JOURNAL OF COMPUTATIONAL PHYSICS 53, 152–172 (1984)

A Finite Difference Galerkin Formulation for the Incompressible Navier–Stokes Equations*

A. B. STEPHENS, J. B. BELL,† J. M. SOLOMON, AND L. B. HACKERMAN

Applied Mathematics Branch, Code R44,
Naval Surface Weapons Center, Silver Spring, Maryland 20910

Received September 8, 1982; revised July 1, 1983

The development of a new computational method for solving the incompressible Navier–Stokes equations in primitive variable form is presented. It is found that certain finite difference approximations for these equations can be transformed into an equivalent system which efficiently determines the discrete velocity field and which completely eliminates the pressure. Two such difference schemes for two dimensional problems are examined and some preliminary numerical results are discussed for the steady driven cavity problem.

1. INTRODUCTION

Of the numerous numerical solutions of the incompressible Navier–Stokes equations that have been reported, most are for two-dimensional laminar problems using the vorticity-stream function formulation. For the two-dimensional case, numerical methods based on this formulation have apparently been regarded as more efficient then those based on the primitive variable form of the equations which have the velocity components and pressure as dependent variables. Numerical methods using primitive variables, however, have been developed and are considered important primarily because this form of the equations provides, at least in principle, a relatively straightforward extension to three dimensions and to turbulent flows. The stream function-vorticity formulation, on the other hand, represents a differentiated form of the equations which for certain models of turbulent flow leads to second derivatives of the rather crudely modeled eddy viscosity. Furthermore, in three dimensions, the vorticity is three dimensional; thus its use as a dependent variable offers no inherent advantage.

In this paper, we consider finite difference methods for the primitive variable form of the equations. The main problem generally associated with the development of such methods is to find workable and efficient algorithms for determining the discrete

* This work was supported jointly by the Naval Sea Systems Command, the Naval Surface Weapons Center Independent Research Fund and the Office of Naval Research.

† Present address: Exxon Production Research, P. O. Box 2189, Houston, Texas 77001.

0021-9991/84 $3.00

pressure field in a manner consistent with the discrete divergence-free velocity field. To accomplish this in many of the existing finite difference methods, the pressure is coupled to the velocity by means of some modified form of the differential equations. For example, in the MAC method of Harlow and Welch [9] a Poisson equation for pressure, obtained by taking the divergence of the momentum equations, is used. In Dodge's method [5], a different Poisson equation is used for a scalar related to pressure. Another technique, often used in unsteady methods where only the asymptotic steady state is desired, is to introduce an appropriate artificial time derivative of pressure which of course vanishes in the steady state limit; see, e.g., Chorin [3]. From the above works, it is apparent that the pressure depends on the velocity in a complicated manner (often involving a differentiated form of the equations). Standard methods which directly treat pressure in this way are limited by this complicated dependency. However, the mathematical theory suggests a much weaker dependency of the velocity on the pressure. In fact, Temam [15] and Ladyzhenskaya [11] in considering the weak formulation of the incompressible Navier–Stokes equations show that for certain problems the pressure can be completely eliminated from the determination of the velocity field. Indeed, if the divergence condition is regarded as a constraint on the velocity field and the subspace of divergence-free vectors subject to appropriate boundary conditions is considered, then the momentum equations when projected into this subspace via an appropriate inner product provide a system of Galerkin equations which determine the velocity field and in which the pressure does not appear. We note that in [4], Chorin uses an iterative technique to determine pressure and velocity which is, in effect, a numerical construction of the appropriate projection operator. In [4], as with most other finite difference methods using primitive variables, the pressure is still coupled to the velocity in an essential way and thus must be determined along with the velocity at each step of the calculation. In principle, a more efficient computational method would result if the pressure were completely eliminated from the determination of the velocity field as suggested by the theory in [11, 15].

This decoupling of the velocity and pressure has been achieved in various finite element methods (we refer the reader to the books by Temam [15] and Thomasset [16] for recent surveys). Two approaches have been used. In one, the Galerkin methods of [11, 15] are applied to vector function spaces consisting of piecewise polynomials which are divergence-free (or "weakly" divergent-free). The other approach, referred to as penalty methods, is an alternative to satisfying the discrete divergence-free constraint. In these methods, the divergence condition is perturbed by the additions of a term involving a "small" parameter multiplying the pressure. This perturbed divergence condition is then used to eliminate the pressure in each element. The penalty method idea has also been considered in the framework of finite difference methods; e.g., see Temam [14].

The decoupling of pressure from the velocity in certain finite difference methods while satisfying a discrete divergence condition has been achieved recently by Amit, Hall, and Porsching [1] using an ingenious application of network theory. In [1], network theory is applied to the difference scheme proposed by Krzhivitski and

Ladyzhenskaya [10] and a scheme using the discrete divergence and gradient operators of the MAC method [9]. When these schemes are reformulated in terms of network variables, a computationally equivalent method results for which the pressure variable is eliminated.

In this paper, we consider a different computational method for finite difference approximations to the primitive variable form of the Navier–Stokes equations which also decouples pressure from the velocity while satisfying the discrete divergence condition. The approach used here is, in the context of finite difference approximations, the discrete analog of the Galerkin techniques used in [11, 15]. The methodology used here is therefore quite different from that of the network approach, although later we will indicate how the two methods are related for a few specific difference schemes. Conceptually, the present method is similar to the finite element Galerkin methods using weakly divergence-free subspaces; however, there is an essential difference. Indeed, the present method applies a Galerkin technique directly to the discrete finite difference approximating equations using certain subspaces of *mesh vectors* and *mesh scalars* (i.e., vectors and scalars defined only at node points of some finite difference mesh); whereas, the finite element Galerkin approach uses various subspaces of piecewise polynomials in the nondiscretized weak form of the equations. By design, the present approach is intended for convenient application to various finite difference approximations for the primitive variable equations in order to obtain an equivalent system, referred to here as the *discrete Galerkin formulation*, in which the discrete pressure unknowns do not appear.

The present method has been used in the work of Krzhivitski and Ladyzhenskaya [10] where a specific finite difference scheme was considered. In [10], the discrete Galerkin formulation was used solely as a device for obtaining a convergence proof for the proposed finite difference scheme—the computational advantages of the formulation were not mentioned. Temam [15] has also formulated a finite difference method in a Galerkin framework similar to that used here. This work deals primarily with the theoretical aspects of the formulation and, thus, does not consider in any detail its computational implementation. The finite difference scheme considered by Temam uses the same forward–backward difference approximations for the divergence and gradient operators as those in the Krzhivitski–Ladyzhenskaya scheme. We also note that discrete divergence-free subspaces based on forward differences have been considered by Cea [2] in his study of certain linear elliptic problems with a divergence-free constraint.

In this paper, the discrete Galerkin formulation is developed so that a general class of finite difference schemes can be considered and which focuses on the various computational aspects of the formulation needed for its implementation. We find that, in general, finite difference methods which admit an equivalent discrete Galerkin formulation are those for which the discrete divergence and gradient operators applied to certain spaces of mesh vectors and scalars are formally adjoint. (This condition is a discrete analog of integration by parts for ordinary functions.) Apart from this requirement, the discrete Galerkin formulation retains much of the flexibility of usual finite difference methods in that there is essentially no restriction on

the way the convective and diffusive terms are differenced. This is in contrast to finite element Galerkin approximations where the discretization of these terms is generally dictated by the choice of the approximating function spaces. For certain discrete divergence-gradient pairs, the subspace of discrete divergence-free mesh vectors is relatively easy to construct. As a result, the discrete Galerkin method associated with these schemes is computationally very efficient. In contrast, the divergence free subspaces associated with finite element methods using conforming piecewise polynomials are more difficult to construct. The equivalence of certain finite difference schemes to their discrete Galerkin formulation also provides a theoretical foundation for analyzing the existence, uniqueness, and convergence of solutions to these schemes. This topic will be pursued in future work.

In this paper, we shall concentrate on the conceptional and computational aspects of the discrete Galerkin approach. We begin, in the next section, with a description of the formulation of the method in rather general terms. Then, in Section 3, two finite difference schemes for the divergence-gradient operator pair in the two-dimensional case that admit discrete Galerkin formulations are examined and the relationship with their network interpretations are briefly discussed. One of these schemes is the forward–backward operators considered in [1, 5, 10]. In Section 4, we indicate how general boundary conforming mesh generating transformations can be implemented in the approach. To illustrate the utility of the method, example calculations for the steady driven cavity problem are described in Section 5. Finally, in the last section, some concluding remarks are presented.

