INT. J. NUM. METH. HEAT FLUID FLOW, VOL 2, 483–502 (1992)

SOLUTION OF INCOMPRESSIBLE NAVIER–STOKES EQUATIONS ON UNSTRUCTURED GRIDS USING DUAL TESSELLATIONS

J. C. CAVENDISH

Mathematics Department, General Motors Research Laboratories, 30500 Mound Road, Box 9055, Warren, Michigan 48090-9055, USA

C. A. HALL* AND T. A. PORSCHING*

University of Pittsburgh, Pittsburgh, Pennsylvania 15260, USA

ABSTRACT

We describe a novel mathematical approach to deriving and solving covolume models of the incompressible 2-D Navier–Stokes flow equations. The approach integrates three technical components into a single modelling algorithm: 1. *Automatic Grid Generation.* An algorithm is described and used to automatically discretize the flow domain into a Delaunay triangulation and a dual Voronoi polygonal tessellation. 2. *Covolume Finite Difference Equation Generation.* Three covolume discretizations of the Navier–Stokes equations are presented. The first scheme conserves mass over triangular control volumes, the second scheme over polygonal control volumes and the third scheme conserves mass over both. Simple consistent finite difference equations are derived in terms of the primitive variables of velocity and pressure. 3. *Dual Variable Reduction.* A network theoretic technique is used to transform each of the finite difference systems into equivalent systems which are considerably smaller than the original primitive finite difference system.

KEY WORDS Incompressible flow problem Dual tessellations Covolume discretization

INTRODUCTION

One of the oldest methods for numerically approximating boundary value problems is that of integrating the defining differential equations over control volumes associated with structured or unstructured decompositions of the spatial domain. For example, nearly 40 years ago, MacNeal[12] derived finite difference models for elliptic problems (electric potential problems) by integrating the continuum equations over *Delaunay triangles* and dual *Voronoi tiles.* Interestingly, rather than solve these equations on a digital computer, he used this approach to calculate ohmic resistance values for use in an analog computer composed of a simple linear electric circuit. Since then, a variety of terms have been coined to signify this approach to discretization of boundary value problems[2,13,14] including the *complementary volume* or *covolume method*[10,11]. Discretizations of fluid flow problems based on networks using dual unstructured partitions of the flow domain were introduced by Porsching[16] and Chou[6].

A new covolume discretization approach to modelling two-dimensional incompressible Navier–Stokes flow equations was investigated[10,11,13,14]. That approach produced finite

* Research supported by the Electric Power Research Institute, Grant No. RP-8006-24.

0961–5539/92/060483–20$2.00

Received December 1991
Revised June 1992

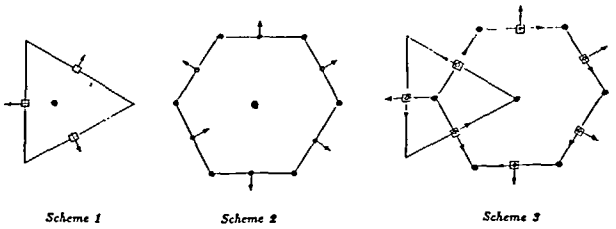


Figure 1 The placement of primitive variables. ●, Pressure; □, velocity normal to triangle side; ○, velocity to normal polygon side

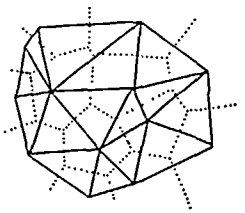


Figure 2 Voronoi tesselation (.....) and Delaunay triangulation (––) of 13 points

difference equations by spatially integrating the continuity equation over each triangle in a Delaunay triangulation of the flow domain, applying Green's theorem to replace area integrals with boundary line integrals, and finally approximating each boundary line integral by an expression involving *scalar* velocity components normal to the midpoints of triangle sides [see *Figure 1* (scheme 1)].

The scheme also made use of spatial integrations over dual Voronoi polygons to generate finite difference discretizations of the momentum equations involving the same scalar velocity components as well as unknown pressure variables positioned at triangle circumcentres. As a result of this approach, the vector valued transient Navier–Stokes equations were replaced by a continuous time, discrete space finite difference approximation in terms of pressures at triangle circumcentres and scalar velocity components in the direction normal to the midpoint of each triangle side.

Collectively, the scalar velocity components and pressure variables are called *primitive* variables and for a Delaunay triangulation involving N_V triangle vertices, the continuum flow equations were approximated by $O(5N_V)$ differential/algebraic equations in a like number of unknowns. Also described[10,11] was a dual variable approach based on elementary network theory whereby the original system of $O(5N_V)$ primitive variables is transformed into an equivalent system of only $O(N_V)$ ordinary differential equations. A pivotal element in the approach[10,11] is the approximation of tangential components of flow along the sides of a triangular cell as a linear combination of the normal components of flow. It is this approximation that permits the replacement of the vector velocity field in the Navier–Stokes equations by a scalar normal velocity field.

An alternative approach to that described[10,11,13,14] is to reverse the roles of the Delaunay triangles and the Voronoi polygons when deriving covolume approximations to the flow equations. In this second scheme, mass is conserved by integrating the continuity equations over each Voronoi polygon in the dual Voronoi tessellation. Green's theorem is applied to reduce area integrals to line integrals over polygonal boundaries and then line integrals are approximated by an expression involving scalar velocity components that are normal to the midpoints of polygonal edges [see *Figure 1* (Scheme 2)]. Because of the mutual orthogonality of the Delaunay and Voronoi constructs, these scalar velocities are both normal to polygon edges and tangential to triangle sides. In this scheme we make use of spatial integrations over the Delaunay triangles to derive finite difference discretizations of the momentum equations involving the same velocity components normal to sides of polygons as well as unknown pressure variables positioned now at triangle vertices. For a triangulation of N_V points, this second approach produces an approximation to the Navier–Stokes equations involving $O(4N_V)$ differential/algebraic equations. We can also use results from network theory to transform this primitive system of

$O(4N_V)$ equations into an entirely equivalent system of $O(2N_V)$ ordinary differential equations. As before, the dual variables can be used to recover the original primitive normal velocity and pressure variables. Key to this approach is the ability to construct an approximation of tangential components of flow along the sides of polygonal flow cells by appropriate combinations of scalar normal flow components.

A third approach, and the one emphasized here, is to combine the two covolume approaches outlined above to derive and solve covolume models of the transient incompressible flow equations. In this third scheme, components of velocity normal and tangential to triangle sides (and hence normal to polygonal sides) are treated as independent variables, thereby coupling the equations resulting from the first and second schemes [see *Figure 1* (Scheme 3)]. For a Delaunay/Voronoi tessellation of N_V points, this approach generates a system of $O(9N_V)$ differential/algebraic equations in terms of $O(9N_V)$ primitive variables (tangential and normal flows at midpoints of triangle sides and pressure variables located at triangle vertices and triangle circumcentres). Combining these results derived from network theory applied to the two companion approaches, the original system of $O(9N_V)$ equations is transformed into an equivalent system of only $O(3N_V)$ ordinary differential equations. *Note that this reduced dual variable system is smaller than the primitive systems of either of the first two approaches.*

First, we briefly recall an efficient method for constructing tessellations of mutually orthogonal Delaunay/Voronoi covolumes (dismissed as a rather routine task by MacNeal[12]). Next we describe the particular form of the incompressible Navier–Stokes equations that will be used later to generate continuous time discrete space finite difference equations for the continuum flow problem. In later sections we outline an approach based on elementary network theory whereby the systems of equations involving primitive flow variables are transformed into equivalent systems of dual variables of considerably reduced dimensionality. This is followed by generalizations of the dual mesh approach to other tessellations. Finally, we present conclusions.

AUTOMATIC GENERATION OF DUAL TESSELLATIONS

A geometric construction which has proven useful for generating finite element discretizations of planar or solid regions into well-proportioned triangular or tetrahedral simplicies is the so-called Delaunay triangulation of a collection of points $\{p_i\}_{i=1}^{N_V}$. The triangulation is most easily described when the points are confined to R^2. Let the set $V_i, i = 1, 2, \ldots, N_V$, be defined by:

$$V_i = \{x: \|x - p_i\| < \|x - p_j\| \text{ for all } j \neq i\}$$

where $\|*\|$ denotes Euclidean distance. Because V_i represents a region whose points are nearer to point p_i than to any other point, V_i is an open convex polygon (called a *Voronoi polygon* or *tile*) whose boundary edges are portions of the perpendicular bisectors of the lines joining p_i to p_j when V_i and V_j are contiguous. The collection of Voronoi polygons forms a Voronoi tessellation of R^2. In general, a vertex of a Voronoi polygon is shared by two other neighbouring polygons so that connecting the three generating points associated with such adjacent polygons forms a triangle, say T_k. The set of triangles is called the *Delaunay triangulation* which can also be shown to be a triangulation of the convex hull of the generating points. The Delaunay triangulation and dual Voronoi tessellation is illustrated in *Figure 2* for a set of 13 points. What makes the triangulation popular for finite element analyses of structural problems is that (1) there are very efficient algorithms for computing it[5,8], and (2) it produces triangles that are as close to equilateral as possible for a given set of points in a sense made precise in Reference 8. Although the dual Voronoi tessellation is a construct that can be realized by simply connecting circumcentres of adjacent Delaunay triangles, no use is made of this dual structure in the finite element method

when applied to structural problems. Such is not the case[10,11,13,14] or here where, for incompressible flow discretizations, the two dual tessellations are closely linked.

