

Computation of incompressible thermal flows using Hermite finite elements

Jonas T. Holdeman^{*,1}

1056 Lovell Road, Knoxville, Tennessee 37932, USA

Jin Whan Kim

Department of Mechanical Engineering, Dong-Eui University, Busanjin-Gu, Busan
614-714, South Korea

Abstract

Using Hermit basis functions with the finite element method offers a remarkably simple way to compute non-isothermal buoyancy-driven incompressible flow. The Hermite bases we use simplify the governing equation and strongly enforce the continuity equation. For this problem, we use a fourth-order C^1 stream function defined on rectangles here, but other higher and lower-order Hermite elements on rectangles and triangles can easily be derived or modified from elements found in the plate-bending literature. Hermite elements are also used for the temperature and pressure. We conclude with results from application of the method to the square thermal cavity at moderate to high Rayleigh numbers.

Key words: thermal flow incompressible flow Hermite elements
divergence-free elements divergence-free methods

1. Introduction

Recalling that a vector field can be orthogonally decomposed into divergence-free (solenoidal) and irrotational parts, and noting the velocity \mathbf{u} and its time derivative $\partial\mathbf{u}/\partial t$ are solenoidal for incompressible flow, it follows from the incompressible Navier-Stokes equation that the governing equation for the velocity can be written [1][2][3],

$$\frac{\partial}{\partial t}\mathbf{u} = \pi^S(-\mathbf{u} \cdot \nabla\mathbf{u} + \nu\nabla^2\mathbf{u}) + \mathbf{f}^S, \quad (1)$$

^{*}Corresponding author

¹The first author thanks Brain Korea 21 and the Department of Mechanical Engineering at Dong-Eui University for the gracious support and hospitality shown while he was in residence, Spring 2003.

where π^S is the solenoidal projection operator and \mathbf{f}^S is the solenoidal (nonconservative) part of the body force (if present). Only solenoidal body forces can directly influence the incompressible fluid flow. In (1) we have used the fact that $\pi^S \nabla P \equiv 0$ to eliminate the pressure P . With respect to the absence of pressure-gradient forces, (1) is understood to be a kinematic equation with the incompressibility condition serving as a conservation law [3]. The presence of the operator π^S is inconvenient, and can be avoided by introducing the weak form of (1) with divergence-free test functions, as we shall see later.

While incompressibility assumes that the fluid density is independent of pressure, buoyancy forces arise from a change in density with temperature, $\rho = \rho_0(1 - \beta \Delta T)$, where β is the thermal expansion coefficient of the fluid, and ΔT (hereafter referred to simply as T) is the relative temperature. Density changes are small, so it is found that the density can be assumed to be constant, with the introduction of a temperature-dependent force $\mathbf{f}^S = -\beta \mathbf{g} T$, known as the Boussinesque approximation. For closure we introduce another (energy) equation for the unknown temperature T . In dimensionless form, the equations might be written,

$$\begin{aligned} \frac{\partial}{\partial t} \mathbf{u} &= \pi^S(-\mathbf{u} \cdot \nabla \mathbf{u} + \sqrt{P_r/R_a} \nabla^2 \mathbf{u} + \hat{\mathbf{y}} T), \\ \frac{\partial}{\partial t} T &= -\mathbf{u} \cdot \nabla T + 1/\sqrt{R_a P_r} \nabla^2 T + Q_i, \end{aligned} \quad (2)$$

where P_r and R_a are the Prandtl and Rayleigh numbers, $\hat{\mathbf{y}}$ is a unit vector in the vertical direction, and Q_i is any thermal source term (which we will neglect). The dimensionless form (2) is not unique. The dimensionless form above [4] is better conditioned for high Rayleigh number flow, but stream function, velocities and vorticities differ by a scale factor $\sqrt{R_a P_r}$ from the small Rayleigh number dimensionless form more commonly used for benchmarks and comparisons [5][6][7][8], and displayed in the Appendix as (17).

Choosing weight/test functions that are divergence-free, the orthogonality of solenoidal and irrotational spaces replaces the projection function in the weak form,

$$\begin{aligned} (\mathbf{v}, \frac{\partial}{\partial t} \mathbf{u}) &= -(\mathbf{v}, \mathbf{u} \cdot \nabla \mathbf{u}) + \sqrt{P_r/R_a} (\mathbf{v}, \nabla^2 \mathbf{u}) + (\mathbf{v} \cdot \hat{\mathbf{g}}, T), \quad \mathbf{u} \in \mathcal{V}_0, \forall \mathbf{v} \in \mathcal{V}_0, \\ (q, \frac{\partial}{\partial t} T) &= -(q, \mathbf{u} \cdot \nabla T) + 1/\sqrt{R_a P_r} (q, \nabla^2 T), \quad T \in H^1, \forall q \in H^1, \end{aligned} \quad (3)$$

where (\cdot, \cdot) indicates the inner product over the problem domain Ω , and the function spaces are given by $\mathcal{V} = \{\mathbf{u} \in \mathbf{H}^1 : \text{div } \mathbf{u} = 0\}$ and $\mathcal{V}_0 = \{\mathbf{u} \in \mathcal{V} : \mathbf{u}|_{\partial\Omega} = 0\}$.

