NU(LJX,11)  = LZ + NALPH3 - N33ON2 - 3
      DO 373 ND4=12,27
      JU(LJX,ND4)  = 0.0
      NU(LJY,ND4)  = 1
373   CONTINUE
      JU(LJY,1)  = 0.0
      NU(LJY,1)  = LZ
      JU(LJY,2)  = (D2*DA/2.0)-((D3+D4)*REINV)
      NU(LJY,2)  = LY + NALPH3
      JU(LJY,3)  = JL(LJY,9)
```

```
      NU(LJY,3) = LX + NALPH3 - N33
      JU(LJY,4) = JL(LJY,8)
      NU(LJY,4) = LX + NALPH3 - N33ON2
      JU(LJY,5) = 0.0
      NU(LJY,5) = LY + NALPH3 - N33
      JU(LJY,6) = 0.0
      NU(LJY,6) = LY + NALPH3 - N33ON2
      DO 374 ND5=7,27
      JU(LJY,ND5) = 0.0
      NU(LJY,ND5) = 1
  374 CONTINUE
      JU(LJZ,1) = (D2*DA/2.0)+((-D3-D4)*REINV)
      NU(LJZ,1) = LZ + NALPH3
      JU(LJZ,2) = 0.0
      NU(LJZ,2) = LZ + NALPH3 - N33
      JU(LJZ,3) = 0.0
      NU(LJZ,3) = LZ + NALPH3 - N33ON2
      JU(LJZ,4) = -JL(LJY,14)
      NU(LJZ,4) = LX + NALPH3 - N33 + 3
      JU(LJZ,5) = JL(LJX,14)
      NU(LJZ,5) = LX + NALPH3 - N33ON2 + 3
      JU(LJZ,6) = JL(LJY,14)
      NU(LJZ,6) = LX + NALPH3 - 3
      JU(LJZ,7) = -JL(LJX,14)
      NU(LJZ,7) = LX + NALPH3 - N33ON2 - 3
      DO 375 ND6=8,27
      JU(LJZ,ND6) = 0.0
      NU(LJZ,ND6) = 1
  375 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MATRY (NN1,NN2,NN3,RRI,RRO,NNEQNY)
      REAL JL, JD, JU, MAL, MAD, MAU
      COMMON/COMY1/D1,D2,D5,D6,D7,D8,D9,D10,D11,D12,D13,D19,D20,
     1D21,D22,D23,D24,D36
      COMMON/COMY2/LX,LY,LZ,LJX,LJY,LJZ,N33,N330N2,N330N4,N390N4
      COMMON/COMY3/R,ST,TT,H1,H3,NALPH3,NALPH6
      COMMON/COMY4/NC2,NC23,NCN2,NCN23,CP,SP,CPP,SPP,CPM,SPM
      COMMON YL(112,20), YD(112), YU(112,20), NYL(112,20), NYU(112,20)
      COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84,27)
      COMMON MAL(84,27), MAD(84), MAU(84,27)
C
C
C
C        THIS SUBROUTINE SETS UP THE MATRIX EQUATION ANALOGUE OF
C        POISSON'S EQUATION FOR THE VECTOR POTENTIAL A OF THE
C        ROTATIONAL COMPONENT OF THE VELOCITY FIELD UPOT FOR
C        LINEAR-SHEAR (AT INFINITY) FLOW PAST A SPHERE OF RADIUS
C        RI. THE BOUNDARY CONDITIONS AT INFINITY ARE REPLACED BY
C        SIMILAR CONDITIONS ON A SPHERE, RADIUS RO, CONCENTRIC
C        WITH THE SPHERE OF RADIUS RI. SECOND-ORDER FINITE-
C        DIFFERENCE APPROXIMATIONS ARE USED BOTH FOR THE VECTOR
C        LAPLACIAN AND FOR THE (MIXED) BOUNDARY CONDITIONS.
C        SUBROUTINES MATRY1-9 SET UP THE MATRIX COEFFICIENTS
C        AND NODE LOCATIONS YL/YD/YU AND NYL/NYU RESPECTIVELY.
C
C
C
      N1 = NN1
      N2 = NN2
      N3 = NN3
      RI = RRI
      RO = RRO
      NEQNY = NNEQNY
      PI = 3.14159265358979
      XN1 = N1
      XN2 = N2
      XN3 = N3
      H1 = (RO - RI)/XN1
      H2 = PI/XN2
      H3 = (2.0 * PI)/XN3
      NALPHA = (N3*(N2 - 1)) + 2
      NODES = (N1 + 1) * NALPHA
      NALPH3 = 3*NALPHA
      NALPH6 = 6*NALPHA
      N33 = 3*N3
      N330N2 = N33/2
      N330N4 = N33/4
      N390N4 = 3*N330N4
      R = RI
      L = 1
      LX = 1
      LJX = 1
      D1 = 2.0/R
      D2 = 1.0/H1
C        INNER BOUNDARY CONDITIONS
    1 CALL MATRY1
      L = L + 1
      LX = (3*L) - 2
      LJX = LJX + 1
      IF(L.EQ.NALPHA+1) GO TO 4
      GO TO 1
    4 I = 2
      J = 1
      K = 1
      R = RI + H1
      THETA = 0.0
      PHI = 0.0
      PHIP = H3
      PHIM =-H3
      LJY = LJX + 1
      LJZ = LJY + 1
      LY = LX + 1
      LZ = LY + 1
    5 D5 = 1.0/(R*R*H2*H2)
      D6 = D5/(H3*H3)
      D7 = 1.0/(H1*H1)
      D8 = 1.0/(R*H1)
      D9 = D8/H2
      D10 = D9/(8.0*H3)
      D11 = 1.0/(R*R*H2)
      D12 = D11/H3
      D13 = D12/H2
```

```
      NC2 = (NALPHA*(I-1)) + 1
      NC23 = 3*NC2
      NCN2 = (NALPHA*I)
      NCN23 = 3*NCN2
C          INTERIOR NODES (J = 1)
      CALL MATRY2
      J = 2
      THETA = PI/XN2
      ST = SIN(THETA)
      TT = TAN(THETA)
      L = L + 1
      LX = (3*L) - 2
      LY = LX + 1
      LZ = LY + 1
      LJX = LJX + 3
      LJY = LJX + 1
      LJZ = LJY + 1
      D19 = 1.0/(R*R*ST*ST)
      D20 = D19/(H3*H3)
      D21 = 1.0/(R*R*TT*H2)
      D22 = 1.0/(R*ST*H1*H3)
      D23 = 1.0/(R*R*ST*TT*H3)
      D24 = 1.0/(R*R*ST*H2*H3)
      CP = COS(PHI)
      SP = SIN(PHI)
      CPP = COS(PHIP)
      SPP = SIN(PHIP)
      CPM = COS(PHIM)
      SPM = SIN(PHIM)
C          INTERIOR NODES (GENERAL CASE)
   11 CALL MATRY3
      IF(K.NE.1) GO TO 18
C          INTERIOR NODES (K = 1)
      CALL MATRY4
   18 IF(K.NE.N3) GO TO 185
C          INTERIOR NODES (K = N3)
      CALL MATRY5
  185 IF(J.EQ.2) GO TO 19
      IF(J.EQ.N2) GO TO 20
      GO TO 21
C          INTERIOR NODES (J = 2)
   19 CALL MATRY6
      GO TO 21
C          INTERIOR NODES (J = N2)
   20 CALL MATRY7
   21 L = L + 1
      LX = (3*L) - 2
      LY = LX + 1
      LZ = LY + 1
      LJX = LJX + 3
      LJY = LJX + 1
      LJZ = LJY + 1
      IF(K.EQ.N3) GO TO 22
      K = K + 1
      Z = K - 1
      PHI = (2.0*Z*PI)/XN3
      CP = COS(PHI)
      SP = SIN(PHI)
      PHIP = PHI + H3
      PHIM = PHI - H3
      CPP = COS(PHIP)
      SPP = SIN(PHIP)
      CPM = COS(PHIM)
      SPM = SIN(PHIM)
      GO TO 11
   22 K = 1
      PHI = 0.0
      CP = COS(PHI)
      SP = SIN(PHI)
      PHIP = H3
      PHIM = -H3
      CPP = COS(PHIP)
      SPP = SIN(PHIP)
      CPM = COS(PHIM)
      SPM = SIN(PHIM)
      J = J + 1
      IF(J.EQ.N2+1) GO TO 23
      Y = J - 1
      THETA = (Y*PI)/XN2
      ST = SIN(THETA)
      TT = TAN(THETA)
      D19 = 1.0/(R*R*ST*ST)
      D20 = D19/(H3*H3)
```

```
      D21 = 1.0/(R*R*TT*H2)
      D22 = 1.0/(R*ST*H1*H3)
      D23 = 1.0/(R*R*ST*TT*H3)
      D24 = 1.0/(R*R*ST*H2*H3)
      GO TO 11
C         INTERIOR NODES (J = N2+1)
   25 CALL MATRY8
      I = I + 1
      J = 1
      K = 1
      R = R + H1
      THETA = 0.0
      PHI = 0.0
      PHIP = H3
      PHIM = -H3
      L = L + 1
      LX = (3*L) - 2
      LY = LX + 1
      LZ = LY + 1
      LJX = LJX + 3
      LJY = LJX + 1
      LJZ = LJY + 1
      IF(I.EQ.N1+1) GO TO 29
      GO TO 5
   29 D36 = 2.0/R0
C         OUTER BOUNDARY CONDITIONS
   30 CALL MATRY9
      L = L + 1
      LX = (3*L) - 2
      LJX = LJX + 1
      IF(L.LT.NODES+1) GO TO 30
      RETURN
      END


      SUBROUTINE MATRY1
      REAL JL, JD, JU, MAL, MAD, MAU
      COMMON/COMY1/D1,D2,D5,D6,D7,D8,D9,D10,D11,D12,D13,D19,D20,
     1D21,D22,D23,D24,D36
      COMMON/COMY2/LX,LY,LZ,LJX,LJY,LJZ,N33,N330N2,N330N4,N390N4
      COMMON/COMY3/R,ST,TT,H1,H3,NALPH3,NALPH6
      COMMON YL(112,20), YD(112), YU(112,20), NYL(112,20), NYU(112,20)
      COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84,27)
      COMMON MAL(84,27), MAD(84), MAU(84,27)
C
C
C
C        THIS SUBROUTINE SETS UP THE MATRIX COEFFICIENTS AND
C        NODE LOCATIONS FOR NODES ON THE INNER BOUNDARY.
C
C
C
      YD(LJX) = D1 - (3.0*D2/2.0)
      DO 2 ND3=1,20
      YL(LJX,ND3) = 0.0
      NYL(LJX,ND3) = 2
    2 CONTINUE
      YU(LJX,1) = 2.0*D2
      NYU(LJX,1) = LY + NALPH3
      YU(LJX,2) = -D2/2.0
      NYU(LJX,2) = LX + NALPH6
      DO 3 ND4=3,20
      YU(LJX,ND4) = 0.0
      NYU(LJX,ND4) = 2
    3 CONTINUE
      RETURN
      END
      SUBROUTINE MATRY2
      REAL JL, JD, JU, MAL, MAD, MAU
      COMMON/COMY1/D1,D2,D5,D6,D7,D8,D9,D10,D11,D12,D13,D19,D20,
     1D21,D22,D23,D24,D36
      COMMON/COMY2/LX,LY,LZ,LJX,LJY,LJZ,N33,N330N2,N330N4,N390N4
      COMMON/COMY3/R,ST,TT,H1,H3,NALPH3,NALPH6
      COMMON YL(112,20), YD(112), YU(112,20), NYL(112,20), NYU(112,20)
      COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84,27)
      COMMON MAL(84,27), MAD(84), MAU(84,27)
C
C
C
C        THIS SUBROUTINE SETS UP THE MATRIX COEFFICIENTS AND
C        NODE LOCATIONS FOR INTERIOR NODES ON THE THETA = 0
C        POLAR AXIS.
```

```
C
C
C
      YD(LJX) = -(2.0*D6) + (4.0*D5)
      YD(LJY) = -(2.0*D6) + (2.0*D7)
      YD(LJZ) =  (2.0*D7) + (3.0*D5)
      YL(LJX,1) = -D9/2.0
      NYL(LJX,1) = LY - NALPH3 + 3
      YL(LJX,2) =  D9/2.0
      NYL(LJX,2) = LY - NALPH3 + 3 + N33ON2
      YL(LJX,3) = -D10
      NYL(LJX,3) = LZ - NALPH3 + 6
      YL(LJX,4) = D10
      NYL(LJX,4) = LZ - NALPH3 + 6 + N33ON2
      YL(LJX,5) = D10
      NYL(LJX,5) = LZ - NALPH3 + N33
      YL(LJX,6) = -D10
      NYL(LJX,6) = LZ - NALPH3 + N33ON2
      DO 6 ND14=7,20
      YL(LJX,ND14) = 0.0
      NYL(LJX,ND14) = 2
    6 CONTINUE
      YL(LJY,1) = D8 - D7
      NYL(LJY,1) = LY - NALPH3
      YL(LJY,2) = -D9/4.0
      NYL(LJY,2) = LX - NALPH3 + 3
      YL(LJY,3) =  D9/4.0
      NYL(LJY,3) = LX - NALPH3 + 3 + N33ON2
      DO 7 ND15=4,20
      YL(LJY,ND15) = 0.0
      NYL(LJY,ND15) = 2
    7 CONTINUE
      YL(LJZ,1) = D8 - D7
      NYL(LJZ,1) = LZ - NALPH3
      YL(LJZ,2) = -D10
      NYL(LJZ,2) = LX - NALPH3 + 6
      YL(LJZ,3) = D10
      NYL(LJZ,3) = LX - NALPH3 + 6 + N33ON2
      YL(LJZ,4) = D10
      NYL(LJZ,4) = LX - NALPH3 + N33
      YL(LJZ,5) = -D10
      NYL(LJZ,5) = LX - NALPH3 + N33ON2
      DO 8 ND16=6,20
      YL(LJZ,ND16) = 0.0
      NYL(LJZ,ND16) = 2
    8 CONTINUE
      YU(LJX,1) = -(2.0*D5) + D6
      NYU(LJX,1) = LX + 3
      YU(LJX,2) = D11
      NYU(LJX,2) = LY + 3
      YU(LJY,3) = D5
      NYU(LJX,3) = LX + N33ON4 + 3
      YU(LJX,4) = D9/2.0
      NYU(LJX,4) = LY + NALPH3 + 3
      YU(LJX,5) = -D9/2.0
      NYU(LJX,5) = LY + NALPH3 + N33ON2 + 3
      YU(LJY,6) = -D6/2.0
      NYU(LJX,6) = LY + 6
      YU(LJX,7) = -D6/2.0
      NYU(LJX,7) = LX + N33
      YU(LJX,8) = D12/4.0
      NYU(LJX,8) = LZ + 6
      YU(LJX,9) = -D12/4.0
      NYU(LJX,9) = LZ + N33
      YU(LJX,10) = D10
      NYU(LJX,10) = LZ + NALPH3 + 6
      YU(LJX,11) = -D10
      NYU(LJX,11) = LZ + NALPH3 + 6 + N33ON2
      YU(LJY,12) = -D10
      NYU(LJX,12) = LZ + NALPH3 + N33
      YU(LJX,13) = D10
      NYU(LJX,13) = LZ + NALPH3 + N33ON2
      YU(LJX,14) = YU(LJY,1)
      NYU(LJX,14) = LX + N33ON2 + 3
      YU(LJX,15) = -YU(LJX,2)
      NYU(LJX,15) = LY + N33ON2 + 3
      YU(LJX,16) = YU(LJY,3)
      NYU(LJX,16) = LX + N33ON4 + 3
      YU(LJX,17) = YU(LJX,6)
      NYU(LJX,17) = LX + N33ON2 + 6
      YU(LJX,18) = YU(LJY,9)
      NYU(LJX,18) = LX + N33ON2
      YU(LJX,19) = -YU(LJY,8)
```