2. The Discrete Galerkin Approach

In this section, we describe, in abstract terms, the basic formulation of the discrete Galerkin approach for fairly general problems. In the following sections, details of the implementation for specific two-dimensional examples will be given.

We consider the incompressible Navier–Stokes equations in a bounded region Ω with boundary $\partial\Omega$. The problem, in the primitive variable form, is to find a velocity field $\mathbf{v}$ and a pressure field p satisfying

$$N\mathbf{v} \equiv -\nu\Delta\mathbf{v} + (\mathbf{v}\cdot\nabla)\,\mathbf{v} = -\nabla p + \mathbf{F}(\mathbf{v}) \quad \text{in } \Omega, \tag{2.1}$$

$$\nabla\cdot\mathbf{v} = 0 \quad \text{in } \Omega, \tag{2.2}$$

$$\mathbf{v} = \boldsymbol{\beta} \quad \text{on } \partial\Omega, \tag{2.3}$$

where ∇ and $\nabla\cdot$ are the gradient and divergence operators, respectively, and Δ is the Laplacian. The specified boundary data $\boldsymbol{\beta}$ must satisfy

$$\int_{\Omega} \nabla\cdot\mathbf{v} = \int_{\partial\Omega} \boldsymbol{\beta}\cdot\mathbf{n} = 0, \tag{2.4}$$

where $\mathbf{n}$ is the normal to $\partial\Omega$. For simplicity, we have taken the operator N corresponding to laminar flow with unit density. For the present discussion, however, more complex forms of the incompressible Navier–Stokes equations can be considered; in fact, we only require that N and $\mathbf{F}$ be independent of pressure. Thus, for example, other forms of N corresponding to certain turbulence models where ν depends on the velocity gradient tensor can be considered. In addition, the analysis remains valid when $\mathbf{F}$ contains temporal derivatives of $\mathbf{v}$ for treating unsteady problems.

We first give a discrete form of (2.1)–(2.3) based on finite difference approximations. For this purpose, we consider a mesh covering of Ω such that the interior nodes $\Omega_h \subset \Omega$ and the boundary nodes $\partial\Omega_h$ are on (or very close to) $\partial\Omega$. On this mesh, we define mesh vectors $\mathbf{u}^h$. We also need to introduce the spaces

$$V_h = \{\mathbf{u}^h \text{ defined on } \Omega_h \cup \partial\Omega_h\},$$
$$V_h^0 = \{\mathbf{u}^h \mid \mathbf{u}^h \in V_h, \mathbf{u}^h = 0 \text{ on } \partial\Omega_h\},$$
$$V_h^1 = \{\mathbf{u}^h \text{ defined on } \Omega_h\}.$$

The spaces V_h^0 and V_h^1 with restriction and extension to zero, respectively, are isomorphic and, thus, can be used interchangeably. Let $\tilde{\Omega}_h$ be a mesh covering of Ω (and, possibly, portions of $\partial\Omega$) on which mesh scalars are defined and let W_h be the space of mesh scalars defined on $\tilde{\Omega}_h$.

Finally, let $\mathscr{G}_h$, $\mathscr{D}_h$, N_h, and $\mathbf{F}_h$ denote discrete finite difference operators approximating ∇, $\nabla\cdot$, N, and $\mathbf{F}$, respectively. These operators are defined such that

$$\mathscr{D}_h : V_h \to W_h, \qquad \mathscr{G}_h : W_h \to V_h^1 \tag{2.5}$$

and

$$N_h : V_h \to V_h^1, \qquad \mathbf{F}_h : V_h \to V_h^1.$$

The difference equations approximating (2.1)–(2.3) are then given by

$$N_h \mathbf{u}^h = -\mathscr{G}_h p^h + \mathbf{F}_h(\mathbf{u}^h) \qquad \text{on} \quad \Omega_h, \tag{2.6}$$
$$\mathscr{D}_h \mathbf{u}^h = 0 \qquad \text{on} \quad \tilde{\Omega}_h, \tag{2.7}$$
$$\mathbf{u}^h = \boldsymbol{\beta}^h \qquad \text{on} \quad \partial\Omega_h, \tag{2.8}$$

where $\boldsymbol{\beta}^h$ is an appropriate discretization of the data $\boldsymbol{\beta}$.

We now show that a discrete Galerkin formulation can be obtained which is equivalent to (2.6)–(2.8) provided that $\mathscr{D}_h$ and $\mathscr{G}_h$ satisfy

$$(\mathscr{D}_h \mathbf{u}^h, \phi^h)_{W_h} = (\mathbf{u}^h, -\mathscr{G}_h \phi^h)_{V_h^0}, \qquad \mathbf{u}^h \in V_h^0, \tag{2.9}$$

where $(\cdot,\cdot)_{V_h^0}$ and $(\cdot,\cdot)_{W_h}$ are inner products for V_h^0 and W_h, respectively. Note that (2.9) is a discrete analogue of integration by parts for ordinary functions with L^2

inner products. However, not all finite difference schemes for $\mathscr{D}_h$ and $\mathscr{G}_h$ satisfy (2.9). The condition (2.9) implies that $\mathscr{D}_h$ and $-\mathscr{G}_h$ are adjoint and thus (see, e.g., [8, p. 48]), we have the following decomposition of V_h^0:

THEOREM 2.1. $V_h^0 = D_h \oplus G_h$, *where*

$$D_h = \{\mathbf{u}^h \in V_h^0 \mid \mathscr{D}_h \mathbf{u}^h = 0\},$$
$$G_h = \{\mathbf{u}^h \in V_h^0 \mid \exists \phi^h \in W_h \ni \mathbf{u}^h = \mathscr{G}_h \phi^h\}.$$

We note that this result is, in some sense, a finite dimensional analogue of a well known decomposition for L^2 vector fields (see, e.g., [11, p. 28]). A decomposition similar to that of Theorem 2.1 was used by Chorin [4] in defining an iterative scheme for the unsteady problem with periodic boundary conditions.

The decomposition of V_h^0 leads directly to a discrete Galerkin approximation. Indeed, let $\{\mathbf{\Phi}_1^h, \mathbf{\Phi}_2^h, \ldots, \mathbf{\Phi}_m^h\}$ be a basis for D_h and let $\mathbf{a}^h \in V_h$ satisfy $\mathscr{D}_h \mathbf{a}^h = 0$ on $\tilde{\Omega}_h$ with $\mathbf{a}^h = \boldsymbol{\beta}^h$ on $\partial\Omega_h$. Then the discrete Galerkin approximation, $\mathbf{w}^h = \sum_{i=1}^m \alpha_i \mathbf{\Phi}_i^h$, is defined as the solution of

$$(N_h(\mathbf{w}^h + \mathbf{a}^h), \mathbf{\Phi}_i^h)_{V_h^0} = (\mathbf{F}_h(\mathbf{w}^h + \mathbf{a}^h), \mathbf{\Phi}_i^h)_{V_h^0}, \qquad i = 1, \ldots, m. \tag{2.10}$$

The system (2.10) represents m (scalar) equations for the m (scalar) coefficients α_i and is equivalent to the finite difference system (2.6)–(2.8) in the following sense:

THEOREM 2.2. *If* $\mathbf{u}^h$, p^h *satisfies* (2.6)–(2.8), *then* $\mathbf{w}^h = \mathbf{u}^h - \mathbf{a}^h$ *satisfies* (2.10). *Conversely, if* $\mathbf{w}^h$ *satisfies* (2.10), *then there exists a* $p^h \in W_h$ *such that* $(\mathbf{w}^h + \mathbf{a}^h)$, p^h *satisfies* (2.6)–(2.8)

Proof. Suppose $\mathbf{u}^h$, p^h satisfies (2.6)–(2.8). Then $\mathbf{w}^h = \mathbf{u}^h - \mathbf{a}^h \in V_h^0$ and $\mathbf{w}^h \in D_h$. Also $N_h(\mathbf{w}^h + \mathbf{a}^h) = N_h \mathbf{u}^h = -\mathscr{G}_h p^h + \mathbf{F}_h(\mathbf{w}^h + \mathbf{a}^h)$ which implies that $N_h(\mathbf{w}^h + \mathbf{a}^h) - (\mathbf{F}_h(\mathbf{w}^h + \mathbf{a}^h) \in G_h$ which implies that $\mathbf{w}^h$ satisfies (2.10). For the converse, suppose $\mathbf{w}^h$ satisfies (2.10). Then $\mathbf{u}^h = \mathbf{w}^h + \mathbf{a}^h$ satisfies $\mathscr{D}_h \mathbf{u}^h = 0$ on $\tilde{\Omega}_h$ with $\mathbf{u}^h = \mathbf{a}^h = \boldsymbol{\beta}^h$ on $\partial\Omega_h$. Also, $N_h \mathbf{u}^h - \mathbf{F}_h(\mathbf{u}^h) \in D_h^{\perp} = G_h$. Hence there exists a $p^h \in W_h$ such that $-N_h \mathbf{u}^h + F_h(\mathbf{u}^h) = \mathscr{G}_h p^h$.

