INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 2, 225–238 (1982)

THE IMPLEMENTATION OF NORMAL AND/OR TANGENTIAL BOUNDARY CONDITIONS IN FINITE ELEMENT CODES FOR INCOMPRESSIBLE FLUID FLOW

M. S. ENGELMAN AND R. L. SANI

CIRES/NOAA, Department of Chemical Engineering, University of Colorado, Boulder, Colorado 80309, U.S.A.

P. M. GRESHO

Lawrence Livermore National Laboratory, Livermore, California 94550, U.S.A.

SUMMARY

Various techniques for implementing normal and/or tangential boundary conditions in finite element codes are reviewed. The principle of global conservation of mass is used to define a unique direction for the outward pointing normal vector at any node on an irregular boundary of a domain containing an incompressible fluid. This information permits the consistent and unambiguous application of essential or natural boundary conditions (or any combination thereof) on the domain boundary regardless of boundary shape or orientation with respect to the co-ordinate directions in both two and three dimensions. Several numerical examples are presented which demonstrate the effectiveness of the recommended technique.

KEY WORDS Normal Direction Boundary Conditions Incompressible Mass Conservation

1. INTRODUCTION

In the finite element analysis of structural and fluid dynamics problems it is often required to specify normal or tangential displacements or velocities along element boundaries which are not level surfaces of the co-ordinate system for the problem. Typically, this arises in structural problems when a sliding contact is imposed between a body and an inclined surface or plane, i.e. the normal displacement of all nodes along the surface are specified to be zero. Similarly, in fluid dynamics the necessity to impose specified normal or tangential velocities arises in many situations; for example, in most steady simulations which include free surfaces the boundary condition at the free surface is that u_n, the normal velocity component, is zero.[1]

Since a Cartesian co-ordinate system is employed in the typical FEM formulation, the degrees of freedom are the x and y components of velocity (or displacement) and the usual methods of treating a constrained degree of freedom are no longer applicable when we wish to specify the normal or tangential component of a variable at a boundary which is not parallel to the X or Y axis. Herein we will be concerned with the details of the implementation of such boundary conditions, particularly in finite element codes for incompressible fluid flow. Such an implementation will to some extent be code dependent; however, the basic techniques to be discussed should be applicable in most cases. We will first review several of

0271–2091/82/030225–14$01.40

Received 1 August 1981
Revised 5 November 1981

the techniques used in some of the current finite element fluid mechanics literature, and then detail the implementation used in most structural FEM codes, and by a few workers in fluid mechanics (e.g. see Pinder and Gray, Reference 2, p. 281). The technique is quite general, and not subject to the restrictions or drawbacks that afflict the methods that have been used to date in many fluid mechanical applications.

We shall also focus on the method of computing the normal and tangential directions at a node point on the boundary along which the normal or tangential boundary condition is to be applied and demonstrate that if the normal vector computation is not consistent with the incompressibility constraint, erroneous results, especially pressure, will be obtained owing to an inconsistent set of equations.

2. IMPLEMENTATION

We consider the problem of a boundary along which either the normal or tangential velocity component is specified while the remaining component is free, as depicted in Figure 1. Our aim will be to detail a general technique for implementing the boundary conditions (u_t specified, u_n free) or (u_t free, u_n specified), where the specified value may be zero or non-zero, which does not require treating the different sets of conditions in a separate manner; i.e. the implementation technique should be independent of whether u_t or u_n is being constrained. The method should also require minimum alterations to the existing FEM code, and preferably only at element level.

Suppose initially we wish to impose $u_n = f$ and u_t free along the boundary, then the constraint equation for node i can be written

$$\mathbf{n} \cdot \mathbf{u} = n_x u_x + n_y u_y = f \tag{1}$$

where (u_x, u_y) are the x and y velocity components at node i and (n_x, n_y) are the x and y components of the outward pointing unit normal vector at node i.

Let $\mathbf{M} = \mathbf{0}$ be the momentum equation for node i. One approach is to resolve the vector momentum equation into its x and y components, $M_x = 0$ and $M_y = 0$, and then solve for u_x and u_y by discarding one of the momentum equations at node i, say the y equation, and replacing it by the constraint equation (1). This approach, adopted for example by Silliman,[3] breaks down when the tangent at node i approaches the y axis (since the 'wrong' momentum equation has been discarded).

Another procedure, which is free of the previous limitation, is to form the tangential momentum equation at node i,

$$0 = M_t = \mathbf{t} \cdot \mathbf{M} = t_x M_x + t_y M_y, \tag{2}$$

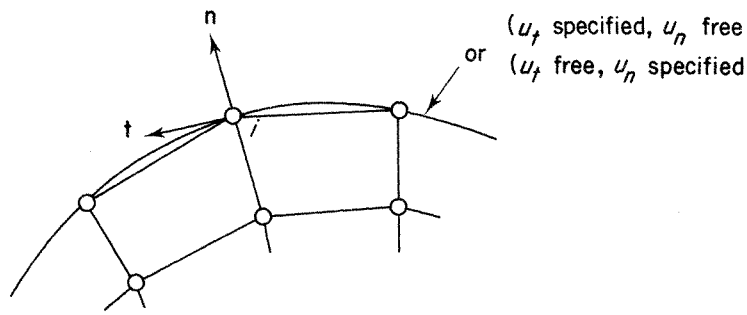

Figure 1. Boundary conditions

and to use this equation together with the constraint equation (1) to solve for u_x and u_y. This method is generally quite satisfactory, the main objection being that its implementation is not independent of the type of boundary condition being applied. If u_t is constrained then the normal momentum equation must be formed instead of the tangential one and the tangential constraint equation used. More importantly, the implementation of this technique in a *general* FEM code is not straightforward, requiring a great deal of manipulation due to the necessity of discarding an existing equation and introducing a new one for a particular degree of freedom. This method has been used by Saito and Scriven[1] in free surface problems where the constraint $u_n = 0$ was the only one required.