```
      NYU(LJX,19) = LZ + N33ON2 + 6
      YU(LJX,20) = YU(LJX,8)
      NYU(LJX,20) = LZ + N33ON2
      YU(LJY,1) = -D8 - D7
      NYU(LJY,1) = LY + NALPH3
      YU(LJY,2) = D6
      NYU(LJY,2) = LY + 3
      YU(LJY,3) = D6
      NYU(LJY,3) = LY + N33ON4 + 7
      YU(LJY,4) = -3.0*D13/2.0
      NYU(LJY,4) = LZ + N33ON4 + 3
      YU(LJY,5) = D9/4.0
      NYU(LJY,5) = LX + NALPH3 + 7
      YU(LJY,6) = -D9/4.0
      NYU(LJY,6) = LX + NALPH3 + N33ON2 + 3
      YU(LJY,7) = -D6/2.0
      NYU(LJY,7) = LY + 6
      YU(LJY,8) = -D6/2.0
      NYU(LJY,8) = LY + N33
      YU(LJY,9) = 3.0*D13/4.0
      NYU(LJY,9) = LZ + 6
      YU(LJY,10) = -3.0*D13/4.0
      NYU(LJY,10) = LZ + N33
      YU(LJY,11) = YU(LJY,2)
      NYU(LJY,11) = LY + N33ON2 + 3
      YU(LJY,12) = YU(LJY,3)
      NYU(LJY,12) = LY + N39ON4 + 3
      YU(LJY,13) = -YU(LJY,4)
      NYU(LJY,13) = LZ + N39ON4 + 3
      YU(LJY,14) = YU(LJY,7)
      NYU(LJY,14) = LY + N33ON2 + 6
      YU(LJY,15) = YU(LJY,7)
      NYU(LJY,15) = LY + N33ON2
      YU(LJY,16) = YU(LJY,9)
      NYU(LJY,16) = LZ + N33ON2 + 6
      YU(LJY,17) = -YU(LJY,9)
      NYU(LJY,17) = LZ + N33ON2
      DO 9 ND17=18,20
      YU(LJY,ND17) = 0.0
      NYU(LJY,ND17) = 2
    9 CONTINUE
      YU(LJZ,1) = -D8 - D7
      NYU(LJZ,1) = LZ + NALPH3
      YU(LJZ,2) = -3.0*D5/2.0
      NYU(LJZ,2) = LZ + 3
      YU(LJZ,3) = -D13/2.0
      NYU(LJZ,3) = LY + N33ON4 + 3
      YU(LJZ,4) = D13/4.0
      NYU(LJZ,4) = LY + 6
      YU(LJZ,5) = -D13/4.0
      NYU(LJZ,5) = LY + N33
      YU(LJZ,6) = D10
      NYU(LJZ,6) = LX + NALPH3 + 6
      YU(LJZ,7) = -D10
      NYU(LJZ,7) = LX + NALPH3 + N33ON2 + 6
      YU(LJZ,8) = -D10
      NYU(LJZ,8) = LX + NALPH3 + N33
      YU(LJZ,9) = D10
      NYU(LJZ,9) = LX + NALPH3 + N33ON2
      YU(LJZ,10) = YU(LJZ,2)
      NYU(LJZ,10) = LZ + N33ON2 + 3
      YU(LJZ,11) = -YU(LJZ,3)
      NYU(LJZ,11) = LY + N39ON4 + 3
      YU(LJZ,12) = YU(LJZ,4)
      NYU(LJZ,12) = LY + N33ON2 + 6
      YU(LJZ,13) = -YU(LJZ,4)
      NYU(LJZ,13) = LY + N33ON2
      DO 10 ND18=14,20
      YU(LJZ,ND18) = 0.0
      NYU(LJZ,ND18) = 2
   10 CONTINUE
      RETURN
      END
      SUBROUTINE MATRYS
      REAL JL, JD, JU, MAL, MAD, MAU
      COMMON/COIY1/D1,D2,D5,D6,D7,D8,D9,D10,D11,D12,D13,D19,D20,
     1D21,D22,D23,D24,D26
      COMMON/COIY2/LX,LY,LZ,LJX,LJY,LJZ,N33,N33ON2,N33ON4,N39ON4
      COMMON/COIY3/R,ST,TT,H1,H7,NALPH3,NALPH6
      COMMON YL(112,20), YD(112), YU(112,20), NYL(112,20), NYU(112,20)
      COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84,27)
      COMMON MAL(84,27), MAD(84), MAU(84,27)
```

```
C
C
C
C     THIS SUBROUTINE SETS UP THE MATRIX COEFFICIENTS AND
C     NODE LOCATIONS FOR INTERIOR NODES NOT ON THE POLAR AXIS.
C
C
C
C
      YD(LJX) = (2.0*D5) + (2.0*D20)
      YD(LJY) = (2.0*D20) + (2.0*D7)
      YD(LJZ) = (2.0*D7) + (2.0*D5) + D19
      YL(LJX,1) = -1.0/(2.0*R*TT*H1)
      NYL(LJX,1) = LY - NALPH3
      YL(LJX,2) = -D5 + (D21/2.0)
      NYL(LJX,2) = LX - N33
      YL(LJX,3) = -D11/2.0
      NYL(LJX,3) = LY - N33
      YL(LJX,4) = -D20
      NYL(LJX,4) = LX - 3
      YL(LJX,5) = -1.0/(2.0*R*R*ST*H3)
      NYL(LJX,5) = LZ - 3
      YL(LJX,6) = -D9/4.0
      NYL(LJX,6) = LY - NALPH3 + N33
      YL(LJX,7) = D9/4.0
      NYL(LJX,7) = LY - NALPH3 - N33
      YL(LJX,8) = -D22/4.0
      NYL(LJX,8) = LZ  - NALPH3  + 3
      YL(LJX,9) = D22/4.0
      NYL(LJX,9) = LZ - NALPH3 - 3
      DO 12 ND25=10,20
      YL(LJX,ND25) = 0.0
      NYL(LJX,ND25) = 2
   12 CONTINUE
      YL(LJY,1) = D8 - D7
      NYL(LJY,1) = LY - NALPH3
      YL(LJY,2) = -D20
      NYL(LJY,2) = LY - 3
      YL(LJY,3) = -D23/2.0
      NYL(LJY,3) = LZ - 3
      YL(LJY,4) = -D9/4.0
      NYL(LJY,4) = LX - NALPH3 + N33
      YL(LJY,5) = D9/4.0
      NYL(LJY,5) = LX - NALPH3 - N33
      YL(LJY,6) = -D24/4.0
      NYL(LJY,6) = LZ - N33 + 3
      YL(LJY,7) = D24/4.0
      NYL(LJY,7) = LZ - N33 - 3
      DO 13 ND26=8,20
      YL(LJY,ND26) = 0.0
      NYL(LJY,ND26) = 2
   13 CONTINUE
      YL(LJZ,1) = D8 - D7
      NYL(LJZ,1) = LZ - NALPH3
      YL(LJZ,2) = -D5 + (D21/2.0)
      NYL(LJZ,2) = LZ - N33
      YL(LJZ,3) = D23/2.0
      NYL(LJZ,3) = LY - 3
      YL(LJZ,4) = -D22/4.0
      NYL(LJZ,4) = LX - NALPH3 + 3
      YL(LJZ,5) = D22/4.0
      NYL(LJZ,5) = LX - NALPH3 - 3
      YL(LJZ,6) = -D24/4.0
      NYL(LJZ,6) = LY - N33 + 3
      YL(LJZ,7) = D24/4.0
      NYL(LJZ,7) = LY - N33 - 3
      DO 14 ND27=8,20
      YL(LJZ,ND27) = 0.0
      NYL(LJZ,ND27) = 2
   14 CONTINUE
      YU(LJX,1) = 1.0/(R*R*TT)
      NYU(LJX,1) = LY
      YU(LJX,2) = -YL(LJX,1)
      NYU(LJX,2) = LY + NALPH3
      YU(LJX,3) = -D5 - (D21/2.0)
      NYU(LJX,3) = LX + N33
      YU(LJX,4) = D11/2.0
      NYU(LJX,4) = LY + N33
      YU(LJX,5) = -D20
      NYU(LJX,5) = LX + 3
      YU(LJX,6) = 1.0/(2.0*R*R*ST*H3)
      NYU(LJX,6) = LZ + 3
      YU(LJX,7) = -YL(LJY,6)
```

```
      NYU(LJX,7) = LY + NALPH3 + N33
      YU(LJX,8) = YL(LJX,6)
      NYU(LJX,8) = LY + NALPH3 - N33
      YU(LJX,9) = -YL(LJX,8)
      NYU(LJX,9) = LZ + NALPH3 + 3
      YU(LJX,10) = YL(LJX,8)
      NYU(LJX,10) = LZ + NALPH3 - 3
      DO 15 ND28=11,20
      YU(LJX,ND28) = 0.0
      NYU(LJX,ND28) = 2
   15 CONTINUE
      YU(LJY,1) = -D8 - D7
      NYU(LJY,1) = LY + NALPH3
      YU(LJY,2) = -D20
      NYU(LJY,2) = LY + 3
      YU(LJY,3) = D23/2.0
      NYU(LJY,3) = LZ + 3
      YU(LJY,4) = D9/4.0
      NYU(LJY,4) = LX + NALPH3 + N33
      YU(LJY,5) = -D9/4.0
      NYU(LJY,5) = LX + NALPH3 - N33
      YU(LJY,6) = D24/4.0
      NYU(LJY,6) = LZ + N33 + 3
      YU(LJY,7) = -D24/4.0
      NYU(LJY,7) = LZ + N33 - 3
      DO 16 ND29=8,20
      YU(LJY,ND29) = 0.0
      NYU(LJY,ND29) = 2
   16 CONTINUE
      YU(LJZ,1) = -D8 - D7
      NYU(LJZ,1) = LZ + NALPH3
      YU(LJZ,2) = -D5 - (D21/2.0)
      NYU(LJZ,2) = LZ + N33
      YU(LJZ,3) = -D23/2.0
      NYU(LJZ,3) = LY + 3
      YU(LJZ,4) = D22/4.0
      NYU(LJZ,4) = LX + NALPH3 + 3
      YU(LJZ,5) = -D22/4.0
      NYU(LJZ,5) = LX + NALPH3 - 3
      YU(LJZ,6) = D24/4.0
      NYU(LJZ,6) = LY + N33 + 3
      YU(LJZ,7) = -D24/4.0
      NYU(LJZ,7) = LY + N33 - 3
      DO 17 ND30=8,20
      YU(LJZ,ND30) = 0.0
      NYU(LJZ,ND30) = 2
   17 CONTINUE
      RETURN
      END
      SUBROUTINE MATRY4
      REAL JL, JD, JU, MAL, MAD, MAU
      COMMON/COMY1/D1,D2,D5,D6,D7,D8,D9,D10,D11,D12,D13,D19,D20,
     1D21,D22,D23,D24,D26
      COMMON/COMY2/LX,LY,LZ,LJX,LJY,LJZ,N33,N330N2,N330N4,N390N4
      COMMON/COMY3/A,ST,TT,H1,H3,NALPH3,NALPH6
      COMMON YL(112,20), YD(112), YU(112,20), NYL(112,20), NYU(112,20)
      COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84,27)
      COMMON MAL(84,27), MAD(84), MAU(84,27)
C
C
C
C        THIS SUBROUTINE MAKES THE NECESSARY ALTERATIONS TO
C        THE MATRIX COEFFICIENTS AND NODE LOCATIONS FOR
C        INTERIOR NODES WHERE PHI VARIATIONS MOVE COEFFICIENTS
C        FROM THE LOWER TO THE UPPER TRIANGULAR MATRIX.
C
C
C
      YL(LJX,4) = 0.0
      NYL(LJX,4) = 2
      YL(LJX,5) = 0.0
      NYL(LJX,5) = 2
      NYL(LJX,9) = NYL(LJX,9) + N33
      YL(LJY,2) = 0.0
      NYL(LJY,2) = 2
      YL(LJY,3) = 0.0
      NYL(LJY,3) = 2
      NYL(LJY,7) = NYL(LJY,7) + N33
      YL(LJZ,3) = 0.0
      NYL(LJZ,3) = 2
      NYL(LJZ,5) = NYL(LJZ,5) + N33
      NYL(LJZ,7) = NYL(LJZ,7) + N33
      NYU(LJX,10) = NYU(LJX,10) + N33
      YU(LJX,11) = -D20
```

```
      NYU(LJX,11) = LX - 3 + N33
      YU(LJX,12) = -1.0/(2.0*R*R*ST*H3)
      NYU(LJX,12) = LZ - 3 + N33
      NYU(LJY,7) = NYU(LJY,7) + N33
      YU(LJY,8) = -D20
      NYU(LJY,8) = LY - 3 + N33
      YU(LJY,9) = -D23/2.0
      NYU(LJY,9) = LZ - 3 + N33
      NYU(LJZ,5) = NYU(LJZ,5) + N33
      NYU(LJZ,7) = NYU(LJZ,7) + N33
      YU(LJZ,8) = D23/2.0
      NYU(LJZ,8) = LY - 3 + N33
      RETURN
      END
      SUBROUTINE MATRY5
      REAL JL, JD, JU, MAL, MAD, MAU
      COMMON/COMY1/D1,D2,D5,D6,D7,D8,D9,D10,D11,D12,D13,D19,D20,
     1D21,D22,D23,D24,D36
      COMMON/COMY2/LX,LY,LZ,LJX,LJY,LJZ,N33,N33ON2,N33ON4,N39ON4
      COMMON/COMY3/R,ST,TT,H1,H3,NALPH3,NALPH6
      COMMON YL(112,20), YD(112), YU(112,20), NYL(112,20), NYU(112,20)
      COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84,27)
      COMMON MAL(84,27), MAD(84), MAU(84,27)
C
C
C
C        THIS SUBROUTINE MAKES THE NECESSARY ALTERATIONS TO
C        THE MATRIX COEFFICIENTS AND NODE LOCATIONS FOR
C        INTERIOR NODES WHERE PHI VARIATIONS MOVE COEFFICIENTS
C        FROM THE UPPER TO THE LOWER TRIANGULAR MATRIX.
C
C
C
      NYL(LJX,8) = NYL(LJX,8) - N33
      YL(LJX,10) = -D20
      NYL(LJX,10) = LX + 3 - N33
      YL(LJX,11) = 1.0/(2.0*R*R*ST*H3)
      NYL(LJX,11) = LZ + 3 - N33
      NYL(LJY,6) = NYL(LJY,6) - N33
      YL(LJY,8) = -D20
      NYL(LJY,8) = LX + 3 - N33
      YL(LJY,9) = D23/2.0
      NYL(LJY,9) = LZ + 3 - N33
      NYL(LJZ,4) = NYL(LJZ,4) - N33
      NYL(LJZ,6) = NYL(LJZ,6) - N33
      YL(LJZ,8) = -D23/2.0
      NYL(LJZ,8) = LY + 3 - N33
      YU(LJX,5) = 0.0
      NYU(LJX,5) = 2
      YU(LJX,6) = 0.0
      NYU(LJX,6) = 2
      NYU(LJX,9) = NYU(LJX,9) - N33
      YU(LJY,2) = 0.0
      NYU(LJY,2) = 2
      YU(LJY,3) = 0.0
      NYU(LJY,3) = 2
      NYU(LJY,6) = NYU(LJY,6) - N33
      YU(LJZ,3) = 0.0
      NYU(LJZ,3) = 2
      NYU(LJZ,4) = NYU(LJZ,4) - N33
      NYU(LJZ,6) = NYU(LJZ,6) - N33
      RETURN
      END
      SUBROUTINE MATRY6
      REAL JL, JD, JU, MAL, MAD, MAU
      COMMON/COMY1/D1,D2,D5,D6,D7,D8,D9,D10,D11,D12,D13,D19,D20,
     1D21,D22,D23,D24,D36
      COMMON/COMY2/LX,LY,LZ,LJX,LJY,LJZ,N33,N33ON2,N33ON4,N39ON4
      COMMON/COMY3/R,ST,TT,H1,H3,NALPH3,NALPH6
      COMMON/COMY4/NC2,NC23,NCN2,NCN23,CP,SP,CPP,SPP,CPM,SPM
      COMMON YL(112,20), YD(112), YU(112,20), NYL(112,20), NYU(112,20)
      COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84,27)
      COMMON MAL(84,27), MAD(84), MAU(84,27)
C
C
C
C        THIS SUBROUTINE MAKES THE NECESSARY ALTERATIONS TO
C        THE MATRIX COEFFICIENTS AND NODE LOCATIONS FOR
C        INTERIOR NODES ADJACENT TO THE THETA = 0 POLAR AXIS.
C
C
C
      YL(LJX,2) = 0.0
      NYL(LJX,2) = 2
```

