

## II. SOME HERMITE INTERPOLATION FUNCTIONS FOR SOLENOIDAL AND IRROTATIONAL VECTOR FIELDS

JONAS T. HOLDEMAN

ABSTRACT. Some remarkable new Hermite interpolation functions on rectangular Cartesian meshes in two dimensions are developed. The examples are cubic-complete for scalar fields and quadratic-complete for vector fields. These are extended to orthogonal curvilinear coordinate systems, and affine meshes in those systems. The pair provide a new paradigm for interpolating divergence-free vector fields. After revisiting the incompressible Navier-Stokes equation, the functions are used with the finite element method (FEM) to solve the equation of motion for incompressible flow.

### 1. INTRODUCTION

In a previous paper [9], some Lagrange interpolation functions for vector fields were introduced, with the property that they exhibited a constant divergence or constant curl in Cartesian coordinates. They were extended to curvilinear coordinates, where they exhibited a common or consistent (non-constant) divergence or curl in each subdomain of the mesh. It was shown how strongly-solenoidal fields would result using a simple constraint on each subdomain of the mesh. In application to unsteady and nonlinear partial differential equations, the constraint equations have to be satisfied at each time step or nonlinear iteration. It would be more efficient to solve the constraint equation once, producing solenoidal basis functions to interpolate the vector field. This approach leads from Lagrange to Hermite interpolation functions.

It was asserted that the Lagrange functions provided a basis from which scalar Hermite interpolation functions could be developed, which functions have sufficient continuity that they can interpolate divergence-free and irrotational vector fields in a strong or pointwise sense. In the following sections these Hermite interpolation functions will be introduced, and it will be shown how they can be derived. It will be shown how meshes can be generated to take advantage of the generalized form of these functions. After revisiting the equation of motion for incompressible flow, it will be shown how strongly solenoidal solutions to the equation of motion for incompressible flow may be found.

Similar methods can be used with the three-dimensional Lagrange functions given in [9], and extensions thereof, to generate three-dimensional divergence-free Hermite interpolation functions. Extension of the methods, and the “solenoidal”

---

Prepared January 27, 2003.

2000 *Mathematics Subject Classification*. Primary 65D05; Secondary 65N30, 76D05.

results of this paper, to higher dimension may not entirely straightforward, but extension of the “irrotational” Hermite functions to three-dimensions is rather trivial and will be quoted later.

## 2. HERMITE FUNCTIONS ON RECTILINEAR CARTESIAN MESHES

Assume the problem domain  $\Omega$ , is partitioned into non-overlapping rectangular subdomains  $\Omega^e$  which cover  $\Omega$ . Consider a typical subdomain or element of the partition or mesh, a rectangle with dimensions  $\{h_x, h_y\}$  and center  $(x_c, y_c)$ , and with nodal indices  $\hat{x}_i, \hat{y}_i = \pm 1$ . For simplicity and economy of display, we give the interpolation functions in terms of dimensionless or normalized variables with some dimensional constants,

$$(2.1) \quad \begin{aligned} \xi &\equiv \frac{2}{h_x}(x - x_c), & \xi_0 &\equiv \hat{x}_i \xi, & \xi_h &\equiv \frac{2}{h_x}, & \xi_i &\equiv \hat{x}_i \xi_h, \\ \eta &\equiv \frac{2}{h_y}(y - y_c), & \eta_0 &\equiv \hat{y}_i \eta, & \eta_h &\equiv \frac{2}{h_y}, & \eta_i &\equiv \hat{y}_i \eta_h, \end{aligned}$$

We introduce the first new scalar Hermite interpolation function (which has its *curl* associated with the interpolation of solenoidal vector fields),

$$(2.2) \quad \mathbf{s}_i(x, y) = \begin{bmatrix} \frac{1}{8}(1 + \xi_0)(1 + \eta_0) & -\frac{1}{8}\eta_i^{-1}(1 + \xi_0) & \frac{1}{8}\xi_i^{-1}(1 + \eta_0) \\ (2 + \xi_0(1 - \xi_0) & (1 + \eta_0)^2(1 - \eta_0) & (1 + \xi_0)^2(1 - \xi_0) \\ + \eta_0(1 - \eta_0)) & & \end{bmatrix}.$$

Piecewise interpolations with this function are continuous, and interpolation of the monomials  $x^k y^m$ ,  $k + m \leq 3$  are exact. The interpolated scalar field on a typical rectangular subdomain  $e$  is given by,

$$(2.3) \quad \psi^e(x, y) = \sum_{i \in e} \mathbf{s}_i(x, y) \begin{bmatrix} \psi_i \\ u_i \\ v_i \end{bmatrix},$$

where the degrees of freedom are the scalar field  $\psi_i$  (stream function) and components of its curl  $\{u_i, v_i\}$ , and the sum is over the corner mesh nodes  $i$  on the element. Taking the curl of (2.3) gives the (solenoidal) vector interpolation function,

$$(2.4) \quad \mathbf{S}_i(x, y) = \begin{bmatrix} \frac{1}{8}\eta_i(1 + \xi_0) & \frac{1}{8}(1 + \xi_0)(1 + \eta_0) & \frac{1}{8}\eta_i \xi_i^{-1}(1 + \xi_0)^2 \\ (\xi_0(1 - \xi_0) + 3(1 - \eta^2)) & (-1 + 3\eta_0) & (1 - \xi_0) \\ -\frac{1}{8}\xi_i(1 + \eta_0) & \frac{1}{8}\xi_i \eta_i^{-1}(1 + \eta_0)^2 & \frac{1}{8}(1 + \xi_0)(1 + \eta_0) \\ (\eta_0(1 - \eta_0) + 3(1 - \xi^2)) & (1 - \eta_0) & (-1 + 3\xi_0) \end{bmatrix}.$$

Inspection of the columns of  $\mathbf{S}_i$  shows that the normal components of the column vectors vanish on boundaries of the rectangle that do not contain the controlling node, but the tangential components do not. Thus the normal components of piecewise interpolations are continuous across subdomain interfaces, and so the interpolation is divergence-free everywhere. The tangential vector components are potentially discontinuous at subdomain interfaces. Consequently, interpolations lie in a divergence-free subspace of  $H(\text{div}, \Omega)$ . This function exactly interpolates all solenoidal vector fields in which the field components are quadratic polynomials. Consequently, a necessary condition for the tangential components to be discontinuous at internal subdomain boundaries is that the field components vary more

rapidly than quadratically. The interpolated solenoidal vector field on a rectangular subdomain  $e$  is given by,

$$(2.5) \quad \mathbf{V}^e = \sum_{i \in e} \mathbf{S}_i(\xi, \eta) \begin{bmatrix} \psi_i \\ u_i \\ v_i \end{bmatrix},$$

The second Hermite interpolation function, which has its *gradient* associated with interpolation of irrotational fields, is

$$(2.6) \quad \mathbf{g}_i(x, y) = \begin{bmatrix} \frac{1}{8}(1 + \xi_0)(1 + \eta_0) & -\frac{1}{8}\xi_i^{-1}(1 + \eta_0) & -\frac{1}{8}\eta_i^{-1}(1 + \xi_0) \\ (2 + \xi_0(1 - \xi_0) & (1 + \xi_0)^2(1 - \xi_0) & (1 + \eta_0)^2(1 - \eta_0) \\ + \eta_0(1 - \eta_0)) & & \end{bmatrix}.$$

Piecewise interpolations using this function are continuous, and interpolation of the monomials  $x^k y^m$ ,  $k + m \leq 3$  are exact. The interpolated scalar (potential) field on a typical rectangular subdomain  $e$  is given by,

$$(2.7) \quad \phi^e(x, y) = \sum_{i \in e} \mathbf{g}_i(x, y) \begin{bmatrix} \phi_i \\ u_i \\ v_i \end{bmatrix},$$

where the degrees of freedom are the scalar field  $\phi_i$  (potential function), and components of its gradient  $\{u_i, v_i\}$ , and the sum is over the corner mesh nodes  $i$  on the element. Taking the gradient of (2.7) gives the (irrotational) vector interpolation function,

$$(2.8) \quad \mathbf{G}_i(x, y) = \begin{bmatrix} \frac{1}{8}\xi_i(1 + \eta_0) & \frac{1}{8}(1 + \xi_0)(1 + \eta_0) & -\frac{1}{8}\xi_i\eta_i^{-1}(1 + \eta_0)^2 \\ (\eta_0(1 - \eta_0) + 3(1 - \xi^2)) & (-1 + 3\xi_0) & (1 - \eta_0) \\ \frac{1}{8}\eta_i(1 + \xi_0) & -\frac{1}{8}\eta_i\xi_i^{-1}(1 + \xi_0)^2 & \frac{1}{8}(1 + \xi_0)(1 + \eta_0) \\ (\xi_0(1 - \xi_0) + 3(1 - \eta^2)) & (1 - \xi_0) & (-1 + 3\eta_0) \end{bmatrix}.$$

Inspection of the columns of  $\mathbf{G}_i$  shows that the *tangential* components of the column vectors vanish on boundaries of the rectangle that do not contain the controlling node, but the normal components do not. Thus the tangential components of piecewise interpolations are continuous across subdomain interfaces, and so the interpolation is irrotational everywhere. The normal vector components are potentially discontinuous at subdomain interfaces. Consequently, interpolations lie in an irrotational subspace of  $H(\mathbf{curl}, \Omega)$ . This function exactly interpolates all irrotational vector fields in which the field components are quadratic polynomials. Consequently, a necessary condition for the normal components to be discontinuous at internal subdomain boundaries is that the irrotational vector field vary more rapidly than quadratically.

Inspection of the scalar functions (2.2) and (2.6) shows that they are closely related. One can be obtained from the other by interchanging the rightmost two columns, with a change in sign of one column. This results from the close connection between the forms of the gradient and curl in two dimensions. Of course this simple relation does not extend to three dimensions.