An alternative approach is rather than working with u_x and u_y as the degrees of freedom for node i on the boundary, the local x–y co-ordinate system at node i is rotated to coincide with the tangential and normal directions so that u_t and u_n now become the degrees of freedom at node i. Noting that

$$\begin{aligned} u_t &= t_x u_x + t_y u_y \\ u_n &= n_x u_x + n_y u_y, \end{aligned} \tag{3}$$

the conversion from (u_x, u_y) to (u_t, u_n) is achieved at element level by post-multiplying the total element stiffness matrix $\mathbf{K}$ ($\mathbf{K}$ includes contributions from both the momentum and continuity equations) of any element containing node i by the orthogonal rotation matrix $\mathbf{R}$, i.e. $\mathbf{R}^{-1} = \mathbf{R}^{\mathrm{T}}$, where the entries of $\mathbf{R}$ at j, k (the positions of the u_x and u_y degrees of freedom in the element stiffness matrix) are

$$\begin{matrix} & j & k \\ \begin{matrix} j \\ k \end{matrix} & \begin{bmatrix} t_x & t_y \\ n_x & n_y \end{bmatrix} \end{matrix} .$$

The remaining diagonal entries of $\mathbf{R}$ are unity and all other elements are zero. The rotation matrix, which transforms u_x and u_y to u_t and u_n, is determined once $\mathbf{n}$ is known because $\mathbf{t} = \mathbf{k} \times \mathbf{n}$, where $\mathbf{k}$ is the unit vector in the z direction, i.e. $t_x = -n_y$ and $t_y = n_x$. Note that this describes a transformation from a right-handed (x–y) co-ordinate system to a left-handed (t–n) co-ordinate system; we chose this approach so that the positive tangent direction is anti-clockwise. It is of course possible (and perhaps even preferable) to define the tangent in the clockwise sense, via $\mathbf{t} = \mathbf{n} \times \mathbf{k} = -\mathbf{k} \times \mathbf{n}$.

This procedure has the great advantage that constrained values for u_n or for u_t can, once $\mathbf{K}$ is modified to $\mathbf{K}\mathbf{R}^{\mathrm{T}}$, be treated in exactly the same manner as constrained values for u_x and u_y, and without any additional code modifications. The method, however, suffers from the same limitation as the first one discussed. If the tangent at the surface approaches the X or Y axis, then depending on the component being specified, u_n or u_t, the method will break down.

The natural modification to the above procedure which eliminates this potential problem is to transform the x and y momentum equations to the tangential and normal momentum equations *before* transforming u_x and u_y to u_t and u_n. This is the procedure adopted in most structural FEM codes[4] and in some fluids applications. Noting that

$$\begin{aligned} M_t &= t_x M_x + t_y M_y \\ M_n &= n_x M_x + n_y M_y, \end{aligned}$$

we see that this is simply achieved by pre-multiplying the element stiffness of any element containing node i by the inverse (transpose) of the same orthogonal rotation matrix $\mathbf{R}$ used to transform u_x, u_y to u_t, u_n.

So, in summary, to apply a constrained normal or tangential velocity at node i, we simply pre-multiply the element stiffness matrix and corresponding right-hand side vector by the appropriate matrix $\mathbf{R}$ and post-multiply the stiffness matrix by its transpose $\mathbf{R}^T$, and then apply the specified value in the same manner as a specified value for u_x, u_y (i.e. the solution at this node will be in terms of the normal and tangential components of velocity). Once the system of equations has been solved, the u_x and u_y components of velocities at node i can be recovered using equation (3)—or its inverse (see equation (10)).

This procedure satisfies all the criteria which we originally required. It has the additional advantage that because the normal and tangential equations are available at node i, the application of normal or tangential stress (or natural) boundary conditions at the boundary is now straightforward.

3. COMPUTATION OF NORMAL DIRECTION

Crucial to the successful implementation of any of the procedures in the previous section is the computation of the normal components n_x, n_y at a boundary node. Along a curved boundary which is approximated using isoparametric elements (see Figure 1), there is often a discontinuity in the slope of the approximated boundary and hence, the normal direction at such a node is not uniquely defined (such non-uniqueness can also occur on the exact boundary). An appropriate (and unique) normal direction may be determined, however, by invoking mass conservation arguments when dealing with an incompressible fluid.

When using a standard Galerkin FEM procedure, the weak form of $\nabla \cdot \mathbf{u} = 0$ is:

$$\int_\Omega \psi_i \nabla \cdot \mathbf{u} \, d\Omega = 0 \quad \text{for all } \psi_i, \tag{4}$$

where ψ_i is any basis function for the pressure and Ω is the entire computational domain. Since the space of pressure basis functions must always contain the constant function (hydrostatic pressure level), it follows from equation (4) that a global mass balance is always satisfied, i.e.

$$\int_\Omega \nabla \cdot \mathbf{u} \, d\Omega = 0. \tag{5}$$

Remark: For the common case in which the $\{\psi_i\}$ are composed of Lagrange polynomials, equation (5) follows from equation (4) by direct summation of all the weak constraints since $\sum_j \psi_j = 1$.