In order to obtain a computational algorithm based on the discrete Galerkin approach embodied in (2.10), one must have a convenient means of constructing the basis mesh vectors $\{\mathbf{\Phi}_i^h;\ i = 1, \ldots, m\}$ and the mesh vector $\mathbf{a}^h$ which annihilates the boundary data. The construction of these quantities will be discussed in detail in the following section where we give examples of specific schemes for $\mathscr{D}_h$, $\mathscr{G}_h$ in two dimensions for which the associated $\{\mathbf{\Phi}_i^h\}$ are obtained in closed form and $m = \dim(D_h) = O(M)$, where M is the number of nodes of Ω_h. In such cases, (2.10) not only uncouples the pressure from the velocity but also determines the velocity in terms of $O(M)$ scalars. This represents a sizeable reduction in variables over (2.6)–(2.8) which involves $O(3M)$ coupled unknowns for two-dimensional flow. The discrete Galerkin approach presented here is similar in spirit to the work of Amit *et*

al. [1] who use network theory to transform the two-dimensional primitive variable equations to a system involving $O(M)$ network variables in which pressure is eliminated. More discussion of the connection between the discrete Galerkin and network theory approaches will be given in the following section.

Since N_h is nonlinear, (2.10) represents a system of nonlinear equations for the coefficients α_i. To numerically solve this system, we have used in our example calculations an iterative technique based on Newton's method with continuation in Reynolds number $R = LU_0/\nu$, where L and U_0 are characteristic length and velocity scales, respectively. Details of this procedure will be given in Section 5. More sophisticated continuation and quasi-Newton iteration (see, e.g., [5, 11]) would probably be more efficient but will not be considered here. Also, other iterative strategies could be used; for example, the discrete operators N_h and $\mathbf{F}_h$ could be taken as appropriate linearized approximations to N and $\mathbf{F}$.

After the solution $\mathbf{w}^h$ of (2.10) has been determined, the pressure p^h can, if desired, be obtained in a separate calculation using the linear equations

$$\mathscr{G}_h p^h = \mathbf{F}_h(\mathbf{w}^h + \mathbf{a}^h) - N_h(\mathbf{w}^h + \mathbf{a}^h), \tag{2.11}$$

where the right-hand side is now known. The linear system (2.11) is consistent but will, in general, be singular since $\dim(\ker \mathscr{G}_h) \geqslant 1$. (Recall that a consistent finite difference approximation for ∇p will be zero for a constant mesh scalar.) Thus to obtain p^h uniquely either a minimum norm least squares procedure can be used or the system (2.11) must be augmented with constraints.

3. Particular Choices of $\mathscr{L}_h$, $\mathscr{G}_h$

In this section we consider two different choices of the operators $\mathscr{L}_h$ and $\mathscr{G}_h$ for use in applications of the discrete Galerkin method to two-dimensional planar flows. For simplicity, in this section, we assume Ω is the unit square and we consider meshes with uniform spacing. In the next section, we discuss the implementation of non-uniform meshes on more general domains using mesh generating transformations.

3.1. *Scheme* I

On the unit square Ω, we define the meshes

$$\begin{aligned}
\Omega_h &= \{(ih, jh) \mid i = 1,..., n; j = 1,..., n\},\\
\partial\Omega_h &= \{(ih, jh) \mid i = 0,..., n+1 \text{ when } j = 0 \text{ and } n+1;\\
&\qquad j = 0,..., n+1 \text{ when } i = 0 \text{ and } n+1\},\\
\tilde{\Omega}_h &= \{((i+\tfrac{1}{2})h, (j+\tfrac{1}{2})h) \mid i = 0,..., n; j = 0,..., n\},
\end{aligned}$$

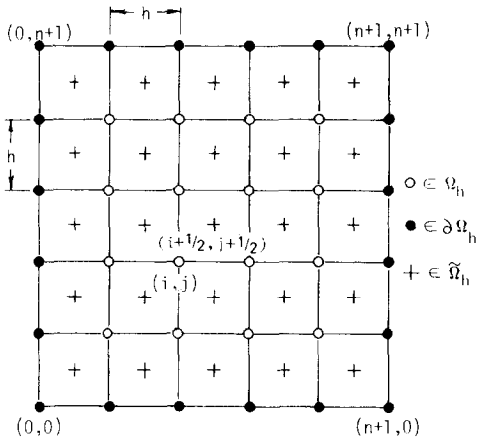

FIG. 1. Meshes for Scheme I.

where $h = 1/(n+1)$. Note that the mesh nodes of $\tilde{\Omega}_h$ are defined to be the centers of the computational cells induced by Ω_h, see Fig. 1. For these meshes, the spaces V_h, V_h^0, V_h^1, and W_h defined in Section 2 have dimensions given by

(i) $\dim V_h = 2(n+2)^2$,

(ii) $\dim V_h^0 = \dim V_h^1 = 2n^2$,

(iii) $\dim W_h = (n+1)^2$.

The operators $\mathscr{D}_h$ and $\mathscr{G}_h$ are defined on V_h and W_h, respectively, as

$$(\mathscr{D}_h \mathbf{u}^h)_{i+1/2,j+1/2} = \frac{1}{2h}\left|(u_{i+1,j+1} - u_{i,j+1}) + (u_{i+1,j} - u_{ij}) + (v_{i+1,j+1} - v_{i+1,j}) + (v_{i,j+1} - v_{i,j})\right| \tag{3.1}$$

where $\mathbf{u}^h = \binom{u}{v} \in V_h$ and

$$(\mathscr{G}_h \phi)_{i,j} = \begin{pmatrix} \frac{1}{2h}\left|(\phi_{i+(1/2),j+(1/2)} - \phi_{i-(1/2),j+(1/2)}) + (\phi_{i+(1/2),j-(1/2)} - \phi_{i-(1/2),j-(1/2)})\right| \\ \frac{1}{2h}\left|(\phi_{i+(1/2),j+(1/2)} - \phi_{i+(1/2),j-(1/2)}) + (\phi_{i-(1/2),j+(1/2)} - \phi_{i-(1/2),j-(1/2)})\right| \end{pmatrix}. \tag{3.2}$$

These definitions obviously satisfy (2.5). We note that the staggered grid system and the above operators have previously been proposed for standard finite difference calculations, see [7]. As pointed out by one of the reviewers, these operators can also be obtained from a finite element expansion using bi-linear velocity and piecewise constant pressure on square elements; e.g., see Sani *et al.* [12].

On V_h^0 and W_h, we consider the standard l^2 inner products

$$(\mathbf{u}, \mathbf{v})_{V_h^0} = \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbf{u}_{ij} \cdot \mathbf{v}_{ij}$$

and

$$(\phi, \psi)_{W_h} = \sum_{i=0}^{n} \sum_{j=0}^{n} \phi_{i+(1/2),j+(1/2)} \psi_{i+(1/2),j+(1/2)}.$$

It is then straightforward to show that (2.9) is satisfied and hence we have the decomposition of V_h^0 given in Theorem 2.1. We now show

Remark 3.1. (iv) $\dim \ker \mathcal{G}_h = 2$,

(v) $\dim G_h = (n+1)^2 - 2$,

(vi) $\dim D_h = (n-1)^2$.