## 3. AFFINE MESHES AND CURVILINEAR COORDINATES

Returning to the divergence-free Hermite function (2.4), the geometric parameters can be factored out into pre- and post-multiplying matrices as,

$$(3.1) \quad \mathbf{S}_i(x, y) = \begin{bmatrix} \xi_h^{-1} & 0 \\ 0 & \eta_h^{-1} \end{bmatrix} \hat{\mathbf{S}}_i(T(\xi, \eta)) \begin{bmatrix} 1 & 0 & 0 \\ 0 & \xi_h & 0 \\ 0 & 0 & \eta_h \end{bmatrix}.$$

Here,  $\hat{\mathbf{S}}_i$  is the function  $\mathbf{S}_i$  evaluated on the reference square  $[-1, 1]^2$ , obtained from (2.4) by setting  $\xi_i, \eta_i = \pm 1$ , and  $T(\xi, \eta)$  is the transformation from the local coordinates  $(\xi, \eta)$  to the global coordinates  $(x, y)$ . The pre- and post-multiplying matrices can be recognized as related to the Jacobian matrix of the transformation from local coordinates on  $[-1, 1]^2$  to the element centered at  $(x_c, y_c)$  with dimensions  $h_x, h_y$ . We choose,

$$(3.2) \quad \mathbf{J} = \begin{bmatrix} x_{,\xi} & y_{,\xi} \\ x_{,\eta} & y_{,\eta} \end{bmatrix}, \quad \mathbf{J}^{-1} = \Delta^{-1} \begin{bmatrix} y_{,\eta} & -y_{,\xi} \\ -x_{,\eta} & x_{,\xi} \end{bmatrix}, \quad \Delta = x_{\xi}y_{\eta} - x_{\eta}y_{\xi},$$

where the comma notation indicates differentiation with respect to the following variable. It can be verified that the pre-multiplying matrix in (3.1) is then  $\Delta^{-1}\mathbf{J}^T$ , where  $\Delta$  is the determinant of  $\mathbf{J}$ , and the post-multiplying matrix involves the inverse of this product evaluated at the node  $i$ . Then (3.1) can be expressed as,

$$(3.3) \quad \mathbf{S}_i(x, y) = \Delta^{-1}\mathbf{J}^T \hat{\mathbf{S}}_i(T(\xi, \eta)) \mathbf{T}_i^{-1}, \quad \mathbf{T}_i^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & \Delta_i(\mathbf{J}_i^T)^{-1} \end{bmatrix}$$

This form effects a generalization to non-rectangular subdomains. It can be shown that on affine meshes this exactly interpolates all solenoidal fields with quadratic components and it is quadratic-complete under general invertible transformations. (The first can be proved by enumeration using a symbolic mathematics program such as Maple, and the second by using the chain rule for differentiation.) A similar result is found for the scalar function (2.2).

In a similar way, the Hermite gradient function (2.8) can be factored as,

$$(3.4) \quad \mathbf{G}_i(x, y) = \begin{bmatrix} \xi_h & 0 \\ 0 & \eta_h \end{bmatrix} \hat{\mathbf{G}}_i(T(\xi, \eta)) \begin{bmatrix} 1 & 0 & 0 \\ 0 & \xi_h^{-1} & 0 \\ 0 & 0 & \eta_h^{-1} \end{bmatrix}.$$

As before, it can be shown that this generalizes to,

$$(3.5) \quad \mathbf{G}_i(x, y) = \mathbf{J}^{-1} \hat{\mathbf{G}}_i(T(\xi, \eta)) \bar{\mathbf{T}}_i^{-1}, \quad \bar{\mathbf{T}}_i^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{J}_i \end{bmatrix},$$

Again, it can be verified that this function exactly interpolates all irrotational fields with quadratic vector components under affine transformations. A similar generalization for the scalar potential form (2.6) will be given later.

In the previous paper [9], it was shown that the Lagrange interpolation functions from which these Hermite functions are derived (as will be shown below) can be further generalized to curvilinear coordinate systems (and affine transformations in the natural coordinates of these systems) by pre- and post-multiplying by certain diagonal matrices  $\mathbf{h}$  and  $\bar{\mathbf{h}}$  (or their inverses) involving singular points of the coordinate systems, and are given in Table 1. The same results apply here, so that the

TABLE 1. Diagonal matrices  $\mathbf{h}$  and  $\bar{\mathbf{h}}$  for several curvilinear coordinate systems.

	Cartesian ( $x, y$ )	Polar ( $r, \theta$ )	Axisymmetric ( $r, z$ )	Spherical surface ( $\vartheta, \varphi$ )	Cylindrical ( $r, \vartheta, z$ )	Spherical ( $r, \vartheta, \varphi$ )
$\mathbf{h}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} r^{-1} & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} r^{-1} & 0 \\ 0 & r^{-1} \end{bmatrix}$	$\begin{bmatrix} (\sin \vartheta)^{-1} & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} r^{-1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & r^{-1} \end{bmatrix}$	$\begin{bmatrix} (r^2 \sin \vartheta)^{-1} & 0 & 0 \\ 0 & (r \sin \vartheta)^{-1} & 0 \\ 0 & 0 & r^{-1} \end{bmatrix}$
$\bar{\mathbf{h}}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & r^{-1} \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & (\sin \vartheta)^{-1} \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & r^{-1} & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & r^{-1} & 0 \\ 0 & 0 & (r \sin \vartheta)^{-1} \end{bmatrix}$

full general forms of the interpolation functions are,

$$\begin{aligned}
\mathbf{s}_i(x', y') &= \hat{\mathbf{s}}_i(T(\xi, \eta)) \mathbf{T}_i^{-1} \mathbf{H}_i^{-1}, \\
\mathbf{S}_i(x', y') &= \mathbf{h} \Delta^{-1} \mathbf{J}^T \hat{\mathbf{S}}_i(T(\xi, \eta)) \mathbf{T}_i^{-1} \mathbf{H}_i^{-1}, \\
\mathbf{g}_i(x', y') &= \hat{\mathbf{g}}_i(T(\xi, \eta)) \bar{\mathbf{T}}_i^{-1} \bar{\mathbf{H}}_i^{-1}, \\
\mathbf{G}_i(x', y') &= \bar{\mathbf{h}} \mathbf{J}^{-1} \hat{\mathbf{G}}_i(T(\xi, \eta)) \bar{\mathbf{T}}_i^{-1} \bar{\mathbf{H}}_i^{-1},
\end{aligned}
\tag{3.6}$$

where

$$\mathbf{H}_i = \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{h}_i \end{bmatrix}, \quad \bar{\mathbf{H}}_i = \begin{bmatrix} 1 & 0 \\ 0 & \bar{\mathbf{h}}_i \end{bmatrix}.
\tag{3.7}$$

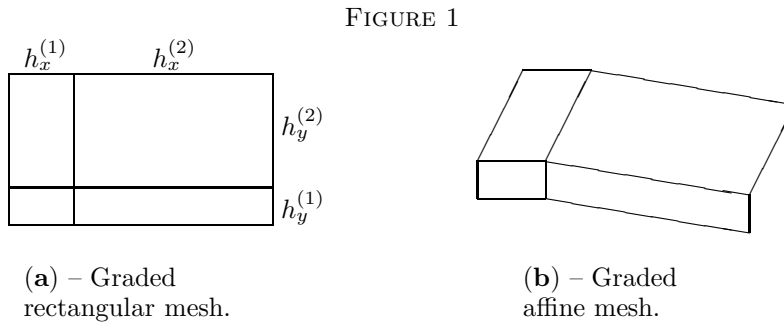
If we define  $\mathfrak{S}_i(\xi, \eta) = [\mathbf{s}_i, \mathbf{S}_i]^T$  and  $\mathfrak{G}_i(\xi, \eta) = [\mathbf{g}_i, \mathbf{G}_i]^T$ , then these can be written in the form of similarity-like transformations,

$$\begin{aligned}
\mathfrak{S}_i(x, y) &= \mathbf{H} \mathbf{T} \hat{\mathfrak{S}}_i(\xi, \eta) \mathbf{T}_i^{-1} \mathbf{H}_i^{-1}, \\
\mathfrak{G}_i(x, y) &= \bar{\mathbf{H}} \bar{\mathbf{T}} \hat{\mathfrak{G}}_i(\xi, \eta) \bar{\mathbf{T}}_i^{-1} \bar{\mathbf{H}}_i^{-1}.
\end{aligned}
\tag{3.8}$$

#### 4. GEOMETRIC TRANSFORMATION AND MESH GENERATION

To take advantage of these compact forms, the Jacobian matrix or first derivatives must be known at the four nodes of each subdomain. In many cases it is more convenient and efficient to precompute  $\mathbf{J}$  and store it with the nodal data, as will be discussed later. However, even for simple graded rectangular meshes or grids (and for unstructured quadrilateral meshes), there is no single  $\mathbf{J}$  associated with each node. To see this, refer to Figure 1a. On each rectangular subdomain or element,  $\mathbf{J}$  is constant, given by  $\mathbf{J} = \begin{bmatrix} h_x/2 & 0 \\ 0 & h_y/2 \end{bmatrix}$ . Thus it is different for each of the four elements sharing the internal node. So for graded rectangular meshes, it may be more effective to use the implicit forms of Section 2, and compute  $h_x, h_y$  on each rectangle from the nodal coordinates as needed, or the Jacobian can be associated with the element rather than the node. While the four Jacobians at the node are different, the tangential components are continuous across the rectangle interfaces (as they must be) because those boundaries have common lengths. The graded affine mesh shown in Figure 1b is not rectangular, but is the result of transformations involving scaling, skewing and global rotation for each element. The

Jacobians might be pre-computed during mesh generation and the constant values associated with the elements. Finally, for general quadrilateral and unstructured meshes, compatible Jacobians might be computed and associated with the nodes of each element.



For complex geometries, one can often generate an appropriate mesh with a unique  $\mathbf{J}$  at *most* nodal points, and store the values of  $\mathbf{J}$  as part of the nodal data. These values can be interpolated to interior points of the subdomain as needed.

The Hermite geometry functions can be used to generate and interpolate the data for the mesh. As an example, consider the problem domain for fluid flow over a cylinder, where one quarter of the domain is shown in Figures 2 and 3.