As usual, we apply Green's theorem

$$(g, \nabla^2 f)_\Omega = -(\nabla g, \nabla f)_\Omega + \langle g, \mathbf{n} \cdot \nabla f \rangle_\Gamma \quad (4)$$

to reduce continuity requirements,

$$\begin{aligned} (\mathbf{v}, \frac{\partial}{\partial t} \mathbf{u}) &= -(\mathbf{v}, \mathbf{u} \cdot \nabla \mathbf{u}) + \sqrt{P_r/R_a} (-(\nabla \mathbf{v}, \nabla \mathbf{u}) + \langle \mathbf{v}, \mathbf{n} \cdot \nabla \mathbf{u} \rangle_\Gamma) + (\mathbf{v} \cdot \hat{\mathbf{g}}, T), \\ (q, \frac{\partial}{\partial t} T) &= -(q, \mathbf{u} \cdot \nabla T) + 1/\sqrt{R_a P_r} (-(\nabla q, \nabla T) + \langle q, \mathbf{n} \cdot \nabla T \rangle_\Gamma), \end{aligned} \quad (5)$$

where $\langle \cdot, \cdot \rangle_\Gamma$ indicates the inner product over the boundary of the problem domain Ω .

2. Buoyancy-driven flow

In the remainder of this paper we focus on stationary flow (where $\frac{\partial}{\partial t} \mathbf{u}$ and $\frac{\partial}{\partial t} T$ are zero) in a square cavity. We are using the notation that a comma in a subscript indicates differentiation with respect to any variables following the comma, unless specified otherwise.

The obvious dependent variables in (2) are the solenoidal velocity \mathbf{u} and the temperature T . However, any divergence-free field can be written as the curl of a stream function ψ , so ψ is implicitly a dependent variable. Further, the vanishing of the divergence reflects the relation $\psi_{,xy} = \psi_{,yx}$ and interest in the vorticity $\omega = v_{,x} - u_{,y} = -(\psi_{,xx} + \psi_{,yy})$ highlights the possible interest in the second derivatives of ψ as data. In the standard (dimensionless) thermal cavity problem, the vertical boundaries of the cavity are kept at specified temperatures ($T = 1$ on the left and $T = 0$ on the right), and the horizontal boundaries (top and bottom) are assumed to be adiabatic (thermally insulated), where $T_{,y} = 0$. The velocity \mathbf{u} is assumed to obey the usual no-flow and non-slip conditions on the boundary, so the associated stream function is constant there, and we will take this constant to be zero. Since the velocity is constant on the boundaries, the change in \mathbf{u} along the boundaries is also zero. In summary then, the assumed boundary conditions are,

$$\begin{aligned} \psi = u = u_{,x} = u_{,y} = v = v_{,y} = T_{,y} = 0, \quad T = T_h = 1, \quad x = 0, \quad y \in [0, 1], \\ \psi = u = u_{,x} = u_{,y} = v = v_{,y} = T_{,y} = 0, \quad T = T_c = 0, \quad x = 1, \quad y \in [0, 1], \quad (6) \\ \psi = u = u_{,x} = v = v_{,x} = v_{,y} = T_{,y} = 0, \quad y = 0, 1, \quad x \in (0, 1). \end{aligned}$$

The geometry of the problem is shown in Figure (1) along with these boundary conditions. We recall that (in two dimensions) the velocity components are

Figure 1: Boundary conditions for natural convection in a square cavity

$$\begin{array}{c} \psi = u = u_{,x} = v = v_{,x} = v_{,y} = 0, T_{,y} = 0 \\ \begin{array}{c} \psi = 0 \\ u = 0 \\ u_{,x} = 0 \\ u_{,y} = 0 \\ v = 0 \\ v_{,y} = 0 \\ T = 1 \\ T_{,y} = 0 \end{array} \begin{array}{c} \boxed{\begin{array}{cc} (0, 1) & (1, 1) \\ & \\ & \\ & \\ & \\ & \\ (0, 0) & (1, 0) \end{array}} \\ \psi = u = u_{,x} = v = v_{,x} = v_{,y} = 0, T_{,y} = 0 \end{array} \begin{array}{c} \psi = 0 \\ u = 0 \\ u_{,x} = 0 \\ u_{,y} = 0 \\ v = 0 \\ v_{,y} = 0 \\ T = 0 \\ T_{,y} = 0 \end{array} \end{array}$$

simply first derivatives of the stream function (components of the curl). First derivatives of the velocity components can be expressed as second derivatives of the stream function. Thus the data for the problem (dependent variables and boundary conditions) consist of the stream function and its first and second derivatives, and the temperature and its gradient (first derivatives). If we were to choose a set of basis functions to expand the velocity and temperature fields (as we shall do in the following sections), it seems natural to choose *Hermite* bases which reflect much of the data of the problem.

Since we are applying no-flow, no-slip boundary conditions, $\mathbf{n} \cdot \nabla \mathbf{v}|_{\Gamma} = 0$, and the velocity boundary integral in (5) vanishes. As we have applied adiabatic (Neumann) boundary conditions on the top and bottom (horizontal) surfaces, $\mathbf{n} \cdot \nabla T|_{\Gamma_N} = 0$, so the temperature boundary integral vanishes on these parts of the surface.

3. Divergence-free Hermite finite element method

As usual, the continuous problem domain Ω is discretized or decomposed into non-overlapping subdomains Ω_e (in this case rectangles), each characterized by a diameter h , which cover the problem domain. These subdomains or elements are characterized by a set of nodes (on the corners of the rectangles). The dependent variables are expanded in terms of basis functions, where the coefficients are the values of the variables at the nodes.

$$\begin{aligned} \mathbf{u} &\approx \mathbf{u}^h = \sum_i \mathbf{S}_i \mathbf{U}_i \\ T &\approx T^h = \sum_i \mathbf{g}_i \mathbf{T}_i \end{aligned} \quad (7)$$

Each component of the vector coefficients $\{\mathbf{U}_i, \mathbf{T}_i\}$ constitutes a degree-of-freedom of the computed field(s). Provided we can find sets of finite elements which are divergence-free, the discrete algebraic form of the weak statement (5) is given by,

$$\begin{bmatrix} -\mathbf{C}(\mathbf{U}) + \sqrt{P_r/R_a} \mathbf{D} & \mathbf{B} \\ \mathbf{0} & -\bar{\mathbf{C}}(\mathbf{U}) + 1/\sqrt{P_r R_a} \bar{\mathbf{D}} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{T} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix}, \quad (8)$$

where the submatrices are defined in the Appendix.