Proof. (iv) implies (v) since $\dim G_h = \text{rank } \mathcal{G}_h$, $\dim W_h = \dim \ker \mathcal{G}_h + \text{rank } \mathcal{G}_h$, and (iii). From Theorem 2.1 and (ii), it follows that (v) implies (vi). To prove (iv), we show that

$$\begin{aligned} \psi^1_{i+(1/2),j+(1/2)} &= 1, \qquad i+j \text{ even}, \\ &= 0, \qquad i+j \text{ odd}, \end{aligned}$$

and

$$\begin{aligned} \psi^2_{i+(1/2),j+(1/2)} &= 0, \qquad i+j \text{ even}, \\ &= 1, \qquad i+j \text{ odd}, \end{aligned}$$

span ker $\mathcal{G}_h$. It is obvious that ψ^1 and ψ^2 are independent and belong to ker $\mathcal{G}_h$. Now let $\gamma \in \ker \mathcal{G}_h$ and consider $\phi = \gamma - \gamma_{(1/2),(1/2)} \psi^1 - \gamma_{(1/2),3/2} \psi^2$. Clearly $\phi \in \ker \mathcal{G}_h$ and $\phi_{(1/2),(1/2)} = \phi_{(1/2),3/2} = 0$. But then, $(\mathcal{G}_h \phi)_{1,1} = 0$ implies that $\phi_{3/2,(1/2)} = \phi_{3/2,3/2} = 0$ and by an induction argument we see that $\phi \equiv 0$. Thus, $\gamma = \gamma_{(1/2),(1/2)} \psi^1 + \gamma_{1/2,3/2} \psi^2$ which proves that ψ^1, ψ^2 span ker $\mathcal{G}_h$.

We now show

Remark 3.2. The mesh vectors defined by

$$\begin{aligned} \Phi_{ij}^{l+(1/2),m+(1/2)} &= (1,-1)^t, && i=l, j=m, \\ &= (1,1)^t, && i=l+1, j=m, \\ &= (-1,1)^t, && i=l+1, j=m+1, \\ &= (-1,-1)^t, && i=l, \quad j=m+1, \\ &= (0,0)^t, && \text{all other } i,j, \end{aligned}$$

$$l = 1,\ldots, n-1; \quad m = 1,\ldots, n-1, \qquad (3.3)$$

are a basis for D_h. Here the index $l+(1/2)$, $m+(1/2)$ corresponds to the center of the cell that forms the support of the particular mesh vector.

Proof. For each l, m it can be directly verified that $\mathscr{D}_h \Phi^{l+(1/2),m+(1/2)} = 0$ on $\tilde{\Omega}_h$. To see that the $(n-1)^2$ mesh vectors are independent, we note that each mesh vector is a translation of the same basic mesh vector. For a formal proof of independence, it suffices to examine the associated Gram–Schmidt matrix. A computation shows that this matrix is irreducibly diagonally dominant with positive diagonal entries and non-positive off-diagonal entries. Therefore it is a M-matrix and hence non-singular.

Consider now the determination of boundary mesh vectors $\mathbf{a}^h \in V_h$ satisfying (2.7) and (2.8) for this scheme. In order for such a mesh vector to exist, the discrete boundary values $\boldsymbol{\beta}^h$ must satisfy certain conditions analogous to (2.4). Indeed, we note that (2.7) implies that

$$\sum_{\substack{i,j=0 \\ (i+j,\text{even})}}^{n} 2h(\mathscr{D}_h \mathbf{a}^h)_{i+(1/2),j+(1/2)} = \sum_{\substack{i,j=0 \\ (i+j,\text{odd})}}^{n} 2h((\mathscr{D}_h \mathbf{a}^h)_{i+(1/2),j+(1/2)} = 0$$

and, because of the telescoping nature of the summands, each sum involves only the components of the boundary mesh vector. Let $\boldsymbol{\beta}_{ij}^h = (u_{ij}, v_{ij})^t$, then adding and subtracting these sums gives

$$-\left[\tfrac{1}{2}(u_{0,0}+u_{0,n+1}) + \sum_{\mu=1}^{n} u_{0,\mu}\right] + \left[\tfrac{1}{2}(u_{n+1,0}+u_{n+1,n+1}) + \sum_{\mu=1}^{n} u_{n+1,\mu}\right]$$
$$-\left[\tfrac{1}{2}(v_{0,0}+v_{n+1,0}) + \sum_{\mu=1}^{n} v_{\mu,0}\right] + \left[\tfrac{1}{2}(v_{0,n+1}+v_{n+1,n+1}) + \sum_{\mu=1}^{n} v_{\mu,n+1}\right] = 0 \tag{3.4}$$

and

$$\sum_{\mu=0}^{n} (-1)^{\mu}(u_{\mu+1,0}-u_{\mu,0}) + (-1)^n \sum_{\mu=0}^{n} (-1)^{\mu}(u_{\mu+1,n+1}-u_{\mu,n+1})$$
$$+ \sum_{\mu=0}^{n} (-1)^{\mu}(v_{0,\mu+1}-v_{0,\mu}) + (-1)^n \sum_{\mu=0}^{n} (-1)^{\mu}(v_{n+1,\mu+1}-v_{n+1,\mu}) = 0. \tag{3.5}$$

Condition (3.4) involves only normal components of $\boldsymbol{\beta}^h$ and in fact, represents the trapezoidal rule approximation to (2.4) (for the unit square Ω). On the other hand, condition (3.5) involves only tangential components of $\boldsymbol{\beta}^h$. A condition similar to (3.5) has also been given for the finite element method of Sani *et al.* [12]. In problems involving smooth data, $\boldsymbol{\beta}$, it may be necessary to make small $O(h^2)$ adjustments in the discretization of $\boldsymbol{\beta}$ in order to satisfy (3.4) and (3.5).

Conditions (3.4) and (3.5) are also sufficient to ensure the existence of a mesh vector $\mathbf{a}^h$ satisfying (2.7) and (2.8). To see this, we seek $\mathbf{a}^h$ in the form

$$\mathbf{a}^h = \sum_{l=-1}^{n+2} \sum_{m=-1}^{n+2} \alpha^{l+(1/2),m+(1/2)} \mathbf{\Phi}^{l+(1/2),m+(1/2)}, \tag{3.6}$$

where the mesh vectors $\mathbf{\Phi}^{l+(1/2),m+(1/2)}$ are as defined in (3.3) with the understanding that those mesh vectors with support outside of $\Omega_h \cup \partial\Omega_h$ are restricted to $\partial\Omega_h$. Clearly, $\mathbf{a}^h \in V_h$ and satisfies (2.7). If $\mathbf{a}^h$ satisfies (2.8), then it follows from (3.3) and (3.6) that

$$\begin{aligned}\boldsymbol{\beta}_{ij}^h = (u_{ij}, v_{ij})^t = {} & \alpha^{i-(1/2),j-(1/2)}(-1, 1)^t + \alpha^{i-(1/2),j+(1/2)}(1, 1)^t \\ & + \alpha^{i+(1/2),j+(1/2)}(1, -1)^t + \alpha^{i+(1/2),j-(1/2)}(-1, -1)^t\end{aligned} \tag{3.7}$$

or

$$\alpha^{i-(1/2),j+(1/2)} - \alpha^{i+(1/2),j-(1/2)} = (u_{i,j} + v_{i,j})/2 \tag{3.8}$$

and

$$\alpha^{i+(1/2),j+(1/2)} - \alpha^{i-(1/2),j-(1/2)} = (u_{i,j} - v_{i,j})/2, \tag{3.9}$$

where the i, j indices correspond to the nodes of $\partial\Omega_h$ and, of course, the components of $\boldsymbol{\beta}_{ij}^h$ are given. It can be shown that (3.8) and (3.9) can be solved for the $\alpha^{i+(1/2),j+(1/2)}$ corresponding to the centers of cells bordering $\partial\Omega_h$ on both sides if and only if $\boldsymbol{\beta}^h$ satisfies (3.4) and (3.5). These $\alpha^{i+(1/2),j+(1/2)}$ are determined along two distinct zig-zag paths around $\partial\Omega_h$ as illustrated in Fig. 2. One $\alpha^{i+(1/2),j+(1/2)}$ on each path is arbitrary. Also, all other coefficients in (3.6) (i.e., $l = 1,..., n$: $m = 1,..., n$) are arbitrary and for simplicity, can be taken to be zero.

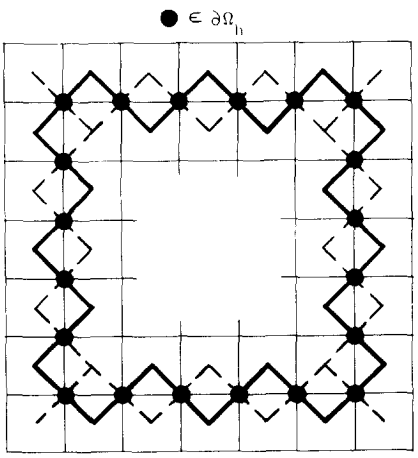

FIG. 2. Paths for the solution of Eqs. (3.8) and (3.9).