Inserting the approximate (FEM) solution,

$$\mathbf{u} = \sum_{i=1}^{N} \mathbf{u}_i \varphi_i \tag{6}$$

into equation (5) gives

$$\sum_{i=1}^{N} \left(u_{x_i} \int_\Omega \frac{\partial \varphi_i}{\partial x} \, d\Omega + u_{y_i} \int_\Omega \frac{\partial \varphi_i}{\partial y} \, d\Omega \right) = 0 \tag{7}$$

where N is the *total* number of velocity nodes (in Ω and on $\partial\Omega$) and φ_i is a basis function for velocity. We show next that the summation in equation (7) is, effectively, a boundary node constraint equation; i.e. all internal nodes cancel, as stated by Sani *et al.*[5] To do this we

invoke Green's theorem in the following form

$$\int_\Omega \frac{\partial \varphi_i}{\partial x}\,dx\,dy = \int_{\partial\Omega} \varphi_i n_x\,dl, \tag{8a}$$

$$\int_\Omega \frac{\partial \varphi_i}{\partial y}\,dx\,dy = \int_{\partial\Omega} \varphi_i n_y\,dl, \tag{8b}$$

where dl is an element of arc length on $\partial\Omega$. Thus, from equations (7) and (8),

$$\sum_{i=1}^{N} \left(u_{x_i} \int_{\partial\Omega} \varphi_i n_x\,dl + u_{y_i} \int_{\partial\Omega} \varphi_i n_y\,dl \right) = 0. \tag{9}$$

But $\varphi_i = 0$ on $\partial\Omega$ for *all* internal nodes and therefore the sum over N is, effectively, collapsed to a sum over N_B, the number of *boundary* nodes.

The next step is to relate u_x and u_y to the normal and tangential velocities. Inverting equation (3) and using $\mathbf{t} = \mathbf{k} \times \mathbf{n}$ we obtain

$$u_x = n_x u_n - n_y u_t \tag{10a}$$

$$u_y = n_y u_n + n_x u_t. \tag{10b}$$

In general, of course, n_x and n_y are functions of l, the location on $\partial\Omega$ and may therefore be ambiguous at certain points. Now let u_{n_i} be the normal component of velocity at the boundary node i, and similarly for u_{t_i}. Finally, we introduce *nodal* components of the unit normal at node i, n_{x_i} and n_{y_i}; our goal is to determine these quantities uniquely (and mass consistently), even though the geometric normal may be undefined at node i. Thus we have

$$u_{x_i} = n_{x_i} u_{n_i} - n_{y_i} u_{t_i} \tag{11a}$$

$$u_{y_i} = n_{y_i} u_{n_i} + n_{x_i} u_{t_i}. \tag{11b}$$

Inserting equation (11) into equation (7) yields, considering the results from equation (9),

$$\sum_{i=1}^{N_B} \left[(n_{x_i} u_{n_i} - n_{y_i} u_{t_i}) \int_\Omega \frac{\partial \varphi_i}{\partial x}\,d\Omega + (n_{y_i} u_{n_i} + n_{x_i} u_{t_i}) \int_\Omega \frac{\partial \varphi_i}{\partial y}\,d\Omega \right] = 0$$

which involves only those nodes on $\partial\Omega$ and can be rearranged to

$$\sum_{i=1}^{N_B} \left[\left(n_{x_i} \int_\Omega \frac{\partial \varphi_i}{\partial x}\,d\Omega + n_{y_i} \int_\Omega \frac{\partial \varphi_i}{\partial y}\,d\Omega \right) u_{n_i} + \left(n_{x_i} \int_\Omega \frac{\partial \varphi_i}{\partial y}\,d\Omega - n_{y_i} \int_\Omega \frac{\partial \varphi_i}{\partial x}\,d\Omega \right) u_{t_i} \right] = 0. \tag{12}$$

The mass consistent definition of the normal direction follows from equation (12) which, being a statement of global mass conservation, must be *independent* of the nodal tangential velocities. Hence, it must follow that

$$n_{y_i} \int_\Omega \frac{\partial \varphi_i}{\partial x}\,d\Omega = n_{x_i} \int_\Omega \frac{\partial \varphi_i}{\partial y}\,d\Omega, \tag{13}$$

which, when combined with the normalization requirement, $n_{x_i}^2 + n_{y_i}^2 = 1$, gives

$$n_{x_i} = \frac{1}{n_i} \int_\Omega \frac{\partial \varphi_i}{\partial x}\,d\Omega, \qquad n_{y_i} = \frac{1}{n_i} \int_\Omega \frac{\partial \varphi_i}{\partial y}\,d\Omega; \tag{14}$$

where

$$n_i = \left[\left(\int_\Omega \frac{\partial \varphi_i}{\partial x}\,d\Omega \right)^2 + \left(\int_\Omega \frac{\partial \varphi_i}{\partial y}\,d\Omega \right)^2 \right]^{1/2},$$

as the consistent normal direction at node i. Note that the domain integrals are replaced, in practice, by Ω_i, the support of φ_i, and the integrations are performed in the standard way via isoparametric mapping.

It is noteworthy that this result applies to any type of element, the only restriction being an incompressible medium. Also, this result is not new; it was previously derived in a less formal and simpler manner by Gray,[6] based on the argument that a zero normal flow condition can only be satisfied in an averaged sense. Our derivation thus complements and substantiates that of Gray.