We now define the bases used for the computation in following sections. For the temperature, we seek a Hermite element with the scalar function and first-derivative degrees-of-freedom at the four corner nodes. A C^0 continuous, cubic-complete element with 12 degrees-of-freedom was found independently by Melosh [9] and Zienkiewicz [10] and is given by (19) in the Appendix.

In the previous section we provided motivation for the use of finite elements for the velocity from the point of view of the problem data. However to apply (8) it is necessary to go beyond the data and require that the finite elements be divergence-free in a strong sense. Other authors have proposed and used methods which are divergence-free in a weak sense. Here we require that for any choice of nodal parameters, (i) the divergence vanishes on the element,

(*ii*) the elements are sufficiently conforming that the divergence vanishes on all inter-element boundaries, and (*iii*) the element degrees-of-freedom include the physical variables of the problem.

One could, of course, satisfy (*i*) and (*ii*) but not(*iii*) by taking the curl of a continuous Lagrangian stream function element. All attempts to satisfy all three conditions with Lagrange finite elements have failed. As we have noted, any divergence-free vector field implies the existence of a stream function/vector potential. So we can satisfy these conditions by taking the curl of an appropriate Hermite stream function/vector potential element.

For the stream function-velocity element, we will use a four-node C^1 element having 24 degrees-of-freedom, with DOFs $\{\psi_i, u_i, v_i, \psi_{i,xx}, \psi_{i,xy}, \psi_{i,yy}\}$ on the corners. The stream function element is a modified form of a quartic element published by Gopalacharyulu [11][12] and Watkins [13][14] and others. The modification consists of the interchange of the first derivative terms with a change in sign of one of them as shown in (24) in the Appendix. The divergence-free vector element is found by taking the curl, (25) in the Appendix. For these elements we have,

$$\begin{aligned}\mathbf{U}_i &= [\psi_i \ u_i \ v_i \ \psi_{i,xx} \ \psi_{i,xy} \ \psi_{i,yy}]^T, \\ \mathbf{T}_i &= [T_i \ T_{i,x} \ T_{i,y}]^T.\end{aligned}\tag{9}$$

We could have used other four-node elements for the velocity, for instance the simple cubic C^0 stream function element (22) with three degrees-of-freedom per node and its curl, or a number of others, but the two chosen are arguably a canonical pair for this problem.

4. Pressure and vorticity

Applying a solenoidal projection operator π^S to the scaled Navier-Stokes equation, we found the pressureless governing equation for the velocity (1). If, instead, we apply the irrotational projection operator π^I , we get the equation for recovering the pressure from the velocity,

$$\nabla p = \pi^I (-\mathbf{u} \cdot \nabla \mathbf{u} + \sqrt{P_r/R_a} \nabla^2 \mathbf{u}) + \mathbf{f}^I.\tag{10}$$

where the body force \mathbf{f}^I is conservative, typically the gradient of a potential function. The resemblance of (10) to the ‘‘pressure Poisson’’ equation is more than coincidental. We now choose a space of scalar functions $\{q\}$ and use the gradients $\{\nabla q\}$ as the weight functions. Then orthogonality replaces the irrotational projection operator again, and the weak form is given by,

$$(\nabla q, \nabla p) = -(\nabla q, \mathbf{u} \cdot \nabla \mathbf{u}) + \sqrt{P_r/R_a} (\nabla q, \nabla^2 \mathbf{u}) + (\nabla q, \mathbf{f}^I), \quad \forall \nabla q \in \mathbf{w},\tag{11}$$

where \mathbf{w} is the space of irrotational functions. The irrotational force \mathbf{f}^I might include the gradient of the gravitational *potential*.

The discrete form, the algebraic equations to be solved, is given by,

$$\bar{\bar{\mathbf{K}}} \mathbf{P} = (-\bar{\bar{\mathbf{C}}}(\mathbf{U}) + \sqrt{P_r/R_a} \bar{\bar{\mathbf{D}}}) \mathbf{U} + \bar{\bar{\mathbf{F}}},\tag{12}$$

Any type of finite element may be used for the evaluation of the pressure, Lagrangian or Hermite. Since we may also be interested in the pressure forces, which are related to the negative of the pressure gradient, we choose the same Hermite element (19) as is used for the temperature.

The vorticity ω of the flow is defined as the curl of the velocity. In two dimensions, $\omega = v_{,x} - u_{,y} = -(\psi_{,xx} + \psi_{,yy})$, as we have noted. But the second derivatives of ψ at the nodes are degrees-of-freedom of the quartic element, and are calculated when evaluating the velocity, so the vorticities at the nodes,

$$\omega_i = -(\psi_{i,xx} + \psi_{i,yy}), \quad (13)$$

are a byproduct of the velocity computation. However, the second derivatives are not continuous at element boundaries. A continuous vorticity can be found by interpolating the vorticities at the nodes with a bilinear function, or by a least-squares fit to the curl of the velocity by the simple cubic Hermite. Likewise the heat flux and pressure gradient can be calculated at each node from nodal temperature and (derived) pressure data.