```
      YL(LJX,3) = 0.0
      NYL(LJX,3) = 2
      YL(LJX,7) = 0.0
      NYL(LJX,7) = 2
      YL(LJX,12) = -D5 + (D21/2.0)
      NYL(LJX,12) = NC23 - 2
      YL(LJX,13) = -D11*CP/2.0
      NYL(LJX,13) = NC23 - 1
      YL(LJX,14) = -D11*SP/2.0
      NYL(LJX,14) = NC23
      YL(LJX,15) = D9*CP/4.0
      NYL(LJX,15) = NC23 - NALPH3 - 1
      YL(LJX,16) = D9*SP/4.0
      NYL(LJX,16) = NC23 - NALPH3
      YL(LJY,5) = 0.0
      NYL(LJY,5) = 2
      YL(LJY,6) = 0.0
      NYL(LJY,6) = 2
      YL(LJY,7) = 0.0
      NYL(LJY,7) = 2
      YL(LJY,10) = D9/4.0
      NYL(LJY,10) = NC23 - NALPH3 - 2
      YL(LJY,11) = D24*SPP/4.0
      NYL(LJY,11) = NC23 - 1
      YL(LJY,12) = -D24*CPP/4.0
      NYL(LJY,12) = NC23
      YL(LJY,13) = -D24*SPM/4.0
      NYL(LJY,13) = NC23 - 1
      YL(LJY,14) = D24*CPM/4.0
      NYL(LJY,14) = NC23
      YL(LJZ,2) = 0.0
      NYL(LJZ,2) = 2
      YL(LJZ,6) = 0.0
      NYL(LJZ,6) = 2
      YL(LJZ,7) = 0.0
      NYL(LJZ,7) = 2
      YL(LJZ,9) = (D5 - (D21/2.0))*SP
      NYL(LJZ,9) = NC23 - 1
      YL(LJZ,10) = (-D5 + (D21/2.0))*CP
      NYL(LJZ,10) = NC23
      YL(LJZ,11) = -D24*CPP/4.0
      NYL(LJZ,11) = NC23 - 1
      YL(LJZ,12) = -D24*SPP/4.0
      NYL(LJZ,12) = NC23
      YL(LJZ,13) = D24*CPM/4.0
      NYL(LJZ,13) = NC23 - 1
      YL(LJZ,14) = D24*SPM/4.0
      NYL(LJZ,14) = NC23
      YU(LJX,8) = 0.0
      NYU(LJX,8) = 2
      YU(LJX,13) = -D9*CP/4.0
      NYU(LJX,13) = NC23 - 1 + NALPH3
      YU(LJX,14) = -D9*SP/4.0
      NYU(LJX,14) = NC23 + NALPH3
      YU(LJY,5) = 0.0
      NYU(LJY,5) = 2
      YU(LJY,10) = -D9/4.0
      NYU(LJY,10) = NC23 - 2 + NALPH3
      RETURN
      END
      SUBROUTINE MATRY3
      REAL JL, JD, JU, MAL, MAD, MAU
      COMMON/COMY1/D1,D2,D5,D6,D7,D8,D9,D10,D11,D12,D13,D19,D20,
     1D21,D22,D23,D24,D26
      COMMON/COMY2/LX,LY,LZ,LJX,LJY,LJZ,N33,N33ON2,N33ON4,N79ON4
      COMMON/COMY3/R,ST,TT,H1,H3,NALPH3,NALPH5
      COMMON/COMY4/NC2,NC23,NCN2,NCN23,CP,SP,CPP,SPP,CPM,SPM
      COMMON YL(112,20), YD(112), YU(112,20), NYL(112,20), NYU(112,20)
      COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84,27)
      COMMON MAL(84,27), MAD(84), MAU(84,27)
C
C
C
C        THIS SUBROUTINE MAKES THE NECESSARY ALTERATIONS TO
C        THE MATRIX COEFFICIENTS AND NODE LOCATIONS FOR
C        INTERIOR NODES ADJACENT TO THE THETA = PI POLAR AXIS.
C
C
C
      YL(LJX,6) = 0.0
      NYL(LJX,6) = 2
      YL(LJX,12) = D9*CP/4.0
      NYL(LJX.12) = NCN23 - 1 - NALPH3
```

```
      YL(LJX,13) = D9*SP/4.0
      NYL(LJX,13) = NCN23 - NALPH3
      YL(LJY,4) = 0.0
      NYL(LJY,4) = 2
      YL(LJY,10) = -D9/4.0
      NYL(LJY,10) = NCN23 - NALPH3 - 2
      YU(LJX,3) = 0.0
      NYU(LJX,3) = 2
      YU(LJX,4) = 0.0
      NYU(LJX,4) = 2
      YU(LJX,7) = 0.0
      NYU(LJX,7) = 2
      YU(LJX,13) = -D5 - (D21/2.0)
      NYU(LJX,13) = NCN23 - 2
      YU(LJX,14) = -D11*CP/2.0
      NYU(LJX,14) = NCN23 - 1
      YU(LJX,15) = -D11*SP/2.0
      NYU(LJX,15) = NCN23
      YU(LJX,16) = -D9*CP/4.0
      NYU(LJX,16) = NCN23 - 1 + NALPH3
      YU(LJX,17) = -D9*SP/4.0
      NYU(LJX,17) = NCN23 + NALPH3
      YU(LJY,4) = 0.0
      NYU(LJY,4) = 2
      YU(LJY,6) = 0.0
      NYU(LJY,6) = 2
      YU(LJY,7) = 0.0
      NYU(LJY,7) = 2
      YU(LJY,10) = D9/4.0
      NYU(LJY,10) = NCN23 - 2 + NALPH3
      YU(LJY,11) = -D24*SPP/4.0
      NYU(LJY,11) = NCN23 - 1
      YU(LJY,12) = D24*CPP/4.0
      NYU(LJY,12) = NCN23
      YU(LJY,13) = D24*SPM/4.0
      NYU(LJY,13) = NCN23 - 1
      YU(LJY,14) = -D24*CPM/4.0
      NYU(LJY,14) = NCN23
      YU(LJZ,2) = 0.0
      NYU(LJZ,2) = 2
      YU(LJZ,6) = 0.0
      NYU(LJZ,6) = 2
      YU(LJZ,7) = 0.0
      NYU(LJZ,7) = 2
      YU(LJZ,9) = (D5 + (D21/2.0))*SP
      NYU(LJZ,9) = NCN23 - 1
      YU(LJZ,10) = -(D5 + (D21/2.0))*CP
      NYU(LJZ,10) = NCN23
      YU(LJZ,11) = -D24*CPP/4.0
      NYU(LJZ,11) = NCN23 - 1
      YU(LJZ,12) = -D24*SPP/4.0
      NYU(LJZ,12) = NCN23
      YU(LJZ,13) = D24*CPM/4.0
      NYU(LJZ,13) = NCN23 - 1
      YU(LJZ,14) = D24*SPM/4.0
      NYU(LJZ,14) = NCN23
      RETURN
      END
      SUBROUTINE MATRY8
      REAL JL, JD, JU, MAL, MAD, MAU
      COMMON/COMY1/D1,D2,D5,D6,D7,D8,D9,D10,D11,D12,D13,D19,D20,
     1D21,D22,D23,D24,D36
      COMMON/COMY2/LX,LY,LZ,LJX,LJY,LJZ,N33,N330M2,N330N4,N790N4
      COMMON/COMY3/R,ST,TT,H1,H3,NALPH3,NALPH6
      COMMON YL(112,20), YD(112), YU(112,20), NYL(112,20), NYU(112,20)
      COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84,27)
      COMMON MAL(84,27), MAD(84), MAU(84,27)
C
C
C        THIS SUBROUTINE SETS UP THE MATRIX COEFFICIENTS AND
C        NODE LOCATIONS FOR INTERIOR NODES ON THE THETA = PI
C        POLAR AXIS.
C
C
C
      YD(LJX) = -(2.0*D6) + (4.0*D5)
      YD(LJY) = -(2.0*D5) + (2.0*D7)
      YD(LJZ) =  (2.0*D7) + (3.0*D5)
      YL(LJX,1) = -(2.0*D5) + D6
      NYL(LJX,1) = LX - N330N2
      YL(LJX,2) = -D11
      NYL(LJX,2) = LY - N330N2
```

```
      YL(LJX,3) = D6
      NYL(LJX,3) = LX - N33ON4
      YL(LJX,4) = -D9/2.0
      NYL(LJX,4) = LY - NALPH3 - N33
      YL(LJX,5) = D9/2.0
      NYL(LJX,5) = LY - NALPH3 - N33ON2
      YL(LJX,6) = -D6/2.0
      NYL(LJX,6) = LX - N33ON2 + 3
      YL(LJX,7) = -D6/2.0
      NYL(LJX,7) = LX - N33ON2 - 3
      YL(LJX,8) = D12/4.0
      NYL(LJX,8) = LZ - N33ON2 + 3
      YL(LJX,9) = -D12/4.0
      NYL(LJX,9) = LZ - N33ON2 - 3
      YL(LJX,10) = D10
      NYL(LJX,10) = LZ - NALPH3 - N33 + 3
      YL(LJX,11) = -D10
      NYL(LJX,11) = LZ - NALPH3 - N33ON2 + 3
      YL(LJX,12) = -D10
      NYL(LJX,12) = LZ - NALPH3 - 3
      YL(LJX,13) = D10
      NYL(LJX,13) = LZ - NALPH3 - N33ON2 - 3
      YL(LJX,14) = YL(LJX,1)
      NYL(LJX,14) = LX - N33
      YL(LJX,15) = -YL(LJX,2)
      NYL(LJX,15) = LY - N33
      YL(LJX,16) = YL(LJX,3)
      NYL(LJX,16) = LX - N39ON4
      YL(LJX,17) = YL(LJX,6)
      NYL(LJX,17) = LX - N33 + 3
      YL(LJX,18) = YL(LJX,6)
      NYL(LJX,18) = LX - 3
      YL(LJX,19) = -YL(LJX,8)
      NYL(LJX,19) = LZ - N33 + 3
      YL(LJX,20) = YL(LJX,8)
      NYL(LJX,20) = LZ - 3
      YL(LJY,1) = D8 - D7
      NYL(LJY,1) = LY - NALPH3
      YL(LJY,2) = D5
      NYL(LJY,2) = LY - N33ON2
      YL(LJY,3) = D6
      NYL(LJY,3) = LY - N33ON4
      YL(LJY,4) = -3.0*D13/2.0
      NYL(LJY,4) = LZ - N33ON4
      YL(LJY,5) = -D9/4.0
      NYL(LJY,5) = LX - NALPH3 - N33
      YL(LJY,6) = D9/4.0
      NYL(LJY,6) = LX - NALPH3 - N33ON2
      YL(LJY,7) = -D6/2.0
      NYL(LJY,7) = LY - N33ON2 + 3
      YL(LJY,8) = -D6/2.0
      NYL(LJY,8) = LY - N33ON2 - 3
      YL(LJY,9) = -3.0*D13/4.0
      NYL(LJY,9) = LZ - N33ON2 + 3
      YL(LJY,10) = 3.0*D13/4.0
      NYL(LJY,10) = LZ - N33ON2 - 3
      YL(LJY,11) = YL(LJY,2)
      NYL(LJY,11) = LY - N33
      YL(LJY,12) = YL(LJY,3)
      NYL(LJY,12) = LY - N39ON4
      YL(LJY,13) = -YL(LJY,4)
      NYL(LJY,13) = LZ - N39ON4
      YL(LJY,14) = YL(LJY,7)
      NYL(LJY,14) = LY - N33 + 3
      YL(LJY,15) = YL(LJY,7)
      NYL(LJY,15) = LY - 3
      YL(LJY,16) = YL(LJY,9)
      NYL(LJY,16) = LZ - N33 + 3
      YL(LJY,17) = -YL(LJY,9)
      NYL(LJY,17) = LZ - 3
      DO 24 ND31=18,20
      YL(LJY,ND31) = 0.0
      NYL(LJY,ND31) = 0
   24 CONTINUE
      YL(LJZ,1) = D8 - D7
      NYL(LJZ,1) = LZ - NALPH3
      YL(LJZ,2) = -3.0*D5/2.0
      NYL(LJZ,2) = LZ - N33ON2
      YL(LJZ,3) = -D13/2.0
      NYL(LJZ,3) = LY - N33ON4
      YL(LJZ,4) = -D13/4.0
      NYL(LJZ,4) = LY - N33ON2 + 3
      YL(LJZ,5) = D13/4.0
```

```
      NYL(LJZ,5) = LY - N33ON2 - 3
      YL(LJZ,6) = D10
      NYL(LJZ,6) = LX - NALPH3 - N33 + 3
      YL(LJZ,7) = -D10
      NYL(LJZ,7) = LX - NALPH3 - N33ON2 + 3
      YL(LJZ,8) = -D10
      NYL(LJZ,8) = LX - NALPH3 - 3
      YL(LJZ,9) = D10
      NYL(LJZ,9) = LX - NALPH3 - N33ON2 - 3
      YL(LJZ,10) = YL(LJZ,2)
      NYL(LJZ,10) = LZ - N33
      YL(LJZ,11) = -YL(LJZ,3)
      NYL(LJZ,11) = LY - N39ON4
      YL(LJZ,12) = YL(LJZ,4)
      NYL(LJZ,12) = LY - N33 + 3
      YL(LJZ,13) = -YL(LJZ,4)
      NYL(LJZ,13) = LY - 3
      DO 25 ND32=14,20
      YL(LJZ,ND32) = 0.0
      NYL(LJZ,ND32) = 2
   25 CONTINUE
      YU(LJX,1) = -YL(LJX,4)
      NYU(LJX,1) = LY + NALPH3 - N33
      YU(LJX,2) = YL(LJX,4)
      NYU(LJX,2) = LY + NALPH3 - N33ON2
      YU(LJX,3) = - D10
      NYU(LJX,3) = LZ + NALPH3 + 3- N33
      YU(LJX,4) = D10
      NYU(LJX,4) = LZ + NALPH3 - N33ON2 + 3
      YU(LJX,5) = D10
      NYU(LJX,5) = LZ + NALPH3 - 3
      YU(LJX,6) = -D10
      NYU(LJX,6) = LZ + NALPH3 - N33ON2 - 3
      DO 26 ND33=7,20
      YU(LJX,ND33) = 0.0
      NYU(LJX,ND33) = 2
   26 CONTINUE
      YU(LJY,1) = -D8 - D7
      NYU(LJY,1) = LY + NALPH3
      YU(LJY,2) = -YL(LJY,5)
      NYU(LJY,2) = LX + NALPH3 - N33
      YU(LJY,3) = YL(LJY,5)
      NYU(LJY,3) = LX + NALPH3 - N33ON2
      DO 27 ND34=4,20
      YU(LJY,ND34) = 0.0
      NYU(LJY,ND34) = 2
   27 CONTINUE
      YU(LJZ,1) = -D8 - D7
      NYU(LJZ,1) = LZ + NALPH3
      YU(LJZ,2) = -D10
      NYU(LJZ,2) = LX + NALPH3 - N33 + 3
      YU(LJZ,3) = D10
      NYU(LJZ,3) = LX + NALPH3 - N33ON2 + 3
      YU(LJZ,4) = D10
      NYU(LJZ,4) = LX + NALPH3 - 3
      YU(LJZ,5) = -D10
      NYU(LJZ,5) = LX + NALPH3 - N33ON2 - 3
      DO 28 ND35=6,20
      YU(LJZ,ND35) = 0.0
      NYU(LJZ,ND35) = 2
   28 CONTINUE
      RETURN
      END
      SUBROUTINE MATRY9
      REAL JL, JD, JU, MAL, MAD, MAU
      COMMON/COMY1/D1,D2,D5,D6,D7,D8,D9,D10,D11,D12,D18,D19,D20,
     1D21,D22,D23,D24,D36
      COMMON/COMY2/LX,LY,LZ,LJX,LJY,LJZ,N33,N33ON2,N33ON4,N39ON4
      COMMON/COMY3/R,ST,TT,H1,H3,NALPH3,NALPH5
      COMMON YL(112,20), YD(112), YU(112,20), NYL(112,20), NYU(112,20)
      COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84,27)
      COMMON MAL(84,27), MAD(84), MAU(84,27)
C
C
C
C         THIS SUBROUTINE SETS UP THE MATRIX COEFFICIENTS AND
C         NODE LOCATIONS FOR NODES ON THE OUTER BOUNDARY.
C
C
C
      YD(LJX) = -D36 -(3.0*D2/2.0)
      YL(LJX,1) = 2.0*D2
      NYL(LJX,1) = LX - NALPH3
```