3.2. *Scheme* II

As a second choice of the operators $\mathscr{D}_h$, $\mathscr{G}_h$, we consider the operators used in the time dependent scheme of Krzhivitski and Ladyzhenskaya [10] and also considered in [1] using network theory. For this scheme, the mesh Ω_h is the same as for Scheme I; however, $\partial\Omega_h$ and $\tilde{\Omega}_h$ are different. In particular we define

$$\partial\Omega_h = \{(ih, jh) \mid i = 1,..., n+1 \text{ when } j = 0 \text{ and } n+1;\\ j = 1,..., n+1 \text{ when } i = 0 \text{ and } n+1\}$$

and

$$\tilde{\Omega}_h = \Omega_h \cup \Gamma_h,$$

where

$$\Gamma_h = \{(ih, jh) \mid i = 1,..., n \text{ when } j = 0; j = 1,..., n \text{ when } i = 0\}.$$

The spaces V_h, V_h^0, V_h^1, and W_h are as defined in Section 2 for these meshes. The operators $\mathscr{D}_h : V_h \to W_h$ and $\mathscr{G}_h : W_h \to V_h^1 \sim V_h^0$ are defined, respectively, as

$$(\mathscr{D}_h \mathbf{u}^h)_{i,j} = \frac{1}{h}[(u_{i+1,j} - u_{i,j}) + (v_{i,j+1} - v_{i,j})], \tag{3.10}$$

where $\mathbf{u}^h = (u, v)^t \in V_h$ and

$$(\mathscr{G}_h \phi)_{i,j} = \begin{pmatrix} \frac{1}{h}(\phi_{i,j} - \phi_{i\ \ 1,j}) \\ \frac{1}{h}(\phi_{i,j} - \phi_{i,j-1}) \end{pmatrix}. \tag{3.11}$$

On V_h^0 and W_h, we define the standard l^2 inner products as before. Then, it can be directly verified that $\mathscr{D}_h$, $\mathscr{G}_h$ defined by (3.10) and (3.11) satisfy (2.9) and hence the decomposition of V_h^0 given in Theorem 2.1 is valid.

For this scheme, we have that $\dim V_h = 2(n+2)^2 - 2$, $\dim V_h^0 = \dim V_h^1 = 2n^2$, and $\dim W_h = (n+1)^2 - 1$. Further, it is easy to see that $\mathscr{G}_h \phi = 0$ on Ω_h if and only if $\phi =$ constant on $\tilde{\Omega}_h$; hence, $\dim \ker \mathscr{G}_h = 1$. As in the previous example, it can be shown that the subspaces defined in Theorem 2.1 have $\dim D_h = (n-1)^2$ and $\dim G_h = (n+1)^2 - 2$. A basis for D_h is given by

$$\begin{aligned} \Phi_{i,j}^{l+1/2,m+1/2} &= (0,1)^t, && i = l, j = m+1, \\ &= (-1,0)^t, && i = l+1, j = m, \\ &= (1,-1)^t, && i = l+1, j = m+1, \\ &= (0,0)^t, && \text{all other } i,j, \quad l = 1,..., n-1, \quad m = 1,..., n-1. \end{aligned} \tag{3.12}$$

Note that the $(n-1)^2$ mesh vectors defined by (3.12) are translations of the same basic mesh vector.

For this scheme, a necessary and sufficient condition for the existence of mesh vectors $\mathbf{a}^h$ satisfying (2.7) and (2.8) is

$$v_{0,1} + u_{1,0} + \sum_{\mu=1}^{n} (v_{\mu,0} + u_{0,\mu}) - \sum_{\mu=0}^{n} (v_{\mu,n+1} + u_{n+1,\mu}) = 0, \tag{3.13}$$

where $\boldsymbol{\beta}^h = (u, v)^t$. This condition arises from the summation of $(\mathscr{D}_h \mathbf{a}^h)_{i,j} = 0$ over all nodes of $\tilde{\Omega}_h$. When $\boldsymbol{\beta}^h$ satisfies (3.13), $\mathbf{a}^h$ can be constructed in a similar manner as given for Scheme I; viz. we use the representation (3.6) for $\mathbf{a}^h$ but with $\{\boldsymbol{\Phi}^{l+(1/2),m+(1/2)}\}$ defined as in (3.12) and with $\alpha^{-(1/2),-(1/2)} = \alpha^{n+3/2,n+3/2} = 0$. As for the other coefficients $\{\alpha^{l+(1/2),m+(1/2)}\}$ corresponding to the centers of cells bordering the boundary on both sides, one is arbitrary and the others can be determined in terms of the given data β^h using

$$\begin{aligned} \beta_{ij}^h = (u_{ij}, v_{ij})^t = {} & \alpha^{i+(1/2),j-(1/2)}(0,1)^t + \alpha^{i-(1/2),j-(1/2)}(1,-1)^t \\ & + \alpha^{i-(1/2),j+(1/2)}(-1,0)^t \qquad (i,j \neq 0,0). \end{aligned} \tag{3.14}$$

Note that, for this scheme, $\partial\Omega_h$ does not include the node $(0, 0)$. Thus no data is given at $(0, 0)$ and $\mathbf{a}^h$ is not defined at $(0, 0)$. In applications where the discretization of N in the momentum equations requires the velocity at node $(0, 0)$, it will be necessary to include this node in the definition of $\mathbf{a}^h$. In such cases the above procedure is slightly modified. The data at $(0, 0)$ must satisfy $v_{0,1} + u_{1,0} - v_{0,0} - u_{0,0} = 0$ and the coefficient $\alpha^{-(1/2),-(1/2)}$ must be included in (3.6) and is determined using (3.14) with $(i, j) = (0, 0)$.

3.3. *Connection with Network Theory*

There is an interesting relationship between the discrete Galerkin formulation and the network theory approach of Amit *et al.* [1]. Indeed, following [1], both Schemes I and II (and probably others) can be interpreted as a network consisting of a system of nodes with directed connecting links. The incidence matrix $\mathbf{A}$ of the network and its negative transpose $-\mathbf{A}^t$ are representations of the discrete divergence and gradient operators $\mathscr{D}_h$, $\mathscr{G}_h$, respectively. Further, the fundamental matrix $\mathbf{C}$ associated with the network and formed from elementary cycles has the property that $\mathbf{AC} = \mathbf{0}$. Thus, the columns of $\mathbf{C}$ form a basis for the space D_h.

Although Scheme I was not considered in [1], it can be interpreted as a network consisting of two disjoint, connected components. For each component, the pressures are the "states" at the nodes which are isomorphic to mesh points of $\tilde{\Omega}_h$ with $i+j$ of the same parity and the appropriate velocities are the "flows" on the connecting links; cf., see Fig. 3. The disjoint character of this network is indicative of the fact that the null space of $\mathscr{G}_h$ has dimension 2 for this scheme.

The network interpretation of Scheme II has been given in [1] and need not be discussed here. Also given in [1] is an application of network theory to a finite

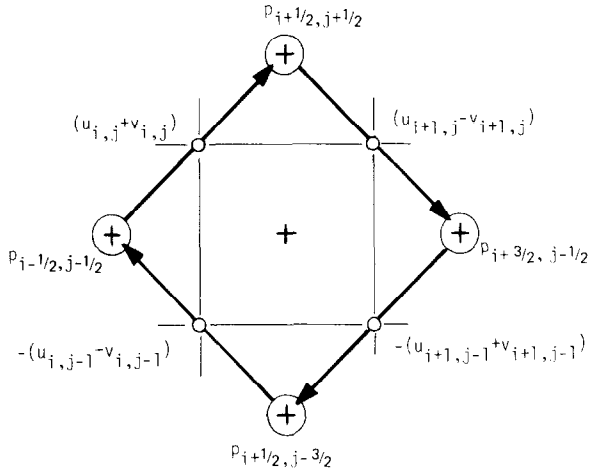

FIG. 3. Molecule for each component of the Scheme I network.

difference method utilizing the discrete divergence and gradient operators given by Harlow and Welch [9] for the MAC method. For this scheme, $\mathscr{D}_h$ and $\mathscr{G}_h$ are defined essentially the same as Scheme II but with the understanding that (u_{ij}, v_{ij}) represents $(u_{i-(1/2),j}, v_{i,j-(1/2)})$. Although not considered in detail here, we note that the operators $\mathscr{D}_h$, $\mathscr{G}_h$ so defined can be used in a discrete Galerkin approach similar to that associated with Scheme II. The space V_h^0 is different at certain boundaries than that of Scheme II but the basis mesh vectors for the subspace D_h are of the same form as (3.12) (with the above interpretation).