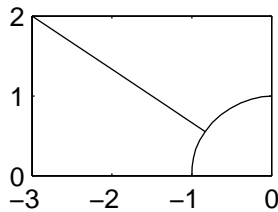


FIGURE 2. Domain partitioned into blocks

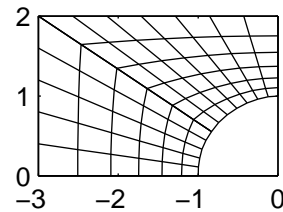


FIGURE 3. Blocks partitioned into grid

It is a common practice to subdivide the complex domain into simpler blocks. Having determined the coordinates of the nodal points at the corners of these blocks, the Jacobians (derivatives) at the nodes are adjusted until the interpolated boundaries match the problem domain boundaries, as shown in Figure 2, subject to the condition that the derivatives of the coordinate transformation (components of  $\mathbf{J}$ ) are equal on either side of each common block interface. Global nodal coordinates of the mesh nodes are computed by partitioning the reference space of each block with a (perhaps graded) grid as in Figure 3. Then using the geometry gradient function, the Jacobians are evaluated by interpolation at the internal grid nodal points. The same subdivision and grading must be used on either side of a block interface, of course. It is suggested that the nodes on block interfaces be duplicated on each block because the normal components of  $\mathbf{J}$  will likely be different on each block, and this allows one to have only one Jacobian per node. The additional degrees of freedom need not be a penalty.

For finite element applications, the stiffness matrix can be assembled separately on each block (which can be done in parallel) and copied into the global stiffness matrix. After assembly, the degrees-of-freedom on either side of the interfaces can be constrained to be equal by adjusting the stiffness matrix. The simple constraint equations can be removed so that the global matrix to be solved is smaller, and the constraints patched up after the algebraic equations are solved.

Another way to provide a mesh with the required geometric data, similar to the method described by Petera and Pittman [14], is to augment the mesh produced by a standard mesh generator. Suppose one has generated an invertible, structured quadrilateral mesh covering all or part of the problem domain. This defines a mapping from the computational space with local coordinates  $(\xi, \eta)$  to the global space with transformed coordinates  $(\hat{x}, \hat{y})$

$$(4.1) \quad \begin{aligned} \hat{x}(\xi, \eta; \mathbf{x}_j) &= \sum_e \sum_{i \in e} n_i(\xi, \eta) x_i, \\ \hat{y}(\xi, \eta; \mathbf{x}_j) &= \sum_e \sum_{i \in e} n_i(\xi, \eta) y_i. \end{aligned}$$

Now let  $(\check{x}, \check{y})$  be a piecewise Hermite mapping with the same nodal coordinates and as yet unknown derivatives  $\gamma, \lambda$ ,

$$(4.2) \quad \begin{aligned} \check{x}(\xi, \eta; \mathbf{x}_j, \gamma) &= \sum_e \sum_{i \in e} g_i^{(0)} x_i + g_i^{(1)} \gamma_i^{(1)} + g_i^{(2)} \gamma_i^{(2)}, \\ \check{y}(\xi, \eta; \mathbf{y}_j, \lambda) &= \sum_e \sum_{i \in e} g_i^{(0)} y_i + g_i^{(1)} \lambda_i^{(1)} + g_i^{(2)} \lambda_i^{(2)}. \end{aligned}$$

One then evaluates the derivatives  $\gamma, \lambda$  by minimizing the functionals,

$$(4.3) \quad \begin{aligned} F_1(\check{x}) &= \int_{\Omega} (\hat{x} - \check{x})^2 d\Omega, \\ F_2(\check{y}) &= \int_{\Omega} (\hat{y} - \check{y})^2 d\Omega. \end{aligned}$$

over the set of derivatives. This leads to a set of algebraic equations of the form

$$(4.4) \quad \begin{aligned} \mathbf{A}\gamma &= \mathbf{b}_1, \\ \mathbf{A}\lambda &= \mathbf{b}_2. \end{aligned}$$

to be solved for the derivatives or Jacobians at the nodes. This is a projection which yields the Hermite mapping which is closest, in the  $L^2$  norm, to the original invertible mapping, but an  $\mathbf{H}^1$  norm can be used as well [14].

Finally, the boundary of the computational domain can be mapped to the boundary of the physical domain. Then the seminorms  $|(\nabla \check{x})|_2$  and  $|(\nabla \check{y})|_2$  can be minimized over the mesh parameters, subject to the boundary constraint, to determine the necessary mesh parameters.

This range of meshing options provides considerable opportunities for application of the Hermite interpolation functions to interpolation or computation of divergence-free fields. This also illustrates how a general object-oriented implementation may need multiple options for nodal/element data association.

## 5. BICUBIC HERMITE INTERPOLATIONS

We now consider the classical bicubic Hermite interpolation function, which is formed as the tensor product of one-dimensional Hermite functions, casting it into the form of the previous Hermite functions. The degrees of freedom are the scalar

field, the two components of its gradient, and the second cross-derivative. The function is given by,

$$(5.1) \quad \mathbf{g}_i^b(x, y) = \begin{bmatrix} \frac{1}{16}(1 + \xi_0)^2(1 + \eta_0)^2 & -\frac{1}{16}\xi_i^{-1}(1 + \xi_0)^2(1 + \eta_0)^2 \\ (2 - \xi_0)(2 - \eta_0) & (1 - \xi_0)(2 - \eta_0) \\ -\frac{1}{16}\eta_i^{-1}(1 + \xi_0)^2(1 + \eta_0)^2 & \frac{1}{16}\xi_i^{-1}\eta_i^{-1}(1 + \xi_0)^2(1 + \eta_0)^2 \\ (2 - \xi_0)(1 - \eta_0) & (1 - \xi_0)(1 - \eta_0) \end{bmatrix}.$$

This function is of sixth degree with sixteen degrees-of-freedom, and it exactly interpolates all scalar fields through third degree on rectangular subdomains. The earlier Hermite function was also exact through third degree, but was only of fourth degree with twelve degrees-of-freedom. What is gained with this added complexity is that *both* components of the gradient are continuous on subdomain interfaces.

Following the preceding work, this function will be used for geometric transformations, but may be referred to as a *potential* function. The gradient is given by,

$$(5.2) \quad \mathbf{G}_i^b(x, y) = \begin{bmatrix} \frac{3}{16}\xi_i(1 + \eta_0)^2 & -\frac{1}{16}(1 + \eta_0)^2(1 + \xi_0) \\ (2 - \eta_0)(1 - \xi^2) & (2 - \eta_0)(1 - 3\xi_0) \\ \frac{3}{16}\eta_i(1 + \xi_0)^2 & -\frac{3}{16}\eta_i\xi_i^{-1}(1 + \xi_0)^2 \\ (2 - \xi_0)(1 - \eta^2) & (1 - \eta^2)(1 - \xi_0) \\ -\frac{3}{16}\xi_i\eta_i^{-1}(1 + \eta_0)^2 & \frac{1}{16}\eta_i^{-1}(1 + \xi_0)(1 + \eta_0) \\ (1 - \eta_0)(1 - \xi^2) & (1 - 3\xi_0)(1 - \eta^2) \\ -\frac{1}{16}(1 + \eta_0)(1 + \xi_0)^2 & \frac{1}{16}\xi_i^{-1}(1 + \xi_0)(1 + \eta_0) \\ (1 - 3\eta_0)(2 - \xi_0) & (1 - \xi^2)(1 - 3\eta_0) \end{bmatrix}.$$

It will be also convenient to have the function for interpolating cross derivatives,

$$(5.3) \quad \mathbf{g}_{i,xy}^b(x, y) = \begin{bmatrix} \frac{9}{16}\xi_i\eta_i(1 - \xi^2)(1 - \eta^2) & -\frac{3}{16}\eta_i(1 + \xi_0)(1 - \eta^2) \\ (1 - 3\xi_0) & \\ -\frac{3}{16}\xi_i(1 - \xi^2)(1 + \eta_0) & \frac{1}{16}(1 + \xi_0)(1 + \eta_0) \\ (1 - 3\eta_0) & (1 - 3\xi_0)(1 - 3\eta_0) \end{bmatrix}.$$

A cubic-complete/exact *stream* function, with stream function, curl components, and second cross derivative degrees-of-freedom is found by interchanging the second and third (derivative) columns, along with a sign change, to give,

$$(5.4) \quad \mathbf{s}_i^b(x, y) = \begin{bmatrix} \frac{1}{16}(1 + \xi_0)^2(1 + \eta_0)^2 & -\frac{1}{16}\eta_i^{-1}(1 + \xi_0)^2(1 + \eta_0)^2 \\ (2 - \xi_0)(2 - \eta_0) & (2 - \xi_0)(1 - \eta_0) \\ \frac{1}{16}\xi_i^{-1}(1 + \xi_0)^2(1 + \eta_0)^2 & \frac{1}{16}\xi_i^{-1}\eta_i^{-1}(1 + \xi_0)^2(1 + \eta_0)^2 \\ (1 - \xi_0)(2 - \eta_0) & (1 - \xi_0)(1 - \eta_0) \end{bmatrix}.$$

This also exactly interpolates all scalar (stream function) fields of third degree. Taking the curl of this function gives an interpolation function for divergence-free

vector fields,

$$(5.5) \quad \mathbf{S}_i^b(x, y) = \begin{bmatrix} \frac{3}{16}\eta_i(1+\xi_0)^2 & -\frac{1}{16}(1+\xi_0)^2(1+\eta_0) \\ (2-\xi_0)(1-\eta^2) & (2-\xi_0)(1-3\eta_0) \\ -\frac{3}{16}\xi_i(1+\eta_0)^2 & \frac{3}{16}\xi_i\eta_i^{-1}(1+\eta_0)^2 \\ (2-\eta_0)(1-\xi^2) & (1-\xi^2)(1-\eta_0) \\ \frac{3}{16}\eta_i\xi_i^{-1}(1+\xi_0)^2 & \frac{1}{16}\xi_i^{-1}(1+\xi_0)(1+\eta_0) \\ (1-\xi_0)(1-\eta^2) & (1-\xi^2)(1-3\eta_0) \\ -\frac{1}{16}(1+\xi_0)(1+\eta_0)^2 & -\frac{1}{16}\eta_i^{-1}(1+\xi_0)(1+\eta_0) \\ (1-3\xi_0)(2-\eta_0) & (1-3\xi_0)(1-\eta^2) \end{bmatrix},$$

but the cross-derivative is not problem data. The interpolations are continuous across subdomain interfaces, and so is divergence-free there too. As before, the geometric quantities can be factored out to give,