```
      YL(LJX,2) = -D2/2.0
      NYL(LJX,2) = LX - NALPHS
      DO 31 ND37=3,20
      YL(LJX,ND37) = 0.0
      NYL(LJX,ND37) = 2
   31 CONTINUE
      DO 32 ND38=1,20
      YU(LJX,ND38) = 0.0
      NYU(LJX,ND38) = 2
   32 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PRESS(VORTEX,NODE3,RRI,RRO,RRE,NN1,NN2,NN3,P,NNALPH,
     1SUM,PTEST,ERRMAX)
      DIMENSION P(NNALPH), VORTEX(NODE3), SUM(NN3), PTEST(NN3)
C
C
C
C        THIS SUBROUTINE DETERMINES THE PRESSURE FIELD ON THE
C        SURFACE OF THE SPHERE (R = RI). SINCE PRESSURE IS
C        ARBITRARY UP TO A CONSTANT, THE PRESSURE ON THE SPHERE
C        IS SET EQUAL TO ZERO AT THE THETA = 0 NODE.
C
C
C
      N1 = NN1
      N2 = NN2
      N3 = NN3
      RI = RRI
      RO = RRO
      RE = RRE
      RIONRE = RI/RE
      XN1 = N1
      XN2 = N2
      XN3 = N3
      PI = 3.14159265358 98
      H1 = (RO - RI)/XN1
      H12INV = 1.0/(2.0*H1)
      H2 = PI/XN2
      H3 = 2.0*PI/XN3
      NALPHA = NNALPH
      NALPH3 = 3*NALPHA
      NALPH6 = 6*NALPHA
C        PRESSURE AT THETA = 0 POLAR AXIS.
      P(1) = 0.0
C        PRESSURE AT NON-POLAR NODES.
      PHI = 0.0
      DO 800 K=1,N3
      SP = SIN(PHI)
      CP = COS(PHI)
      DA = -(VORTEX(2)*SP) + (VORTEX(3)*CP)
      DAB = DA/RI
      DB = -3.0*DA
      DC = -(VORTEX(2+NALPH3)*SP) + (VORTEX(3+NALPH3)*CP)
      DD = 4.0*DC
      DE = -(VORTEX(2+NALPH6)*SP) + (VORTEX(3+NALPH6)*CP)
      DF = H12INV*(DB + DD - DE)
      D = (DAB + DF)*RIONRE*H2
      SUM(K) = D/2.0
      PHI = PHI + H3
  800 CONTINUE
      J = 2
      K = 1
      L = 2
      LZ = 6
  801 DA = VORTEX(LZ)
      DAB = DA/RI
      DB = -3.0*DA
      DC = 4.0*VORTEX(LZ+NALPH3)
      DD = -VORTEX(LZ+NALPH6)
      DE = H12INV*(DB + DC + DD)
      D = (DAB + DE)*RIONRE*H2
      DON2 = D/2.0
      P(L) = SUM(K) + DON2
      SUM(K) = SUM(K) + D
      L = L + 1
      LZ = LZ + 3
      IF(K.EQ.N3) GO TO 802
      K = K + 1
      GO TO 801
  802 J = J + 1
      K = 1
      IF(J.EQ.N2+1) GO TO 803
      GO TO 801
C        PRESSURE AT THETA = PI POLAR AXIS.
  803 PHI = 0.0
      PL = 0
  804 SP = SIN(PHI)
      CP = COS(PHI)
      DA = -(VORTEX(LZ-1)*SP) + (VORTEX(LZ)*CP)
      DAB = DA/RI
      DB = -3.0*DA
      DC = -(VORTEX(LZ+NALPH3-1)*SP) + (VORTEX(LZ+NALPH3)*CP)
      DD = 4.0*DC
```

```
      DE = -(VORTEX(LZ+NALPH6-1)*SP) + (VORTEX(LZ+NALPH6)*CP)
      DF = H12INV*(DR + DD - DE)
      D = (DAB + DF)*RIONRE*H2
      DON2 = D/2.0
      PTEST(K) = SUM(K) + DON2
      PL = PL + PTEST(K)
      IF(K.EQ.N3) GO TO 805
      K = K + 1
      PHI = PHI + H3
      GO TO 804
  805 P(L) = PL/XN3
C        TEST THAT PRESSURE IS SINGLE-VALUED.
      PMAX = PTEST(1)
      PMIN = PTEST(1)
      DO 806 K=2,N3
      IF(PTEST(K).GT.PMAX) PMAX = PTEST(K)
      IF(PTEST(K).LT.PMIN) PMIN = PTEST(K)
  806 CONTINUE
      ERRMAX = PMAX - PMIN
      RETURN
      END
```

```
      SUBROUTINE SCRIBE (N1,N2,N3,RI,RO,U0,SHEAR,NNODES,W,M,DELTA,H,
     1REST,DIFFCE,NODE2,NODE3,U)
      DIMENSION H(NNODES),U(NODE3)
C
C
C
C        THIS SUBROUTINE WRITES OUT ALL THE PRINCIPAL RESULTS
C        OBTAINED BY PROGRAM UTRANS.
C
C
C
      NODES = NNODES
      WRITE (6,400) RI,RO,U0,SHEAR
  400 FORMAT(1H1/////1X,19HRADIUS OF SPHERE =,F7.4,32X,
     126HRADIUS OF OUTER BOUNDARY =,F7.4/1X,
     240HCENTRE-LINE VELOCITY AT OUTER BOUNDARY =,F7.4,18X,
     338HMAGNITUDE OF SHEAR AT OUTER BOUNDARY =,F7.4)
      WRITE (6,401) N1,N2,N3,NODES
  401 FORMAT(1H0,27HNUMBER OF RADIAL SPACINGS =,I3,27X,
     132HNUMBER OF POLAR ANGLE SPACINGS =,I3/1X,
     236HNUMBER OF AZIMUTHAL ANGLE SPACINGS =,I3,18X,
     317HNUMBER OF NODES =,I5)
      WRITE (6,402) W,DELTA,DIFFCE,M,REST
  402 FORMAT(1H0,26HRELAXATION FACTOR CHOSEN =,F7.4/1X,
     157HMAXIMUM SQUARE DIFFERENCE BETWEEN ELEMENTS IN SUCCESSIVE ,
     228HITERATIONS FOR CONVERGENCE =,F12.10/1X,
     357HMAXIMUM SQUARE DIFFERENCE BETWEEN ELEMENTS IN SUCCESSIVE ,
     428HITERATIONS AT CONVERGENCE  =,F12.10/1X,
     538HNUMBER OF ITERATIONS FOR CONVERGENCE =,I4/1X,
     640HMAXIMUM SQUARE RESIDUAL AT CONVERGENCE =,F12.10/////)
      WRITE (6,403)
  403 FORMAT(1X,10X,22HSCALAR POTENTIAL FIELD/)
      WRITE (6,404) (K,H(K),K=1,NODES)
  404 FORMAT((1X,5(I4,4H ---,F9.5,5X,)I4,4H ---,F9.5))
      WRITE (6,405)
  405 FORMAT(1H0////11X,31HVELOCITY FIELD (UR,UTHETA,UPHI)/)
      WRITE (6,406) (K,U(K),U(K+NODES),U(K+NODE2),K=1,NODES)
  406 FORMAT((1X,2(I4,4H ---,F9.5,6X,F9.5,6X,F9.5,15X,)1X))
      WRITE (6,407)
  407 FORMAT(1H1)
      RETURN
      END
```

```
      SUBROUTINE SOR (E,F,NE,NF,B,H,HGUESS,NNEQNS,NODES,WW,M,MMMAX,
     1DDELTA,RREST,DDIFFC)
      DIMENSION E(4,NNEQNS), F(4,NNEQNS), NE(4,NNEQNS), NF(4,NNEQNS)
      DIMENSION B(NNEQNS), H(NODES), HGUESS(NNEQNS)
C
C
C
C        THIS SUBROUTINE SOLVES THE MATRIX EQUATION SET UP BY
C        SUBROUTINE MATRIX BY SUCCESSIVE POINT OVER-RELAXATION.
C
C
C
      DELTA = DDELTA
      MMAX = MMMAX
      NEQNS = NNEQNS
      W = WW
      H(NEQNS + 1) = 0.0
C        SET THE SCALAR POTENTIAL FIELD H EQUAL TO THE
C        GUESSED FIELD HGUESS.
      DO 199 I = 1,NEQNS
      H(I) = HGUESS(I)
  199 CONTINUE
      M = 1
  200 DIFFCE = 0.0
C        UPDATE THE SCALAR POTENTIAL FIELD H.
      DO 202 I = 1,NEQNS
      D = 0.0
      DO 201 J = 1,4
      JNF = NF(J,I)
      JNE = NE(J,I)
      D = D + (F(J,I)*H(JNF)) + (E(J,I)*H(JNE))
  201 CONTINUE
      HI = H(I)
      H(I) = - (W*D) + ((1.0 - W)*HI) + (B(I)*W)
      SQD = (H(I) - HI)*(H(I) - HI)
      DIFFCE = AMAX1(SQD,DIFFCE)
  202 CONTINUE
      DDIFFC = DIFFCE
C        TEST FOR CONVERGENCE.
      IF (DIFFCE.LE.DELTA) GO TO 203
C        TEST WHETHER THE MAXIMUM ALLOWABLE NUMBER OF
C        ITERATIONS HAS BEEN EXCEEDED.
      IF(M.EQ.MMAX) GO TO 203
      M = M + 1
      GO TO 200
C        CALCULATE THE MAXIMUM SQUARE RESIDUAL.
  203 REST = 0.0
      DO 205 I =1,NEQNS
      D = 0.0
      DO 204 J=1,4
      JNF = NF(J,I)
      JNE = NE(J,I)
      D = D + (F(J,I)*H(JNF)) + (E(J,I)*H(JNE))
  204 CONTINUE
      V = D + H(I) - B(I)
      SQR = V*V
      REST = AMAX1(SQR,REST)
  205 CONTINUE
      RREST = REST
      RETURN
      END




      SUBROUTINE SORJMA(NNEQN,VORTEX,VORNEW,WW1,DDELT1,MM1,MMMAX1,
     1RREST1,DDIFF1,NNODE3,NNALPH,DDELTA)
      DIMENSION VORTEX(NNODE3), VORNEW(NNODE3)
      REAL JL, JD, JU, MAL, MAD, MAU
      COMMON YL(112,20), YD(112), YU(112,20), NYL(112,20), NYU(112,20)
      COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84,27)
      COMMON MAL(84,27), MAD(84), MAU(84,27)
C
C
C
C        THIS SUBROUTINE SOLVES THE MATRIX EQUATION SET UP BY
C        SUBROUTINES MATRJ AND MATMA BY SUCCESSIVE POINT
C        OVER-RELAXATION.
```

```
C
C
C
      DELT1 = DDELT1
      DELTAT = DDELTA
      MMAX1 = MMMAX1
      NEQN = NNEQN
      NALPHA = NNALPH
      NALPH3 = 3*NALPHA
      NODE3 = NNODE3
      W1 = WW1
C         SET THE VECTOR VORNEW EQUAL TO THE VORTICITY FIELD VORTEX
C         (1) AT THE PREVIOUS TIME-STEP FOR INTERIOR NODES, AND
C         (2) AT THE PRESENT TIME-STEP FOR BOUNDARY NODES.
      DO 389 I=1,NODE3
      VORNEW(I) = VORTEX(I)
  389 CONTINUE
C         UPDATE THE VORTICITY FIELD.
      M1 = 1
  390 DIFF1 = 0.0
      DO 392 I=1,NEQN
      IP = I + NALPH3
      D = 0.0
      DO 391 J=1,27
      JNU = NU(I,J)
      JNL = NL(I,J)
      D1 = VORNEW(JNU)*JU(I,J)
      D2 = VORNEW(JNU)*MAU(I,J)
      D3 = VORNEW(JNL)*JL(I,J)
      D4 = VORNEW(JNL)*MAL(I,J)
      D5 = DELTAT*(D1+D2+D3+D4)
      D = D + D5
  391 CONTINUE
      VORI = VORNEW(IP)
      D6 = ((JD(I)+MAD(I))*DELTAT) + 1.0
      VORNEW(IP) = - (W1*(D-VORTEX(IP))/D6) + ((1.0-W1)*VORI)
      D7 = VORNEW(IP) - VORI
      SQD = D7*D7
      DIFF1 = AMAX1(DIFF1,SQD)
  392 CONTINUE
      DDIFF1 = DIFF1
C         TEST FOR CONVERGENCE.
      IF(DIFF1.LE.DELT1) GO TO 393
C         TEST WHETHER THE MAXIMUM ALLOWABLE NUMBER OF
C         ITERATIONS HAS BEEN EXCEEDED.
      IF(M1.EQ.MMAX1) GO TO 393
      M1 = M1 + 1
      GO TO 390
C         CALCULATE THE MAXIMUM SQUARE RESIDUAL.
  393 REST1 = 0.0
      DO 395 L=1,NEQN
      LP = L + NALPH3
      D = 0.0
      DO 394 M=1,27
      JNU = NU(L,M)
      JNL = NL(L,M)
      D1 = VORNEW(JNU)*JU(L,M)
      D2 = VORNEW(JNU)*MAU(L,M)
      D3 = VORNEW(JNL)*JL(L,M)
      D4 = VORNEW(JNL)*MAL(L,M)
      D5 = DELTAT*(D1+D2+D3+D4)
      D = D + D5
  394 CONTINUE
      D6 = ((JD(L)+MAD(L))*DELTAT) + 1.0
      D8 = (D6*VORNEW(LP)) + D - VORTEX(LP)
      SQR = D8*D8
      REST1 = AMAX1(REST1,SQR)
  395 CONTINUE
C         SET UP THE VORTICITY FIELD VORTEX FOR ALL INTERIOR
C         NODES FROM THE VECTOR VORNEW.
      DO 396 K=1,NEQN
      KP = K + NALPH3
      VORTEX(KP) = VORNEW(KP)
  396 CONTINUE
      MM1 = M1
      RREST1 = REST1
      RETURN
      END
```

```
      SUBROUTINE SORY(VORTEX,A,ANEW,ASTAR,NNALPH,NNEQNY,NNODE3,WW,
     1DDELTY,MM,MMMAX,RRESTY,DDIFFY,DIFF1,DIFF2,DDELT,DDIFFC)
      DIMENSION VORTEX(NNODE3), A(NNODE3), ANEW(NNODE3), ASTAR(NNEQNY)
      DIMENSION DIFF1(NNEQNY), DIFF2(NNEQNY)
      REAL JL, JD, JU, MAL, MAD, MAU
      COMMON YL(112,20), YD(112), YU(112,20), NYL(112,20), NYU(112,20)
      COMMON JL(84,27), JD(84), JU(84,27), NL(84,27), NU(84,27)
      COMMON MAL(84,27), MAD(84), MAU(84,27)
C
C
C
C        THIS SUBROUTINE SOLVES THE MATRIX EQUATION SET UP BY
C        SUBROUTINES MATRY AND MATRY1-9 BY SUCCESSIVE POINT
C        OVER-RELAXATION.
C
C
C
C
      DELT = DDELT
      DELTY = DDELTY
      MMAX = MMMAX
      NALPHA = NNALPH
      NALPH2 = 2*NALPHA
      NALPH3 = 3*NALPHA
      NEQNY = NNEQNY
      NODE3 = NNODE3
      W = WW
C        INITIALISE THE VECTOR OF DIFFERENCES BETWEEN ELEMENTS
C        OF ASTAR IN THE PENULTIMATE SUCCESSIVE ITERATIONS.
      DO 48 IN=1,NEQNY
      DIFF1(IN) = 0.0
   48 CONTINUE
C        SET UP THE VECTOR ASTAR CORRESPONDING TO THE INITIAL
C        GUESS A OF THE VECTOR POTENTIAL FIELD ANEW.
      DO 49 I=1,NALPHA
      NDA = (3*I) - 2
      ASTAR(I) = A(NDA)
   49 CONTINUE
      NDB = NALPHA+1
      NDC = NEQNY-NALPHA
      DO 50 J=NDB,NDC
      NDD = J+NALPH2
      ASTAR(J) = A(NDD)
   50 CONTINUE
      NDE = NDC + 1
      DO 51 K=NDE,NEQNY
      NDF = (3*K) - 2 - NODE3
      ASTAR(K) = A(NDF)
   51 CONTINUE
C        SET THE VECTOR POTENTIAL FIELD ANEW EQUAL TO THE
C        GUESSED FIELD A.
      DO 52 L=1,NODE3
      ANEW(L) = A(L)
   52 CONTINUE
      M = 1
   53 DIFFY = 0.0
      N = 1
C        UPDATE THE VECTORS ASTAR AND ANEW (INNER BOUNDARY NODES).
   54 DG = 0.0
      DO 55 I=1,20
      MU = NYU(N,I)
      ML = NYL(N,I)
      DG = DG + (YL(N,I)*ANEW(ML)) + (YU(N,I)*ANEW(MU))
   55 CONTINUE
      ASTARN = ASTAR(N)
      ASTAR(N) = - (W*DG/YD(N)) + ((1.0-W)*ASTARN)
      D = ASTAR(N) - ASTARN
      DIFF2(N) = D
      SQDY = D*D
      DIFFY = AMAX1(SQDY,DIFFY)
      NDH = (3*N)
      ANEW(NDH-2) = ASTAR(N)
      ANEW(NDH-1) = 0.0
      ANEW(NDH) = 0.0
      N = N + 1
      IF(N.LT.NDB) GO TO 54
C        UPDATE THE VECTORS ASTAR AND ANEW (INTERIOR NODES).
   56 DG = 0.0
      DO 57 I=1,20
      MU = NYU(N,I)
      ML = NYL(N,I)
      DG = DG + (YL(N,I)*ANEW(ML)) + (YU(N,I)*ANEW(MU))
   57 CONTINUE
```