Finally, inserting these results, equations (13) and (14), into equation (12) gives

$$\sum_{i=1}^{N_B} n_i u_{n_i} = 0 \quad \text{or} \quad \sum_{i=1}^{N_B} n_i (n_{x_i} u_{x_i} + n_{y_i} u_{y_i}) = 0, \tag{15}$$

which is the appropriate discretized analogue of the global mass balance equation,

$$\int_{\partial\Omega} \mathbf{u} \cdot \mathbf{n} \, dl = 0, \tag{16}$$

obtained from equation (5) via the divergence theorem. Note that n_i has units of length.

This technique is simple to implement and is generally quite effective. It does not, however, result in a physically acceptable normal when the node i is contained in a single element on the corner of the computational domain. In this case, any rectangular element (i.e. φ_i could be bilinear, biquadratic, etc.) will generate a normal direction given by

$$\theta = \tan^{-1}(n_y/n_x) = \tan^{-1}(l/h)$$

for the rectangular element shown in Figure 2 (which turns out to be the same normal as that of a circumscribed ellipse):

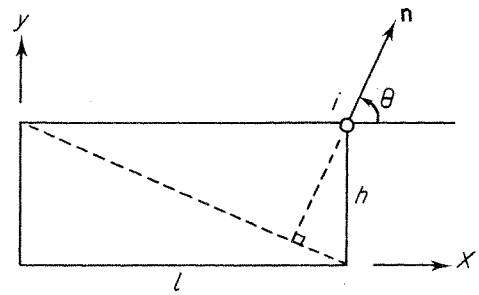

Figure 2. Consistent normal at element corner

While still consistent from the viewpoint of mass conservation, this normal is quite awkward in many cases; for example, suppose that the boundary conditions require that $u_n = 1$ on the right vertical face and $u_n = 0$ on the top surface (in general, of course, the element need not be rectangular nor aligned in the co-ordinate directions). We consider here two ways of handling such a case: (a) the consistently computed normal may be used in combination with the known (desired) resolution of the specified values in the geometric sense and the appropriate changes made in specifying u_n and u_t; or (b) the nearest boundary Gauss point may be used on the side whose normal is desired, i.e. the normal direction is computed at the Gauss point and applied at the node. We use the latter procedure in the examples below; partly because the first procedure can be used only when both the desired normal and tangential velocity components are known.

Although perhaps not obvious at first glance, the use of normal and tangential directions which are not consistent with the incompressibility condition (including, for example, the *analytic* normals for the boundary being modelled) can result in erroneous results. As demonstrated earlier, mass conservation arguments lead to the global mass balance

$$\int_{\Omega} \nabla \cdot \mathbf{u} \, d\Omega = \int_{\partial\Omega} \mathbf{u} \cdot \mathbf{n} \, dl = 0.$$

The right-hand side of this equation represents a constraint equation on the normal velocities (as determined by the consistently computed normals) along the boundary $\partial\Omega$, which is implicitly contained within the system of equations to be solved; in other words, it is possible to linearly combine the m discretized continuity equations so as to reproduce this constraint equation (see equation (15)).

Consider a flow for which the normal velocities are properly specified *everywhere* on the boundary, (i.e. equation (15) is satisfied, see also Reference 5) e.g. any contained flow. If the normal velocities are imposed via the procedure detailed in the previous section, together with the consistently computed normals, the pressure will be (appropriately) determined only up to an additive arbitrary constant (of order one if Gaussian elimination is employed). This occurs because the imposed velocities now duplicate the implicit constraint equation and so we have a system of $m-1$ linearly independent equations for m pressure unknowns. When the Gaussian elimination equation solver reaches the final pressure equation, it will encounter an equation of the form

$$r_1^e p_m = r_2^e,$$

where p_m is the last pressure unknown and r_i^e is an $O(1)$ random number multiplied by the round-off level of the computer. The solution of this equation results in $p_m \sim O(1)$ and a well-behaved back substitution process; the arbitrary hydrostatic pressure level has been set at r_2^e/r_1^e.

On the other hand, suppose that the same normal velocities are imposed, but now using normals which are not consistent from the point of view of mass conservation, e.g. geometrically computed normals. We will now have, contained within the algebraic system of equations, a constraint equation of the form

$$\int_{\Omega} \nabla \cdot \mathbf{u} \, d\Omega = \int_{\partial\Omega} \mathbf{u} \cdot \mathbf{n} \, dl = \varepsilon \tag{17}$$

where ε represents the mass imbalance due to the inconsistent normals. In this case, when the equation solver reaches the final pressure equation, it will encounter an equation of the form

$$r^e p_m = O(\varepsilon).$$

On our CDC computer, $r^e \sim 10^{-14}$, so that if $\varepsilon \sim 1$ (or even 10^{-2}), the pressure results may be completely meaningless due to overflow conditions which can occur during the back substitution solution phase for the remaining pressures. As pointed out in Reference 5, if $\varepsilon \ll 1$, the velocities will be little-affected; a 'large' ε, however, will also lead to an incorrect velocity solution and a very large constant pressure field. The algebraic system is 'numerically' ill-posed if equation (15) is violated to an extent which is recognizable by the computer.