For all schemes $\mathscr{D}_h$, $\mathscr{G}_h$ considered here both the discrete Galerkin and the network approach represent the discrete velocity field as certain linear combinations of basis mesh vectors or elements of a fundamental matrix both of which are known in closed form. Thus for both approaches the discrete velocity will satisfy the discrete divergence condition (2.7) to within machine round-off error in actual calculations. Also the coefficients determined by both approaches, when interpreted as discrete scalar fields (with appropriate scale factors), correspond to a discrete stream function. To see this for the discrete Galerkin approach, consider the representation of the discrete velocity components in terms of the coefficients $\{\alpha\}$ given by (3.7) or (3.14) for any mesh point.

4. MESH TRANSFORMATIONS

The schemes for $\mathscr{D}_h$, $\mathscr{G}_h$ given in the previous section for the unit square with uniform meshes can be extended to more general domains and nonuniform meshes using boundary conforming, mesh generating transformations. We briefly sketch here

how this can be accomplished for domains Ω which are topologically equivalent to the unit square.

Let

$$\mathscr{E} : x = x(\xi, \eta), \qquad y = y(\xi, \eta)$$

be a sufficiently smooth, one-to-one mapping of the unit square $\hat{\Omega}$ onto the given domain Ω with sufficiently smooth inverse, $\mathscr{E}^{-1}$. The transformation $\mathscr{E}$ can be given in closed form or can be approximated numerically using a mesh generating procedure; e.g., see [17]. The ξ, η plane is where computations are performed and, thus, all quantities are considered as functions of (ξ, η). Accordingly, the divergence operator can be expressed in the ξ, η plane using the identity

$$\nabla \cdot \mathbf{u} = (1/J)(\hat{u}_\xi + \hat{v}_\eta), \tag{4.1}$$

where

$$J = \det(\mathbf{T}) \neq 0 \qquad \text{on} \quad \hat{\Omega} \cup \partial\hat{\Omega}, \mathbf{T} = \begin{pmatrix} y_\eta & -x_\eta \\ -y_\xi & x_\xi \end{pmatrix},$$

and $\hat{u}$, $\hat{v}$ are the components of the *transformed velocity vector* defined by

$$\hat{\mathbf{u}} = (\hat{u}, \hat{v})^t = \mathbf{T}\mathbf{u} = \mathbf{T}(u, v)^t.$$

The transformed components of $\nabla\phi$ (ϕ a scalar) are given by

$$\mathrm{T}\nabla\phi = (1/J)\,\mathbf{T}\mathbf{T}^t(\phi_\xi, \phi_\eta)^t \tag{4.2}$$

and the transformed components of the momentum equation (2.1) can be expressed in the form

$$\mathbf{T}|\hat{N}\mathbf{u} - \mathbf{F}(\mathbf{u})| = \frac{-1}{J}\,\mathbf{T}\mathbf{T}^t(p_\xi, p_\eta)^t, \tag{4.3}$$

where $\hat{N}$ denotes the operator N with x, y derivatives expressed in terms of ξ, η derivatives. $\hat{N}$ will therefore contain elements of $\mathbf{T}$ and their partial derivatives.

In the ξ, η-plane, we define on $\hat{\Omega}$ the meshes Ω_h, $\partial\Omega_h$ and $\tilde{\Omega}_h$ as in Section 3 and let $\hat{\mathbf{u}}$ denote the transformed vector mesh function, $\hat{\mathbf{u}}^h = \mathbf{T}_h\mathbf{u}^h$, where $\mathbf{T}_h$ is the restriction of $\mathbf{T}$ to $\Omega_h \cup \partial\Omega_h$. Further, let $\hat{V}_h$, $\hat{V}_h^0$, $\hat{V}_h^1$ be the spaces of transformed vector mesh functions, $\hat{\mathbf{u}}^h$, defined analogously to the spaces V_h, V_h^0, V_h^1 introduced in Section 2. On $\hat{V}_h^0$, we consider the inner product

$$(\hat{\mathbf{u}}^h, \hat{\mathbf{v}}^h)_{\hat{V}_h^0} = \sum_{\Omega_h} \{J(\mathbf{T}_h^{-1}\hat{\mathbf{u}}^h) \cdot (\mathbf{T}_h^{-1}\hat{\mathbf{v}}^h)\} = \sum_{\Omega_h} \{J\hat{\mathbf{u}}^h \cdot |(\mathbf{T}_n^t)^{-1}\,\mathbf{T}_h^{-1}\hat{\mathbf{v}}^h|\},$$

where the sum is over all nodes of Ω_h. The space of scalar mesh functions, $\hat{W}_h$, is the same as W_h defined in Section 2 but with the inner product

$$(\phi^h, \psi^h)_{\hat{W}_h} = \sum_{\tilde{\Omega}_h} \{J\phi^h\psi^h\},$$

where the sum is over all nodes of $\tilde{\Omega}_h$. Discrete divergence, $\hat{\mathscr{D}}_h : \hat{V}_h \to \hat{W}_h$, and gradient $\hat{\mathscr{G}}_h : \hat{W}_h \to \hat{V}_h^1$ operators corresponding to the right-hand sides of (4.1) and (4.2), respectively, are defined by

$$\hat{\mathscr{D}}_h \hat{\mathbf{u}}^h = \frac{1}{J} \mathscr{D}_h \hat{\mathbf{u}}^h \qquad \text{(for nodes of } \tilde{\Omega}_h\text{)} \tag{4.4}$$

and

$$\hat{\mathscr{G}}_h \phi^h = \frac{1}{J} \mathbf{T}_h \mathbf{T}_h^t (\mathscr{G}_h \phi^h) \qquad \text{(for nodes of } \Omega_h\text{)}, \tag{4.5}$$

where the discrete operators $\mathscr{D}_h$, $\mathscr{G}_h$ are the same as those given in Section 3. Since $\mathscr{D}_h$, $\mathscr{G}_h$ satisfy (2.9) (for the inner products considered in Section 3), we have that

$$\begin{aligned}(\hat{\mathscr{D}}_h \hat{\mathbf{u}}^h, \phi^h)_{\hat{W}_h} &= \sum_{\Omega_h} (\mathscr{D}_h \hat{\mathbf{u}}^h)\, \phi^h = -\sum_{\Omega_h} \hat{\mathbf{u}}^h \cdot \mathscr{G}_h \phi^h \\ &= (\hat{\mathbf{u}}^h, -\hat{\mathscr{G}}_h \phi^h)_{\hat{V}_h^0}.\end{aligned}$$

Hence, the operators $\hat{\mathscr{D}}_h$, $\hat{\mathscr{G}}_h$ induce the decomposition $\hat{V}_h^0 = \hat{D}_h \oplus \hat{G}_h$ where the subspaces $\hat{D}_h$ and $\hat{G}_h$ are analogous to D_h and G_h defined in Theorem 2.1. Further, the algebraic structure of the subspaces $\hat{D}_h$ and $\hat{G}_h$ are the same as indicated in Section 3 for D_h and G_h. In particular, the basis mesh vectors for D_h given in Section 3 are also a basis for $\hat{D}_h$.

It follows from (4.1)–(4.5) that the finite difference equations approximating (2.1)–(2.3) in the ξ, η-plane are

$$\mathbf{T}_h \hat{N}_h \mathbf{u}^h = -\hat{\mathscr{G}}_h p^h + \mathbf{T}_h \mathbf{F}_h(\mathbf{u}^h) \qquad \text{on} \quad \Omega_h, \tag{4.6}$$

$$\hat{\mathscr{D}}_h(\mathbf{T}_h \mathbf{u}^h) = 0 \qquad \text{on} \quad \tilde{\Omega}_h, \tag{4.7}$$

$$\mathbf{u}^h = \boldsymbol{\beta}^h \qquad \text{on} \quad \partial\Omega_h, \tag{4.8}$$

where N_h and $\mathbf{F}_h$ are discrete finite difference operators approximating N and $\mathbf{F}$ respectively. To obtain a discrete Galerkin formulation which is equivalent to (4.6)–(4.8) in the sense of Theorem 2.2, we first construct a mesh vector $\mathbf{a}^h$ satisfying (4.7) and (4.8). For this purpose, we consider $\hat{\mathbf{a}}^h = \mathbf{T}_h \mathbf{a}^h$ satisfying $\hat{\mathscr{D}}_h \hat{\mathbf{a}}^h = 0$ on Ω_h and $\hat{\mathbf{a}}^h = \mathbf{T}_h \boldsymbol{\beta}^h$ on $\partial\Omega_h$. The construction of $\hat{\mathbf{a}}^h$ is the same as the boundary mesh vectors in Section 3.