```
      ASTARN = ASTAR(N)
      ASTAR(N) = - (W*(DG - VORTEX(N+NALPH2))/YD(N)) + ((1.0-W)*ASTARN)
      D = ASTAR(N) - ASTARN
      DIFF2(N) = D
      SQDY = D*D
      DIFFY = AMAX1(SQDY,DIFFY)
      ANEW(N+NALPH2) = ASTAR(N)
      N = N + 1
      IF(N.LT.NDE) GO TO 56
C         UPDATE THE VECTORS ASTAR AND ANEW (OUTER BOUNDARY NODES).
   58 DG = 0.0
      DO 59 I=1,20
      MU = NYU(N,I)
      ML = NYL(N,I)
      DG = DG + (YL(N,I)*ANEW(ML)) + (YU(N,I)*ANEW(MU))
   59 CONTINUE
      ASTARN = ASTAR(N)
      ASTAR(N) = - (W*DG/YD(N)) + ((1.0-W)*ASTARN)
      D = ASTAR(N) - ASTARN
      DIFF2(N) = D
      SQDY = D*D
      DIFFY = AMAX1(SQDY,DIFFY)
      NDH = (3*N) - NODE3
      ANEW(NDH-2) = ASTAR(N)
      ANEW(NDH-1) = 0.0
      ANEW(NDH) = 0.0
      N = N + 1
      IF(N.LE.NEQNY) GO TO 58
C         TEST FOR CONVERGENCE.
      IF(DIFFY.LE.DELTY) GO TO 60
C         TEST WHETHER THE MAXIMUM ALLOWABLE NUMBER OF
C         ITERATIONS HAS BEEN EXCEEDED.
      IF(M.EQ.MMAX) GO TO 60
C         TEST WHETHER THE VECTOR ASTAR IS BEING UPDATED
C         BY A CONSTANT VECTOR IN SUCCESSIVE ITERATIONS.
      DIFFCE = 0.0
      DO 591 IM=1,NEQNY
      DK = DIFF1(IM) - DIFF2(IM)
      DIFF1(IM) = DIFF2(IM)
      DL = ABS(DK)
      DIFFCE = AMAX1(DIFFCE,DL)
  591 CONTINUE
      IF(DIFFCE.LE.DELT) GO TO 60
      M = M + 1
      GO TO 53
   60 DDIFFY = DIFFY
      M1 = M
      DDIFFC = DIFFCE
C         CALCULATE THE MAXIMUM SQUARE RESIDUAL.
      RESTY = 0.0
      N = 1
   61 DI = 0.0
      DO 62 I=1,20
      MU = NYU(N,I)
      ML = NYL(N,I)
      DI = DI + (YL(N,I)*ANEW(ML)) + (YU(N,I)*ANEW(MU))
   62 CONTINUE
      DJ = DI + (YD(N)*ASTAR(N))
      SQRY = DJ*DJ
      RESTY = AMAX1(SQRY,RESTY)
      N = N + 1
      IF(N.LT.NDB) GO TO 61
   63 DI = 0.0
      DO 64 I=1,20
      MU = NYU(N,I)
      ML = NYL(N,I)
      DI = DI + (YL(N,I)*ANEW(ML)) + (YU(N,I)*ANEW(MU))
   64 CONTINUE
      DJ = DI + (YD(N)*ASTAR(N)) - VORTEX(N+NALPH2)
      SQRY = DJ*DJ
      RESTY = AMAX1(SQRY,RESTY)
      N = N + 1
      IF(N.LT.NDE) GO TO 63
   65 DI = 0.0
      DO 66 I=1,20
      MU = NYU(N,I)
      ML = NYL(N,I)
      DI = DI + (YL(N,I)*ANEW(ML)) + (YU(N,I)*ANEW(MU))
   66 CONTINUE
      DJ = DI + (YD(N)*ASTAR(N))
      SQRY = DJ*DJ
      RESTY = AMAX1(SQRY,RESTY)
      N = N + 1
      IF(N.LE.NEQNY) GO TO 65
```

```
      RRESTY = RESTY
C        SET THE VECTOR POTENTIAL FIELD A EQUAL TO THE FIELD ANEW.
      DO 67 I=1,NODE3
      A(I) = ANEW(I)
   67 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SPEED(A,V,NNODE3,NN1,NN2,NN3,RRI,RRO)
      DIMENSION A(NNODE3), V(NNODE3)
C
C
C
C        THIS SUBROUTINE CONVERTS THE VECTOR POTENTIAL FIELD
C        A DETERMINED BY SUBROUTINE SORY INTO THE THREE-
C        DIMENSIONAL ROTATIONAL VELOCITY FIELD V.
C
C
C
      N1 = NN1
      N2 = NN2
      N3 = NN3
      N33 = 3*N3
      N33ON2 = N33/2
      NODE3 = NNODE3
      NALPH3 = NODE3/(N1 + 1)
      NALPH6 = 2*NALPH3
      NALPHA = NALPH3/3
      RI = RRI
      RO = RRO
      PI = 3.14159265358 98
      XN1 = N1
      XN2 = N2
      XN3 = N3
      H1 = (RO - RI)/XN1
      H2 = PI/XN2
      H3 = (2.0*PI)/XN3
C        INNER BOUNDARY NODES.
      I = 1
      J = 1
      K = 1
      LX = 1
      LY = 2
      LZ = 3
      R = RI
      D1 = 1.0/(R*H2)
      D2 = D1/(4.0*H3)
      D3 = 1.0/(2.0*H1)
      D4 = 1.0/R
      NC2 = (NALPHA*(I-1)) + 1
      NC23 = 3*NC2
      NCN2 = NALPHA*I
      NCN23 = 3*NCN2
      V(LX) = 0.0
      DA = A(LX+6) - A(LX+N33ON2+6) - A(LX+N33) + A(LX+N33ON2)
      DB = -(4.0*A(LZ+NALPH3)) + A(LZ+NALPH6)
      V(LY) = (DA*D2) + (DB*D3)
      DC = (4.0*A(LY+NALPH3)) - A(LY+NALPH6)
      DD = (-A(LX+3) + A(LX+N33ON2+3))/2.0
      V(LZ) = (DC*D3) + (DD*D1)
  448 J = J + 1
      Y = J - 1
      THETA = Y*PI/XN2
      ST = SIN(THETA)
      TT = TAN(THETA)
      D5 = D4/TT
      D6 = D4/(2.0*ST*H3)
  449 LX = LX + 3
      LY = LY + 3
      LZ = LZ + 3
      V(LX) = 0.0
      IF(K.EQ.1) GO TO 450
      IF(K.EQ.N3) GO TO 451
      DA = A(LX+3) - A(LX-3)
      GO TO 452
  450 DA = A(LX+3) - A(LX-3+N33)
      GO TO 452
```

```
  451 DA = A(LX+3-N33) - A(LX-3)
  452 DB = -(4.0*A(LZ+NALPH3)) + A(LZ+NALPH6)
      V(LY) = (DA*D6) + (DB*D3)
      DC = (4.0*A(LY+NALPH3)) - A(LY+NALPH6)
      IF(J.EQ.2) GO TO 453
      IF(J.EQ.N2) GO TO 454
      DD = (-A(LX+N33) + A(LX-N33))/2.0
      GO TO 455
  453 DD = (-A(LX+N33) + A(NC23-2))/2.0
      GO TO 455
  454 DD = (-A(NCN23-2) + A(LX-N33))/2.0
  455 V(LZ) = (DC*D3) + (DD*D1)
      IF(K.EQ.N3) GO TO 456
      K = K + 1
      GO TO 449
  456 IF(J.EQ.N2) GO TO 457
      K = 1
      GO TO 448
  457 LX = LX + 3
      LY = LY + 3
      LZ = LZ + 3
      V(LX) = 0.0
      DA = -A(LX-N33+3) + A(LX-N33ON2+3) + A(LX-3) - A(LX-N33ON2-3)
      DB = -(4.0*A(LZ+NALPH3)) + A(LZ+NALPH6)
      V(LY) = (DA*D2) + (DB*D3)
      DC = (4.0*A(LY+NALPH3)) - A(LY+NALPH6)
      DD = (-A(LX-N33) + A(LX-N33ON2))/2.0
      V(LZ) = (DC*D3) + (DD*D1)
C         INTERIOR NODES.
 4575 I = I + 1
      J = 1
      K = 1
      R = R + H1
      LX = LX + 3
      LY = LY + 3
      LZ = LZ + 3
      D1 = 1.0/(R*H2)
      D2 = D1/(4.0*H3)
      D3 = 1.0/(2.0*H1)
      D4 = 1.0/R
      NC2 = (NALPHA*(I-1)) + 1
      NC23 = 3*NC2
      NCN2 = NALPHA*I
      NCN23 = 3*NCN2
      DE = A(LZ+3) - A(LZ+N33ON2+3)
      DF = -A(LY+6) + A(LY+N33ON2+6) + A(LY+N33) - A(LY+N33ON2)
      V(LX) = (DE*D1) + (DF*D2)
      DG = -A(LZ)
      DH = -A(LZ+NALPH3) + A(LZ-NALPH3)
      DI = A(LX+6) - A(LX+N33ON2+6) - A(LX+N33) + A(LX+N33ON2)
      V(LY) = (DG*D4) + (DH*D3) + (DI*D2)
      DJ = A(LY)
      DK = A(LY+NALPH3) - A(LY-NALPH3)
      DL = (-A(LX+3) + A(LX+N33ON2+3))/2.0
      V(LZ) = (DJ*D4) + (DK*D3) + (DL*D1)
  458 J = J + 1
      Y = J - 1
      THETA = Y*PI/XN2
      ST = SIN(THETA)
      TT = TAN(THETA)
      D5 = D4/TT
      D6 = D4/(2.0*ST*H3)
  459 Z = K - 1
      PHI = PI*Z*2.0/XN3
      CP = COS(PHI)
      SP = SIN(PHI)
      LX = LX + 3
      LY = LY + 3
      LZ = LZ + 3
      DE = A(LZ)
      DF = A(LY)
      DG = -A(LZ+NALPH3) + A(LZ-NALPH3)
      DH = A(LY+NALPH3) - A(LY-NALPH3)
      IF(K.EQ.1) GO TO 460
      IF(K.EQ.N3) GO TO 461
      DI = -A(LY+3) + A(LY-3)
      DJ = A(LX+3) - A(LX-3)
      GO TO 462
  460 DI = -A(LY+3) + A(LY-3+N33)
      DJ = A(LX+3) - A(LY-3+N33)
      GO TO 462
  461 DI = -A(LY+3-N33) + A(LY-3)
      DJ = A(LX+3-N33) - A(LY-3)
```

```
  462 IF(J.EQ.2) GO TO 463
      IF(J.EQ.N2) GO TO 464
      DK = (A(LZ+N33) - A(LZ-N33))/2.0
      DL = (-A(LX+N33) + A(LX-N33))/2.0
      GO TO 465
  463 DK = (A(LZ+N33) + (A(NC23-1)*SP) - (A(NC23)*CP))/2.0
      DL = (-A(LX+N33) + A(NC23-2))/2.0
      GO TO 465
  464 DK = ((-A(NCN23-1)*SP) + (A(NCN23)*CP) - A(LZ-N33))/2.0
      DL = (-A(NCN23-2) + A(LX-N33))/2.0
  465 V(LX) = (DE*D5) + (DK*D1) + (DI*D6)
      V(LY) = -(DE*D4) + (DG*D3) + (DJ*D6)
      V(LZ) = (DF*D4) + (DH*D3) + (DL*D1)
      IF(K.EQ.N3) GO TO 466
      K = K + 1
      GO TO 459
  466 IF(J.EQ.N2) GO TO 467
      K = 1
      GO TO 458
  467 LX = LX + 3
      LY = LY + 3
      LZ = LZ + 3
      DE = A(LZ-N33) - A(LZ-N33ON2)
      DF = A(LY-N33+3) - A(LY-N33ON2+3) - A(LY-3) + A(LY-N33ON2-3)
      V(LX) = (DE*D1) + (DF*D2)
      DG = -A(LZ)
      DH = -A(LZ+NALPH3) + A(LZ-NALPH3)
      DI = -A(LY-N33+3) + A(LX-N33ON2+3) + A(LX-3) - A(LX-N33ON2-3)
      V(LY) = (DG*D4) + (DH*D3) + (DI*D2)
      DJ = A(LY)
      DK = A(LY+NALPH3) - A(LY-NALPH3)
      DL = (-A(LX-N33) + A(LX-N33ON2))/2.0
      V(LZ) = (DJ*D4) + (DK*D3) + (DL*D1)
      IF(I.EQ.N1) GO TO 468
      GO TO 4575
C        OUTER BOUNDARY NODES.
  468 I = N1 + 1
      J = 1
      K = 1
      R = RO
      D1 = 1.0/(R*H2)
      D2 = D1/(4.0*H3)
      D3 = 1.0/(2.0*H1)
      D4 = 1.0/2
      NC2 = (NALPHA*(I-1)) + 1
      NC23 = 3*NC2
      NCN2 = NALPHA*I
      NCN23 = 3*NCN2
      LX = LX + 3
      LY = LY + 3
      LZ = LZ + 3
      V(LX) = 0.0
      DM = A(LX+6) - A(LX+N33ON2+6) - A(LX+N33) + A(LX+N33ON2)
      DN = (4.0*A(LZ-NALPH3)) - A(LZ-NALPH6)
      V(LY) = (DM*D2) + (DN*D3)
      DP = -(4.0*A(LY-NALPH3)) + A(LY-NALPH6)
      DQ = (-A(LX+3) + A(LX+N33ON2+3))/2.0
      V(LZ) = (DP*D3) + (DQ*D1)
  469 J = J + 1
      Y = J - 1
      THETA = Y*PI/XN2
      ST = SIN(THETA)
      TT = TAN(THETA)
      D5 = D4/TT
      D6 = D4/(2.0*ST*H3)
  470 LX = LX + 3
      LY = LY + 3
      LZ = LZ + 3
      V(LY) = 0.0
      IF(K.EQ.1) GO TO 471
      IF(K.EQ.N3) GO TO 472
      DA = A(LX+3) - A(LX-3)
      GO TO 473
  471 DA = A(LX+3) - A(LX-3+N33)
      GO TO 473
  472 DA = A(LX+3-N33) - A(LX-3)
  473 DB = (4.0*A(LZ-NALPH3)) - A(LZ-NALPH6)
      V(LY) = (DA*D6) + (DB*D3)
      DC = -(4.0*A(LY-NALPH3)) + A(LY-NALPH6)
      IF(J.EQ.2) GO TO 474
      IF(J.EQ.N2) GO TO 475
      DD = (-A(LX+N33) + A(LX-N33))/2.0
      GO TO 476
```

```
474 DD = (-A(LX+N33) + A(NC23-2))/2.0
    GO TO 476
475 DD = (-A(JCN23-2) + A(LY-N33))/2.0
476 V(LZ) = (DC*D3) + (DD*D1)
    IF(K.EQ.N3) GO TO 477
    K = K + 1
    GO TO 470
477 IF(J.EQ.N2) GO TO 478
    K = 1
    GO TO 469
478 LX = LX + 3
    LY = LY + 3
    LZ = LZ + 3
    V(LX) = 0.0
    DA = -A(LX-N33+3) + A(LX-N33ON2+3) + A(LY-3) - A(LX-N33ON2-3)
    DB = (4.0*A(LZ-NALPH3)) - A(LZ-NALPH6)
    V(LY) = (DA*D2) + (DB*D3)
    DC = -(4.0*A(LY-NALPH3)) + A(LY-NALPH6)
    DD = (-A(LX-N33) + A(LX-N33ON2))/2.0
    V(LZ) = (DC*D3) + (DD*D1)
    RETURN
    END