4. EXTENSION TO 3-D

The procedures described in the previous sections can be extended to three dimensional problems in a straightforward manner. Instead of transforming the x and y momentum equations to the normal and tangential momentum equations, the x, y and z momentum equations will be transformed to a normal and two tangential momentum equations. Similarly the u_x, u_y and u_z velocity components are transformed to the corresponding tangential and normal velocities u_{t_1}, u_{t_2} and u_n, where $\mathbf{t}_1, \mathbf{t}_2$ are two linearly-independent tangent vectors and $\mathbf{n}$ the normal vector of the surface passing through the node in question.

In practice, this is achieved by pre-multiplying by the matrix $\mathbf{R}$ and post-multiplying by $\mathbf{R}^T$, the total element stiffness matrix of any element containing node i, where $\mathbf{R}$ is the orthogonal rotation matrix with the following entries at j, k, l (the positions of the three boundary node degrees of freedom in the element stiffness matrix):

$$\begin{matrix} & j & k & l \\ \begin{matrix} j \\ k \\ l \end{matrix} & \begin{bmatrix} t_x^1 & t_y^1 & t_z^1 \\ t_x^2 & t_y^2 & t_z^2 \\ n_x & n_y & n_z \end{bmatrix} \end{matrix},$$

where $\mathbf{t}_1 = (t_x^1, t_y^1, t_z^1)$, $\mathbf{t}_2 = (t_x^2, t_y^2, t_z^2)$. The remaining diagonal elements of $\mathbf{R}$ are unity and all other elements are zero. The right hand side vector is similarly pre-multiplied by $\mathbf{R}$ (at element level).

It now remains to compute the normal and tangent vectors $\mathbf{n}, \mathbf{t}_1, \mathbf{t}_2$. The arguments of the previous section carry over identically to three dimensions resulting in $\mathbf{n} = (n_x, n_y, n_z)$ where (suppressing the index for node i for convenience)

$$n_x = \frac{1}{n}\int_\Omega \frac{\partial \varphi_i}{\partial x}\,d\Omega, \qquad n_y = \frac{1}{n}\int_\Omega \frac{\partial \varphi_i}{\partial y}\,d\Omega, \qquad n_z = \frac{1}{n}\int_\Omega \frac{\partial \varphi_i}{\partial z}\,d\Omega;$$

and

$$n = \left[\left(\int_\Omega \frac{\partial \varphi_i}{\partial x}\,d\Omega\right)^2 + \left(\int_\Omega \frac{\partial \varphi_i}{\partial y}\,d\Omega\right)^2 + \left(\int_\Omega \frac{\partial \varphi_i}{\partial z}\,d\Omega\right)^2\right]^{1/2}.$$

However, whereas in two dimensions for any given normal vector there exists a unique tangent vector, in three dimensions there exists only a unique tangent *plane*, with an infinite number of possibilities for the pair $(\mathbf{t}_1, \mathbf{t}_2)$. We now outline a procedure for calculating such a pair given a normal $\mathbf{n} = (n_x, n_y, n_z)$ to the surface S at a node i:

(a) First compute $\max\{|\mathbf{n}\times\mathbf{i}|, |\mathbf{n}\times\mathbf{j}|, |\mathbf{n}\times\mathbf{k}|\}$ at the node point in question and choose that unit vector associated with the maximum value, say $\mathbf{k}$, to form $\mathbf{t}_1$, a unit vector perpendicular to the plane determined by $\mathbf{n}$ and $\mathbf{k}$, i.e.

$$\mathbf{t}_1 = \frac{\mathbf{n}\times\mathbf{k}}{|\mathbf{n}\times\mathbf{k}|} = \left(\frac{n_y}{n^*}, \frac{-n_x}{n^*}, 0\right); \qquad n^* = (n_x^2 + n_y^2)^{1/2}$$

(b) Complete a right-handed triad of vectors using $\mathbf{n}$ and $\mathbf{t}_1$, i.e.

$$\mathbf{t}_2 = \mathbf{n}\times\mathbf{t}_1 = \left(\frac{n_x n_z}{n^*}, \frac{n_y n_z}{n^*}, -n^*\right).$$

The triad $(\mathbf{t}_1, \mathbf{t}_2, \mathbf{n})$ defines the normal and two tangential directions at the node i on the surface S.

5. NUMERICAL EXPERIMENTS

The technique described in the previous sections was implemented in the FEM code FIDAP,[7] which is a general purpose code for solving the two or three dimensional equations of fluid flow including the effect of heat transfer using a mixed or penalty type formulation.

To illustrate the capabilities of the implementation, three numerical simulatons will be presented. The first is Hamel flow in which fluid flows between intersecting infinite plates—see Figure 3. This flow is particularly well suited for our purposes since it has an analytic solution for comparison and, since only a finite length of wedge can be modelled, it requires the imposition of normal and tangential boundary conditions along curved boundaries.