$$(5.6) \quad \mathbf{S}_i^b(x, y) = \xi_h \eta_h \begin{bmatrix} \xi_h^{-1} & 0 \\ 0 & \eta_h^{-1} \end{bmatrix} \hat{\mathbf{S}}_i^b(T(\xi, \eta)) \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \eta_h & 0 & 0 \\ 0 & 0 & \xi_h & 0 \\ 0 & 0 & 0 & \xi_h^{-1} \eta_h^{-1} \end{bmatrix},$$

where, as before,  $\hat{\mathbf{S}}_i^b$  is defined on  $[-1, 1]^2$ . This suggests the generalizations,

$$(5.7) \quad \begin{aligned} \mathbf{s}_i^b(x, y) &= \hat{\mathbf{s}}_i^b(T(\xi, \eta)) (\mathbf{T}_i^b)^{-1}, \\ \mathbf{S}_i^b(x, y) &= \Delta^{-1} \mathbf{J} \hat{\mathbf{S}}_i^b(T(\xi, \eta)) (\mathbf{T}_i^b)^{-1}, \\ (\mathbf{T}_i^b)^{-1} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & y_{,\eta} & -x_{,\eta} & 0 \\ 0 & -y_{,\xi} & x_{,\xi} & 0 \\ 0 & y_{,\xi\eta} & -x_{,\xi\eta} & (x_{,\xi}y_{,\eta} + y_{,\xi}x_{,\eta}) \end{bmatrix}, \end{aligned}$$

obtained by the chain rule, where  $(\mathbf{T}_i^b)^{-1}$  has been expressed in terms of computable quantities. If the block-meshing strategy using (5.1), (5.2) and (5.3) is used, all of these quantities are available at each node.

The scalar function  $\mathbf{s}_i^b$  remains complete, but exactly interpolates only linear scalar fields on general affine meshes because of the missing second derivative degrees-of-freedom. Yet it might approximate these fields quite well, so its effectiveness for these meshes needs to be evaluated.

## 6. DERIVATION OF HERMITE FUNCTIONS

It is instructive to understand how solenoidal Hermite interpolation functions might be derived. In the earlier paper [9], some constant-divergence Lagrange interpolation functions were introduced. A set of these functions forms a basis for the Hermite functions.

First, there are two vector-valued functions on the corner nodes of a rectangle,

$$(6.1) \quad \mathbf{N}_i^{4D}(\xi, \eta) = \begin{bmatrix} \frac{1}{4}(1+\xi_0)(1+\eta_0) & \frac{1}{8}\eta_i\xi_i^{-1}(1-\xi^2) \\ \frac{1}{8}\xi_i\eta_i^{-1}(1-\eta^2) & \frac{1}{4}(1+\xi_0)(1+\eta_0) \end{bmatrix} \begin{matrix} \xi_i = \pm 1, \\ \eta_i = \pm 1, \end{matrix}$$

Then there are two functions corresponding to normal flow, defined on mid-side nodes,

$$(6.2) \quad \mathbf{N}_i^{\text{SD}}(\xi, \eta) = \begin{cases} \begin{bmatrix} \cdots & \cdots \\ \frac{1}{2}(1 + \xi_0)(1 - \eta^2) & 0 \\ -\frac{1}{6}\xi_i\eta_h^{-1}\eta(1 - \eta^2) & 0 \end{bmatrix} & \begin{matrix} \xi_i = \pm 1, \\ \eta_i = 0, \end{matrix} \\ \begin{bmatrix} 0 & -\frac{1}{6}\eta_i\xi_h^{-1}\xi(1 - \xi^2) \\ 0 & \frac{1}{2}(1 + \eta_0)(1 - \xi^2) \end{bmatrix} & \begin{matrix} \xi_i = 0, \\ \eta_i = \pm 1. \end{matrix} \end{cases}$$

We rename the parts as follows,

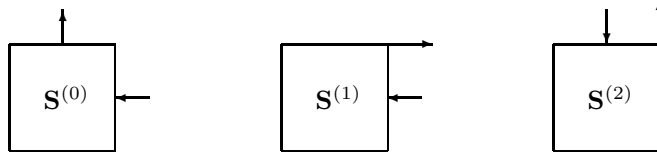
$$(6.3) \quad \begin{aligned} \mathbf{n}_1 &= \begin{bmatrix} \frac{1}{4}(1 + \xi_0)(1 + \eta_0) \\ \frac{1}{8}\xi_i\eta_i^{-1}(1 - \eta^2) \end{bmatrix} & \begin{matrix} \xi_i = \pm 1, \\ \eta_i = \pm 1, \end{matrix} & \text{div } \mathbf{n}_1 &= \frac{1}{4}\xi_i, \\ \mathbf{n}_2 &= \begin{bmatrix} \frac{1}{8}\eta_i\xi_i^{-1}(1 - \xi^2) \\ \frac{1}{4}(1 + \xi_0)(1 + \eta_0) \end{bmatrix} & \begin{matrix} \xi_i = \pm 1, \\ \eta_i = \pm 1, \end{matrix} & \text{div } \mathbf{n}_2 &= \frac{1}{4}\eta_i, \\ \mathbf{m}_1 &= \begin{bmatrix} \frac{1}{2}(1 + \xi_0)(1 - \eta^2) \\ -\frac{1}{6}\xi_i\eta_h^{-1}\eta(1 - \eta^2) \end{bmatrix} & \begin{matrix} \xi_i = \pm 1, \\ \eta_i = 0, \end{matrix} & \text{div } \mathbf{m}_1 &= \frac{1}{3}\xi_i, \\ \mathbf{m}_2 &= \begin{bmatrix} -\frac{1}{6}\eta_i\xi_h^{-1}\xi(1 - \xi^2) \\ \frac{1}{2}(1 + \eta_0)(1 - \xi^2) \end{bmatrix} & \begin{matrix} \xi_i = 0, \\ \eta_i = \pm 1, \end{matrix} & \text{div } \mathbf{m}_2 &= \frac{1}{3}\eta_i. \end{aligned}$$

Then the following combinations,

$$(6.4) \quad \begin{aligned} \mathbf{S}^{(0)} &= \frac{3}{4}\eta_i\mathbf{m}_1 - \frac{3}{4}\xi_i\mathbf{m}_2, \\ \mathbf{S}^{(1)} &= \mathbf{n}_1 - \frac{3}{4}\mathbf{m}_1, \\ \mathbf{S}^{(2)} &= \mathbf{n}_2 - \frac{3}{4}\mathbf{m}_2, \end{aligned}$$

are divergence-free, and are the three columns of (2.4). This construction is shown schematically in Figure 4. This approach is similar to that of Griffiths [7]. One then integrates this by inspection to find the scalar function (2.2) whose curl yields the terms (6.4).

FIGURE 4. Degrees-of-freedom used to construct  $\mathbf{S}_i$



An alternate way [8] of deriving this result is to consider a  $2 \times 2$  mesh of four rectangles with nodes at the center of each of the four internal edges. Applying homogeneous boundary conditions and using the basis functions (6.3) with constant test or weight functions, one assembles the discrete divergence matrix. One then finds a basis for the null space of the matrix. The dimension of the null space is three, six degrees of freedom minus three constraints. The combinations (6.4) form a basis for the null space.

The form (2.6) can be found by interchanging the two columns on the right along with a change in sign of one column as described in [3]. An independent

development of (2.6) proceeds as follows. From (I.2.5) and (I.3.2), (I refers to [9]) one defines the functions,

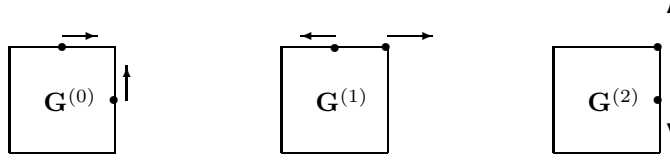
$$(6.5) \quad \begin{aligned} \mathbf{n}_1 &= \begin{bmatrix} \frac{1}{4}(1 + \xi_0)(1 + \eta_0) \\ -\frac{1}{8}\eta_i\xi_i^{-1}(1 - \xi^2) \end{bmatrix} \begin{array}{l} \xi_i = \pm 1, \\ \eta_i = \pm 1, \end{array} & \nabla \times \mathbf{n}_1 &= \frac{1}{4}\eta_i, \\ \mathbf{n}_2 &= \begin{bmatrix} -\frac{1}{8}\xi_i\eta_i^{-1}(1 - \eta^2) \\ \frac{1}{4}(1 + \xi_0)(1 + \eta_0) \end{bmatrix} \begin{array}{l} \xi_i = \pm 1, \\ \eta_i = \pm 1, \end{array} & \nabla \times \mathbf{n}_2 &= -\frac{1}{4}\xi_i, \\ \mathbf{m}_1 &= \begin{bmatrix} \frac{1}{2}(1 + \eta_0)(1 - \xi^2) \\ \frac{1}{6}\eta_i\xi_i^{-1}\xi(1 - \xi^2) \end{bmatrix} \begin{array}{l} \xi_i = 0, \\ \eta_i = \pm 1, \end{array} & \nabla \times \mathbf{m}_1 &= \frac{1}{3}\eta_i, \\ \mathbf{m}_2 &= \begin{bmatrix} \frac{1}{6}\xi_i\eta_i^{-1}\eta(1 - \eta^2) \\ \frac{1}{2}(1 + \xi_0)(1 - \eta^2) \end{bmatrix} \begin{array}{l} \xi_i = \pm 1, \\ \eta_i = 0, \end{array} & \nabla \times \mathbf{m}_2 &= -\frac{1}{3}\xi_i. \end{aligned}$$

Then irrotational combinations give the gradient vector form,

$$(6.6) \quad \mathbf{G}_i = \left[ \frac{3}{4}(\xi_i\mathbf{m}_1 + \eta_i\mathbf{m}_2) \quad \mathbf{n}_1 - \frac{3}{4}\mathbf{m}_1 \quad \mathbf{n}_2 - \frac{3}{4}\mathbf{m}_2 \right],$$

which can be integrated to give (2.6). This construction is shown schematically in Figure (5).

FIGURE 5. Degrees-of-freedom used to construct  $\mathbf{G}_i$



After deriving this function using the methods just described, the author recognized the scalar geometric function (2.6) in a classic text by O. C. Zienkiewicz ([18] p179) where it appeared in the guise of a plate element. Its origin actually goes back to a 1963 paper by R. J. Melosh [13].