    SUBROUTINE START1(V,VORTEX,U,NODE3,NN1,NN2,NN3,RRI,RRO)
    DIMENSION V(NODE3), VORTEX(NODE3), U(NODE3)
C
C
C
C      THIS SUBROUTINE SETS UP THE INITIAL CONDITIONS ON V,
C      THE ROTATIONAL COMPONENT OF THE VELOCITY FIELD, AND
C      VORTEX, THE VORTICITY FIELD, CORRESPONDING TO UNIFORM
C      STOKES FLOW PAST A SPHERE.
C
C
C
    N1 = NN1
    N2 = NN2
    N3 = NN3
    RI = RRI
    RO = RRO
    NODES = NODE3/3
    XN1 = N1
    XN2 = N2
    H1 = (RO - RI)/XN1
    PI = 3.1415926535898
    H2 = PI/XN2
    LX = 1
    LY = 2
    LZ = 3
    MX = 1
    MY = 1 + NODES
    DO 552 I=1,N1+1
    XI = I - 1
    R = RI + (XI*H1)
    RA = RI/R
    RB = RA*RA*RA
    THETA = 0.0
C      THETA = 0 POLAR NODE.
    V(LX) = (1.0 - (1.5*RA) + (0.5*RB)) - U(MX)
    V(LY) = 0.0
    V(LZ) = 0.0
    VORTEX(LX) = 0.0
    VORTEX(LY) = 0.0
    VORTEX(LZ) = 0.0
    LX = LX + 3
    LY = LY + 3
    LZ = LZ + 3
    MX = MX + 1
    MY = MY + 1
C      NON-POLAR NODES.
    DO 551 J=2,N2
    THETA = THETA + H2
    ST = SIN(THETA)
```

```
      CT = COS(THETA)
      DO 550 K=1,N3
      V(LX) = ((1.0 - (1.5*RA) + (0.5*RB))*CT) - U(MX)
      V(LY) = -((1.0 - (0.75*RA) - (0.25*RB))*ST) - U(MY)
      V(LZ) = 0.0
      VORTEX(LX) = 0.0
      VORTEX(LY) = 0.0
      VORTEX(LZ) = -1.5*RA*ST/R
      LX = LX + 3
      LY = LY + 3
      LZ = LZ + 3
      MX = MX + 1
      MY = MY + 1
  550 CONTINUE
  551 CONTINUE
C         THETA = PI POLAR NODE.
      V(LX) = -(1.0 - (1.5*RA) + (0.5*RB)) - U(MX)
      V(LY) = 0.0
      V(LZ) = 0.0
      VORTEX(LX) = 0.0
      VORTEX(LY) = 0.0
      VORTEX(LZ) = 0.0
      LX = LX + 3
      LY = LY + 3
      LZ = LZ + 3
      MX = MX + 1
      MY = MY + 1
  552 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE STARI2(V,VORTEX,U,NODE3,NN1,NN2,NN3,RRI,RRO,SSHEAR,UU0
      DIMENSION V(NODE3), VORTEX(NODE3), U(NODE3)
C
C
C
C        THIS SUBROUTINE SETS UP THE INITIAL CONDITIONS ON V,
C        THE ROTATIONAL COMPONENT OF THE VELOCITY FIELD, AND
C        VORTEX, THE VORTICITY FIELD, CORRESPONDING TO A LINEAR
C        VARIATION BETWEEN THE BOUNDARY CONDITIONS ON THE
C        SPHERE (R=RI) AND THE BOUNDARY CONDITIONS ON THE
C        OUTER BOUNDARY (R=RO).
C
C
C
C
      N1 = NN1
      N2 = NN2
      N3 = NN3
      RI = RRI
      RO = RRO
      SHEAR = SSHEAR
      U0 = UU0
      NODES = NODE3/3
      NALPHA = NODES/(N1+1)
      NALPH3 = NALPHA*3
      XN1 = N1
      XN2 = N2
      XN3 = N3
      DELTAR = RO - RI
      H1 = DELTAR/XN1
      PI = 3.14159265358998
      H2 = PI/XN2
      H3 = 2.0*PI/XN3
      LX = 1
      LY = 2
      LZ = 3
      MX = 1
      MY = 1 + NODES
      MZ = 1 + NODES + NODES
      DO 562 I=1,N1+1
      XI = I - 1
      R = RI + (XI*H1)
```

```
      RA = (R - RI)/DELTAR
      RB = RA*SHEAR*RO/R
      THETA = 0.0
C        THETA = 0 POLAR MODE.
      V(LX) = (RA*U0) - U(MX)
      V(LY) = -U(MY)
      V(LZ) = -U(MZ)
      VORTEX(LX) = 0.0
      VORTEX(LY) = 0.0
      VORTEX(LZ) = -RB
      LX = LX + 3
      LY = LY + 3
      LZ = LZ + 3
      MX = MX + 1
      MY = MY + 1
      MZ = MZ + 1
C        NON-POLAR MODES
      DO 561 J=2,N2
      THETA = THETA + H2
      ST = SIN(THETA)
      CT = COS(THETA)
      PHI = 0.0
      DO 560 K=1,N3
      SP = SIN(PHI)
      CP = COS(PHI)
      RC = U0 + (SHEAR*RO*ST*CP)
      V(LX) = (RA*RC*CT) - U(MX)
      V(LY) = -(RA*RC*ST) - U(MY)
      V(LZ) = -U(MZ)
      VORTEX(LX) = -RB*ST*SP
      VORTEX(LY) = -RB*CT*SP
      VORTEX(LZ) = -(RB*CT*CT*CP) - (RC*ST/DELTAR)
      LX = LX + 3
      LY = LY + 3
      LZ = LZ + 3
      MX = MX + 1
      MY = MY + 1
      MZ = MZ + 1
      PHI = PHI + H3
  560 CONTINUE
  561 CONTINUE
C        THETA = PI POLAR MODE.
      V(LX) = -(RA*U0) - U(MX)
      V(LY) = -U(MY)
      V(LZ) = -U(MZ)
      VORTEX(LX) = 0.0
      VORTEX(LY) = 0.0
      VORTEX(LZ) = -RB
      LX = LX + 3
      LY = LY + 3
      LZ = LZ + 3
      MX = MX + 1
      MY = MY + 1
      MZ = MZ + 1
  562 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TYPIST(P,NNALPH,CDP,CDV,CLP,CLV,CM,NSTEP,TIME,IOPT4,
     1ERRMAX)
      DIMENSION P(NNALPH)
C
C
C
C         THIS SUBROUTINE WRITES OUT THE FORM, FRICTION AND
C         TOTAL DRAG AND LIFT COEFFICIENTS (CDP, CDV, CD
C         AND CLP, CLV, CL, RESPECTIVELY), AND THE MOMENT
C         COEFFICIENT (CM). THE PRESSURE FIELD (P) ON THE
C         SURFACE OF THE SPHERE (R=RI) IS WRITTEN OUT ALSO,
C         UNLESS THE OUTPUT SELECTION PARAMETER IOPT4 = 0.
C
C
C
      NALPHA = NNALPH
      CD = CDP + CDV
      CL = CLP + CLV
      WRITE(6,980)
  980 FORMAT(1H1/////)
      WRITE(6,981) NSTEP,TIME
  981 FORMAT(1X,16HTIME-STEP NUMBER,I6,16X,21H(DIMENSIONLESS TIME =,
     1F9.4,1H))
      WRITE(6,982) CDP,CDV,CD
  982 FORMAT(1H0,17HDRAG COEFFICIENTS/1X,10X,
     127HFORM DRAG COEFFICIENT      =,F7.3/1X,10X,
     227HFRICTION DRAG COEFFICIENT  =,F7.3/1X,10X,
     327HTOTAL DRAG COEFFICIENT     =,F7.3)
      WRITE(6,983) CLP,CLV,CL
  983 FORMAT(1H0,17HLIFT COEFFICIENTS/1X,10X,
     127HFORM LIFT COEFFICIENT      =,F7.3/1X,10X,
     227HFRICTION LIFT COEFFICIENT  =,F7.3/1X,10X,
     327HTOTAL LIFT COEFFICIENT     =,F7.3)
      WRITE(6,9835) CM
 9835 FORMAT(1H0,20HMOMENT COEFFICIENT =,F7.3)
      IF(IOPT4.EQ.0) GO TO 987
      WRITE(6,984)
  984 FORMAT(1H0///1X,10X,35HPRESSURE FIELD ON SURFACE OF SPHERE/)
      WRITE(6,985) (K,P(K),K=1,NALPHA)
  985 FORMAT((1X,3(I4,4H ---,F9.5,9X,)1X))
      WRITE(6,986) ERRMAX
  986 FORMAT(1H0,10X,43HDIFFERENCE BETWEEN THE MAXIMUM AND MINIMUM ,
     112HPRESSURES AT/1X,10X,37HTHE THETA = PI POLAR NODE CALCULATED ,
     217HALONG EACH OF THE/1X,10X,32HCONSTANT-PHI INTEGRATION PATHS =,
     3F9.5)
  987 RETURN
      END




      SUBROUTINE VELOX(H,B,NNODES,NNEQNS,U,UU0,NN1,NN2,NN3,RRI,RRO,
     1NNODE2,NNODE3)
      DIMENSION H(NNODES),B(NNEQNS),U(NNODE3)
C
C
C
C         THIS SUBROUTINE CONVERTS THE SCALAR POTENTIAL FIELD
C         H DETERMINED BY SUBROUTINE SOR INTO THE THREE-
C         DIMENSIONAL TRANSLATIONAL VELOCITY FIELD U.
C
C
C
      N1 = NN1
      N2 = NN2
      N3 = NN3
      N3OV2 = N3/2
      RI = RRI
      RO = RRO
      U0 = UU0
      NODES = NNODES
      NODE2 = NNODE2
      NODE3 = NNODE3
      PI = 3.1415926535898
```

```
      XN1 = N1
      XN2 = N2
      XN3 = N3
      H1 = (RO - RI)/XN1
      H2 = PI/XN2
      H3 = 2.0*PI/XN3
      NALPHA = (N3*(N2-1)) + 2
C        INNER BOUNDARY NODES.
      U(1) = 0.0
      T = H(2) - H(N3ON2+2)
      U(1+NODES) = T/(2.0*RI*H2)
      U(1+NODE2) = 0.0
      J=2
      THETA = PI/XN2
      S = SIN(THETA)
      K=1
      M=2
  499 U(M) = 0.0
      T = H(N+N3) - H(1)
      U(N+NODES) = T/(2.0*RI*H2)
      IF(K.EQ.1) GO TO 500
      IF(K.EQ.N3) GO TO 501
      T = H(N+1) - H(M-1)
      GO TO 502
  500 T = H(N+1) - H(M-1+N3)
      GO TO 502
  501 T = H(N+1-N3) - H(M-1)
  502 U(M+NODE2) = T/(2.0*RI*H3*S)
      K = K+1
      M = M+1
      IF(K.LT.N3+1) GO TO 499
      J = 3
      THETA = (2.0*PI)/XN2
      S = SIN(THETA)
      K = 1
  503 U(N) = 0.0
      T = H(N+N3) - H(M-N3)
      U(N+NODES) = T/(2.0*RI*H2)
      IF(K.EQ.1) GO TO 504
      IF(K.EQ.N3) GO TO 505
      T = H(N+1) - H(M-1)
      GO TO 506
  504 T = H(N+1) - H(N-1+N3)
      GO TO 506
  505 T = H(N+1-N3) - H(M-1)
  506 U(N+NODE2) = T/(2.0*RI*H3*S)
      K = K+1
      M = M+1
      IF(K.LT.N3+1) GO TO 503
      K = 1
      J = J + 1
      Y = J - 1
      THETA = (Y*PI)/XN2
      S = SIN(THETA)
      IF(J.LT.N2) GO TO 503
  507 U(M) = 0.0
      T = H(NALPHA) - H(N-N3)
      U(N+NODES) = T/(2.0*RI*H2)
      IF(K.EQ.1) GO TO 508
      IF(K.EQ.N3) GO TO 509
      T = H(N+1) - H(M-1)
      GO TO 510
  508 T = H(N+1) - H(M-1+N3)
      GO TO 510
  509 T = H(N+1-N3) - H(M-1)
  510 U(N+NODE2) = T/(2.0*RI*H3*S)
      K = K+1
      M = M+1
      IF(K.LT.N3+1) GO TO 507
      U(M) = 0.0
      T = H(M-N3) - H(M-N3ON2)
      U(N+NODES) = T/(2.0* I*H2)
      U(N+NODE2) = 0.0
      I = 2
      R = RI + H1
      J = 1
C        INTERIOR NODES (THETA = 0 POLAR AXIS).
      K = 1
      M = M+1
  511 T = H(N+NALPHA) - H(M-NALPHA)
      U(M) = T/(2.0*H1)
      T = H(N+1) - H(N+N3ON2+1)
```

```
      U(N+NODES) = T/(2.0*R*H2)
      U(N+NODE2) = 0.0
      J = 2
C        INTERIOR NODES (ADJACENT TO THE THETA = 0 POLAR AXIS).
      THETA = PI/XN2
      S = SIN(THETA)
      N = N+1
  512 T = H(N+NALPHA) - H(N-NALPHA)
      U(N) = T/(2.0*H1)
      NQ = (NALPHA*(I-1)) + 1
      T = H(N+N3) - H(NQ)
      U(N+NODES) = T/(2.0*R*H2)
      IF(K.EQ.1) GO TO 513
      IF(K.EQ.N3) GO TO 514
      T = H(N+1) - H(N-1)
      GO TO 515
  513 T = H(N+1) - H(N-1+N3)
      GO TO 515
  514 T = H(N+1-N3) - H(N-1)
  515 U(N+NODE2) = T/(2.0*R*H3*S)
      K = K+1
      N = N+1
      IF(K.LT.N3+1) GO TO 512
      J = 3
C        INTERIOR NODES (GENERAL CASE).
      THETA = (2.0*PI)/XN2
      S = SIN(THETA)
      K = 1
  516 T = H(N+NALPHA) - H(N-NALPHA)
      U(N) = T/(2.0*H1)
      T = H(N+N3) - H(N-N3)
      U(N+NODES) = T/(2.0*R*H2)
      IF(K.EQ.1) GO TO 517
      IF(K.EQ.N3) GO TO 518
      T = H(N+1) - H(N-1)
      GO TO 519
  517 T = H(N+1) - H(N-1+N3)
      GO TO 519
  518 T = H(N+1-N3) - H(N-1)
  519 U(N+NODE2) = T/(2.0*R*H3*S)
      K = K+1
      N = N+1
      IF(K.LT.N3+1) GO TO 516
      J = J+1
      Y = J-1
      THETA = (Y*PI)/XN2
      S = SIN(THETA)
      K = 1
      IF(J.LT.N2) GO TO 516
C        INTERIOR NODES (ADJACENT TO THE THETA = PI POLAR AXIS).
  520 T = H(N+NALPHA) - H(N-NALPHA)
      U(N) = T/(2.0*H1)
      NQ = NALPHA*I
      T = H(NQ) -H(N-N3)
      U(N+NODES) = T/(2.0*R*H2)
      IF(K.EQ.1) GO TO 521
      IF(K.EQ.N3) GO TO 522
      T = H(N+1) - H(N-1)
      GO TO 523
  521 T = H(N+1) - H(N-1+N3)
      GO TO 523
  522 T = H(N+1-N3) - H(N-1)
  523 U(N+NODE2) = T/(2.0*R*H3*S)
      K = K+1
      N = N+1
      IF(K.LT.N3+1) GO TO 520
C        INTERIOR NODES (THETA = PI POLAR AXIS).
      T = H(N+NALPHA) - H(N-NALPHA)
      U(N) = T/(2.0*H1)
      T = H(N-N3) - H(N-N3ON2)
      U(N+NODES) =  T/(2.0*R*H2)
      U(N+NODE2) = 0.0
      I = I+1
      Y = I-1
      R = (Y*H1) + RI
      K = 1
      N = N+1
      IF(I.LT.N1+1) GO TO 511
C        OUTER BOUNDARY NODES.
      U(N) = U0
      T = H(N+1) - H(N+N3ON2+1)
      U(N+NODES) = T/(2.0*R*H2)
```