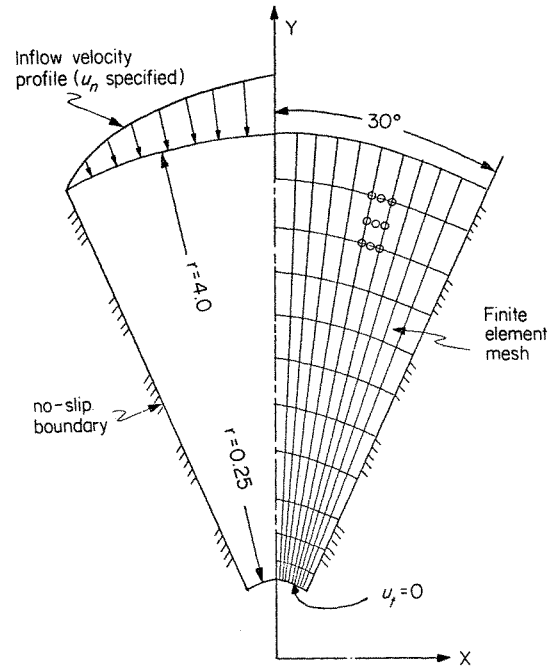

Figure 3. Hamel problem

Table I. Hamel solution at $r = 1$

	Analytic		Numerical	
Angle	u_y	u_x	u_y	u_x
0	−4·6560	0·	−4·6146	0·
3	−4·6282	0·2414	−4·6091	0·2355
6	−4·6003	0·4828	−4·5860	0·4799
9	−4·5724	0·7242	−4·5413	0·7209
12	−4·5101	0·9586	−4·4707	0·9561
15	−4·3971	1·1781	−4·3359	1·1779
18	−4·2075	1·3671	−4·1650	1·3654
21	−3·8687	1·4851	−3·8258	1·4810
24	−3·2388	1·4420	−3·2012	1·4312
27	−2·0640	1·0517	−2·0375	1·0409

Table II. Exact and computed normals for Hamel flow at outlet ($r = 0{\cdot}25$)

	Computed		Exact	
Angle	n_y	n_x	n_y	n_x
0	0·99998	0·00704	1·	0·
3	0·99867	0·05156	0·99863	0·05234
6	0·99451	0·10461	0·99452	0·10453
9	0·98761	0·15690	0·98769	0·15643
12	0·97816	0·20778	0·97815	0·20791
15	0·96589	0·25895	0·96593	0·25882
18	0·95121	0·30854	0·95106	0·30902
21	0·93352	0·35852	0·93358	0·35837
24	0·91389	0·40596	0·91355	0·40674
27	0·89042	0·45514	0·89101	0·45399
30	0·86968	0·49362	0·86603	0·50000

For our simulation the flow was taken to be inward (sink flow). The exact normal velocities were applied at the radius $r = 4$, and at the exit, $r = 1/4$, the condition $u_t = 0$ was applied. The intersecting plates were set at a half angle of 30° with only half the wedge being modelled due to the symmetry of the problem. The element mesh used is shown in Figure 3. The Reynolds number was set to 61 and the analytic solution computed using a program provided by Gartling *et al.*[8] A nine-node biquadratic velocity–discontinuous linear pressure quadrilateral element with a penalty formulation was used for the simulation.

Table I compares the analytic and numerical solutions at the radius $r = 1$, for 3° increments of ϑ. The error in the solution at any node is less than 1 per cent. In Table II the exact and