One of the primary advantages of basing flow discretizations on Delaunay/Voronoi covolumes is the capability of easily defining discrete partitions of the flow domain that can be locally graded or refined, and which are mutually orthogonal in the sense described above. Local grading permits efficient resolution and control of spatial discretization errors while orthogonal control volumes produce small discrete systems (especially when combined with the dual variable method of reducing system dimensionality) and improved treatment of flow boundary conditions. A main objective of the current work has been to link efficient control volume generation with covolume flow discretization and the dual variable method to produce an approach capable of delivering accurate approximations from computational models that are comparatively small in size.

For the purpose of this paper we use a Delaunay triangulator described in detail in Reference 8. That mesh generator simultaneously integrates two tasks: generation of a distribution of well-placed nodes on all user-defined boundaries (both interior and exterior boundaries) and within the interior of the planar flow domain, and construction of a Delaunay triangulation of these nodes suitable for finite volume analysis applications. By ensuring that no interior points are placed inside circumcircles of each boundary edge, the mesh generator produces *boundary conforming* triangulations where boundaries include those of empty holes as well as closed or open interior boundary segments. (Closed internal boundaries differ from hole boundaries in

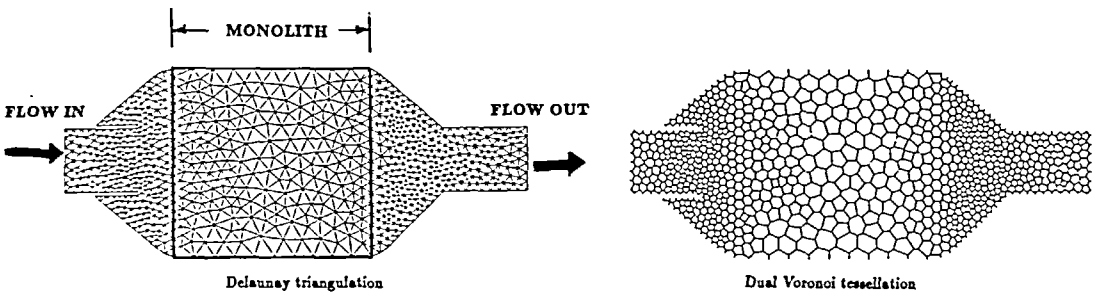


Figure 3 The Delaunay triangulation and Voronoi polygonal tessellation of an automotive converter

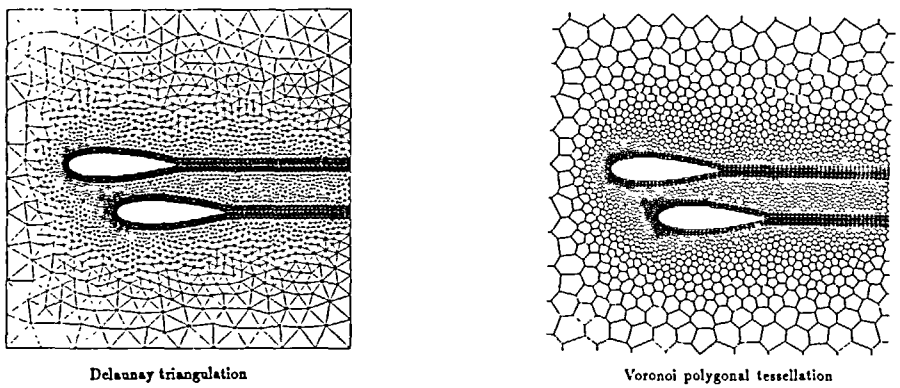


Figure 4 Dual tessellations for two airfoils

that triangles are generated in their interiors.) This variety of internal boundaries permits mesh generation on multiply connected planar domains and it provides a useful means for controlling local element density when modelling boundary conditions or flow obstructions associated with any continuum flow problem. Mesh generation turns on Watson's algorithm[17] for constructing a Delaunay triangulation of a set of points, followed by a post-processing application of Laplacian smoothing. Boundary and interior nodes are automatically inserted into the mesh by making use of an implicitly defined node separating function (see Reference 8 for complete details).

Figure 3 contains dual tessellations generated by the mesh generator[8] for a monolith automotive converter. The monolith was automatically subdivided as a separate subregion so that resistance terms could be added to momentum equations associated with triangles contained in this subregion. For this particular example, the mesh generator produced a total of 1122 triangles from 601 node points. All but 25 of these triangles are acute. The minimum angle in the triangulation is 32 degrees; the maximum angle is 107 degrees. For the purpose of covolume finite difference equation generation, it is desirable—though not essential—that triangles be acute which guarantees that they contain their circumcentres. We are not aware of any mesh generator that assures creation of entirely acute meshes. Extensive experience with the one used in this report indicates that the mesh generator seldom produces triangulations with more than 3% obtuse triangles.

Figure 4 contains the dual tessellations for two airfoils, illustrating the ability to achieve local mesh refinement and also to force such refinements to contain near right triangles in the boundary layers. Right triangles are desirable so that velocity components normal to sides of triangles are either normal or tangential to airfoil surfaces. If desired, the diagonals of the quadrilaterals adjacent to the airfoils could be deleted and a grid containing both triangles and quadrilaterals used for the discretization of the Navier–Stokes problem.

THE INCOMPRESSIBLE FLOW PROBLEM

We consider a bounded polygonal flow domain Ω with boundary $\partial\Omega$. The continuous problem is to find a velocity vector $\mathbf{q} = (u(x, y, t), v(x, y, t))^T$ and a (reduced) pressure $p(x, y, t)$ which satisfy the continuity, or conservation of mass, equation

$$\nabla \cdot \mathbf{q} = 0 \qquad (x, y) \in \Omega, t > 0 \tag{1}$$

and the vector-valued conservation of momentum equation:

$$\frac{\partial \mathbf{q}}{\partial t} - \nu \nabla^2 \mathbf{q} + (\mathbf{q} \cdot \nabla)\mathbf{q} = -\nabla p + \mathbf{F}(\mathbf{x}, t) \tag{2}$$

where $\mathbf{x} = (x, y)$, ν is the kinematic viscosity and $\mathbf{F}(\mathbf{x}, t)$ is a source term. We assume boundary conditions of the form

$$p(\mathbf{x}, t) = p_b(\mathbf{x}) \qquad \mathbf{x} \in \partial\Omega_1, t > 0 \tag{3}$$

and

$$\mathbf{q}(\mathbf{x}, t) = \mathbf{q}_b(\mathbf{x}) \qquad \mathbf{x} \in \partial\Omega_2, t > 0 \tag{4}$$

where $\partial\Omega = \partial\Omega_1 \cup \partial\Omega_2$ and $\partial\Omega_1 \cap \partial\Omega_2 = \phi$. In the case of a pressure specified condition, (3), we also assume a velocity distribution outside of the boundary $\partial\Omega_1$ which is continuative in nature. This assumption allows us to invoke the condition that the velocity field is divergence free in a neighbourhood of $\partial\Omega_1$, a fact which is used in the deviation of (6) below.

Further, we assume initial conditions of the form:

$$\mathbf{q}(\mathbf{x}, 0) = \mathbf{q}_0(\mathbf{x}), p(\mathbf{x}, 0) = p_0(\mathbf{x}) \qquad \mathbf{x} \in \Omega \tag{5}$$

To put (2) in a form more suitable for discretization on a triangular grid, let $\mathbf{n}$ be a constant unit vector (to be chosen later) and take the dot product of (2) with $\mathbf{n}$. For the viscous term, let ω be the scalar vorticity as given in Reference 15. Then, since $\nabla \cdot \mathbf{q} = 0$, it follows that:

$$-(\nu \nabla^2 \mathbf{q}) \cdot \mathbf{n} = \nu \frac{\partial \omega}{\partial \mathbf{s}} \tag{6}$$

where $(\mathbf{n}, \mathbf{s})$ is a right-handed orthogonal coordinate system. The convective term becomes[11]:

$$[(\mathbf{q} \cdot \nabla)\mathbf{q}] \cdot \mathbf{n} = \left[\frac{\partial}{\partial \mathbf{q}}(\mathbf{n} \cdot \mathbf{q})\right] |\mathbf{q}| \tag{7}$$

where $\partial/\partial \mathbf{q}$ denotes the directional derivative in the direction $\mathbf{q}$ and $|\mathbf{q}|$ is its Euclidean length. Equation (2) is thus replaced by the scalar equation:

$$\frac{\partial}{\partial t}(\mathbf{n} \cdot \mathbf{q}) + \nu \frac{\partial \omega}{\partial \mathbf{s}} + \left(\frac{\partial}{\partial \mathbf{q}}(\mathbf{n} \cdot \mathbf{q})\right) |\mathbf{q}| = -\frac{\partial p}{\partial \mathbf{n}} + \mathbf{n} \cdot \mathbf{F} \tag{8}$$

We now apply the complementary volume technique to (1) and (8).