```
      U(N+NODE2) = 0.0
      J = 2
      THETA = PI/XN2
      S = SIN(THETA)
      N = N+1
      N0 = (NALPHA*N1) + 1
  524 U(N) = (3.0*B(N))/(2.0*H1)
      T = H(N+N3) - H(N0)
      U(N+NODES) = T/(2.0*R*H2)
      IF(K.EQ.1) GO TO 525
      IF(K.EQ.N3) GO TO 526
      T = H(N+1) - H(N-1)
      GO TO 527
  525 T = H(N+1) - H(N-1+N3)
      GO TO 527
  526 T = H(N+1-N3) - H(N-1)
  527 U(N+NODE2) = T/(2.0*R*H3*S)
      K = K+1
      N = N+1
      IF(K.LT.N3+1) GO TO 524
      J = 3
      THETA = (2.0*PI)/XN2
      S = SIN(THETA)
      K = 1
  528 U(N) = (3.0*B(N))/(2.0*H1)
      T = H(N+N3) - H(N-N3)
      U(N+NODES) = T/(2.0*R*H2)
      IF(K.EQ.1) GO TO 529
      IF(K.EQ.N3) GO TO 530
      T = H(N+1) - H(N-1)
      GO TO 531
  529 T = H(N+1) - H(N-1+N3)
      GO TO 531
  530 T = H(N+1-N3) - H(N-1)
  531 U(N+NODE2) = T/(2.0*R*H3*S)
      K = K+1
      N = N+1
      IF(K.LT.N3+1) GO TO 528
      J = J+1
      Y = J-1
      THETA = (Y*PI)/XN2
      S = SIN(THETA)
      K = 1
      IF(J.LT.N2) GO TO 528
  532 U(N) = (3.0*B(N))/(2.0*H1)
      T = H(NODES) -H(N-N3)
      U(N+NODES) = T/(2.0*R*H2)
      IF(K.EQ.1) GO TO 533
      IF(K.EQ.N3) GO TO 534
      T = H(N+1) - H(N-1)
      GO TO 535
  533 T = H(N+1) - H(N-1+N3)
      GO TO 535
  534 T = H(N+1-N3) - H(N-1)
  535 U(N+NODE2) = T/(2.0*R*H3*S)
      K = K+1
      N = N+1
      IF(K.LT.N3+1) GO TO 532
      U(N) = -U0
      T = H(N-N3) - H(N-N30N2)
      U(N+NODES) =  T/(2.0*R*H2)
      U(NODE3) = 0.0
      RETURN
      END


      SUBROUTINE VORBC1(VORTEX,NNODE3,SHEAR,NN2,NN3,NNALPH)
      DIMENSION VORTEX(NNODE3)
C
C
C
C         THIS SUBROUTINE SETS THE BOUNDARY CONDITIONS ON
C         VORTICITY WHICH ARE INDEPENDENT OF TIME.
C
C
C
```

```
      NALPHA = NNALPH
      NODE3 = NNODE3
      N2 = NN2
      N3 = NN3
      SHEAR = SSHEAR
      XN2 = N2
      XN3 = N3
      PI = 3.1415926535898
C        INNER BOUNDARY CONDITIONS ON RADIAL VORTICITY COMPONENTS.
      DO 600 ND0=1,NALPHA
      ND1 = (3*ND0) - 2
      VORTEX(ND1) = 0.0
  600 CONTINUE
C        OUTER BOUNDARY CONDITIONS ON ALL VORTICITY COMPONENTS
C        (THETA = 0 POLAR AXIS).
      ND2 = NODE3 -(3*NALPHA)
      VORTEX(ND2+1) = 0.0
      VORTEX(ND2+2) = 0.0
      VORTEX(ND2+3) = -SHEAR
C        OUTER BOUNDARY CONDITIONS ON ALL VORTICITY COMPONENTS
C        (GENERAL CASE).
      LX = 4
      LY = 5
      LZ = 6
      J = 2
  601 K = 1
      Y = J - 1
      THETA = (Y*PI)/XN2
      ST = SIN(THETA)
      CT = COS(THETA)
      PHI = 0.0
      SP = 0.0
      CP = 1.0
  602 VORTEX(ND2+LX) = -SHEAR*ST*SP
      VORTEX(ND2+LY) = -SHEAR*CT*SP
      VORTEX(ND2+LZ) = -SHEAR*CP
      LX = LX + 3
      LY = LY + 3
      LZ = LZ + 3
      IF(K.EQ.N3) GO TO 603
      K = K + 1
      Z = K - 1
      PHI = (Z*2.0*PI)/XN3
      SP = SIN(PHI)
      CP = COS(PHI)
      GO TO 602
  603 IF(J.EQ.N2) GO TO 604
      J = J + 1
      GO TO 601
C        OUTER BOUNDARY CONDITIONS ON ALL VORTICITY COMPONENTS
C        (THETA = PI POLAR AXIS).
  604 VORTEX(ND2+LX) = 0.0
      VORTEX(ND2+LY) = 0.0
      VORTEX(ND2+LZ) = -SHEAR
      RETURN
      END
```

```
      SUBROUTINE VORBC2(VORTEX,NODE3,NNALPH,NN1,RRI,RRO,V,U)
      DIMENSION VORTEX(NODE3), U(NODE3), V(NODE3)
C
C
C
C        THIS SUBROUTINE SETS THE BOUNDARY CONDITIONS ON
C        VORTICITY WHICH ARE FUNCTIONS OF TIME.
C
C
C
      NALPHA = NNALPH
      NALPH3 = 3*NALPHA
      NALPH6 = 6*NALPHA
      N1 = NN1
      NODES = NALPHA*(N1 + 1)
      RI = RRI
      RO = RRO
```

```
      XM1 = N1
      H1 = (RO-RI)/XM1
      CA = 1.0/(2.0*H1)
      CB = 1.0/RI
      DO 606 I=1,NALPHA
      IA = I*3
      IB = IA - 1
      IC = I + NODES
      ID = IC + NODES
      DA = CB*U(ID)
      DB = -3.0*U(ID)
      DC = -4.0*V(IA+NALPH3)
      DD = V(IA+NALPH6)
      DE = -CB*U(IC)
      DF = 3.0*U(IC)
      DG = 4.0*V(IB+NALPH3)
      DH = -V(IB+NALPH6)
      EA = (DB + DC + DD)*CA
      EF = (DF + DG + DH)*CA
      VORTEX(IB) = EA + DA
      VORTEX(IA) = EF + DE
  606 CONTINUE
      RETURN
      END