5. The thermal cavity

A number of authors have used the thermal cavity to compare the results of different methods. An early paper by de Vahl Davis[5] is regarded as a benchmark, but some more-recent papers seem to be slight improvements. All use the parameter value $P_r = .71$ for the Prandtl number, and a range or Rayleigh numbers ($10^3 \leq R_a \leq 10^6$). Computations at low Rayleigh number are unchallenging, but become more difficult at the upper range as the natural convective flow tends to become unstable. We use simple iteration to solve the nonlinearity in the velocity, and for larger R_a we stabilize the iteration using under-relaxation. At $R_a = 10^6$, the condition number of the fully-coupled stress matrix associated with (2) is unmanageably large, and that associated with the alternate dimensionless form (17) is about 50 times greater. Rather than solve the velocity and temperature equations (8) simultaneously, it is advantageous to separate them into two coupled equations to be solved sequentially,

$$\begin{aligned} (-\bar{\mathbf{C}}(\mathbf{U}^{(i)}) + \sqrt{P_r/R_a} \bar{\mathbf{D}}) \mathbf{T}^{(i+1)} &= \mathbf{f}_2, \\ (-\mathbf{C}(\mathbf{U}^{(i)}) + 1/\sqrt{P_r R_a} \mathbf{D}) \mathbf{U}^{(i+1)} &= \mathbf{f}_1 - \mathbf{B} \mathbf{T}^{(i+1)} \end{aligned} \quad (14)$$

within each iteration step i . In this form, the buoyancy term, which contributed to the poor condition of the stress matrix, appears as data in the second equation above. The resulting stress matrix, though not well-conditioned, can be managed by a preconditioner, and is insensitive to interchange of the forms (2) and (17). Relaxing only the velocity equation does not affect the overall iterative convergence rate and produces a smaller temperature correction.

Comparing the diffusion matrix $\bar{\mathbf{D}}$ in the temperature equation above with its definition (42), the boundary integral seems to be missing, as indeed it is. We have already remarked that the integral vanishes on the top and bottom

adiabatic surfaces. On the vertical *isothermal* surfaces, T and $T_{,y}$ are fixed, specified as boundary conditions. The remaining integral of the normal gradient component involves $\int g_i^{(1)} G_i^{(1)}|_{\Gamma_N}$, but the test function $g_i^{(1)}|_{\Gamma_N}$ vanishes on vertical boundaries, as can be seen from its definition (19).

The method was coded in MATLAB[®] v5.3(R11) (1999), and the equations were solved using the sparse iterative solver GMRES with an incomplete LU preconditioner, and run on a 2.2Ghz, dual processor laptop with 3Gb memory.

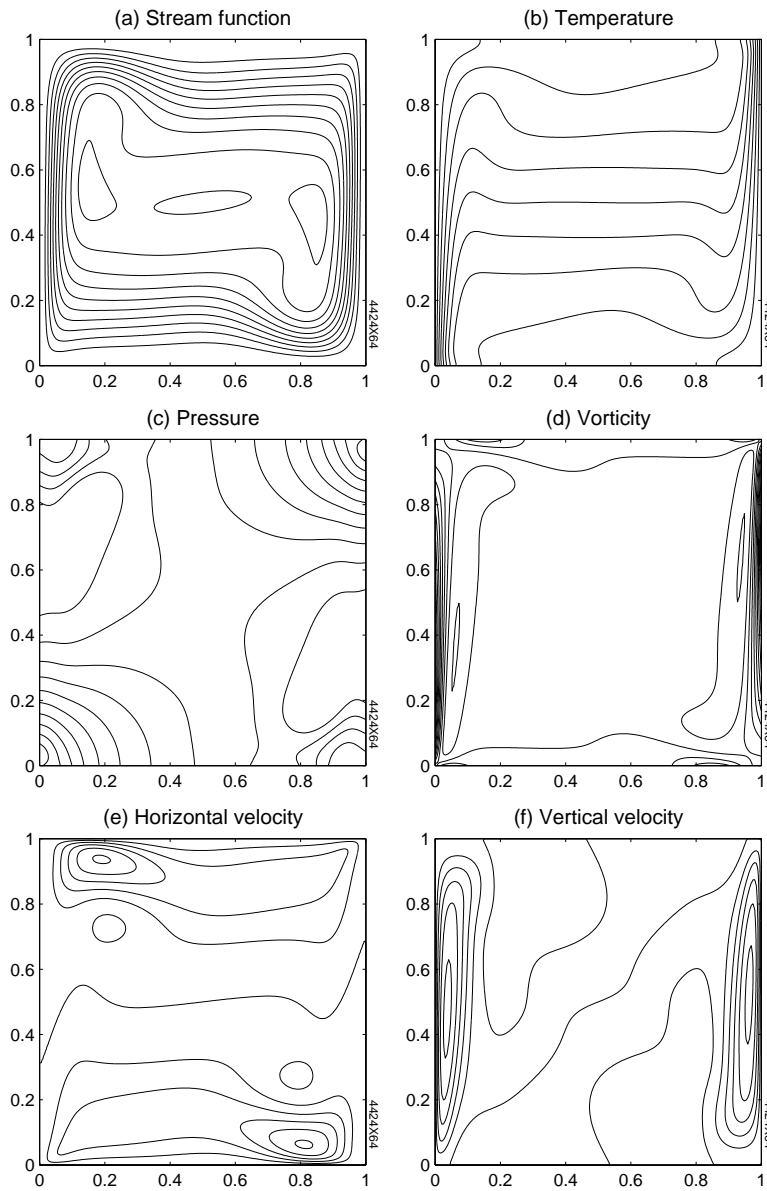
We used two sequences of uniform meshes, $\{ 24 \times 24, 36 \times 36, 48 \times 48, 60 \times 60 \}$, and $\{ 32 \times 32, 48 \times 48, 64 \times 64 \}$ elements to examine accuracy and convergence under refinement at common mesh points of each set. In practice we would use graded meshes to resolve the high velocity and thermal gradients on the sides. We solved the governing equations for Rayleigh numbers $R_a = \{ 10^3, 10^4, 10^5, 10^6 \}$. Relaxation factors used for the solution of the nonlinear velocity equations were $\lambda = \{ .94, .67, .35, .048 \}$ respectively for the Rayleigh numbers given. We summarize some of the benchmark quantities for each R_a in Table 1 and show some of the flow and pressure contours obtained from the finest meshes, but (interpolated) plot results from the coarser meshes would be visually indistinguishable from these. Contour levels are those used by Davis [5] (except for a slight increase in the largest magnitude stream function contour level for $R_a = 10^6$), but adjusted by the factor $1/\sqrt{PrR_a}$ to account for the different nondimensional form (2) used, as mentioned elsewhere. Tabular values presented (except for temperature) have been multiplied by $\sqrt{PrR_a}$ for comparison. We show contours for six computed and derived variables in the problem on a 64×64 element mesh for $R_a = 10^6$ with contour levels in Figure 2. In Table 1, $|\psi_{mid}|$ is the value of the magnitude of the stream function