The first method above has been used to find divergence-free Hermite interpolation functions in three dimensions, but generalization to three dimensions is not straight-forward. This work is still in progress and will be reported in a future paper.

The the scalar potential function (2.6) can be extended to three dimensions using the linear-curl Lagrange functions of [9]. One finds the linear-curl mid-edge

(tangential) functions,

$$(6.7) \quad \mathbf{N}_i^{20C}(x, y, z) = \begin{cases} \begin{bmatrix} \frac{1}{4}(1 + \eta_0)(1 + \zeta_0)(1 - \xi^2) & 0 & 0 \\ \frac{1}{12}\eta_i(1 + \zeta_0)\xi(1 - \xi^2) & 0 & 0 \\ \frac{1}{12}\zeta_i(1 + \eta_0)\xi(1 - \xi^2) & 0 & 0 \end{bmatrix} & \begin{array}{l} \hat{x}_i = 0, \\ \hat{y}_i, \hat{z}_i = \pm 1, \end{array} \\ \\ \begin{bmatrix} 0 & \frac{1}{12}\xi_i(1 + \zeta_0)\eta(1 - \eta^2) & 0 \\ 0 & \frac{1}{4}(1 + \xi_0)(1 + \zeta_0)(1 - \eta^2) & 0 \\ 0 & \frac{1}{12}\zeta_i(1 + \xi_0)\eta(1 - \eta^2) & 0 \end{bmatrix} & \begin{array}{l} \hat{y}_i = 0, \\ \hat{x}_i, \hat{z}_i = \pm 1, \end{array} \\ \\ \begin{bmatrix} 0 & 0 & \frac{1}{12}\xi_i(1 + \eta_0)\zeta(1 - \zeta^2) \\ 0 & 0 & \frac{1}{12}\eta_i(1 + \xi_0)\zeta(1 - \zeta^2) \\ 0 & 0 & \frac{1}{4}(1 + \xi_0)(1 + \eta_0)(1 - \zeta^2) \end{bmatrix} & \begin{array}{l} \hat{z}_i = 0, \\ \hat{x}_i, \hat{y}_i = \pm 1. \end{array} \end{cases}$$

Referring to this and equation (I.2.7), we define the quantities,

$$(6.8) \quad \begin{aligned} \mathbf{n}_1 &= \begin{bmatrix} \frac{1}{8}(1 + \xi_0)(1 + \eta_0)(1 + \zeta_0) \\ -\frac{1}{16}\eta_i\xi_i^{-1}(1 - \xi^2)(1 + \zeta_0) \\ -\frac{1}{16}\zeta_i\xi_i^{-1}(1 - \xi^2)(1 + \eta_0) \end{bmatrix} \begin{array}{l} \hat{x}_i = \pm 1, \\ \hat{y}_i = \pm 1, \\ \hat{z}_i = \pm 1, \end{array} & \nabla \times \mathbf{n}_1 = \begin{bmatrix} 0 \\ \frac{1}{8}\zeta_i(1 + \eta_0) \\ -\frac{1}{8}\eta_i(1 + \zeta_0) \end{bmatrix}, \\ \\ \mathbf{n}_2 &= \begin{bmatrix} -\frac{1}{16}\xi_i\eta_i^{-1}(1 - \eta^2)(1 + \xi_0) \\ \frac{1}{8}(1 + \xi_0)(1 + \eta_0)(1 + \zeta_0) \\ -\frac{1}{16}\zeta_i\eta_i^{-1}(1 - \eta^2)(1 + \xi_0) \end{bmatrix} \begin{array}{l} \hat{x}_i = \pm 1, \\ \hat{y}_i = \pm 1, \\ \hat{z}_i = \pm 1, \end{array} & \nabla \times \mathbf{n}_2 = \begin{bmatrix} -\frac{1}{8}\zeta_i(1 + \eta_0) \\ 0 \\ \frac{1}{8}\xi_i(1 + \zeta_0) \end{bmatrix}, \\ \\ \mathbf{n}_3 &= \begin{bmatrix} -\frac{1}{16}\xi_i\zeta_i^{-1}(1 - \zeta^2)(1 + \eta_0) \\ -\frac{1}{16}\eta_i\zeta_i^{-1}(1 - \zeta^2)(1 + \xi_0) \\ \frac{1}{8}(1 + \xi_0)(1 + \eta_0)(1 + \zeta_0) \end{bmatrix} \begin{array}{l} \hat{x}_i = \pm 1, \\ \hat{y}_i = \pm 1, \\ \hat{z}_i = \pm 1, \end{array} & \nabla \times \mathbf{n}_3 = \begin{bmatrix} \frac{1}{8}\eta_i(1 + \eta_0) \\ -\frac{1}{8}\xi_i(1 + \zeta_0) \\ 0 \end{bmatrix}, \end{aligned}$$

and,

$$(6.9) \quad \begin{aligned} \mathbf{m}_1 &= \begin{bmatrix} \frac{1}{4}(1 + \eta_0)(1 + \zeta_0)(1 - \xi^2) \\ \frac{1}{12}\eta_i(1 + \zeta_0)\xi(1 - \xi^2) \\ \frac{1}{12}\zeta_i(1 + \eta_0)\xi(1 - \xi^2) \end{bmatrix} \begin{array}{l} \hat{x}_i = 0, \\ \hat{y}_i = \pm 1, \\ \hat{z}_i = \pm 1, \end{array} & \nabla \times \mathbf{m}_1 = \begin{bmatrix} 0 \\ \frac{1}{6}\zeta_i(1 + \eta_0) \\ -\frac{1}{6}\eta_i(1 + \zeta_0) \end{bmatrix}, \\ \\ \mathbf{m}_2 &= \begin{bmatrix} \frac{1}{12}\xi_i(1 + \zeta_0)\eta(1 - \eta^2) \\ \frac{1}{4}(1 + \xi_0)(1 + \zeta_0)(1 - \eta^2) \\ \frac{1}{12}\zeta_i(1 + \xi_0)\eta(1 - \eta^2) \end{bmatrix} \begin{array}{l} \hat{y}_i = 0, \\ \hat{x}_i = \pm 1, \\ \hat{z}_i = \pm 1, \end{array} & \nabla \times \mathbf{m}_2 = \begin{bmatrix} -\frac{1}{6}\zeta_i(1 + \eta_0) \\ 0 \\ \frac{1}{6}\xi_i(1 + \zeta_0) \end{bmatrix}, \\ \\ \mathbf{m}_3 &= \begin{bmatrix} \frac{1}{12}\xi_i(1 + \eta_0)\zeta(1 - \zeta^2) \\ \frac{1}{12}\eta_i(1 + \xi_0)\zeta(1 - \zeta^2) \\ \frac{1}{4}(1 + \xi_0)(1 + \eta_0)(1 - \zeta^2) \end{bmatrix} \begin{array}{l} \hat{z}_i = 0, \\ \hat{x}_i = \pm 1, \\ \hat{y}_i = \pm 1, \end{array} & \nabla \times \mathbf{m}_3 = \begin{bmatrix} \frac{1}{6}\eta_i(1 + \eta_0) \\ -\frac{1}{6}\xi_i(1 + \zeta_0) \\ 0 \end{bmatrix}. \end{aligned}$$

One then forms the irrotational combinations,

$$(6.10) \quad \mathbf{G}_i = \left[ \frac{3}{4}(\xi_i\mathbf{m}_1 + \eta_i\mathbf{m}_2 + \zeta_i\mathbf{m}_3) \quad \mathbf{n}_1 - \frac{3}{4}\mathbf{m}_1 \quad \mathbf{n}_2 - \frac{3}{4}\mathbf{m}_2 \quad \mathbf{n}_3 - \frac{3}{4}\mathbf{m}_3 \right].$$

As the columns are irrotational, this can be written as the gradient of a potential function. Integrating, the result is,

$$(6.11) \quad \mathbf{g}_i(x, y, z) = \begin{bmatrix} \frac{1}{16}(1 + \xi_0)(1 + \eta_0)(1 + \zeta_0) & -\frac{1}{16}\xi_i^{-1}(1 + \xi_0)^2(1 + \eta_0) \\ (2 + \xi_0(1 - \xi_0) & (1 + \zeta_0)(1 - \xi_0) \\ + \eta_0(1 - \eta_0) + \zeta_0(1 - \zeta_0)) & \\ -\frac{1}{16}\eta_i^{-1}(1 + \xi_0)(1 + \eta_0)^2 & -\frac{1}{16}\zeta_i^{-1}(1 + \xi_0)(1 + \eta_0) \\ (1 + \zeta_0)(1 - \eta_0) & (1 + \zeta_0)^2(1 - \zeta_0) \end{bmatrix}.$$

This can also be (and was) inferred from the form of (2.6). It is remarkable that it is cubic-complete, the tangential components of its gradient are continuous at subdomain boundaries, and it is only of fifth degree, while the tricubic Hermite function is of ninth degree. It obviously can be generalized in the same manner as the two-dimensional function.

## 7. REVISITING THE NAVIER-STOKES EQUATION

The Navier-Stokes equation is generally recognized as the governing equation for the motion of incompressible flow. It appears to be a dynamic equation, and it is commonly argued that the role of the pressure in the equation is to enforce incompressibility. This interpretation of the equation is open to question and, from first principles, it is argued here that both of these assumptions are false.

The appearance of a pressure term in the equation of motion arises from an incorrect application of first principles in its derivation. In that derivation, one applies Newton's second law to a small volume of fluid. It is assumed that one force on the fluid volume results from a dynamic pressure difference across opposite faces of the volume, leading to a pressure gradient term. But pressure disturbances propagate at infinite speed in an incompressible medium, so no *dynamic* pressure differences can exist for a finite time, and no term of this type can appear in the *equation of motion* of the fluid.