COVOLUME DISCRETIZATIONS OF THE FLOW PROBLEM

In this section we describe three ways to derive finite difference approximations on Delaunay/Voronoi tessellations. In the first scheme, discrete conservation of mass is imposed on each triangle of the triangulation and in the second scheme conservation of mass holds on each Voronoi polygon. In each of these schemes the vector velocity field of the continuum problem is approximated by a scalar field of a single velocity component (components normal to triangle sides in the first scheme and components normal to polygon sides in the second scheme). However, our main objective here is to combine the two sets of difference equations to derive a third finite difference model which generates a discrete *vector* field by simultaneously determining normal and tangential components of flow. Before describing this implementation of the covolume method, we define several terms and give details of the first two schemes. With reference to *Figure 5*, let $\mathbf{n}_i$ be pre-assigned unit vectors normal to the triangle edges $\overline{V_0 V_i}$ (and therefore, tangent to the polygon edge $\overline{Q_{i-1} Q_i}$) and let $\mathbf{s}_i$ be unit vectors tangent to the triangle edge $V_0 V_i$ such that $(\mathbf{n}_i, \mathbf{s}_i)$ forms a right-handed coordinate system. For ease of exposition, we assume that the Delaunay tessellation is such that all triangles are equilateral. In this case the midpoints $P_i \equiv (V_0 + V_i)/2$ and $P_i' \equiv (Q_{i-1} + Q_i)/2$ coincide. We define normal and tangential velocity components:

$$u_i(t) = \mathbf{n}_i \cdot \mathbf{q}|_{P_i}, v_i(t) = \mathbf{s}_i \cdot \mathbf{q}|_{P_i}$$

If the grid or mesh is not equilateral, then the midpoints P_i and P_i' will be distinct, however, the line segments $\overline{V_0 V_i}$ and $\overline{Q_{i-1} Q_i}$ are always perpendicular for Delaunay/Voronoi tessellations. We assume that P_i and P_i' are sufficiently close so that $u_i(t) - \mathbf{n}_i \cdot \mathbf{q}|_{P_i'}$ and $v_i(t) - \mathbf{s}_i \cdot \mathbf{q}|_{P_i'}$.

Finally, we note that each Voronoi polygon corner Q_i in *Figure 5* is the circumcentre of a Delaunay triangle (for example, Q_4 is the centre of the circle passing through V_0, V_3, V_4). Such a circumcentre lies outside its associated Delaunay triangle if said triangle is obtuse. Such is the case for several circumcentres in *Figure 2*. However, the triangulation in *Figure 2* is intended

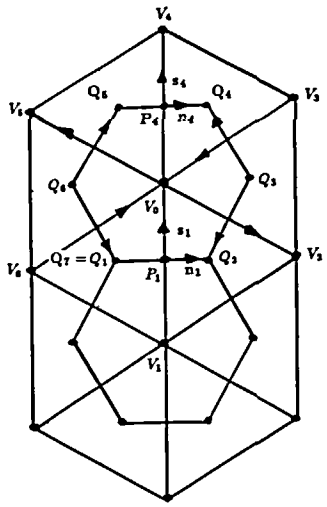
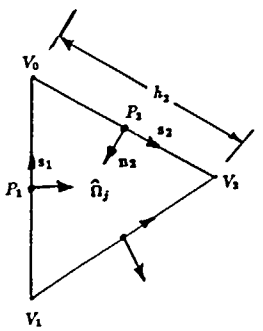
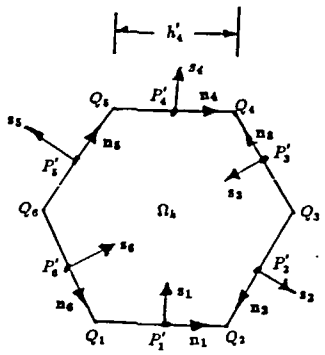


Figure 5 Local coordinate systems



Scheme 1

$u_i(t) = \mathbf{n}_i \cdot \mathbf{q}\,|_{P_i}$



Scheme 2

$v_i(t) = s_i \cdot \mathbf{q}\,|_{P_i}$

Figure 6 Typical control volumes

for illustration purposes only; the points were rather arbitrarily positioned and the mesh was not subject to Laplacian smoothing. The mesh generator used in this report for practical flow problems produces triangulations that are very nearly acute. The maximum angle in the triangulation in *Figure 3* is 107 degrees (the limiting case, a 90 degree triangle, has its circumcentre at the mid point of the hypotenuse). The methods that will be described in the sequel for deriving covolume finite difference equations apply to any triangle, acute or obtuse. What is important is that the Delaunay construction guarantees convex Voronoi polygons, polygon sides are perpendicular bisectors (or segments thereof) of triangle edges, and the mesh generator produces triangles that are acute or close to acute (that is, $P_i - P_i'$).

Scheme 1: Conservation of mass on Delaunay triangles

The continuity equation is integrated[10,11] over each triangular control volume, $\hat{\Omega}_j$ (see *Figure 6*).

The divergence theorem is applied and the normal velocity component, $\mathbf{n} \cdot \mathbf{q}$ is approximated by mid-edge normal velocities. The result is:

$$0 = \int_{\hat{\Omega}_j} \nabla \cdot \mathbf{q}\, dx = \int_{\partial\hat{\Omega}_j} \mathbf{n} \cdot \mathbf{q}\, ds \simeq \sum_{i=1}^{3} u_i(t) h_i (\mathbf{n} \cdot \mathbf{n}_i) \tag{9}$$

where $\mathbf{n}$ is the outward normal vector to $\partial\hat{\Omega}_j$, $\mathbf{n}_i$ is the pre-assigned normal to the ith side of $\partial\hat{\Omega}_j$, h_i is the length of the ith side and $u_i(t) = \mathbf{n}_i \cdot \mathbf{q}|_{P_i}$. Note that $(\mathbf{n} \cdot \mathbf{n}_i) = \pm 1$.

If there are N_T triangles in the triangulation of Ω, then there are N_T such discrete continuity equations which we write as:

$$\mathbf{A}_1 \mathbf{D}_1 \mathbf{u}(t) = \mathbf{b}_1 \tag{10}$$

where $\mathbf{u}(t)$ is an N_S vector of normal velocity components with ith entry $u_i(t)$, and N_S is the number of triangle sides in the triangulation for which the velocity is unknown. The diagonal matrix $\mathbf{D}_1$ has its ith diagonal entry equal to h_i. The vector $\mathbf{b}_1$ contains known boundary data (i.e., specified velocities in (4)). The $N_T \times N_S$ matrix $\mathbf{A}_1\mathbf{D}_1$ is called the *discrete divergence operator* associated with the Delaunay tessellation.

The momentum equation (in the scalar form defined by (8)) is approximated at the midpoint of each side of the N_S edges of the triangulation bearing an unknown normal velocity component as follows: with reference to *Figure 5* and starting with midside node P_1 in *Figure 5*, we take $\mathbf{n} = \mathbf{n}_1$ in (8) where $\mathbf{n}_1$ is the unit normal vector through P_1 normal to $\overline{V_0 V_1}$ and $\mathbf{s} = \mathbf{s}_1$ the unit tangent vector. Then we define the following approximations:

I. *Temporal term:*

$$\frac{\partial}{\partial t}(\mathbf{n}_1 \cdot \mathbf{q})|_{P_1} = \frac{\mathrm{d}}{\mathrm{d}t} u_1(t) \tag{11}$$

II. *Viscous term:*

$$\left.\frac{\partial \omega}{\partial \mathbf{s}}\right|_{P_1} \simeq \frac{\omega(V_0) - \omega(V_1)}{\| V_0 - V_1 \|} \tag{12}$$

As in Reference 13 we next note that we can approximate $\omega(V_0)$ in (12) for example by integrating over the Voronoi tile Ω_k with centre V_0 in *Figure 5* and *Figure 6*. Applying Green's theorem we obtain:

$$\int_{\Omega_k} \omega \, \mathrm{d}\mathbf{x} = \int_{\partial\Omega_k} \mathbf{s} \cdot \mathbf{q} \, \mathrm{d}\mathbf{s} \simeq \sum_{i=1}^{6} u_i(t) h_i' (\mathbf{s} \cdot \mathbf{n}_i) \tag{13}$$

where $\mathbf{s}$ is the unit tangent vector along $\partial\Omega_k$ traversing $\partial\Omega_k$ in the counterclockwise direction, $\mathbf{n}_i$ is the appropriate pre-assigned unit vector normal to a triangle side which is also a unit tangent vector for the ith side of the polygon Ω_k and h_i' is the length of the ith side of the polygon Ω_k. Note that $\mathbf{s} \cdot \mathbf{n}_i = \pm 1$. From (13) we have:

$$\omega(V_0) \simeq \left(\sum_{i=1}^{6} u_i(t) h_i' (\mathbf{s} \cdot \mathbf{n}_i) \right) \Big/ \mathrm{area}(\Omega_k) \tag{14}$$

which is then substituted into (12).