      SUBROUTINE WRITER(VORTEX,A,V,NNODE3)
      DIMENSION VORTEX(NNODE3), A(NNODE3), V(NNODE3)
      COMMON/COMWRI/IOPT1,IOPT2,IOPT3,M1,DIFFJM,REST1,M,DIFFY,RESTY,
     1DIFFCE,TIME,NSTEP,NODE3
C
C
C
C         THIS SUBROUTINE WRITES OUT THE PRINCIPAL RESULTS
C         OBTAINED BY PROGRAM UROT AT EACH TIME-STEP. THE
C         OUTPUT SELECTION PARAMETERS IOPT1-3 FUNCTION AS
C         FOLLOWS - THE VORTICITY FIELD IS WRITTEN OUT UNLESS
C                      IOPT1 = 0.
C                 - THE VECTOR POTENTIAL FIELD IS WRITTEN OUT
C                      UNLESS IOPT2 = 0.
C                 - THE ROTATIONAL COMPONENT OF THE VELOCITY
C                      FIELD IS WRITTEN OUT UNLESS IOPT3 = 0.
C
C
C
      WRITE(6,960)
  960 FORMAT(1H1/////)
      WRITE(6,961)NSTEP,TIME
  961 FORMAT(1X,16HTIME-STEP NUMBER,I6,35X,21H(DIMENSIONLESS TIME =,
     1F9.4,1H))
      WRITE(6,962)M1,DIFFJM,REST1
  962 FORMAT(1H0,28HVORTICITY TRANSPORT EQUATION/1X,10X,
     138HNUMBER OF ITERATIONS FOR CONVERGENCE =,I4/1X,10X,
     257HMAXIMUM SQUARE DIFFERENCE BETWEEN ELEMENTS IN SUCCESSIVE ,
     327HITERATIONS AT CONVERGENCE =,F15.13/1X,10X,
     440HMAXIMUM SQUARE RESIDUAL AT CONVERGENCE =,F15.13)
      WRITE(6,963)M,DIFFY,RESTY,DIFFCE
  963 FORMAT(1H0,33HVECTOR POTENTIAL POISSON EQUATION/1X,10X,
     138HNUMBER OF ITERATIONS FOR CONVERGENCE =,I4/1X,10X,
     257HMAXIMUM SQUARE DIFFERENCE BETWEEN ELEMENTS IN SUCCESSIVE ,
     327HITERATIONS AT CONVERGENCE =,F15.13/1X,10X,
     440HMAXIMUM SQUARE RESIDUAL AT CONVERGENCE =,F15.13/1X,10X,
     557HMAXIMUM SQUARE DIFFERENCE BETWEEN SUCCESSIVE DIFFERENCES ,
     641HIN SUCCESSIVE ITERATIONS AT CONVERGENCE =,F15.13)
      NODES = NODE3/3
      IF(IOPT1.EQ.0) GO TO 966
      WRITE(6,964)
  964 FORMAT(1H0////1X,10X,32HVORTICITY FIELD (WR,WTHETA,WPHI)/)
      WRITE(6,965) (K,VORTEX((3*K)-2),VORTEX((3*K)-1),VORTEX(3*K),
     1K=1,NODES)
  965 FORMAT((1X,2(I4,4H ---,F9.5,6X,F9.5,6X,F9.5,15X,)1X))
  966 IF(IOPT2.EQ.0) GO TO 969
      WRITE(6,967)
  967 FORMAT(1H0////1X,10X,39HVECTOR POTENTIAL FIELD (AR,ATHETA,APHI)/)
      WRITE(6,968) (K,A((3*K)-2),A((3*K)-1),A(3*K),K=1,NODES)
  968 FORMAT((1X,2(I4,4H ---,F9.5,6X,F9.5,6X,F9.5,15X,)1X))
  969 IF(IOPT3.EQ.0) GO TO 972
      WRITE(6,970)
  970 FORMAT(1H0////1X,10X,31HVELOCITY FIELD (VR,VTHETA,VPHI)/)
      WRITE(6,971) (K,V((3*K)-2),V((3*K)-1),V(3*K),K=1,NODES)
  971 FORMAT((1X,2(I4,4H ---,F9.5,6X,F9.5,6X,F9.5,15X,)1X))
  972 RETURN
      END
```

Appendix 5 - Flow Fields

In this appendix, we present numerical values of various fields defining uniform plus centred linear-shear flow past a sphere at unit Reynolds number, for the purpose of comparison with results that may be obtained elsewhere. The fields given are the scalar potential h, the surface pressure p, the vector potential $\underline{A}$ and the vorticity $\underline{w}$. The time-dependent fields are given at effective steady-state, which occurs after about forty time-steps of length $\Delta t = 0.002$. The radius of the outer envelope $r^* = 1.25$. The nodes in the discretised flow field are numbered as follows. On the surface of the sphere, the $\theta = 0$ node is node 1. The $\theta = \pi/4$ and $\phi = 0, \pi/2, \pi$, and $3\pi/2$ nodes are, respectively, nodes 2 to 5. The $\theta = \pi/2$, $\phi = 0, \pi/2, \pi$, and $3\pi/2$ nodes are, respectively, nodes 6 to 9, and so on, so that the $\theta = \pi$ node is node 14. Then, at a distance Δr from the surface, the $\theta = 0$ node is node 15, and so on.

(i) scalar potential field h:-

node 1 ---	2.43100	node 2 ---	2.32911	node 3 ---	2.15141
7 ---	1.45966	8 ---	1.45967	9 ---	1.45969
13 ---	.76800	14 ---	.48861	15 ---	2.49559
19 ---	2.19722	20 ---	1.45970	21 ---	1.45972
25 ---	.72231	26 ---	.93442	27 ---	.72234
31 ---	2.33432	32 ---	2.01915	33 ---	2.33442
37 ---	1.45984	38 ---	.26996	39 ---	.58522
43 ---	2.92042	44 ---	2.95179	45 ---	2.49801
49 ---	1.45981	50 ---	1.45982	51 ---	1.45984
55 ---	.42171	56 ---	0		
4 ---	1.97371	5 ---	2.15144	6 ---	1.45964
10 ---	.59025	11 ---	.76798	12 ---	.94570
16 ---	2.40928	17 ---	2.19719	18 ---	1.98511
22 ---	1.45973	23 ---	1.45975	24 ---	.51020
28 ---	.42420	29 ---	2.68918	30 ---	2.64963
34 ---	1.45975	35 ---	1.45977	36 ---	1.45978
40 ---	.90048	41 ---	.58524	42 ---	.23089
46 ---	2.04422	47 ---	2.49803	48 ---	1.45980
52 ---	-.03212	53 ---	.42169	54 ---	.87550

(ii) surface pressure field p:-

node 1 ---	0	node 2 ---	-1.13015	node 3 ---	.74441
4 ---	2.73339	5 ---	.74820	6 ---	.07868
7 ---	2.71385	8 ---	5.57634	9 ---	2.72071
10 ---	1.43727	11 ---	4.92942	12 ---	8.53833
13 ---	4.93894	14 ---	5.92511		

(iii) vector potential field $\underline{A}$:-

node		A_r	A_θ	A_ϕ
1	---	-.15254	0	0
3	---	-1.24459	0	0
5	---	.94613	0	0
7	---	-1.77916	0	0
9	---	1.40956	0	0
11	---	-1.06055	0	0
13	---	.67049	0	0
15	---	-.04099	.00050	-.09049
17	---	-.38549	-.10913	-.06262
19	---	.30436	.13433	-.05214
21	---	-.54254	.01008	-.08555
23	---	.45306	-.02141	-.06432
25	---	-.54034	.10833	-.05722
27	---	.24430	-.11308	-.05041
29	---	-.01117	.00027	-.06283
31	---	-.29053	-.06456	-.03855
33	---	.27391	.06223	-.03623
35	---	-.39035	.00671	-.05120
37	---	.37283	-.01115	-.05086
39	---	-.31141	.06490	-.03324
41	---	.20839	-.00743	-.03280
43	---	-.00397	0	0
45	---	-.21100	0	0
47	---	.23797	0	0
49	---	-.20865	0	0
51	---	.27022	0	0
53	---	-.23753	0	0
55	---	.25033	0	0
2	---	-.14991	0	0
4	---	-.15422	0	0
6	---	-.16065	0	0
8	---	-.17354	0	0
10	---	-.18537	0	0
12	---	-.19166	0	0
14	---	-.16814	0	0
16	---	-.04042	-.00132	-.18817
18	---	-.04135	-.00196	.06593
20	---	-.04405	-.00321	-.23382
22	---	-.04556	-.00338	.06335
24	---	-.04812	-.00198	-.15757
26	---	-.04951	-.00142	.04154
28	---	-.04974	.00085	-.01913
30	---	-.01155	-.00072	-.11829
32	---	-.01105	-.00104	.04378
34	---	-.00944	-.00176	-.14486
36	---	-.00857	-.00186	.04281
38	---	-.00694	-.00109	-.13000
40	---	-.00621	-.00078	.03173
42	---	-.00665	.00047	-.03359
44	---	-.00153	0	0
46	---	-.00075	0	0
48	---	.00105	0	0
50	---	.00297	0	0
52	---	.00536	0	0
54	---	.00549	0	0
56	---	.00582	0	0

(iv) vorticity field $\underline{\omega}$:-

node	ω_r	ω_θ	ω_ϕ
1 ---	0	-.00413	1.51722
3 ---	0	.06625	-3.49389
5 ---	0	-.02632	-3.49065
7 ---	0	.01695	-5.17896
9 ---	0	.02863	-5.18175
11 ---	0	-.91744	-3.04707
13 ---	0	.97170	-3.35060
15 ---	.00037	-.00140	-.18712
17 ---	-.69010	-.05114	-1.41395
19 ---	.69939	.00708	-1.41439
21 ---	-.95152	-.05657	-1.95865
23 ---	.95150	.07305	-1.93852
25 ---	-.55796	.00045	-1.33035
27 ---	.54655	-.04511	-1.33152
29 ---	.00010	-.00040	-.56754
31 ---	-.00170	-.47568	-.50650
33 ---	.00421	.47933	-.50665
35 ---	-.91452	-.02927	-.71088
37 ---	.91440	.03256	-.71135
39 ---	-.57050	.50814	-.43255
41 ---	.57199	-.50447	-.43287
43 ---	0	0	-1.00000
45 ---	-.70711	-.70711	.00000
47 ---	.70711	.70711	-.00000
49 ---	-1.00000	.00000	.00000
51 ---	1.00000	-.00000	-.00000
53 ---	-.70711	.70711	.00000
55 ---	.70711	-.70711	-.00000
2 ---	0	.00632	-3.32426
4 ---	0	.00781	-3.73326
6 ---	0	.01343	-6.28215
8 ---	0	.01383	-3.87865
10 ---	0	.00903	-3.49631
12 ---	0	.00752	-3.73503
14 ---	0	-.00669	-.98249
16 ---	.00237	.00261	-1.90670
18 ---	.00173	.00297	-.87074
20 ---	-.00006	.00417	-2.91808
22 ---	-.00043	.00440	-.95890
24 ---	-.00165	.00298	-1.87095
26 ---	-.00318	.00262	-.78287
28 ---	-.00358	-.00310	-.57585
30 ---	.00364	.00090	-1.41568
32 ---	.00039	.00083	.33835
34 ---	-.00002	.00102	-1.74889
36 ---	-.00010	.00104	.32742
38 ---	-.00318	.00066	-1.30992
40 ---	-.00062	.00071	.39025
42 ---	.00017	-.00073	-.85488
44 ---	0	0	-1.00000
46 ---	.00000	.00000	1.00000
48 ---	0	0	-1.00000
50 ---	.00000	-.00000	1.00000
52 ---	0	0	-1.00000
54 ---	.00000	-.00000	1.00000
56 ---	0	0	-1.00000

References

(1) Shinbrot, M. (1973): Lectures on Fluid Mechanics; Gordon & Breach; c.f. p108.

(2) Shinbrot, M. (1973): ibid; c.f. chapter 8, pp113-123.

(3) White, F. M. (1974): Viscous Fluid Flow; McGraw Hill; c.f. section 2.11, pp96-101.

(4) Ladyzhenskaya, O. A. (1975): Mathematical Analysis of Navier-Stokes Equations for Incompressible Liquids; Annual Review of Fluid Mechanics, 7, 249; c.f. p251.

(5) Shinbrot, M. (1973): ibid; c.f. p111.

(6) Shinbrot, M. (1973): ibid; c.f. theorem 4.3, p169, and section 12.3, pp182-192.

(7) Ladyzhenskaya, O. A. (1975): ibid; c.f. section 8, pp262-265.

(8) Aziz, K. & Hellums, J. D. (1967): Numerical Solution of the Three-Dimensional Equations of Motion for Laminar Natural Convection; Physics of Fluids, 10, 314; c.f. p320.

(9) Orszag, S. A. (1969): Numerical Methods for the Simulation of Turbulence; Physics of Fluids Supplement II, 12, II-250.

(10) Deardorff, J. W. (1970): A Numerical Study of Three-Dimensional Turbulent Channel Flow at Large Reynolds Numbers; J. Fluid Mech., 41, 453.

(11) Hirt, C. W., Cook, J. L. & Butler, T. D. (1970): A Lagrangian Method for Calculating the Dynamics of an Incompressible Fluid with Free Surface; J. Comp. Physics, 5, 103.

(12) White, F. M. (1974): ibid; c.f. p97.

(13) Aziz, K. & Hellums, J. D. (1967): ibid.

(14) Holst, P. H. & Aziz, K. (1972): Transient Three-Dimensional

Natural Convection in Confined Porous Media; Int. J. Heat & Mass Transfer, 15, 73.

(15) von Mises, R. & Friedrichs, K. O. (1971): Fluid Dynamics; Springer; c.f. p132.

(16) Wu, J. C. & Thompson, J. F. (1973): Numerical Solution of Time-Dependent Incompressible Navier-Stokes Equations using an Integro-Differential Formulation; Computers & Fluids, 1, 197.

(17) Thompson, J. F., Shanks, S. P. & Wu, J. C. (1974): Numerical Solution of Three-Dimensional Navier-Stokes Equations showing Trailing Tip Vortices; A. I. A. A. J., 12, 787.

(18) Témam, R. (1968): Une Méthode d'Approximation de la Solution des Équations de Navier-Stokes; Bull. Soc. Math. France, 96, 115.

(19) Témam, R. (1968): Sur l'Approximation de la Solution des Équations de Navier-Stokes par la Méthode des Pas Fractionnaires (I); Arch. Rat. Mech. Anal., 32, 135.

(20) Chorin, A. J. (1968): Numerical Solution of Incompressible Flow Problems; Symposium on Numerical Solution of Non-Linear Problems, Philadelphia, October 1968; S. I. A. M., 64; c.f. p59.

(21) Chorin, A. J. (1967): A Numerical Method for Solving Incompressible Viscous Flow Problems; J. Comp. Physics, 2, 12.

(22) Plows, W. H. (1968): Some Numerical Results for Two-Dimensional Steady Laminar Benard Convection; Physics of Fluids, 11, 1593.

(23) Chorin, A. J. (1967): The Numerical Solution of the Navier-Stokes Equations for an Incompressible Fluid; Bull. Amer. Math. Soc., 73, 928.

(24) Chorin, A. J. (1969): On the Convergence of Discrete Approximations to the Navier-Stokes Equations; Math. Comp., 23, 341.

(25) Témam, R. (1969): Sur l'Approximation de la Solution des Équations de Navier-Stokes par la Méthode des Pas Fractionnaires (II); Arch. Rat. Mech. Anal., 33, 377.

(26) Chorin, A. J. (1968): Numerical Solution of the Navier-Stokes Equations; Math. Comp., 22, 745.

(27) Veltishchev, N. F. & Zelnin, A. A. (1975): Numerical Simulation of Cellular Convection in Air; J. Fluid Mech., 68, 353.

(28) Crane, C. M. (1974): A new Method for the Solution of Time Dependent Viscous Flow; Applied Scientific Research, 30, 47.

(29) Peskin, C. S. (1973): Flow Patterns around Heart Valves; in Cabannes, H. & Témam, R. (editors): Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics, Paris, July 1972; Springer, Lecture Notes in Physics No.19, 214.

(30) Somerville, R. C. J. (1973): Numerical Simulation of Small-Scale Thermal Convection in the Atmosphere; in Cabannes, H. & Témam, R. (editors): Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics, Paris, July 1972; Springer, Lecture Notes in Physics No.19, 238.

(31) Padmanabhan, H., Ames, W. F., Kennedy, J. F. & Hung, T.-K. (1970): A Numerical Investigation of Wake Deformation in Density Stratified Fluids; J. Eng. Maths., 4, 229.

(32) Patankar, S. V. & Spalding, D. B. (1972): A Calculation Procedure for Heat, Mass & Momentum Transfer in Three-Dimensional Parabolic Flows; Int. J. Heat &

Mass Transfer, 15, 1787.

(33) Kellogg, O. D. (1953): Foundations of Potential Theory; Dover; c.f. chapter IX, sections 3-5, pp236-247.

(34) Lyczowski. R. W. & Gidaspow, D. (1971): Two-Dimensional Inert Accumulation in Space Hydrogen-Oxygen Fuel Cells; A. I. Ch. E. J., 17, 1131.

(35) Williams, G. P. (1969): Numerical Integration of the Three-Dimensional Navier-Stokes Equations for Incompressible Flow; J. Fluid Mech., 37, 727.

(36) Oden, J. T. & Wellford, L. C. (1972): Analysis of Flow of Viscous Fluids by the Finite-Element Method; A. I. A. A. J., 10, 1590.

(37) Bratanow, T. & Ecer, A. (1974): Analysis of Three-Dimensional Unsteady Viscous Flow around Oscillating Wings; A. I. A. A. J., 12, 1577.

(38) Welch, J. E., Harlow, F. H., Shannon, J. P. & Daly, B. J. (1965): The MAC Method - A Computing Technique for Solving Viscous, Incompressible, Transient Fluid-Flow Problems involving Free Surfaces; Los Alamos Scientific Laboratory Report LA-3425.

(39) Roache, P. J. (1975): The LAD, NOS and Split NOS Methods for the Steady-State Navier-Stokes Equations; Computers & Fluids, 3, 179.

(40) Aris, R. (1962): Vectors, Tensors, and the Basic Equations of Fluid Mechanics; Prentice-Hall; c.f. p70.

(41) Lamb, H. (1932): Hydrodynamics; Cambridge University Press.

(42) Kellogg, O. D. (1953): ibid.

(43) Weyl, H. (1940): The Method of Orthogonal Projection in Potential Theory; Duke Math. J., 7, 411.

(44) Kodaira, K. (1949): Harmonic Fields in Riemannian Manifolds

(Generalised Potential Theory); Annals of Maths.,
50, 587; c.f. chapter 2, section 4, pp602-606.

(45) Hodge, W. V. D. (1959): The Theory and Applications of Harmonic Integrals; Cambridge University Press.

(46) Bykhovski, E. B. & Smirnov, N. V. (1960): On Orthogonal Expansions of the Space of Vector Functions which are Square-Summable over a given Domain; Trudy Mat. Inst. Steklov, 59, 5; translation in Krizancic, I.: D. I. C. thesis, Imperial College, London, 1973.

(47) Liusternik, L. A. & Sobolev, V. J. (1961): Elements of Functional Analysis; Ungar; c.f. p73.

(48) Liusternik, L. A. & Sobolev, V. J. (1961): ibid; c.f. lemma 2, p75.

(49) Courant, R. & Hilbert, D. (1953): Methods of Mathematical Physics, volume I; Wiley Interscience; c.f. pp108-111.

(50) Bykhovski, E. B. & Smirnov, N. V. (1960): ibid; c.f. p26.

(51) Bykhovski, E. B. & Smirnov, N. V. (1960): ibid; c.f. theorem 1.2, p13, theorem 5.2, p21, and theorem 1.3, p26.

(52) Bykhovski, E. B. & Smirnov, N. V. (1960): ibid; c.f. p34.

(53) Bykhovski, E. B. & Smirnov, N. V. (1960): ibid; c.f. p22.

(54) Hague, B. & Martin, D. (1970): An Introduction to Vector Analysis; Methuen / Science Paperbacks; c.f. p76.

(55) Stephenson, G. (1969): Mathematical Methods for Science Students; Longmans; c.f. p168.

(56) Moon, P. & Spencer, D. E. (1953): The Meaning of the Vector Laplacian; J. Franklin Inst., 256, 551.

(57) Spiegel, M. R. (1959): Vector Analysis; McGraw Hill; c.f. pp105-107.

(58) Kreysig, E. (1967): Advanced Engineering Mathematics; Wiley; c.f. p337.

(59) Aziz, K. & Hellums, J. D. (1967): ibid; c.f. p320.

(60) Kellogg, O. D. (1953): ibid; c.f. chapter IV, section 8, pp100–112.

(61) Timman, R. (1954): Mémoires offerts à D. Riabouchinsky; Publ. Scient. Tech. Min. Air., 351.

(62) Hirasaki, G. J. & Hellums, J. D. (1968): A General Formulation of the Boundary Conditions on the Vector Potential in Three-Dimensional Hydrodynamics; Quart. Applied Maths., 26, 331.

(63) Hirasaki, G. J. & Hellums, J. D. (1970): Boundary Conditions on the Vector and Scalar Potentials in Viscous Three-Dimensional Hydrodynamics; Quart. Applied Maths., 28, 293.

(64) O'Carroll, M. J. (1972): Computation of Rotational Flows; Quart. Applied Maths., 29, 547.

(65) Courant, R. & Friedrichs, K. O. (1948): Supersonic Flow and Shock Waves; Wiley Interscience; c.f. section 54, pp121–124.

(66) Kreysig, E. (1967): ibid; c.f. p286.

(67) Roache, P. J. (1975): Recent Developments and Problem Areas in Computational Fluid Dynamics; in Oden, J. T. (editor): International Conference on Computational Methods in Nonlinear Mechanics, Austin, September 1974; Springer, Lecture Notes in Mathematics No. 461, 195; c.f. p213.

(68) Chorin, A. J. (1968): Numerical Solution of the Navier-Stokes Equations; Math. Comp., 22, 745; c.f. pp758–761.

(69) Coleman, B. D., Markovitz, H. & Noll. W. (1966): Viscometric Flows of Non-Newtonian Fluids; Springer; c.f. p1.

(70) Beavers, G. S., Sparrow, E. M. & Masha, B. A. (1974): Boundary

Condition at a Porous Surface which bounds a Fluid Flow; A. I. Ch. E. J., <u>20</u>, 596.

(71) Schaaf, S. A. & Chambré, P. L. (1961): Flow of Rarefied Gases; Princeton University Press; c.f. section H,14, pp34-35.

(72) Heywood, J. G. (1970): On Stationary Solutions of the Navier-Stokes Equations as Limits of Non-Stationary Solutions; Arch. Rat. Mech. Anal., <u>37</u>, 48.

(73) Cheng, S.-I. (1969): Accuracy of Difference Formulation of Navier-Stokes Equations; Physics of Fluids Supplement II, <u>12</u>, II-34.

(74) Roache, P. J. (1972): Computational Fluid Dynamics; Hermosa; c.f. section III-C-7, p154.

(75) Stephenson, G. (1969): ibid; c.f. p95.

(76) Roache, P. J. (1972): ibid; c.f. section VI-B, pp292-301.

(77) Griffiths, H. B. & Hilton, P. J. (1970): A Comprehensive Textbook of Classical Mathematics - A Contemporary Interpretation; van Nostrand Reinhold; c.f. p38.

(78) Jeffreys, H. & Jeffreys, B. S. (1966): Methods of Mathematical Physics; Cambridge University Press; c.f. section 1.133, pp50-51.

(79) Roache, P. J. (1972): ibid; c.f. p64.

(80) White, F. M. (1974): ibid; c.f. p221.

(81) Roache, P. J. (1972): ibid; c.f. section III-A-5, pp36-53, and section III-A-14, pp83-87.

(82) Ames, W. F. (1969): Numerical Methods for Partial Differential Equations; Nelson; c.f. section 2.3, pp49-54, section 2.14, pp82-89, and section 5.4, pp245-249.

(83) Varga, R. S. (1962): Matrix Iterative Analysis; Prentice-Hall; c.f. section 8.3, pp262-268.

(84) Roache, P. J. (1972): ibid; c.f. p306.

(85) Pearson, C. E. (1965): A Computational Method for Viscous Flow Problems; J. Fluid Mech., 21, 611.

(86) White, F. M. (1974): ibid; c.f. p218.

(87) Lighthill, M. J. (1963): Introduction. Boundary Layer Theory; in Rosenhead, L. (editor): Laminar Boundary Layers; Oxford, Clarendon Press; c.f. chapter II, section 1.7, pp57-60.

(88) Aziz, K. & Hellums, J. D. (1967): ibid; c.f. pp323-324.

(89) Roache, P. J. (1972): ibid; c.f. p141.

(90) Aziz, K. & Hellums, J. D. (1967): ibid; c.f. pp319-320.

(91) Bird, R. B., Stewart, W. E. & Lightfoot, E. N. (1960): Transport Phenomena; Wiley; c.f. p90.

(92) Stephenson, G. (1969): ibid; c.f. p232.

(93) Peaceman, D. W. & Rachford, H. H. (1955): The Numerical Solution of Parabolic and Elliptic Differential Equations; J. S. I. A. M., 3, 28.

(94) Douglas, J. (1962): Alternating Direction Methods for Three Space Variables; Num. Math., 4, 41.

(95) Samarskii, A. A. (1963): Schemes of High Order Accuracy for the Multi-Dimensional Heat Conduction Equation; U. S. S. R. Comp. Math. & Math. Phys., 3, 1107.

(96) McKee, S. & Mitchell, A. R. (1970): Alternating Direction Methods for Parabolic Equations in Two Space Dimensions with a Mixed Derivative; Computer J., 13, 81.

(97) Roache, P. J. (1972): ibid; c.f. appendix A, pp345-349.

(98) Ames, W. F. (1969): ibid; c.f. section 3.4, pp103-106.

(99) Roache, P. J. (1972): ibid; c.f. p118.

(100) Ames, W. F. (1969): ibid; c.f. section 3.7, pp119-125.

(101) Varga, R. S. (1962): ibid; c.f. section 6.4, pp194-200.

(102) Bird, R. B., Stewart, W. E. & Lightfoot, E. N. (1960): ibid; c.f. p133.

(103) Chwang, A. T. & Wu, T. Y.-T. (1975): Hydromechanics of Low-Reynolds-Number Flow. Part 2. Singularity Method for Stokes Flows; J. Fluid Mech., 67, 787; c.f. section 4, pp799-801.

(104) Roache, P. J. (1972): ibid; c.f. p178.

(105) Roache, P. J. (1972): ibid; c.f. section III-D, pp174-179.

(106) Roache, P. J. (1972): ibid; c.f. p119.

(107) Roache, P. J. (1972): ibid; c.f. section III-A-5, pp36-53.

(108) Bird, R. B., Stewart, W. E. & Lightfoot, E. N. (1960): ibid; c.f.p192.

(109) Bird, R. B., Stewart, W. E. & Lightfoot, E. N. (1960): ibid; c.f. p193.

(110) Chwang, A. T. & Wu, T. Y.-T. (1975): ibid; c.f. p801.

(111) Saffman, P. G. (1965): The Lift on a Small Sphere in a Slow Shear Flow; J. Fluid Mech., 22, 385; c.f. p385 (N.B. correction in J. Fluid Mech., 31, 624, (1968)).