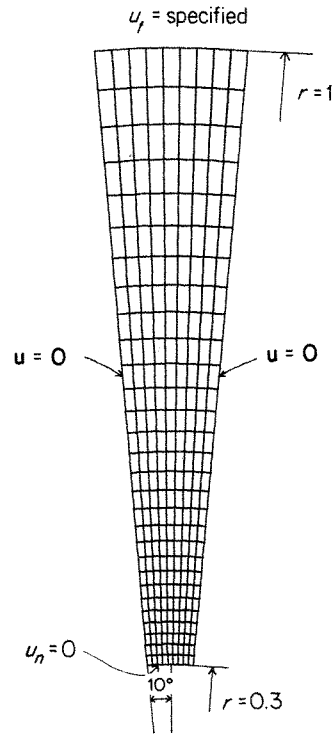

Figure 4(a). Lid-driven wedge

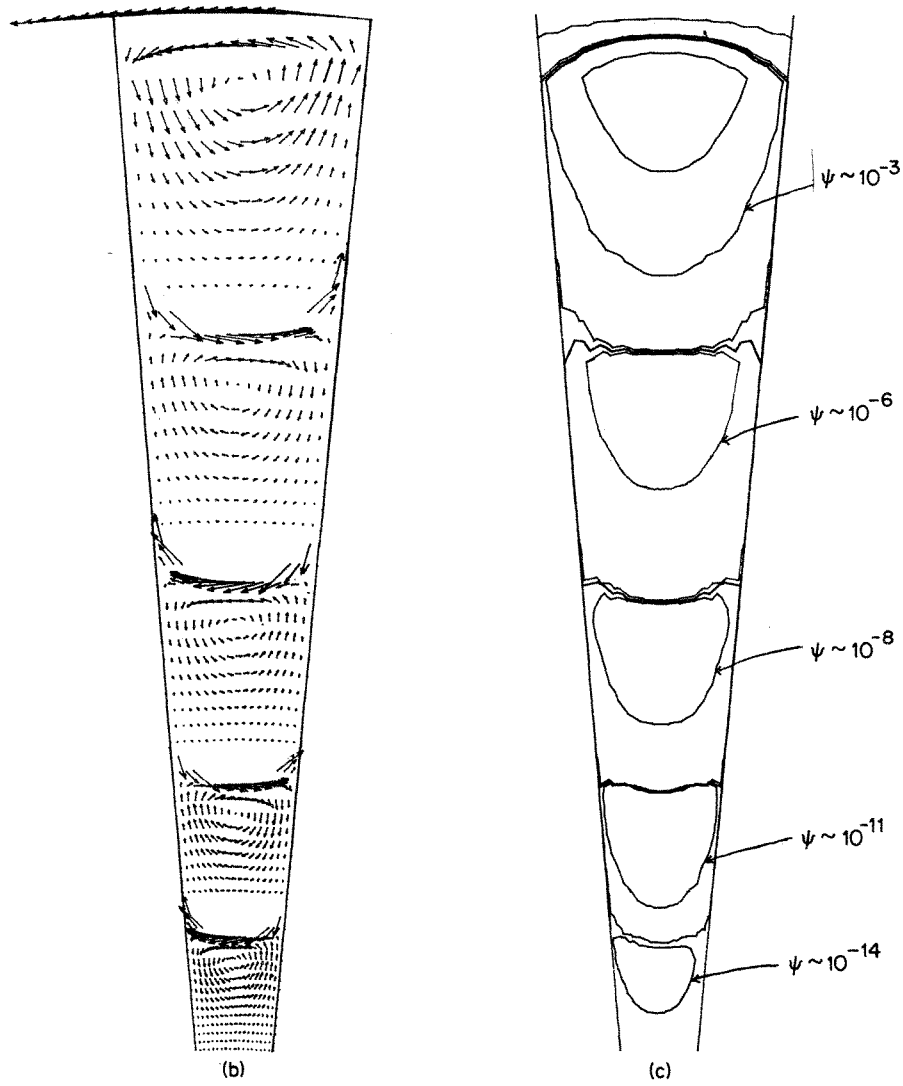

Figure 4. (*Continued*) (b) Velocity vectors; (c) Streamlines

computed normals for different values of ϑ are compared at $r = 1/4$. Note that except for the corner nodes at $\vartheta = 0°$ and $\vartheta = 30°$, the computed normal is on average less than 0·5 per cent in error and that the corner nodes, which use the normal of the nearest Gauss point, are only ≈ 1 per cent in error.

The second simulation, using the same element and method as the first, was of lid-driven flow in a wedge with a base angle of 10°—see Figure 4(a). The solution of the Stokes version of this flow is an infinite series of vortices decreasing in size and magnitude down the wedge. An accurate solution to this problem using streamfunction-vorticity variables and generated by a biorthogonal series technique, can be found in Sanders *et al.*[9] Using this approximate solution we chose the streamline below the 5th eddy as the lower boundary of the region to be modelled.

The exact analytic tangential driving velocities were applied along the lid ($r = 1$) and at the lower streamline boundary the condition $u_n = 0$ was applied. The mesh used is shown in

Figure 4(a) and velocity vector and streamline plots of the solution in Figures 4(b), (c) respectively. Since the maximum velocities in the 5 vortices were of the order $10^{-2}, 10^{-6}, 10^{-9}, 10^{-12}, 10^{-14}$ respectively, each vortex was scaled and plotted separately and then superimposed on a single plot. The streamline values associated with the different vortices compare very favourably with those presented in Reference 9. The definition between the largest and smallest vortices is quite remarkable when the differences in velocity magnitude are considered and it is recalled that a penalty formulation was used. It may be that the successive vortices are largely 'decoupled' from each other even in the matrix, thus permitting good accuracy in each vortex separately.

The final example we present is the Stokes flow resulting from a cylinder rotating with unit tangential velocity within a unit square cavity—$\mathbf{u}$ being zero everywhere on the boundary of the cavity. The radius of the cylinder was 0·15, and the centre of the cylinder was positioned at the point (0·7, 0·6). A nine-node biquadratic velocity–continuous bilinear pressure element using a mixed formulation was employed for the simulation; the results are presented in Figures 5(a)–(d). Figures 5(a), (b), (c) are the computational mesh, velocity vectors and