III. *Convective term*

The convective term can be handled in a variety of ways[11]. Here, we choose the upwind scheme[11], i.e.

$$\left[\frac{\partial}{\partial \mathbf{q}} (\mathbf{q}(P_1) \cdot \mathbf{n}_1) \right] |\mathbf{q}(P_1)| \simeq \begin{cases} \dfrac{\mathbf{q}(P_1) \cdot \mathbf{n}_1 - \mathbf{q}(Q') \cdot \mathbf{n}_1}{|P_1 - Q'|} |\mathbf{q}(P_1)| & \text{if } \mathbf{q}(P_1) \cdot \mathbf{n}_1 \geqslant 0 \\ \dfrac{\mathbf{q}(P_1) \cdot \mathbf{n}_1 - \mathbf{q}(R') \cdot \mathbf{n}_1}{|P_1 - R'|} |\mathbf{q}(P_1)| & \text{if } \mathbf{q}(P_1) \cdot \mathbf{n}_1 < 0 \end{cases} \tag{15}$$

where Q' and R' denote points of intersection of the line through P_1 in the direction of $\mathbf{q}(P_1)$ with the boundaries of the two triangles sharing side $\overline{V_0 V_1}$ (see *Figure 7*). The velocity vectors $\mathbf{q}(Q')$ and $\mathbf{q}(R')$ are approximated by linear interpolation of the three midside velocity vectors[11]. Note that the convective term in (15) is quadratic in the velocity components, $u_i(t)$ and $v_i(t)$.

IV. *Pressure term*

$$\frac{\partial p}{\partial \mathbf{n}_1} \simeq \frac{p_2(t) - p_1(t)}{\| Q_2 - Q_1 \|} = \frac{p_2(t) - p_1(t)}{h_1'} \tag{16}$$

where $p_i(t)$ approximates the pressure at circumcentre Q_i.

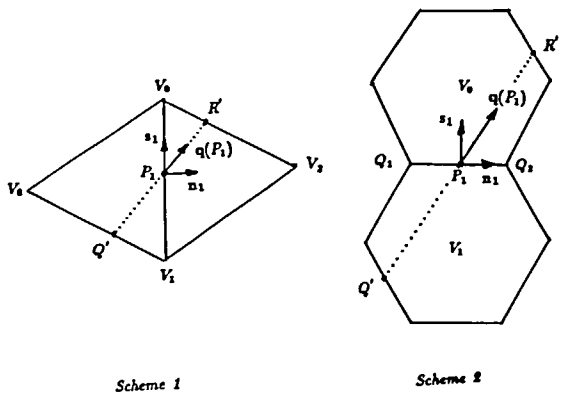


Figure 7 Upwind convection

There is one continuous time, discrete space momentum equation for each of the N_S sides of the triangulation. This system of ordinary differential equations we write as:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{u}(t) + \mathbf{g}_1(\mathbf{u}(t), \mathbf{v}(t)) + \mathbf{D}_2\mathbf{A}_1^{\mathrm{T}}\mathbf{p}_1(t) = \mathbf{c}_1 \tag{17}$$

where $\mathbf{u}(t)$ and $\mathbf{v}(t)$ are respectively N_S-dimensional vectors of normal and tangential flows centred at the midpoints of triangle sides, $\mathbf{p}_1(t)$ is the N_T-dimensional vector of unknown pressures centred at triangle circumcentres and $\mathbf{c}_1$ contains known boundary data. The diagonal matrix $\mathbf{D}_2$ has ith diagonal entry equal to $1/h_i'$. Note that $\mathbf{D}_2\mathbf{A}_1^{\mathrm{T}}$ is the *discrete gradient operator* and that $\mathbf{A}_1^{\mathrm{T}}$ is the transpose of $\mathbf{A}_1$ in (10). Equations (10) and (17) constitute a system of $N_T + N_S$ differential/algebraic equations (a so-called DAE) in $N_T + 2N_S$ unknowns $(\mathbf{p}_1(t), \mathbf{u}(t), \mathbf{v}(t))$. Scaling the normal velocity components $\mathbf{u}(t)$ by $\mathbf{D}_1$ and the tangential velocity components by $\mathbf{D}_2^{-1}$ and defining $\mathbf{U}(t) = \mathbf{D}_1\mathbf{u}(t), \mathbf{V}(t) = \mathbf{D}_2^{-1}\mathbf{v}(t)$, we write this system as follows:

$$\mathbf{A}_1\mathbf{U}(t) = \mathbf{b}_1$$

$$\mathbf{D}_2^{-1}\mathbf{D}_1^{-1}\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}t} + \mathbf{D}_2^{-1}\mathbf{g}_1(\mathbf{D}_1^{-1}\mathbf{U}(t), \mathbf{D}_2\mathbf{V}(t)) + \mathbf{A}_1^{\mathrm{T}}\mathbf{p}_1(t) = \mathbf{D}_2^{-1}\mathbf{c}_1 \tag{18}$$

Scheme 1 has been implemented[10,11] where the time derivatives were approximated by the backward Euler method, the tangential velocities $\mathbf{V}(t)$ were determined using an interpolation of normal components[10], and in the convection term $\mathbf{V}(t)$ was time lagged. The interpolation scheme for the tangential velocities has the potential to introduce temporal instabilities into the calculation. This provides motivation for consideration of the third scheme presented below which avoids this interpolation. We note that for *regular* triangulations (i.e., all triangles have identical circum radii) Scheme 1 exhibits a second order asymptotic spatial convergence rate. For *non-regular* triangulations, the convergence rate slips to first order.

As an illustration of the utility of covolume methods on unstructured grids, *Figure 8* contains a vector field determined by Scheme 1 for flow through a monolithic automotive converter using the dual tessellations in *Figure 3*. For this problem we assumed boundary conditions of zero normal and tangential velocity components across and along impervious walls, $u(x, y, t) =$ constant, $v(x, y, t) = 0$ at the inlet of the converter, and $p(x, y, t) =$ constant at the outlet of the converter.

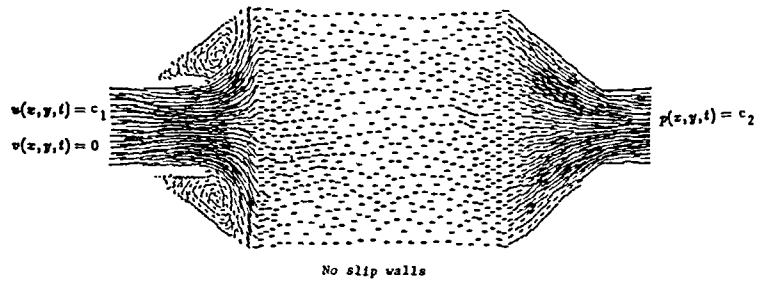


Figure 8 Flow through a monolithic converter

Scheme 2: Conservation of mass on Voronoi tiles

The second approach to deriving finite difference equations by the covolume method is to reverse the roles played by Delaunay triangles and Voronoi polygons. That is, the continuity equation is now integrated over each polygonal Voronoi tile, Ω_k (see *Figure 6.*). The divergence theorem is applied and the normal velocity component $\mathbf{n}\cdot\mathbf{q}$ is approximated by midside normal velocities which are in fact tangential velocities along triangle sides:

$$0 = \int_{\Omega_k} \nabla\cdot\mathbf{q}\,\mathrm{d}\mathbf{x} = \int_{\partial\Omega_k} \mathbf{n}\cdot\mathbf{q}\,\mathrm{d}s \simeq \sum_{i=1}^{6} v_i(t)h_i'(\mathbf{n}\cdot\mathbf{s}_i) \tag{19}$$

where $\mathbf{n}$ is the outward normal vector to $\partial\Omega_k$, $\mathbf{s}_i$ is the prescribed unit tangent vector to the ith side $\overline{V_0V_i}$ (that is, the vector normal to $\overline{Q_iQ_{i+1}}$), and $v_i(t) = \mathbf{s}_i\cdot\mathbf{q}|_{P_i}$. Again, $\mathbf{n}\cdot\mathbf{s}_i = \pm 1$.