Table 1: Summary and comparison of some results

Present					Davis [5]				
R_a	10^3	10^4	10^5	10^6	R_a	10^3	10^4	10^5	10^6
$ \psi_{mid} $	1.1746	5.0737	9.1156	16.386	$ \psi_{mid} $	1.174	5.071	9.111	16.32
$ \psi _{max}$	1.1746	5.0737	9.6168	16.811	$ \psi _{max}$	–	–	9.612	16.750
(x)	0.5000	0.5000	0.2845	0.1504	(x)	–	–	0.285	0.151
(y)	0.5000	0.5000	0.6016	0.5470	(y)	–	–	0.601	0.547
u_{max}	3.6494	16.1833	34.741	64.834	u_{max}	3.649	16.178	34.73	64.63
(y)	0.8132	0.8232	0.8546	0.8499	(y)	0.813	0.823	0.855	0.850
v_{max}	3.6974	19.6282	68.635	220.565	v_{max}	3.697	19.617	68.59	219.36
(x)	0.1783	0.1189	0.0660	0.0378	(x)	0.178	0.119	0.066	0.0379
Nu_0	1.1178	2.2447	4.5204	8.8143	Nu_0	1.117	2.238	4.509	8.817

at the center of the cavity, $|\psi|_{max}$ is the maximum value of the magnitude of the stream function and is followed by the coordinates (x, y) of this maximum, u_{max} is the maximum value of the u velocity component along the vertical centerline and its y -coordinate on that line, and v_{max} is the maximum value of the v velocity component along the horizontal line through the center and its x -coordinate on that line.

The present results in Table 1, calculated on a 64×64 mesh, are essentially



Contour levels: (a) $-16.34, -15.07 (1.675) 0$ (b) $0 (.1) 1$
(c) $0 (26.8/9) 26.8$ (d) $-3178 (1847.1) 15293$
(e) $-125.5 (25.10) 125.5$ (f) $-207.6 (41.52) 207.6$

Figure 2: Thermal cavity contours, $Ra = 10^6$, (64×64) -element uniform mesh.

in agreement with similar results on a 60×60 mesh, and with results calculated with the cubic Hermite stream function elements (22) and (23) in the appendix, but calculated on a 80×80 mesh and having approximately the same number of degrees-of-freedom. All digits in these tabular results are believed to be significant. They are in agreement with those of Davis, but are consistently slightly greater.

6. Extensions

We consider here some generalizations of the method that are beyond the scope of this paper, so we simply list some of them.

For triangular elements we define the Hermite element on a reference right triangle. Then this form is mapped to a general triangle (perhaps with curved sides) in the manner described below for rectangular elements. This can be done quite efficiently for straight-sided triangles.

We mention in (38) in the Appendix that geometric parameters can be factored out of the element on the rectangle. That was done in this paper solely for computational efficiency. But the rectangular element can be extended to a general convex quadrilateral by using the Jacobian of the transformation. This is particularly simple for the quadrilateral with straight sides, but curved sides can be accommodated as well.

The elements can be extended to the usual set of orthogonal curvilinear coordinate systems as well. If \mathbf{R} is a matrix which depends on the particular coordinate system, then (38) generalizes to,

$$\begin{aligned} \mathbf{s}_i(x, y) &= \hat{\mathbf{s}}_i(\xi, \eta) \mathbf{T}_i^{-1} \bar{\mathbf{R}}_i^{-1}, \\ \mathbf{S}_i(x, y) &= \mathbf{R} \Delta^{-1} \mathbf{J}^T \hat{\mathbf{S}}_i(\xi, \eta) \mathbf{T}_i^{-1} \bar{\mathbf{R}}_i^{-1}, \end{aligned} \quad \bar{\mathbf{R}} = \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{R} \end{bmatrix}. \quad (15)$$

For instance, in polar coordinates,

$$\mathbf{R} = \begin{bmatrix} r^{-1} & 0 \\ 0 & 1 \end{bmatrix}. \quad (16)$$

The extension to three dimensions is a little more problematic in that one cannot borrow elements from the existing literature nor directly extend two-dimensional elements to three because we don't have the simple relation (21) between the gradient and curl. Nevertheless, linear and quadratic hexahedral velocity elements have been found which can be used as prototypes for development of further elements. These Hermite elements involve the three vector potential and three velocity-component degrees-of-freedom. They were originally derived from tri-linear and tri-quadratic Lagrange elements, augmented with face-centered functions and a few tricks to eliminate the divergence. Using these as a pattern, one can use more conventional techniques, expanding shape functions for the vector potential with undetermined coefficients, applying FE conditions at the nodes and element boundaries, applying symmetries,

and requiring exact interpolation of an interpolation set to evaluate the coefficients and find other elements. MAPLE® provides an excellent environment and solver for doing this.