In the absence of the pressure term, it is argued that the most fundamental governing equations are integro-differential equations of the form,

$$(7.1) \quad \begin{aligned} \frac{\partial}{\partial t} \mathbf{u} &= \pi^S(-\mathbf{u} \cdot \nabla \mathbf{u} + \nu \nabla^2 \mathbf{u}) + \mathbf{f}^S, & \mathbf{u} \cdot \mathbf{n}|_{\partial\Omega} &= 0, \\ \nabla p &= \pi^I(-\mathbf{u} \cdot \nabla \mathbf{u} + \nu \nabla^2 \mathbf{u}) + \mathbf{f}^I, \end{aligned}$$

where  $\mathbf{u}$  is the fluid velocity,  $p$  is the reduced pressure, and  $\nu$  is the kinematic viscosity. The quantities  $\mathbf{f}^S$  and  $\mathbf{f}^I$  are the non-conservative (solenoidal) and conservative (irrotational) parts of the body force  $\mathbf{f}$ , respectively. Here,  $\pi^S$  and  $\pi^I$  are linear projection operators, which project out the solenoidal and irrotational parts of a vector field, and are given formally by [6],

$$(7.2) \quad \pi^I = \nabla \Delta^{-1} \nabla \cdot, \quad \pi^S = 1 - \pi^I.$$

The operator  $\Delta^{-1}$  (the inverse Laplacian) is the Green's function for the Laplace equation on  $\Omega$ . The first equation, governing the motion of the incompressible fluid, depends only on the nonconservative part of the force and does not depend on the pressure. Likewise, the "pressure" is a functional of the irrotational part of the fluid flow and any conservative forces.

In this view, the incompressible Navier-Stokes equation is not the dynamic equation of motion it seems to be, but is rather the sum of a kinematic equation for

the fluid flow, where incompressibility plays the role of a conservation law, and an algebraic equation (in time) for the *consistent* pressure, which is a function of that flow. The pressure equation is that equation which complements or completes the momentum equation to yield the classical Navier-Stokes equation, a differential algebraic equation, given by,

$$(7.3) \quad \begin{aligned} \frac{\partial}{\partial t} \mathbf{u} &= -\mathbf{u} \cdot \nabla \mathbf{u} - \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}, \\ \nabla \cdot \mathbf{u} &= 0, \end{aligned}$$

Let  $\mathcal{V} = \{\mathbf{u} \in \mathbf{H}^1 : \operatorname{div} \mathbf{u} = 0\}$ ,  $\mathcal{W} = \{\mathbf{u} \in \mathbf{H}^1 : \operatorname{curl} \mathbf{u} = 0\}$ ,  $\mathcal{V}_0 = \{\mathbf{u} \in \mathcal{V} : \mathbf{u}|_{\partial\Omega} = 0\}$  and  $\mathcal{W}_0 = \{\mathbf{u} \in \mathcal{W} : \mathbf{u}|_{\partial\Omega} = 0\}$ . Then the equivalent weak form of (7.1) is given by,

$$(7.4) \quad \begin{aligned} (\mathbf{v}, \frac{\partial}{\partial t} \mathbf{u}) &= -(\mathbf{v}, \mathbf{u} \cdot \nabla \mathbf{u}) - \nu (\nabla \mathbf{v}, \nabla \mathbf{u}) + (\mathbf{v}, \mathbf{f}^S), \quad \mathbf{u} \in \mathcal{V}_0, \forall \mathbf{v} \in \mathcal{V}_0, \\ (\mathbf{w}, \nabla p) &= -(\mathbf{w}, \mathbf{u} \cdot \nabla \mathbf{u}) - \nu (\nabla \mathbf{w}, \nabla \mathbf{u}), \quad \mathbf{u} \in \mathcal{V}_0, p \in H^1/0, \forall \mathbf{w} \in \mathcal{W}_0, \end{aligned}$$

where  $(\cdot, \cdot)$  indicates the inner product over  $\Omega$ . Here, the orthogonality of the solenoidal and irrotational functions replaces the projection operators found in (7.1). This decomposition has been recognized for seventy years [3][11][12][15], but seems to have been dismissed as a “mathematical trick” [4].

It might seem alarming, in light of a class of problems called “pressure-driven flow”, that the flow in (7.1) and (7.4) does not depend on the pressure. What happens is that boundary conditions on the stream function (or vector potential in three-dimensions) replace the pressure gradient as controlling data in these problems. Boundary conditions on the stream function seem more reasonable if the equation of motion is cast in stream function form,

$$(7.5) \quad \frac{\partial}{\partial t} (\nabla \times \psi) = \pi^S (-\nabla \times \psi) \cdot \nabla (\nabla \times \psi) + \nu \nabla^2 (\nabla \times \psi) + \mathbf{f}^S,$$

In fact, the pressure gradient can be directly related to the boundary condition on the stream function in the special case of fully developed flows ([2] p308) by the relation  $\partial p / \partial x = c_P R_e^{-1} \Phi$ , where the flow  $\Phi = \Delta \psi$  is the difference in stream function across the channel,  $R_e$  is the Reynolds number and the geometry-dependent quantity  $c_P$  is known as the Poiseuille constant [2]. For two-dimensional flow in a duct (or between parallel plates),  $c_P = 3$ .

When pressure boundary conditions are specified so that the inverse problem is well-posed, the explicit dependence of the pressure on velocity can be inverted. Then (7.1) or (7.4) can be solved simultaneously to find the velocity as an implicit function of the pressure gradient.

## 8. SOME SIMPLE EXAMPLES

It is useful to revisit some known analytic solutions of incompressible flow in the context of this new formalism. The technique to be used here might be called “projection by inspection.” In the first example, the problem domain is  $R^2$ , with the boundary condition  $\mathbf{u} = 0$  at  $(0, y_i)$ ,  $y_i = 0, \pm 1, \pm 2, \dots$ . The stream function for the tentative solution is  $\psi = y - (2\pi)^{-1} e^{\lambda x} \sin(2\pi y)$ , with velocities,

$$(8.1) \quad u = 1 - e^{\lambda x} \cos(2\pi y), \quad v = \lambda / (2\pi) e^{\lambda x} \sin(2\pi y).$$

This solution, first proposed by L.I.G. Kovasznay [10] and known as Kovasznay flow, is identified with steady incompressible laminar flow behind a grid. A simple

calculation gives,

$$(8.2) \quad \begin{aligned} \mathbf{u} \cdot \nabla \mathbf{u} &= \begin{bmatrix} -\lambda e^{\lambda x} \cos(2\pi y) \\ \lambda^2/(2\pi) e^{\lambda x} \sin(2\pi y) \end{bmatrix} + \begin{bmatrix} \lambda e^{2\lambda x} \\ 0 \end{bmatrix}, \\ \nu \nabla^2 \mathbf{u} &= \begin{bmatrix} (4\pi^2 - \lambda^2) e^{\lambda x} \cos(2\pi y) \\ -\lambda/(2\pi) (4\pi^2 - \lambda^2) e^{\lambda x} \sin(2\pi y) \end{bmatrix}. \end{aligned}$$

The first term of the convection is solenoidal and the second irrotational, while the diffusion term is purely solenoidal. Applying (7.1),

$$(8.3) \quad \begin{aligned} \pi^S(-\mathbf{u} \cdot \nabla \mathbf{u} + \nu \nabla^2 \mathbf{u}) &= (\lambda - \nu(4\pi^2 - \lambda^2)) \begin{bmatrix} e^{\lambda x} \cos(2\pi y) \\ -\lambda/(2\pi) e^{\lambda x} \sin(2\pi y) \end{bmatrix} = 0, \\ \pi^I(-\mathbf{u} \cdot \nabla \mathbf{u} + \nu \nabla^2 \mathbf{u}) &= \begin{bmatrix} -\lambda e^{2\lambda x} \\ 0 \end{bmatrix} = \nabla p, \end{aligned}$$

The requirement that the residual of the momentum term vanish leads to the condition  $\lambda = Re/2 \mp \sqrt{Re^2/4 + 4\pi^2}$ , where the solution with the upper (minus) sign is bounded (and vanishes) at  $x = +\infty$ , while with the lower (minus) sign it is bounded at  $x = -\infty$ . In this expression we have made the replacement  $Re = 1/\nu$ .

The pressure gradient derives from the irrotational residual of the convection term. With the boundary condition  $p(0, y) = 0$  the pressure is given by  $p = \frac{1}{2}(1 - e^{2\lambda x})$ .

For the second example, consider the internal problem domain bounded by  $y = \pm 1$  and the stream function  $\psi = V_0 y(1 - y^2/3)$ . This gives the tentative velocities,

$$(8.4) \quad \mathbf{u} = \begin{bmatrix} V_0(1 - y^2) \\ 0 \end{bmatrix}.$$

This trial solution is associated with fully-developed flow in a channel. Again, by simple computation, we find the decompositions,

$$(8.5) \quad \nu \nabla^2 \mathbf{u} = \begin{bmatrix} -\alpha 2\nu V_0 \\ 0 \end{bmatrix} + \begin{bmatrix} -(1 - \alpha) 2\nu V_0 \\ 0 \end{bmatrix}, \quad \mathbf{u} \cdot \nabla \mathbf{u} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

The arbitrariness in the form of the diffusion term reflects the fact that it could derive from a curl or a gradient, i.e. it could be solenoidal or irrotational. Applying (7.1) again,

$$(8.6) \quad \begin{aligned} \pi^S(-\mathbf{u} \cdot \nabla \mathbf{u} + \nu \nabla^2 \mathbf{u}) &= \begin{bmatrix} -\alpha 2\nu V_0 \\ 0 \end{bmatrix}, \\ \pi^I(-\mathbf{u} \cdot \nabla \mathbf{u} + \nu \nabla^2 \mathbf{u}) &= \begin{bmatrix} -(1 - \alpha) 2\nu V_0 \\ 0 \end{bmatrix} = \nabla p, \end{aligned}$$

Requiring the residual of the momentum term to vanish gives  $\alpha = 0$ , from which  $\partial p/\partial x = -2\nu V_0$ . From this it is usual to assert that the magnitude of the flow depends on the pressure gradient.

However, in light of the relation between the flow and the stream function and that the equation of fluid motion does not depend on the pressure, the flow could more properly be expressed in terms of a boundary condition, the difference of the stream function values across the channel,  $\phi = \Delta\psi$ . In this case  $V_0 = \frac{2}{3}\phi$ . Clearly these two views are related by  $\partial p/\partial x = -3\nu\phi$ .

In the first example the pressure gradient derived from the irrotational residual of the momentum term, and in the second from the residual of the diffusion term.