If there are N_V triangle vertices in the Delaunay triangulation, there are N_V discrete continuity equations in (19), one for each Voronoi polygon, which we write as:

$$\mathbf{A}_2\mathbf{D}_2^{-1}\mathbf{v}(t) = \mathbf{b}_2 \tag{20}$$

where $\mathbf{v}(t)$ is an N_S-dimensional vector with ith entry $v_i(t)$ and N_S is the number of triangle sides in the triangulation for which the tangential velocity is unknown. The $N_S \times N_S$ diagonal matrix $\mathbf{D}_2^{-1}$ has its ith diagonal entry equal to h_i'. The vector $\mathbf{b}_2$ contains known boundary data (i.e., specified tangential velocities in (4)) and the $N_V \times N_S$ matrix $\mathbf{A}_2\mathbf{D}_2^{-1}$ is also called a *discrete divergence operator* associated with the Voronoi tessellation.

In this approach the scalar momentum equation (8) is approximated at the mid-point of each of the N_S-edges of the triangulation bearing an unknown tangential velocity component. We do this by reversing the roles played by $\mathbf{n}_1$ and $\mathbf{s}_1$: in (8) taking $\mathbf{n} = \mathbf{s}_1$, a unit vector through P_1 and normal to $\overline{Q_1Q_2}$, and $\mathbf{s} = \mathbf{n}_1$, a unit tangent vector to $\overline{Q_1Q_2}$, each term in (8) is approximated in turn as:

I. *Temporal term*

$$\frac{\partial}{\partial t}(\mathbf{s}_1\cdot\mathbf{q})|_{P_1} = \frac{\mathrm{d}}{\mathrm{d}t}v_1(t) \tag{21}$$

II. *Viscous term*

$$\left.\frac{\partial\omega}{\partial\mathbf{n}_1}\right|_{P_1} = \frac{\omega(Q_2) - \omega(Q_1)}{\|Q_2 - Q_1\|} \tag{22}$$

whereas previously we can approximate $\omega(Q_2)$ in (22) by integrating ω over a Delaunay triangle,

say Ω_k in *Figure 6*, and again apply Green's thereom to obtain:

$$\int_{\hat{\Omega}_j} \omega \, dx = \int_{\partial\hat{\Omega}_j} \mathbf{s}\cdot\mathbf{q} \, ds \simeq \sum_{i=1}^{3} v_i(t) h_i(\mathbf{s}\cdot\mathbf{s}_i) \tag{23}$$

where $\mathbf{s}$ is the unit clockwise tangent vector along $\partial\hat{\Omega}_j$, $\mathbf{s}_i$ is the pre-assigned unit tangent vector which is also a unit normal vector for the ith side of Ω_k and h_i is the length of the ith side of Delaunay triangle $\hat{\Omega}_j$. Again, $\mathbf{s}\cdot\mathbf{s}_i = \pm 1$. From (23) we have

$$\omega(Q_2) \simeq \left(\sum_{i=1}^{3} v_i(t) h_i(\mathbf{s}\cdot\mathbf{s}_i) \right) \Big/ \text{area}(\hat{\Omega}_j) \tag{24}$$

which is then substituted into (22).

III. *Convective term*

The convective term is handled using an upwind scheme for polygonal control volumes analogous to the upwind scheme[11] for triangular control volumes. With reference to *Figure 7*,

$$\left[\frac{\partial}{\partial \mathbf{q}} (\mathbf{q}(P_1)\cdot\mathbf{s}_1) \right] |\mathbf{q}(P_1)| \simeq \begin{cases} \dfrac{\mathbf{q}(P_1)\cdot\mathbf{s}_1 - \mathbf{q}(Q')\cdot\mathbf{s}_1}{|P_1 - Q'|} |\mathbf{q}(P_1)| & \text{if } \mathbf{q}(P_1)\cdot\mathbf{s}_1 \geqslant 0 \\ \dfrac{\mathbf{q}(P_1)\cdot\mathbf{s}_1 - \mathbf{q}(R')\cdot\mathbf{s}_1}{|P_1 - R'|} |\mathbf{q}(P_1)| & \mathbf{q}(P_1)\cdot\mathbf{s}_1 < 0 \end{cases} \tag{25}$$

where Q' and R' denote points of intersection of the line through P_1 and in the direction of $\mathbf{q}(P_1)$ with the boundaries of the two polygons sharing side $\overline{Q_1 Q_2}$. The velocity vectors $\mathbf{q}(Q')$ and $\mathbf{q}(R')$ are approximated[11] by a linear interpolation of the midside velocity vectors. Note that (25) is quadratic in the velocity components, $u_i(t)$ and $v_i(t)$.

IV. *Pressure term*

$$\frac{\partial p}{\partial \mathbf{s}_1} \simeq \frac{p_0(t) - p_1(t)}{\| V_0 - V_1 \|} = \frac{p_0(t) - p_1(t)}{h_1} \tag{26}$$

where $p_i(t)$ approximates pressure at the triangle vertex, V_i in *Figure 5*.

There is one continuous time-discrete space momentum equation for each of the N_S sides of the triangulation. This system of ordinary differential equations we write as:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{v}(t) + \mathbf{g}_2(\mathbf{u}(t), \mathbf{v}(t)) + \mathbf{D}_1^{-1}\mathbf{A}_2^{\mathrm{T}}\mathbf{p}_2(t) = \mathbf{c}_2 \tag{27}$$

where $\mathbf{p}_2(t)$ is the N-dimensional vector of unknown pressures located at triangle vertices and $\mathbf{c}_2$ contains known boundary data. Note that $\mathbf{D}_1^{-1}\mathbf{A}_2^{\mathrm{T}}$ is a discrete gradient operator and that $\mathbf{A}_2^{\mathrm{T}}$ is the transpose of $\mathbf{A}_2$ in (20). Equations (20) and (27) constitute a system of $N_V + N_S$ differential/algebraic equations in $N_V + 2N_S$ unknowns ($\mathbf{p}_2(t)$, $\mathbf{u}(t)$, $\mathbf{v}(t)$). We write this system as:

$$\mathbf{A}_2\mathbf{V}(t) = \mathbf{b}_2,$$

$$\mathbf{D}_1\mathbf{D}_2\frac{\partial \mathbf{V}}{\partial t} + \mathbf{D}_1\mathbf{g}_2(\mathbf{D}_1^{-1}\mathbf{U}(t), \mathbf{D}_2\mathbf{V}(t)) + \mathbf{A}_2^{\mathrm{T}}\mathbf{p}_2(t) = \mathbf{D}_1\mathbf{c}_2 \tag{28}$$

In order to integrate (28), the velocity components $\mathbf{U}(t)$ need to be determined as some function of the velocity components $\mathbf{V}(t)$. We follow the idea[10,11] and, with reference to

Figure 6, we approximate the velocity within Ω_k by:

$$\mathbf{q}_{\Omega_k} \equiv \frac{1}{6}\sum_{j=1}^{6}(u_j\mathbf{n}_j + v_j\mathbf{s}_j) \tag{29}$$

(Weights other than $\frac{1}{6}$ could be used if Ω_k were not a regular hexagon; see Reference 11 for the triangle case.) The components u_i are now determined so that:

$$\mathbf{q}_{\Omega_k}\cdot\mathbf{n}_i = u_i \qquad i = 1,\ldots,6 \tag{30}$$

This leads to the strictly diagonally dominant system:

$$\sum_{j=1}^{6}\left[\delta_{ij} - \frac{1}{6}\mathbf{n}_j\cdot\mathbf{n}_i\right]u_j = \frac{1}{6}\sum_{j=1}^{6}(\mathbf{s}_j\cdot\mathbf{n}_i)v_j \qquad i = 1,\ldots,6 \tag{31}$$

where δ_{ij} is the Kronecker delta. The system can be solved uniquely for the u-components as functions of the v-components. Using this approximation in (28) reduces (28) to a DAE in the unknowns $\mathbf{V}(t)$ and $\mathbf{p}_2(t)$.

Scheme 3: Combination of Schemes 1 and 2

Schemes 1 and 2 as presented above result in fewer equations than unknowns due to the fact that the convection term involves the velocity vector while only one component of the vector momentum equation has been considered. This deficiency is handled for Scheme 1 by using an interpolation formula[10,11,13] to approximate tangential velocity components in terms of the normal velocity components. A similar interpolation scheme was given above for Scheme 2. We now present an alternative which avoids the need to make such *ad hoc* approximations. We simply combine (18) and (28). This result is $(N_T + N_V + 2N_S)$ differential/algebraic equations in a like number of unknowns. These equations can be written as:

$$\mathbf{A}\mathbf{W}(t) = \mathbf{b} \tag{32}$$

$$D_1 D_2\frac{\mathrm{d}\mathbf{W}}{\mathrm{d}t} + D_1\mathbf{g}(D_2\mathbf{W}) + \mathbf{A}^{\mathrm{T}}\mathbf{p}(t) = D_1\mathbf{c} \tag{33}$$

where $\mathbf{W}(t) = (\mathbf{U}(t), \mathbf{V}(t))^{\mathrm{T}}$, $\mathbf{g} = (\mathbf{g}_1, \mathbf{g}_2)^{\mathrm{T}}$,

$$\mathbf{A} = \begin{bmatrix}\mathbf{A}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{A}_2\end{bmatrix}$$

$$D_1 = \begin{bmatrix}\mathbf{D}_2^{-1} & \mathbf{0}\\ \mathbf{0} & \mathbf{D}_1\end{bmatrix}$$

$$D_2 = \begin{bmatrix}\mathbf{D}_1^{-1} & \mathbf{0}\\ \mathbf{0} & \mathbf{D}_2\end{bmatrix}$$

$\mathbf{p}(t) = (\mathbf{p}_1(t), \mathbf{p}_2(t))^{\mathrm{T}}$, $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2)^{\mathrm{T}}$, and $\mathbf{b} = (\mathbf{b}_1, \mathbf{b}_2)^{\mathrm{T}}$.