These elements can be mapped to a general hexahedron by forms similar to (38), and to curvilinear coordinates by forms similar to (15).

7. Conclusion

The finite element method described in this paper represents a new paradigm for computing incompressible flow. Gone from the vocabulary are terms like “LBB condition”, “saddle points”, and “compatible pressure-velocity pairs”. New terms are found like “pressureless governing equation”, “Hermite stream function elements” and “orthogonal projection”. Extensions from Cartesian to curvilinear coordinate systems follow from generalization of methods described in the Appendix. The role of the pressure as a driving force is diminished, replaced by the ubiquitous stream function and its boundary conditions, though a coupled pressure remains as a control variable. Derivation of the pressureless governing equation from first principles for an idealized incompressible fluid needs to be revisited. In most cases the existing closed-form solutions of classical incompressible flows can be re-derived by application of the Helmholtz decomposition.

For two-dimensions, a rich set of suitable Hermite elements on rectangles and triangles can be found in the literature or easily derived. More challenging is the problem of Hermite vector potential elements for three-dimensions. Here we do not have the simple relation (21) between the gradient and curl operators to draw from. Derivation methods for Hermite elements in two dimensions simply do not extend to three dimensions, but the problem is not insurmountable.

Appendix

We mention here the alternate, small R_a nondimensional form of the buoyancy-driven equation,

$$\begin{aligned}\frac{\partial}{\partial t} \mathbf{u} &= \pi^S(-\mathbf{u} \cdot \nabla \mathbf{u} + P_r \nabla^2 \mathbf{u} + P_r R_a \hat{\mathbf{y}} T), \\ \frac{\partial}{\partial t} T &= -\mathbf{u} \cdot \nabla T + \nabla^2 T + Q_i.\end{aligned}\tag{17}$$

Compared with (2), both use the same nondimensionalization for the temperature, so the two produce the same results for T . Nondimensionalization of the velocities (and the stream function) differ by a factor of $\sqrt{P_r R_a}$. If the computed stream function and velocities from (2) are multiplied by $\sqrt{P_r R_a}$, the results are in agreement with those from (17).

For notational compactness, we introduce the local variables $\{\xi, \eta\}$ defined on the reference square $[-1, 1]^2$, and the notation,

$$\begin{aligned}\xi &\equiv \frac{2}{h_x}(x - x_c), & \xi_0 &\equiv \hat{x}_i \xi, & \xi_h &\equiv \frac{2}{h_x}, & \xi_i &\equiv \hat{x}_i \xi_h, & \hat{\xi}_i &= \hat{x}_i = \pm 1, \\ \eta &\equiv \frac{2}{h_y}(y - y_c), & \eta_0 &\equiv \hat{y}_i \eta, & \eta_h &\equiv \frac{2}{h_y}, & \eta_i &\equiv \hat{y}_i \eta_h, & \hat{\eta}_i &= \hat{y}_i = \pm 1.\end{aligned}\tag{18}$$

The compact notation $\xi_0 = \xi_i \xi$, etc., may seem strange in that the nodal index i is implicit. Further, ξ_0 is to be regarded as a single symbol rather than a subscripted variable. A second complication is that ξ_h and ξ_i have dimensions of inverse length though one might naturally presume the reverse.

We first introduce the simple cubic scalar Hermite element (found independently by Melosh [9], Zienkiewicz [10], and others) for the temperature. We write it as a matrix-valued function with three columns,

$$\mathbf{g}_i(\xi, \eta) = \begin{bmatrix} g_i^{(0)}(\xi, \eta) & g_i^{(1)}(\xi, \eta) & g_i^{(2)}(\xi, \eta) \end{bmatrix},$$

where

$$\begin{aligned} g_i^{(0)}(\xi, \eta) &= \frac{1}{8}(1 + \xi_0)(1 + \eta_0)(2 + \xi_0(1 - \xi_0) + \eta_0(1 - \eta_0)), \\ g_i^{(1)}(\xi, \eta) &= -\frac{1}{8}\xi_i^{-1}(1 + \eta_0)(1 + \xi_0)^2(1 - \xi_0), \\ g_i^{(2)}(\xi, \eta) &= -\frac{1}{8}\eta_i^{-1}(1 + \xi_0)(1 + \eta_0)^2(1 - \eta_0). \end{aligned} \quad (19)$$

The columns correspond to the nodal degrees-of-freedom $[T_i \ T_{i,x} \ T_{i,y}]$. Along edges containing the controlling nodes, the function varies as a Hermite cubic. The function vanishes on edges not containing the controlling node, so piecewise interpolations using this function are continuous. The form above is defined explicitly on the reference square $[1, -1] \times [1, -1]$, where the pair $[\hat{\xi}_i, \hat{\eta}_i]$ are the coordinates of the node i . The mapping to a general rectangle in (x, y) is found by substitution using (18).