Let $\{\boldsymbol{\Phi}_i^h : i = 1,\ldots, m = (n-1)^2\}$ denote the basis mesh functions of D_h defined in Section 3. Since these also form a basis for $\hat{D}_h$, the discrete Galerkin approximation equivalent to (4.6)–(4.8) is defined as $\mathbf{w}^h = \mathbf{T}_h^{-1}(\sum_{k=1}^m \alpha_k \boldsymbol{\Phi}_k^h)$, where $\mathbf{w}^h$ satisfies

$$(\boldsymbol{\Phi}_i^h, \mathbf{T}_h \hat{N}_h(\mathbf{w}^h + \mathbf{a}^h))_{\hat{V}_h^0} = (\boldsymbol{\Phi}_i^h, \mathbf{T}_h \mathbf{F}_h(\mathbf{w}^h + \mathbf{a}^h))_{\hat{V}_h^0}, \qquad i = 1,\ldots, m.$$

We note that with only slight modification the above can be used to treat problems with cylindrical symmetry. Indeed, suppose x is the axial and y the radial coordinates

for describing axially symmetric flow. The divergence condition is then given by $\nabla \cdot u = [(yu)_x + (yv)_y]/y = 0$, where u, v are the axial and radial velocity components, respectively. This case can be treated by introducing appropriate factors of y in the definitions of $\hat{\mathbf{u}}$, $\hat{\mathscr{D}}_h$, $\hat{\mathscr{G}}_h$, and the inner products.

5. Applications to the Driven Cavity

To illustrate the application of the discrete Galerkin approach, we consider the steady, laminar flow of unit density in a square cavity with moving top wall. The flow domain and boundary conditions are normalized as indicated in Fig. 4. Thus, the Reynolds number $R = 1/\nu$ is the only parameter. This problem presents a very simple geometry for application of finite difference schemes. However, it also possesses a difficulty in that the u component of velocity is discontinuous at the two upper corner points (cf., Fig. 4). Since our purpose here is only to illustrate, by way of example, an implementation of the discrete Galerkin formulation, we shall not present a systematic numerical study of this problem. Such a study would only emphasize the properties of the particular underlying difference schemes and the particular numerical implementation of the discrete Galerkin formulation—matters which are not the central issue in this work.

For our sample calculations, we have selected two finite difference schemes which employ the operators $\mathscr{D}_h$, $\mathscr{G}_h$ defined in Section 3. One scheme uses the operators $\mathscr{D}_h$, $\mathscr{G}_h$ defined by Scheme I (viz. Eqs. (3.1), (3.2)) with a centered "conservative" difference approximation to N defined by

$$(N_h \mathbf{v}^h)_{ij} \equiv -\frac{1}{Rh^2}(\hat{\delta}_x + \hat{\delta}_y)\,\mathbf{v}_{ij}^h + \frac{1}{h}\delta_x(u_{ij}\mathbf{v}_{ij}^h) + \frac{1}{h}\delta_y(v_{ij}\mathbf{v}_{ij}^h), \tag{5.1}$$

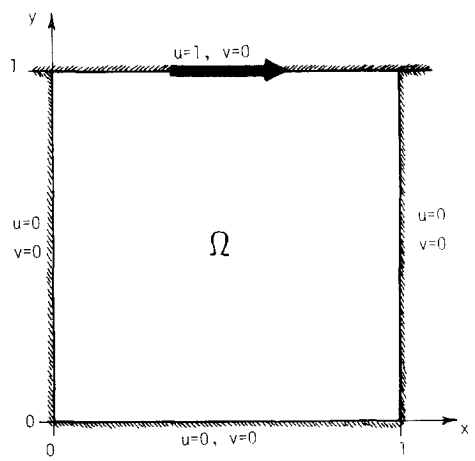

Fig. 4. Driven cavity problem.

where $\mathbf{v}_{ij}^h = (u_{ij}, v_{ij})^t$, $\hat{\delta}_x(\cdot)_{ij} \equiv (\cdot)_{i+1,j} - 2(\cdot)_{ij} + (\cdot)_{i-1,j}$, $\delta_x(\cdot)_{ij} \equiv \frac{1}{2}[(\cdot)_{i+1,j} - (\cdot)_{i-1,j}]$, and $\hat{\delta}_y$, δ_y defined analogously. We will refer to this scheme as Scheme A. The other scheme, which we will refer to as Scheme B, uses $\mathscr{D}_h$, $\mathscr{G}_h$ defined by Scheme II (viz. Eqs. (3.10) and (3.11)) with N approximated by

$$(N_h \mathbf{v}^h)_{ij} \equiv -\frac{1}{Rh^2}(\hat{\delta}_x + \hat{\delta}_y)\mathbf{v}_{ij}^h + \frac{1}{h}\sigma_x(u_{ij}\nabla_x \mathbf{v}_{ij}^h) + \frac{1}{h}\sigma_y(v_{ij}\nabla_y \mathbf{v}_{ij}^h), \tag{5.2}$$

where $\sigma_x(\cdot)_{ij} \equiv \frac{1}{2}[(\cdot)_{i+1,j} + (\cdot)_{ij}]$, $\nabla_x \equiv (\cdot)_{ij} - (\cdot)_{i-1,j}$, and σ_y, ∇_y defined analogously. We note that Scheme B is essentially the nonlinear version for steady flow of the linearized unsteady scheme proposed in [10].

In the driven cavity problem, the discontinuity of u at the upper corners creates an ambiguity in the specification of the discrete boundary data $\boldsymbol{\beta}^h$. Recall that $\boldsymbol{\beta}^h$ enters directly in the discrete Galerkin formulation through the construction of a mesh vector $\mathbf{a}^h$ satisfying (2.7) and (2.8). In our example calculations, we have take $u = 1$ at both upper corner points. This leads to the rather simple choice of $\mathbf{a}^h$ given by

$$\begin{aligned} \mathbf{a}^h &= (1, 0)^t \qquad i = 0,\ldots, n+1;\ j = n+1, \\ &= (0, 0)^t \qquad \text{all other points of } \Omega_h \cup \partial\Omega_h, \end{aligned} \tag{5.3}$$

which satisfies (2.7) and (2.8) for both Schemes A and B. Other choices of u at the upper corner points could be considered; however, care must be taken to ensure that the discrete boundary data satisfy the necessary and sufficient conditions given in Section 3 for the existence of an $\mathbf{a}^h$ satisfying (2.7) and (2.8).

For both the schemes considered here, the discrete Galerkin equations (2.10) represent $(n-1)^2$ nonlinear equations for the unknown coefficients $\boldsymbol{\alpha} = \{\alpha_i : i = 1,\ldots, (n-1)^2\}$. These equations are considered in the form $\mathbf{H}(\boldsymbol{\alpha}; R) = 0$ and solved using Newton's method with continuation in Reynolds number. That is, numerical solutions are obtained for an increasing sequence of Reynolds numbers R starting with $R = 0$ (Stokes flow). For each (fixed) R, $\mathbf{H}(\boldsymbol{\alpha}; R) = 0$ is solved iteratively using

$$[\mathbf{H}_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}^k; R)](\boldsymbol{\alpha}^{k+1} - \boldsymbol{\alpha}^k) = -\mathbf{H}(\boldsymbol{\alpha}^k, R), \qquad k = 0, 1, 2,\ldots, \tag{5.4}$$

where $\boldsymbol{\alpha}^k$ is the kth iterate and $\mathbf{H}_{\boldsymbol{\alpha}}$ is the associated Jacobian matrix. For each R, the iteration is started with $\boldsymbol{\alpha}^0$ taken as the "converged" solution of (5.4) for the previous value of R (for $R = 0$, $\boldsymbol{\alpha}^0$ is taken equal to those which determine $\mathbf{a}^h$). As previously indicated, this procedure is but one of several possible iterative strategies compatible with the discrete Galerkin formulation (for others see, e.g., [6, 13]). To indicate the amount of computational work required for each iteration of (5.4), we note that for both Schemes A and B the $\{\alpha_i\}$ can be ordered so that the matrix $\mathbf{H}_{\boldsymbol{\alpha}}$ is banded with a

band width of $4n + 1$. In fact, $\mathbf{H}_\alpha$ is similar in structure to the matrix arising from the usual differencing of the biharmonic operator.