## 9. THE ALGEBRAIC PROBLEM

Let the problem domain  $\Omega$  with boundary  $\partial\Omega$  be partitioned into non-overlapping affine quadrilateral subdomains  $\Omega^e$  with  $h$  a measure of the diameter of these subdomains, and such that  $\Omega \approx \Omega^h = \cup_e \Omega^e$ . Let  $\mathcal{V}^h = \{u^h \in \mathbf{H}^1 : \operatorname{div} u^h = 0\}$ ,  $\mathcal{W}^h = \{u^h \in \mathbf{H}^1 : \operatorname{curl} u^h = 0\}$ ,  $\mathcal{V}_0^h = \{u^h \in \mathcal{V}^h : u^h|_{\partial\Omega} = 0\}$  and  $\mathcal{W}_0^h = \{u^h \in \mathcal{W}^h : u^h|_{\partial\Omega} = 0\}$ . The discrete weak form of (7.4) is then,

$$(9.1) \quad \begin{aligned} (\mathbf{v}^h, \frac{\partial}{\partial t} \mathbf{u}^h) &= -(\mathbf{v}^h, \mathbf{u}^h \cdot \nabla \mathbf{u}^h) - \nu(\nabla \mathbf{v}^h, \nabla \mathbf{u}^h) + (\mathbf{v}^h, \mathbf{f}^S), \\ \mathbf{u}^h &\in \mathcal{V}_0^h, \forall \mathbf{v}^h \in \mathcal{V}_0^h, \\ (\mathbf{w}^h, \nabla p^h) &= -(\mathbf{w}^h, \mathbf{u}^h \cdot \nabla \mathbf{u}^h) - \nu(\nabla \mathbf{w}^h, \nabla \mathbf{u}^h) + (\mathbf{w}^h, \mathbf{f}^I), \\ \mathbf{u}^h &\in \mathcal{V}_0^h, p \in H^1, \forall \mathbf{w}^h \in \mathcal{W}_0^h. \end{aligned}$$

Approximating  $\mathcal{V}^h$  by  $\operatorname{span}\{\mathbf{S}_i\}$  and  $\mathcal{W}^h$  by  $\operatorname{span}\{\mathbf{G}_i\}$  (which lie in  $H(\operatorname{div}, \Omega^h)$  and  $H(\operatorname{curl}, \Omega^h)$  respectively) results in the algebraic equations,

$$(9.2) \quad \begin{aligned} \mathbf{M}_{ij} \gamma_{j,t} &= -\mathbf{C}_{ij}(\mathbf{u}^h) \gamma_j + \nu \mathbf{D}_{ij} \gamma_j + \mathbf{F}_i, \\ \bar{\mathbf{M}}_{ij} \lambda_j &= -\bar{\mathbf{C}}_{ij}(\mathbf{u}^h) \lambda_j + \nu \bar{\mathbf{D}}_{ij} \lambda_j + \bar{\mathbf{F}}_i, \end{aligned}$$

where  $\gamma_i = [\psi_i, u_i, v_i]^\sim$  and  $\lambda_i = [p_i, p_{i,x}, p_{i,y}]^\sim$ . The global matrices are assembled from,

$$(9.3) \quad \begin{aligned} \mathbf{M}_{ij}^e &= \int_{\Omega^e} \mathbf{S}_i^T \mathbf{S}_j d\Omega, & \mathbf{C}_{ij}^e(\mathbf{u}^h) &= \int_{\Omega^e} \mathbf{S}_i^T (\mathbf{u}^h \cdot \nabla) \mathbf{S}_j d\Omega, \\ \mathbf{D}_{ij}^e &= - \int_{\Omega^e} (\nabla_k \mathbf{S}_i)^T \nabla_k \mathbf{S}_j d\Omega, & \mathbf{F}_i^e &= \int_{\Omega^e} \mathbf{S}_i^T \mathbf{f}^S d\Omega. \end{aligned}$$

and

$$(9.4) \quad \begin{aligned} \bar{\mathbf{M}}_{ij}^e &= \int_{\Omega^e} \mathbf{G}_i^T \mathbf{G}_j d\Omega, & \bar{\mathbf{C}}_{ij}^e(\mathbf{u}^h) &= \int_{\Omega^e} \mathbf{G}_i^T (\mathbf{u}^h \cdot \nabla) \mathbf{S}_j d\Omega, \\ \bar{\mathbf{D}}_{ij}^e &= - \int_{\Omega^e} (\nabla_k \mathbf{G}_i)^T \nabla_k \mathbf{S}_j d\Omega, & \bar{\mathbf{F}}_i^e &= \int_{\Omega^e} \mathbf{G}_i^T \mathbf{f}^I d\Omega. \end{aligned}$$

In the following example we assume two-dimensional Cartesian coordinates and an affine mesh. The general non-affine case would involve derivatives of the Jacobian matrix, determinants and the matrix  $\mathbf{h}$ .

Substituting the general forms of the interpolation functions and transforming to integrate over the square  $[-1, 1]^2$ ,

$$(9.5) \quad \begin{aligned} \mathbf{M}_{ij}^e &= \int_{-1}^1 d\xi \int_{-1}^1 d\eta \Delta (\Delta^{-1} \mathbf{J}^T \hat{\mathbf{S}}_i(\xi, \eta) \mathbf{T}_i^{-1})^T \Delta^{-1} \mathbf{J}^T \hat{\mathbf{S}}_j(\xi, \eta) \mathbf{T}_j^{-1}), \\ \mathbf{C}_{ij}^e(\mathbf{U}) &= \int_{-1}^1 d\xi \int_{-1}^1 d\eta \Delta (\Delta^{-1} \mathbf{J}^T \hat{\mathbf{S}}_i(\xi, \eta) \mathbf{T}_i^{-1})^T \{U_0 \hat{\mathbf{S}}_{j,x}(\xi, \eta) + V_0 \hat{\mathbf{S}}_{j,y}(\xi, \eta)\}, \\ \mathbf{D}_{ij}^e &= - \int_{-1}^1 d\xi \int_{-1}^1 d\eta \Delta (\hat{\mathbf{S}}_{i,x}^T(\xi, \eta) \hat{\mathbf{S}}_{j,x}(\xi, \eta) + \hat{\mathbf{S}}_{i,y}^T(\xi, \eta) \hat{\mathbf{S}}_{j,y}(\xi, \eta)), \\ \mathbf{F}_i^e &= \int_{-1}^1 d\xi \int_{-1}^1 d\eta \Delta (\Delta^{-1} \mathbf{J}^T \hat{\mathbf{S}}_i(\xi, \eta) \mathbf{T}_i^{-1})^T \mathbf{f}^S, \end{aligned}$$

where,

$$(9.6) \quad \begin{aligned} \hat{\mathbf{S}}_{k,x}(\xi, \eta) &= \Delta^{-1} \mathbf{J}^T (J_{11}^{-1} \hat{\mathbf{S}}_{k,\xi}(\xi, \eta) + J_{21}^{-1} \hat{\mathbf{S}}_{k,\eta}(\xi, \eta)) \mathbf{T}_j \\ \hat{\mathbf{S}}_{k,y}(\xi, \eta) &= \Delta^{-1} \mathbf{J}^T (J_{12}^{-1} \hat{\mathbf{S}}_{k,\xi}(\xi, \eta) + J_{22}^{-1} \hat{\mathbf{S}}_{k,\eta}(\xi, \eta)) \mathbf{T}_j \end{aligned} \quad k = i, j.$$

Likewise,

$$\begin{aligned}
\bar{\mathbf{M}}_{ij}^e &= \int_{-1}^1 d\xi \int_{-1}^1 d\eta \Delta ((\mathbf{J}^{-1} \hat{\mathbf{G}}_i(\xi, \eta) \bar{\mathbf{T}}_i^{-1})^T \Delta^{-1} \mathbf{J}^T \hat{\mathbf{S}}_j(\xi, \eta) \mathbf{T}_j^{-1}), \\
\bar{\mathbf{C}}_{ij}^e(\mathbf{U}) &= \int_{-1}^1 d\xi \int_{-1}^1 d\eta \Delta (\mathbf{J}^{-1} \hat{\mathbf{G}}_i(\xi, \eta) \bar{\mathbf{T}}_i)^T \{U_0 \hat{\mathbf{S}}_{j,x}(\xi, \eta) + V_0 \hat{\mathbf{S}}_{j,y}(\xi, \eta)\}, \\
\bar{\mathbf{D}}_{ij}^e &= - \int_{-1}^1 d\xi \int_{-1}^1 d\eta \Delta (\hat{\mathbf{G}}_{i,x}^T(\xi, \eta) \hat{\mathbf{S}}_{j,x}(\xi, \eta) + \hat{\mathbf{G}}_{i,y}^T(\xi, \eta) \hat{\mathbf{S}}_{j,y}(\xi, \eta)), \\
\bar{\mathbf{F}}_i^e &= \int_{-1}^1 d\xi \int_{-1}^1 d\eta \Delta ((\mathbf{J}^{-1} \hat{\mathbf{G}}_i(\xi, \eta) \bar{\mathbf{T}}_i^{-1})^T \mathbf{f}^I),
\end{aligned}
\tag{9.7}$$

where,

$$\begin{aligned}
\hat{\mathbf{G}}_{k,x}(\xi, \eta) &= \mathbf{J}^{-1} (J_{11}^{-1} \hat{\mathbf{G}}_{k,\xi}(\xi, \eta) + J_{21}^{-1} \hat{\mathbf{G}}_{k,\eta}(\xi, \eta)) \bar{\mathbf{T}}_k^{-1} \\
\hat{\mathbf{G}}_{k,y}(\xi, \eta) &= \mathbf{J}^{-1} (J_{12}^{-1} \hat{\mathbf{G}}_{k,\xi}(\xi, \eta) + J_{22}^{-1} \hat{\mathbf{G}}_{k,\eta}(\xi, \eta)) \bar{\mathbf{T}}_k^{-1} \quad k = i, j.
\end{aligned}
\tag{9.8}$$

The expressions above can obviously be simplified. The choice of the Hermite functions  $\mathbf{g}_i$  as the basis for the pressure field results in “balanced” equations. Of course the pressure need not be evaluated at each time step (if at all) in unsteady problems, so the cost of the pressure formulation above need not be too great.