We have need for the gradient of this element. This is given by the 2×3 matrix-valued function,

$$\mathbf{G}_i(\xi, \eta) = \begin{bmatrix} G_i^{(0)}(\xi, \eta) & G_i^{(1)}(\xi, \eta) & G_i^{(2)}(\xi, \eta) \end{bmatrix},$$

where

$$\begin{aligned} G_i^{(0)}(\xi, \eta) &= \begin{bmatrix} \frac{1}{8}\xi_i(1 + \eta_0)(\eta_0(1 - \eta_0) + 3(1 - \xi^2)) \\ \frac{1}{8}\eta_i(1 + \xi_0)(\xi_0(1 - \xi_0) + 3(1 - \eta^2)) \end{bmatrix}, \\ G_i^{(1)}(\xi, \eta) &= \begin{bmatrix} \frac{1}{8}(1 + \xi_0)(1 + \eta_0)(-1 + 3\xi_0) \\ -\frac{1}{8}\eta_i\xi_i^{-1}(1 + \xi_0)^2(1 - \xi_0) \end{bmatrix}, \\ G_i^{(2)}(\xi, \eta) &= \begin{bmatrix} -\frac{1}{8}\xi_i\eta_i^{-1}(1 + \eta_0)^2(1 - \eta_0) \\ \frac{1}{8}(1 + \xi_0)(1 + \eta_0)(-1 + 3\eta_0) \end{bmatrix}. \end{aligned} \quad (20)$$

We note as an aside, that this vector element is suitable for representation of irrotational vector fields.

Many attempts have been made to formulate divergence-free bases, but none seem to have been successful enough to be widely adopted. The method we use here appeared in 1977 in the first edition of a book by R. Temam [1] as method APX4, but seems to have been ignored (there are difficulties in constructing his element).

A modification of the element (19) yields a Hermite stream function element with degrees-of-freedom $[\psi_i \ u_i \ v_i]$. We note that in two dimensions the gradient operator and curl operator are given by

$$\nabla = [\partial_x, \partial_y]^T \quad \text{and} \quad \mathbf{curl} = [\partial_y, -\partial_x]^T, \quad (21)$$

respectively. This close relation suggests that a stream function element can be generated from (19) by an interchange of the last two columns with a change of sign of one of them. This is indeed the case. A simple cubic C^0 stream function element is given by,

$$\begin{aligned} \mathbf{s}_i(\xi, \eta) &= \begin{bmatrix} s_i^{(0)}(\xi, \eta) & s_i^{(1)}(\xi, \eta) & s_i^{(2)}(\xi, \eta) \\ g_i^{(0)}(\xi, \eta) & g_i^{(2)}(\xi, \eta) & -g_i^{(1)}(\xi, \eta) \end{bmatrix}, \\ &= \begin{bmatrix} s_i^{(0)}(\xi, \eta) & s_i^{(1)}(\xi, \eta) & s_i^{(2)}(\xi, \eta) \\ g_i^{(0)}(\xi, \eta) & g_i^{(2)}(\xi, \eta) & -g_i^{(1)}(\xi, \eta) \end{bmatrix}, \end{aligned} \quad (22)$$

with degrees-of-freedom $[\psi_i, u_i, v_i]^T$.

A divergence-free velocity element follows by taking the curl of this C^0 element and gives the 2×3 matrix-valued function,

$$\mathbf{S}_i(\xi, \eta) = \begin{bmatrix} S_i^{(0)}(\xi, \eta) & S_i^{(1)}(\xi, \eta) & S_i^{(2)}(\xi, \eta) \end{bmatrix},$$

where

$$\begin{aligned} S_i^{(0)}(\xi, \eta) &= \begin{bmatrix} \frac{1}{8}\eta_i(1+\xi_0)(\xi_0(1-\xi_0)+3(1-\eta^2)) \\ -\frac{1}{8}\xi_i(1+\eta_0)(\eta_0(1-\eta_0)+3(1-\xi^2)) \end{bmatrix}, \\ S_i^{(1)}(\xi, \eta) &= \begin{bmatrix} \frac{1}{8}(1+\xi_0)(1+\eta_0)(-1+3\eta_0) \\ \frac{1}{8}\xi_i\eta_i^{-1}(1+\eta_0)^2(1-\eta_0) \end{bmatrix}, \\ S_i^{(2)}(\xi, \eta) &= \begin{bmatrix} \frac{1}{8}\eta_i\xi_i^{-1}(1+\xi_0)^2(1-\xi_0) \\ \frac{1}{8}(1+\xi_0)(1+\eta_0)(-1+3\xi_0) \end{bmatrix}. \end{aligned} \quad (23)$$

The tangential component of the resulting vector element is not continuous on element boundaries, but the normal component is, and that is all that is necessary to make the interpolated vector field divergence-free.

For the velocity element to be used here, we seek a four-node Hermite element with first and second derivative DOFs. There are two elements which satisfy these requirements, a quintic C^0 element [15] and a quartic C^1 element. In this paper we have chosen the second option. This element was first published by Gopalacharyulu [11][12], but had an error that was corrected by Watkins [13][14]. It will be referred to here as the ‘‘GW element’’. A *stream function* element is found by interchanging the first derivative shape functions with a change of sign of one as above, so that the degrees-of-freedom are $[\psi_i \ u_i \ v_i \ \psi_{i,xx} \ \psi_{i,xy} \ \psi_{i,yy}]^T$. This element, written as a 1×6 matrix-valued

function, is given by,

$$\mathbf{s}_i(x, y) = \left[s_i^{(0)}(\xi, \eta) \quad s_i^{(1)}(\xi, \eta) \quad s_i^{(2)}(\xi, \eta) \quad s_i^{(3)}(\xi, \eta) \quad s_i^{(4)}(\xi, \eta) \quad s_i^{(5)}(\xi, \eta) \right],$$