Note that the coupling between $\mathbf{U}(t)$ and $\mathbf{V}(t)$ is through the discrete convection term $\mathbf{g}$ and the boundary conditions. These differential/algebraic equations could now be solved using a DAE solver or by discretizing the time derivatives and solving the resulting non-linear equations. Alternatively, as is commonly done, the non-linear convection terms can be linearized, say by time-lagging the convection coefficients and the resulting linear system solved. In the next section the dual variable method is described which economizes on the solution of systems of the form (32)–(33).

REDUCING THE PRIMITIVE SYSTEM

Many discretizations of the incompressible Navier–Stokes equations lead to a system of the generic form (32–33). In fact, this is true for Schemes 1 and 2 themselves. The *dual variable method*[1] introduces a set of variables (called *dual variables*) which transforms systems of the generic form (32–33) into an equivalent system of much lower dimensionality. We outline the steps of the method below:

Step 1. At any value of time, t, a solution to the underdetermined $(N_T + N_V) \times 2N_S$ system (32) must be of the form $\mathbf{W}(t) = \mathbf{W}_P + \mathbf{W}_H(t)$ where $\mathbf{W}_P$ is a time independent particular solution of (32), and $\mathbf{W}_H(t)$ is a solution of the homogeneous system $\mathbf{A}\mathbf{W}_H(t) = 0$. The first step, then is to find a particular solution to (32).

Step 2. Since $\mathbf{A}\mathbf{W}_H(t) = 0$, $\mathbf{W}_H(t)$ is a member of the *null space* of the matrix A. A basis for the null space of A is a set of linearly independent $2N_S$-dimensional vectors $\{\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_R\}$ all of which satisfy $\mathbf{A}\mathbf{C}_i = 0$. Any solution to the homogeneous system can be written as a linear combination of the $\mathbf{C}_i$'s. Hence, in this step we find a basis for the null space of A and form the $2N_S \times R$ matrix C, with $\mathbf{C}_i$ as its ith column. Then,

$$\mathbf{A}\mathbf{C} = 0 \tag{34}$$

and

$$\mathbf{W}_H(t) = \mathbf{C}\mathbf{X}(t) \tag{35}$$

for some yet to be determined $R \times 1$ vector $\mathbf{X}(t)$. The value of R will be determined in the next section.

Step 3. Substitute $\mathbf{W}_P + \mathbf{W}_H(t)$ into (33) to produce:

$$D_1 D_2 \mathbf{C}\frac{d\mathbf{X}}{dt} + D_1 \mathbf{g}(\mathbf{W}_P + \mathbf{C}\mathbf{X}(t)) + \mathbf{A}^T \mathbf{p}(t) = D_1 \mathbf{c} \tag{36}$$

Step 4. Multiply (36) by $\mathbf{C}^T$ and use (34) to obtain the $R \times R$ system:

$$\mathbf{C}^T D_1 D_2 \mathbf{C}\frac{d\mathbf{X}}{dt} + \mathbf{C}^T D_1 \mathbf{g}(\mathbf{W}_P + \mathbf{C}\mathbf{X}(t)) = \mathbf{C}^T D_1 \mathbf{c} \tag{37}$$

The matrix transformation in (37) is called the *dual variable transformation* and (37) itself is called the $R \times R$ *dual variable system.* Note that the pressures have been eliminated.

Step 5. Solve (37) for $\mathbf{X}(t)$ using any ordinary differential equation integrator, recover the velocities $\mathbf{W}(t)$ using (35), and recover pressures $\mathbf{p}(t)$ by using (36).

The dual variable method then involves two major steps: (a) finding a particular solution to (32) and (b) finding a basis for the null space of A. Both of these tasks can be accomplished by using network theoretic properties of the Delaunay/Voronoi meshes. This will be discussed in the next section. However, we remark here that for Scheme 3 the dimension of the dual variable system (37) is of the order $O(3N_V)$. Hence, we have replaced a system of differential/algebraic equations (32–33) involving* $O(9N_V)$ primitive velocity and pressure unknowns by a system of differential equations that is *3 times smaller* involving $O(3N_V)$ dual variables.

Although it is difficult to give a purely physical interpretation to the dual variables, $\mathbf{X}(t)$, they are best thought of as *circulations* about each of the N_V triangle vertices and N_T polygon vertices in the tessellations. A circulation about a vertex is determined from a line integral of the flow along the boundary of the dual tile containing that vertex. An equivalent electric network

*It is known[7] that $N_S = O(3N_V)$ and $N_T = O(2N_V)$.

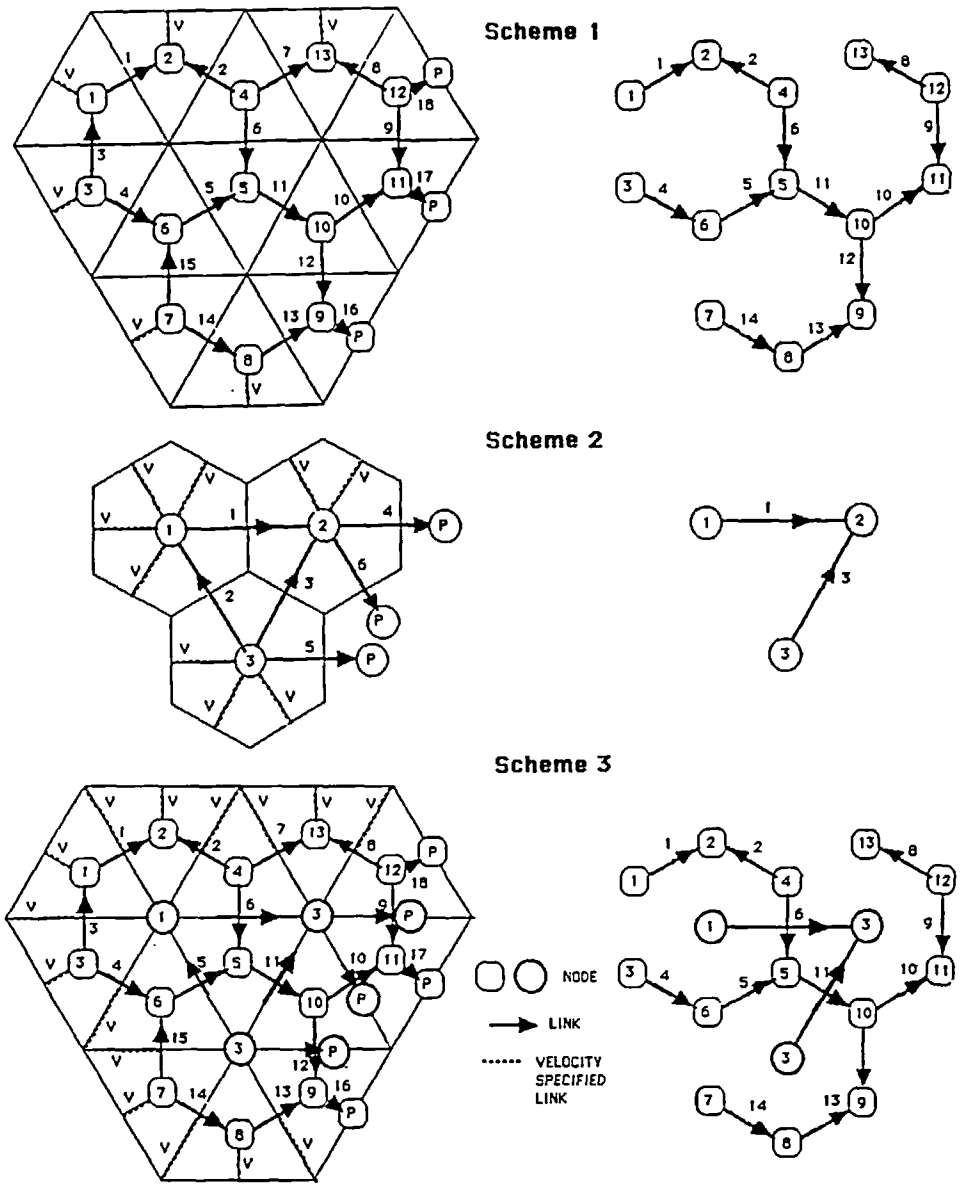


Figure 9 Networks and trees for those networks

interpretation is offered by considering the dual variables to be *mesh currents* around the boundary of each Delaunay triangle or Voronoi tile. See, for example p. 465 of Reference 4.