For the numerical results presented here, the Jacobian matrix $\mathbf{H}_\alpha$ was evaluated numerically (to order of the square root of machine precision) by replacing the derivatives of $\mathbf{H}$ with respect to the components of $\boldsymbol{\alpha}$ by appropriate first order forward differences. The linear system (5.4) was solved using an in core banded solver. The iterations were considered "converged" when the absolute value of all components of $\boldsymbol{\alpha}^{k+1} - \boldsymbol{\alpha}^k$ were less then 10^{-5}. In the continuation process, the increments of Reynolds number were adjusted to obtain convergence within 10 iterations.

Computations have been performed with both schemes using two grid sizes corresponding to $n = 19$ and $n = 39$ for Reynolds numbers to 10,000. Calculations were made at high Reynolds numbers with the purpose of indicating the robustness of the numerical method rather than giving a detailed picture of the flow field. An accurate resolution of boundary layers, corner eddies, etc., would require more mesh points, or mesh clustering and will be a topic of a later work. All calculations were performed on the CRAY-1 at AFWL (Albuquerque, N.M.). No attempt was made to optimize our program for this machine. Each iteration required 0.29 sec of CPU time for $n = 19$ and 2.4 sec of CPU time for $n = 39$. Roughly half of this time was used in solving the banded system (0.13 sec and 1.69 sec for $n = 19$ and 39, respectively). For the $n = 19$ grid, the continuation in Reynolds number for both schemes consisted of $R = 0$, 100, 300, 500, 1,000, 5,000, 10,000. For the $n = 39$ grid, smaller steps in R were needed beyond $R = 1000$ to maintain convergence within 10 iterations for scheme A and beyond $R = 5000$ for Scheme B. Throughout the range of Reynolds numbers considered, no oscillations in converged results were observed for either scheme on either mesh. Typical computer plots of the discrete velocity fields and the streamlines (which are in fact level curves of the scalar field corresponding to $\{\alpha^{i+(1/2),j+(1/2)}\}$) are shown in Figs. 5 and 6.

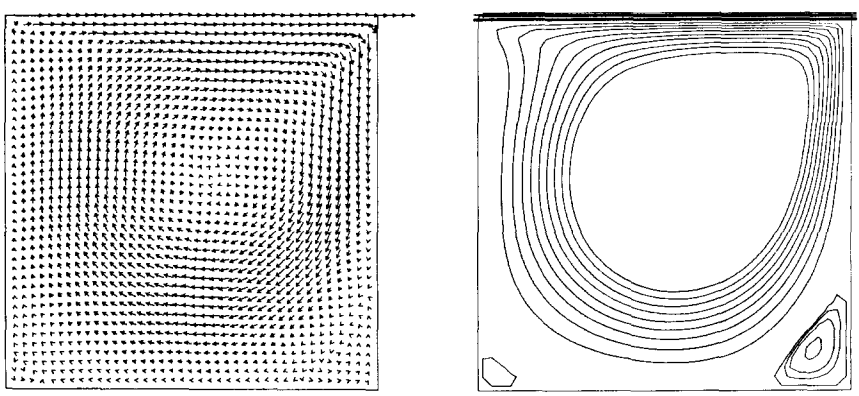

FIG. 5. Velocity field and streamlines for $R = 500$ (Scheme A with $n = 39$).

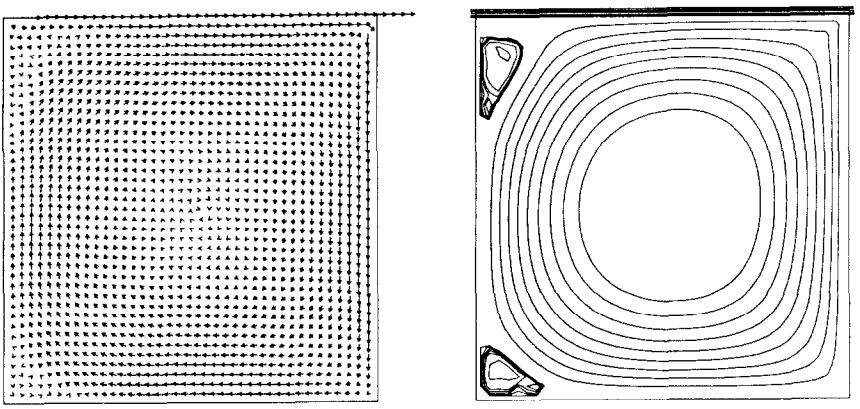

FIG. 6. Velocity field and streamlines for $R = 10{,}000$ (Scheme A with $n = 39$).

6. Concluding Remarks

In this paper we have shown how certain finite difference schemes for the two-dimensional steady incompressible Navier–Stokes equations in primitive variable form with Dirichlet boundary conditions may be reformulated as a system of discrete Galerkin equations. We have also indicated how this reformulation is related to the network theory approach of [1]. The solution of the reformulated equations is computationally efficient since the number of variables is greatly reduced.

In later work we plan to extend this approach to the cases of more complicated boundary conditions, unsteady flows, three-dimensional flows, and turbulence modeling. The reduction of finite difference equations to a discrete Galerkin setting is also useful for examining theoretical topics such as existence, uniqueness, and convergence. These topics will also be explored in a later work.

Acknowledgments

The authors would like to thank Drs. Alan Berger and John Wingate of the Naval Surface Weapons Laboratory for some stimulating and useful discussions concerning this work.

References

1. R. Amit, C. A. Hall, and T. A. Porsching, An application of network theory to the solution of implicit Navier–Stokes difference equations, *J. Comput. Phys.* **40** (1981), 183–201.
2. J. Cea, Approximation variationelle des problems aux limites, *Ann. Inst. Fourier (Grenoble)* **14** (1964), 345–444.
3. A. J. Chorin, A numerical method for solving incompressible viscous flow problems, *J. Comput. Phys.* **2** (1967), 12–26.

4. A. J. CHORIN, Numerical solution of the Navier–Stokes equations, *Math. Comput.* **22** (1968), 745–762.
5. P. R. DODGE, Numerical method for 2D and 3D viscous flows, *AIAA J.* **15** (1977), 961–965.
6. M. S. ENGLEMAN, G. STRANG, AND K. J. BATHE, The application of quasi-Newton methods in fluid dynamics, *Intl. J. Numer. Methods Eng.* **17** (1981), 707–718.
7. M. FORTIN, Numerical solution of steady state Navier–Stokes equations, *in* "Numerical Methods in Fluid Dynamics" (J. J. Smolderen, Ed.), AGARD–LS–48, 1972.
8. B. FRIEDMAN, "Principles and Techniques of Applied Mathematics," Wiley, New York, 1956.
9. F. H. HARLOW AND J. E. WELCH, Numerical calculation of time dependent viscous incompressible flow of fluid with free surface, *Phys. Fluids* **8** (1965), 2182–2189.
10. A. KRZHIVITSKI AND O. A. LADYZHENSKAYA, The method of nets for the nonstationary Navier–Stokes equations, *in* "Boundary Value Problems of Math. Phys.," Amer. Math. Soc. Transl., pp. 105–112, 1968; *Proc. Math. Inst. Steklov* **92** (1966).
11. O. A. LADYZHENSKAYA, "The Mathematical Theory of Viscous Incompressible Flow," Gordon & Breach, New York, 1963.
12. R. SANI, P. GRESHO, R. LEE, D. GRIFFITHS, AND M. ENGLEMAN, The cause and cure(?) of the spurious pressure generated by certain FEM solutions of the incompressible Navier–Stokes equations, Parts 1 and 2, *Intl. J. Numer. Methods Fluids* **1** (1981), 17–43; 171–204.
13. R. SCHREIBER AND H. B. KELLER, "Spurious Solutions in Driven Cavity Calculations," to appear.
14. R. TEMAM, Une method d'approximation de la solution des equations de Navier–Stokes, *Bull. Soc. France* **28** (1968), 115–152.
15. R. TEMAM, "Navier–Stokes Equations: Theory and Numerical Analysis," North-Holland, Amsterdam, 1977.
16. F. THOMASSET, "Implementation of Finite Element Methods for Navier–Stokes Equations," Springer-Verlag, New York, 1981.
17. J. F. THOMPSON, F. C. THAMES, AND C. W. MASTIN, Automatic numerical generation of body-fitted curvilinear coordinate systems for fields containing any number of arbitrary two-dimensional bodies, *J. Comput. Phys.* **15** (1974), 299–319.