One could have used a simpler basis such as bilinear Lagrange functions. One has complete freedom to choose a pressure basis, though one would hope to see no more piecewise-constant “pressures.”

## 10. APPLICATIONS

The following illustrates application of the Hermite interpolation functions to incompressible flow with two finite element examples. The first example is the step-diffuser or backward-facing step. A steady, viscous, incompressible fluid flowing through a duct or channel or between two parallel plates experiences a sudden change in the duct width (a step). As the Reynolds number increases (velocity increases) a recirculation bubble forms behind the step and grows in length. The example assumes  $\text{Re} = 50$ .

This example is of the class of problems called “pressure-driven” flows, but the pressureless equation of motion (9.1) was used in its solution. The problem domain is partitioned by a graded rectangular mesh. The boundary conditions on the walls are  $\mathbf{u} = 0$  and  $\psi = \text{constant}$ . No boundary conditions are applied on the outflow surface (open boundary conditions). The volume of flow (and the Reynolds number) is determined by the difference in the stream function across the duct.

In Figures 6 and 7 a Dirichlet boundary condition is applied to the inlet surface. The inlet flow is assumed to be fully-developed with a parabolic velocity profile and consistent stream function profile specified. Figure 6 shows the stream lines in a computational domain which includes the first recirculation bubble. In Figure 7, the domain has been truncated through the bubble. The remaining part of the bubble is visually indistinguishable from the corresponding part of the full bubble.

In Figure 8, the Dirichlet boundary condition has been removed at the inlet, and no boundary condition is applied at the inflow surface. In this case, fully-developed flow results in the inlet region, and the flow is visually-indistinguishable from the

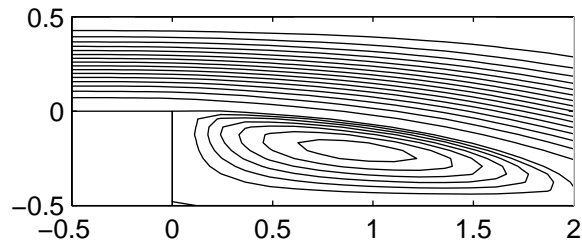


FIGURE 6. Step diffuser, full mesh, Dirichlet inflow BC.

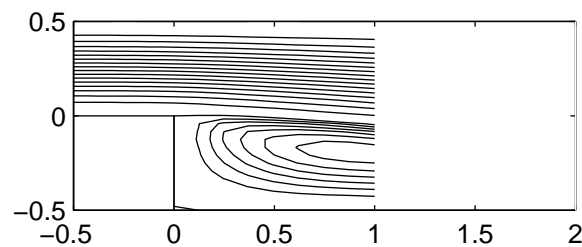


FIGURE 7. Step diffuser, truncated mesh.

Dirichlet case. It is remarkable how robust the results are for this problem with respect to domain truncation and boundary conditions.

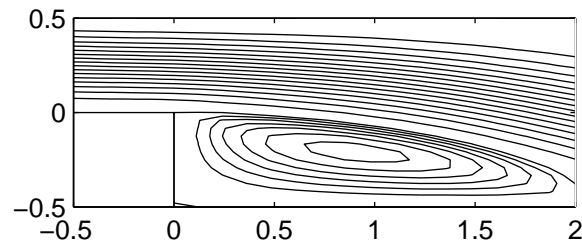


FIGURE 8. Step diffuser, open inflow and outflow BC.

The second problem is steady Stokes flow over a cylinder, and illustrates the fitting of curved boundaries. The problem domain was partitioned by unstructured blocks, which were then partitioned by a structured mesh in the manner shown in Figure 3.

The stiffness matrix was assembled on each block, and then the degrees-of-freedom on either side of the block interfaces were constrained to be equal before solving the system of equations. The stream lines of the computed flow are shown in Figure 9.

## 11. CONCLUDING REMARKS

There are consistency conditions associated with the existence of solutions to the incompressible flow equation. It is required that the net flow into the problem

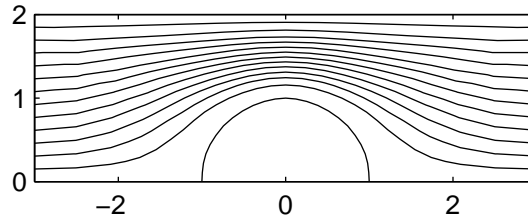


FIGURE 9. Stokes flow over a cylinder.

domain vanish,

$$(11.1) \quad \int_{\partial\Omega} \mathbf{u} \cdot \mathbf{n} = 0.$$

This condition is always satisfied by these Hermite interpolations in Cartesian coordinates. Potentially troubling is the condition,

$$(11.2) \quad \psi(x) - \psi(0) = \int_0^x u_n dl$$

on the boundary when Dirichlet conditions are applied to the stream function *and* velocities. Of the three components of  $\mathbf{S}_i$ , the stream function degree-of-freedom determines the net flow between the nodes. The velocity degrees-of-freedom fix the flow at the nodes, but do not contribute to the net flow. If inconsistent, the velocity will not be smooth, and leakage can even occur.

As remarked earlier, the solenoidal interpolation functions/elements presented here display potential tangential or slip discontinuities between the nodes. This lack of control at boundaries seems (almost) unavoidable. But this may not be perceived as the difficulty it once was. Indeed, the development and study of discontinuous functions, where the *normal* component of flow is only weakly continuous, seems popular now. In fact, the potential for development of discontinuities increases the compliance of the interpolation, and so may suppress to some extent, or delay the onset of oscillations.

The unisolvence sets of the vector functions are quadratically complete and remain so under affine transformations, as can be shown directly [14]. Thus, if the field being interpolated is sufficiently smooth, the tangential discontinuities may not be manifest.

Suppose the vector field being interpolated is sufficiently smooth that at any point  $p$ , for any tolerance  $\epsilon > 0$  there exists a disk (ball) of radius  $\delta$  where the contribution to the Taylor expansion of the vector field about  $p$  from the third and higher order derivatives is less than  $\epsilon$  so that these terms can be neglected. Then if the mesh is refined sufficiently that each adjacent pair of elements lies in some such disk (ball), then the interpolated field will be continuous within error  $\epsilon$ . Consequently, with sufficient mesh refinement the slip discontinuities can be made negligible, and the presence of non-negligible discontinuities signal the need for further mesh refinement.

The above argument fails near the boundary of the problem domain, where the higher order derivatives on the boundary of the interpolated field are unlikely to

vanish. A method for resolution of this problem was discussed in the first paper [9].

An obvious area of application of these Hermite functions  $\{\mathbf{g}_i\}$  is in the advection-diffusion equation. Besides providing divergence-free flow fields and “balanced” representation of the transported material or property, the Hermite form would simultaneously compute the fluxes without projection and post-processing.

There are two other Hermite functions which might be mentioned. The first, reported by Wegmuller [17], has continuous first derivative in the tangential direction at element interfaces, and uses six degrees of freedom per node including all derivatives through second order. It is of sixth degree, complete through fifth order and can be modified for incompressible flow and generalized to affine and complex geometries. A second function with the same degrees of freedom as above, derived by Gopalacharyulu [5] with corrections by Watkins [16], has both derivatives continuous but as it is of eighth degree and quartic complete. These may not be cost-effective for general computations, but should be useful in p-refinement and error studies, and as models for hierarchical refinement for the element discussed in this paper.

#### REFERENCES

1. M. Fortin and A. Fortin, “Newer and newer elements for incompressible flow,” *Finite Elements in Fluids* **6**, John Wiley & Sons Ltd. (1985) 171–187.
2. G. P. Galdi, *An Introduction to the Mathematical Theory of the Navier-Stokes Equations, Vol. I – Linearized Steady Problems*, Springer-Verlag, New York, 1994.
3. V. Girault and P. -A. Raviart, *Finite Element Approximation of the Navier–Stokes Equations, Theory and algorithms*, Springer-Verlag, Berlin, 1979.
4. V. Girault, personal communication.
5. S. Gopalacharyulu, “A higher order conforming rectangular plate element,” *Int. J. Num. Methods Eng.* **6**, (1973) 305–308.
6. P. M. Gresho and R. L. Sani, *Incompressible Flow and the Finite Element Method*, John Wiley & Sons, Ltd., New York, 1998.
7. D. F. Griffiths, “An approximately divergence-free 9-node velocity element (with variations) for incompressible flows,” *Int. J. Num. Meth. Fluids* **1**, (1981) 323–346.
8. C. A. Hall, J. S. Peterson, T. A. Porsching and F. R. Sledge, “The dual variable method for finite element discretizations of Navier/Stokes equations,” *Int. J. Num. Meth. Engr.*, **21** (1985), 883–898.
9. J. T. Holdeman, “I. Some Lagrange interpolation functions for solenoidal and irrotational vector fields,” (Obelisk Research Report).
10. L. I. G. Kovasznay, “Laminar flow behind a two-dimensional grid,” *Proceedings of the Cambridge Philosophical Society*, **44** (1948) 58-62.
11. J. Leray, “Etude de diverses équations intégrales nonlinéaires et de quelques problèmes que pose l’hydrodynamique,” *J. Math. Pures Appl.* **12** (1933) 1–82.
12. J. -L. Lions, *Quelques Méthodes de Résolution des Problèmes aux Limites Non Linéaires*, Dunod, Paris, 1969.
13. R. J. Melosh, “Basis of derivation of matrices for the direct stiffness method,” *J. A. I. A. A.*, **1**, 1631-7, 1963.
14. J. Petera, J. F. T. Pittman, “Isoparametric Hermite elements,” *Int. J. Numer. Methods Engr.*, **37** (1994) 3489-3519.
15. R. Temam, *Navier–Stokes Equations, Theory and Numerical Analysis*, North-Holland Publishing Company, New York, 1977.
16. D. S. Watkins, “A Comment on Gopalacharyulu’s 24 Node Element,” *Int. J. Num. Methods Eng.* **10** (1976) 471-472.
17. A. W. Wegmuller, “A refined plate bending element,” *Int. J. Solids Structures*, **10** (1974) 1173–1178.

18. O. C. Zienkiewicz, *The Finite Element Method in Engineering Science*, McGraw-Hill, London, 1971.

1056 LOVELL ROAD, KNOXVILLE, TENNESSEE 37932  
*E-mail address:* `j.t.holdeman@charter.net`