NETWORK THEORY AND THE DUAL VARIABLE METHOD

Implementation of the dual variable method requires (i) construction of a particular solution of the discrete continuity equation, and (ii) construction of a basis for the null space of the discrete divergence operator. Both these requirements can be met using standard algorithms for network

theory. A network N^* is a set, S, of N *nodes* and set, T^*, of L *links*. These links are of two types. *Interior* links connecting two nodes of S and *boundary* links connecting a node with an ideal node at infinity. *Figure 9* contains three networks associated with the three schemes presented earlier. For Scheme 1, the nodes are triangle circumcentres and the links are the directed (according to the choice of $\mathbf{n}_i$) line segments connecting the circumcentres. For Scheme 2, the nodes are centres of the Voronoi polygons and the links are the directed (according to the choice of $\mathbf{s}_i$) line segments connecting these centres. Note that we do not include any flow path along which a velocity is known; however, there are boundary links to pressure specified boundary (or pendant) nodes. Finally, we observe that the network for Scheme 3 is simply a catenation of the networks for Schemes 1 and 2. We now consider the above two requirements for a general network.

Particular solutions

The particular solution of a discrete continuity equation is obtained using the notion of a spanning tree of the associated network[1]. A spanning tree is a path (chain of links) in the network which contains no cycles (loops) and which connects all nodes. There are algorithms and software available to determine a tree for a given network. *Figure 9* contains sample trees for the three networks given.

Next, recall that there is one discrete continuity equation for each node (excluding pendant nodes). The velocities on the links of N^* which are not in the tree are set to zero. Also, the velocities on boundary links are set to zero. Beginning with the outermost extremities of the network, one proceeds through the nodes of the tree so that as each node is encountered, all but one velocity component associated with links incident on that node has been determined. The continuity equation for that node is used to determine its remaining velocity, taking into account any specified boundary velocities.

In essence, the tree establishes an order for the nodes (equations) and links (variables) so that the discrete continuity equations form a triangular system. For the spanning tree in *Figure 9* (Scheme 1) the nodes are re-ordered according to the tree structure as (1, 2, 4, 3, 6, 5, 13, 12, 11, 10, 9, 8, 7). The unknown velocities are re-ordered as (1, 2, 6, 4, 5, 11, 8, 9, 10, 12, 13, 14). With this new ordering the 13 continuity equations form a lower triangular system which is easily solved. Hence, the construction of a particular solution of the discrete continuity equation is an easy task given a spanning tree. This same tree can be used, knowing the velocities, to recover the pressures (or more accurately pressure drops) from

$$\mathbf{A}^{\mathrm{T}}\mathbf{p}(t) = \mathbf{d}(t) \tag{38}$$

where $\mathbf{d}(t)$ is known [cf. (33)].

Null space

The matrices $\mathbf{A}_1$ in (18), $\mathbf{A}_2$ in (28) and $\mathbf{A}$ in (32) are *incidence* matrices of the networks associated with the respective schemes (see *Figure 9*). The matrix $\mathbf{E} = (e_{ij})$ is an incidence matrix of the directed network N^* if

$$e_{ij} = \begin{cases} +1, & \text{if link } j \text{ is incident from node } i \\ -1, & \text{if link } j \text{ is incident to node } i \\ 0, & \text{otherwise} \end{cases}$$

The incidence matrix for the second network in *Figure 9* is the 3 × 6 matrix

$$\mathbf{E} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$$

It is known[3] that if N^* has at least one boundary link then the incidence matrix **E** is of rank N, the number of nodes in N^*. Hence the dimension of the null space of **E** is $L - N$, the number of links minus the number of nodes. To find a basis for this null space again we turn to network theory and the notion of a *fundamental matrix*, **C**, the columns of which are *cycle vectors.*

A cycle is a chain of links whose extremities coincide and is such that any other node is encountered at most one during a traverse of the chain. A cycle vector $\mathbf{c}_k = (c_{1k}, \ldots, c_{Lk})^T$ is defined by:

$$c_{jk} = \begin{cases} +1, & \text{if link } j \text{ has positive orientation during a clockwise traverse of cycle } k \\ -1, & \text{if link } j \text{ has negative orientation during a clockwise traverse of cycle } k \\ 0, & \text{otherwise} \end{cases}$$

For the second network in *Figure 9* the vector $\mathbf{c}_1 \equiv (1, 1, -1, 0, 0, 0)^T$ is a cycle vector.

One can verify that $\mathbf{E}\mathbf{c}_k = 0$; that is, $\mathbf{c}_k$ is in the null space of **E**. If there are L° interior links in the network then it is known[3] that there are $L^\circ - N + 1$ linearly independent cycle vectors.

With regard to boundary links, there are $L - L^\circ - 1$ chains such that the first and last links are the kth and $(k + 1)$st boundary links. One can define *pseudo cycle* vectors for such chains in a manner similar to the c_{jk} above. For example, the second network in *Figure 9*, the vector $\mathbf{c}_2 = (0, 0, 1, 0, -1, 1)^T$ is a pseudo-cycle vector. It is known that the totality of $L - N$ cycle vectors and pseudo-cycle vectors thus constructed are linearly independent and span the null space of the incidence matrix.

Hence the matrix **C** needed in the dual variable method is simply the $L \times (L - N)$ matrix of cycle (and pseudo-cycle) vectors. For the network in *Figure 9* (Scheme 2), the fundamental matrix is:

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix}$$

The dual variable system for Scheme 2 is 3 × 3 while the primitive system is 9 × 9. In general let there be N_T triangles, N_S triangle sides and, N_V triangle vertices. Then for each of the three schemes the relative size of the dual variable and primitive systems are given in *Table 1.*

Note that the network for Scheme 3 has two disjoint components or sub-networks which are in fact the networks for Schemes 1 and 2. As such the fundamental matrix **C** for Scheme 3 is a

Table 1 Reduction factors

	Scheme 1	Scheme 2	Scheme 3
L = velocity components	N_S	N_S	$2N_S$
N = pressures	N_T	N_V	$N_T + N_V$
$L + N$ = size of primitive systems	$N_S + N_T = O(5N_V)$	$N_S + N_V = O(4N_V)$	$2N_S + N_T + N_V = O(9N_V)$
$L - N$ = size of dual variable system	$N_S - N_T = O(N_V)$	$N_S - N_V = O(2N_V)$	$2N_S - N_T - N_V = O(3N_V)$
Reduction factor	5	2	3

block diagonal matrix

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 & 0 \\ 0 & \mathbf{C}_2 \end{bmatrix}$$

where $\mathbf{C}_1$ and $\mathbf{C}_2$ are the fundamental matrices for Schemes 1 and 2 respectively.

GENERALIZATIONS

The three covolume schemes presented, and in particular the (combined) Scheme 3, can be generalized to other types of dual tessellations. A very general approach to the construction of a tessellation of a flow domain is given elsewhere[16]. Examination of the approach outlined above shows that the key ingredient in the present approach is the *duality* of the tessellations in the sense that the edge segments of the first tessellation are *orthogonal* to the edge segments of the second tessellation.

Once this orthogonality is established[14], the covolume discretization of the Navier–Stokes problem has three main constituents: (1) the approximation of the divergence, (2) the