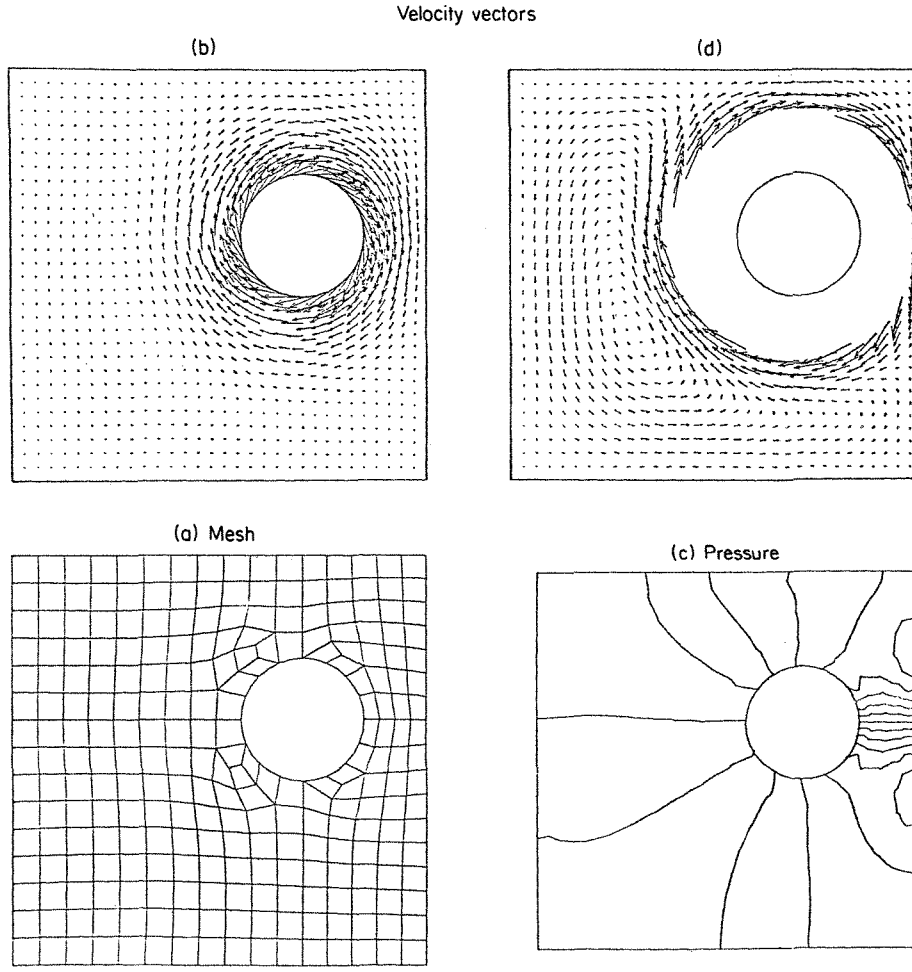

Figure 5. Flow driven by rotating cylinder

Table III. Exact and computed normals for cylinder flow

Angle	Computed		Exact	
	n_x	n_y	n_x	n_y
0·00	−1·0	−0·0	−1·0	−0·0
11·25	−0·9807858	−0·1950876	−0·9807853	−0·1950903
22·50	−0·9238720	−0·3827017	−0·9238795	−0·3826834
33·75	−0·8314663	−0·5555752	−0·8314696	−0·5555702
45·00	−0·7071068	−0·7071068	−0·7071068	−0·7071068
56·25	−0·5555752	−0·8314663	−0·5555702	−0·8314696
67·50	−0·3827017	−0·9238720	−0·3826834	−0·9238795
78·75	−0·1950876	−0·9807858	−0·1950903	−0·9807853
90·00	−0·0	−1·0	−0·0	−1·0

pressure contours respectively, for the flow; Figure 5(d) is a blow up of Figure 5(a) in order to display the eddy structure of the flow—those vectors within a certain radius of the cylinder have not been plotted.

In order to verify the theoretical prediction of the effect of using inconsistent normals, this flow was recomputed using the exact u_x and u_y velocity components (exact to 14 decimal places) of the unit tangential velocity at the boundary nodes of the cylinder. This is equivalent to using a unit tangential velocity and the exact analytic normals rather than the consistent normals for the nodes on the cylinder; Table III presents the exact and computed consistent normals for these nodes on the boundary of one quadrant of the cylinder. The error is seen to be in the sixth or seventh decimal place in all cases, suggesting that the value of ε in equation (17) will be of the order of 10^{-7}. The analysis of Section 3 thus predicts that the pressure solution will be of the order of $\varepsilon/r^e \sim 10^7$. The computed pressures were in fact centred around 22,140 and acceptable (probably) here because the mass inconsistency, ε, is very small. This reflects the (slight) non-satisfaction of the constraint equation associated with the hydrostatic pressure mode. A coarser mesh would cause much larger errors if any but the consistent normals were employed.

6. CONCLUDING REMARKS

The technique for implementing normal or tangential velocities in finite element codes for fluid flows which we have presented (or reviewed, for those who are already familiar with it) is quite general and, we believe, free of some of the inherent difficulties of other methods. Its implementation into existing FEM codes can be achieved in a relatively straightforward manner.

Perhaps more importantly, we have described a method for generating normal and tangential directions in 2-D and 3-D which are consistent with the incompressibility constraint and demonstrated that the use of normal vectors computed by other means can result in erroneous solutions, especially in the pressure.

ACKNOWLEDGEMENTS

MSE and RLS would like to acknowledge support from the U.S. Army Research Office (Grant DAAG29–79–6–0045) and the National Science Foundation (Grant ATM7923354).

REFERENCES

1. H. Saito and L. E. Scriven, 'Study of coating flow by the finite element method', *Proc. Meeting Am. Inst. Chem. Eng.*, Chicago (1980).
2. G. Pinder and W. Gray, *Finite Element Simulation in Surface and Subsurface Hydrology*. Academic Press, 1977, p. 283.
3. W. J. Silliman, 'Viscous film flows with contact lines', *Ph.D. Thesis*, Univ. of Minnesota.
4. K.-J. Bathe, 'Adina—A finite element program for automatic incremental nonlinear analysis', *Report 82448–1*, Acoustics and Vibration Laboratory, Mech. Eng. Dept., M.I.T., December (1978).
5. R. L. Sani, P. M. Gresho, R. L. Lee and D. F. Griffiths, 'The cause and cure (?) of the spurious pressures generated by certain FEM solutions of the incompressible Navier–Stokes equations: Part 1', *Int. j. numer. methods fluids*, **1,** 17–43 (1981).
6. W. G. Gray, 'An efficient finite element scheme for two-dimensional surface water computation', in *Finite Elements in Water Resources* (Ed. W. Gray, G. Pinder and G. Brebbia), Pentech Press, 1977, p. 4.33.
7. M. S. Engelman, *FIDAP Theoretical and Users Manuals*, 1981.
8. D. K. Gartling, R. E. Nickell and R. I. Tanner, 'A finite element convergence study for accelerating flow problems', *Int. j. numer. methods eng.*, **11,** 1155 1174 (1977).
9. J. Sanders, V. Obrien and D. D. Joseph, 'Stokes flow in a driven sector by two different methods', *Trans. ASME*, **47,** 482 (1980).