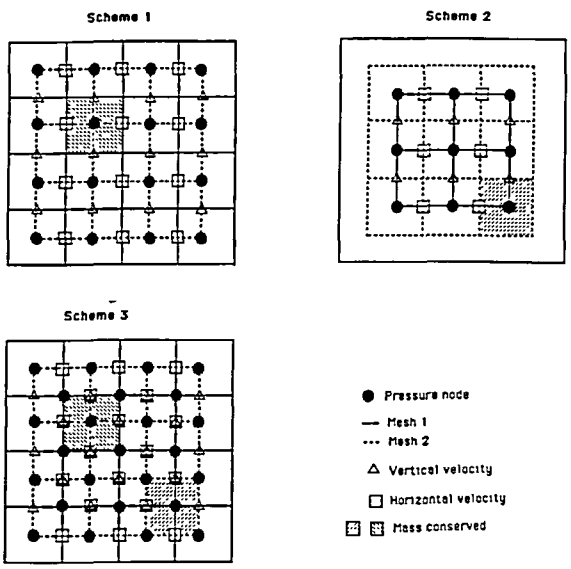


Figure 10 MAC Schemes on rectangular grids

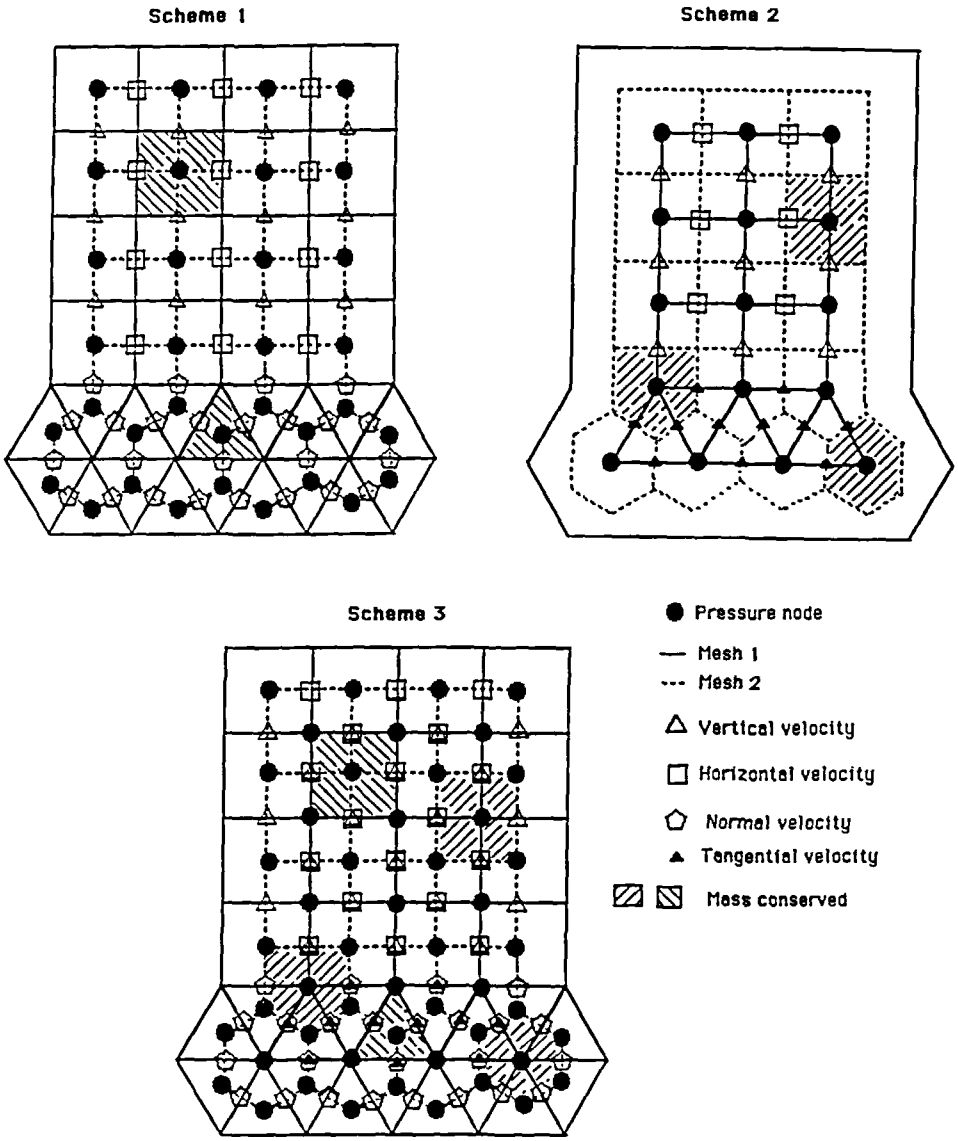


Figure 11 Grid containing rectangles and triangles

approximation of the scalar vorticity, and (3) the approximation of the convective term. The first and second approximations reduce to boundary integrals (cf. (9) and (13)) which have straightforward extensions to other shaped tiles. The third (cf. (15)) can also be extended to other shaped tiles assuming that the appropriate interpolation schemes can be constructed.

We mentioned here two specific generalizations of the notion of dual tessellations; (1) the staggered MAC rectangular tessellations and (2) a combination of rectangular and triangular tiles. The first is illustrated in *Figure 10* and the second in *Figure 11*. We assume in both illustrations that consistent velocities (normal and tangential components) have been specified on the entire boundary so as to simplify the presentation. It is also possible to treat more

elaborate combinations of boundary conditions. Also, in these Figures normal and tangential refer to directions relative to the sides of triangles.

CONCLUSIONS

We presented[10] an approach (called Scheme 1 here) that integrates three computational concepts into a single algorithm for modelling the two-dimensional incompressible Navier–Stokes equations: (1) automatic Delaunay mesh generation, (2) covolume finite difference equation generation, and (3) dual variable reduction of primitive systems. This covolume approach replaces the continuum problem involving a vector velocity field with a discrete problem involving only a single scalar velocity which is normal to triangle sides. Also presented[10] was an interpolation scheme for deriving *a posteriori* approximations of flows tangent to triangle sides in order to produce a discrete vector-valued approximation of velocity.

In this paper we have accomplished three principal objectives. First, we have shown that a simple role reversal of Delaunay triangle and Voronoi polygon in the covolume method for deriving difference equations produces an analogous discrete problem involving a single scalar component of velocity normal to polygon sides. This approach we call Scheme 2. Again, a complete discrete model of the continuous flow field results from the approximation of velocity components tangent to polygon sides.

Second, we have shown that by combining Scheme 1 and Scheme 2 we can avoid the necessity of making any *a posteriori* estimates of velocity components and thereby produce a discrete approximation involving coupled, mutually orthogonal velocity components associated with midpoints of triangle or polygon sides (Scheme 3).

Third, we have shown that given the dual variable transformations for Scheme 1 and Scheme 2, it is a simple matter to construct a complete dual variable reduction for the primitive system arising from Scheme 3.

Finally, we have shown that the dual variable method applied to any one of the three covolume schemes leads to a significant reduction in the size of the primitive variable system (*Table 1*).

REFERENCES

1 Amit, R., Hall, C. A. and Porsching, T. A. An application of network theory to the solution of implicit Navier–Stokes difference equations, *J. Comp. Phys.*, **40**, 183–201 (1981)
2 Bank, R. E. and Rose, D. J. Some error estimates for the box method, *SIAM J. Num. Anal.*, **24**, 777–787 (1987)
3 Berge, C. and Ghouila-Houri, A. *Programming Games and Transportation Networks*, Methuen, London (1965)
4 Branin, F. H. Jr. The Algebraic-Topological Basis for Network Analogies and the Vector Calculus, *Proc. Symp. Gen. Networks* (Ed. J. Fox), Brooklyn Polytechnic Press, New York (1966)
5 Cavendish, J. C., Field, D. A. and Frey, W. H. An approach to automatic three-dimensional finite element mesh generation, *Int. J. Num. Meth. Eng.*, **21**, 329–347 (1985)
6 Chou, S. H. A network model for incompressible two-fluid flow and its numerical solution, *Num. Meth. Part. Diff. Eqs.*, **5**, 1–24 (1989)
7 Ewing, D. J., Fawkes, A. J. and Griffiths, J. R. Rules governing the number of nodes and elements in a finite element mesh, *Int. J. Num. Meth. Eng.*, **2**, 597–601 (1970)
8 Frey, W. H. Selective refinement: a new strategy for automatic node placement in graded triangular meshes, *Int. J. Num. Meth. Eng.*, **24**, 2183–2200 (1987)
9 Hall, C. A. Numerical solution of Navier–Stokes problems by the dual variable method, *SIAM J. Alg. Disc. Meth.*, **6**, 220–236 (1985)
10 Hall, C. A., Cavendish, J. C. and Frey, W. H. The dual variable method for solving fluid flow difference equations on Delaunay triangulations, *Comp. Fluids*, **20**, 145–164 (1991)
11 Hall, C. A., Porsching, T. A. and Mesina, G. On a network method for unsteady incompressible fluid flow on triangular grids, *Int. J. Num. Meth. Fluids* in press.
12 MacNeal, R. H. An asymmetric finite difference network, *Q. Appl. Math.*, **11**, 295–310 (1953)

13 Nicolaides, R. A. Triangular discretization for the vorticity–velocity equations, *Proc. 7th Int. Conf. Fin. Elements in Flow Problems, Huntsville, AL* (1989)
14 Nicolaides, R. A. Flow discretization by complementary volume techniques, *Proc. 9th AIAA CFD Meet., Buffalo, NY*, AIAA paper 89-1978 (1989)
15 Peyret, R. and Taylor, T. D. *Computational Methods for Fluid Flow*, Springer, Berlin (1983)
16 Porsching, T. A. A network model for two-fluid, two phase flow, *Num. Meth. Part. Diff. Eqs.*, **1**, 295–313 (1985)
17 Watson, D. F. Computing the *n*-dimensional Delaunay tessellation with applications to Voronoi polytopes, *Comp. Struct.*, **20**, 211–223 (1985)