where

$$\begin{aligned} s_i^{(0)}(\xi, \eta) &= \frac{1}{64}(1 + \xi_0)^2(1 + \eta_0)^2((1 + \xi_0)(2 - \eta_0)(8 - 9\xi_0 + 3\xi^2) \\ &\quad + (1 + \eta_0)(2 - \xi_0)(8 - 9\eta_0 + 3\eta^2) - 4(2 - \eta_0)(2 - \xi_0)), \\ s_i^{(1)}(\xi, \eta) &= -\frac{1}{64}\eta_i^{-1}(1 + \xi_0)^2(1 + \eta_0)^3(2 - \xi_0)(1 - \eta_0)(5 - 3\eta_0), \\ s_i^{(2)}(\xi, \eta) &= \frac{1}{64}\xi_i^{-1}(1 + \xi_0)^3(1 + \eta_0)^2(2 - \eta_0)(1 - \xi_0)(5 - 3\xi_0), \\ s_i^{(3)}(\xi, \eta) &= \frac{1}{64}\xi_i^{-2}(1 + \xi_0)^3(1 + \eta_0)^2(1 - \xi_0)^2(2 - \eta_0), \\ s_i^{(4)}(\xi, \eta) &= \frac{1}{16}\xi_i^{-1}\eta_i^{-1}(1 + \xi_0)^2(1 + \eta_0)^2(1 - \xi_0)(1 - \eta_0), \\ s_i^{(5)}(\xi, \eta) &= \frac{1}{64}\eta_i^{-2}(1 + \xi_0)^2(1 + \eta_0)^3(1 - \eta_0)^2(2 - \xi_0), \end{aligned} \tag{24}$$

This element is of tenth total polynomial degree, but at most fifth-degree in each coordinate, and varies as a quintic Hermite along the boundaries.

Taking the curl of the stream function element gives a divergence-free vector element with continuous vector components along element boundaries. Written as a 2×6 matrix, it is given by,

$$\mathbf{S}_i(x, y) = \left[S_i^{(0)}(\xi, \eta) \quad S_i^{(1)}(\xi, \eta) \quad S_i^{(2)}(\xi, \eta) \quad S_i^{(3)}(\xi, \eta) \quad S_i^{(4)}(\xi, \eta) \quad S_i^{(5)}(\xi, \eta) \right],$$

where

$$\begin{aligned} S_i^{(0)}(\xi, \eta) &= \left[\begin{array}{l} \frac{3}{64}\eta_i(1 + \xi_0)^2(1 - \eta^2)((1 + \xi_0)(8 - 9\xi_0 + 3\xi^2) + (2 - \xi_0)(1 - 5\eta^2)) \\ -\frac{3}{64}\xi_i(1 + \eta_0)^2(1 - \xi^2)((1 + \eta_0)(8 - 9\eta_0 + 3\eta^2) + (2 - \eta_0)(1 - 5\xi^2)) \end{array} \right], \\ S_i^{(1)}(\xi, \eta) &= \left[\begin{array}{l} -\frac{1}{64}(1 + \xi_0)^2(1 + \eta_0)^2(2 - \xi_0)(7 - 26\eta_0 + 15\eta^2) \\ \frac{3}{64}\xi_i\eta_i^{-1}(1 + \eta_0)^3(1 - \xi^2)(1 - \eta_0)(5 - 3\eta_0) \end{array} \right], \\ S_i^{(2)}(\xi, \eta) &= \left[\begin{array}{l} \frac{3}{64}\eta_i\xi_i^{-1}(1 + \xi_0)^3(1 - \eta^2)(1 - \xi_0)(5 - 3\xi_0) \\ -\frac{1}{64}(1 + \eta_0)^2(1 + \xi_0)^2(2 - \eta_0)(7 - 26\xi_0 + 15\xi^2) \end{array} \right], \\ S_i^{(3)}(\xi, \eta) &= \left[\begin{array}{l} \frac{3}{64}\eta_i\xi_i^{-2}(1 + \xi_0)^3(1 - \eta^2)(1 - \xi_0)^2 \\ -\frac{1}{64}\xi_i^{-1}(1 + \eta_0)^2(1 + \xi_0)^2(1 - \xi_0)(1 - 5\xi_0)(2 - \eta_0) \end{array} \right], \\ S_i^{(4)}(\xi, \eta) &= \left[\begin{array}{l} \frac{1}{16}\xi_i^{-1}(1 + \xi_0)^2(1 + \eta_0)(1 - \xi_0)(1 - 3\eta_0) \\ -\frac{1}{16}\eta_i^{-1}(1 + \eta_0)^2(1 + \xi_0)(1 - \eta_0)(1 - 3\xi_0) \end{array} \right], \\ S_i^{(5)}(\xi, \eta) &= \left[\begin{array}{l} \frac{1}{64}\eta_i^{-1}(1 + \xi_0)^2(1 + \eta_0)^2(1 - \eta_0)(1 - 5\eta_0)(2 - \xi_0) \\ -\frac{3}{64}\xi_i\eta_i^{-2}(1 + \eta_0)^3(1 - \xi^2)(1 - \eta_0)^2 \end{array} \right]. \end{aligned} \tag{25}$$

When computing the combined fluid and temperature stiffness matrix, repeated evaluation of the scalar and vector functions at the quadrature points can be quite consuming of computer resources. This computational effort can be considerably reduced by factoring out the scale-dependent terms ξ_h and η_h from the elements. We first note that the inverse transformations corresponding

to (18) are,

$$\begin{aligned} x &= \frac{h_x}{2}\xi + x_c, & x_{,\xi} &= \frac{h_x}{2} = \xi_h^{-1}, & x_{,\eta} &= 0, \\ y &= \frac{h_y}{2}\eta + y_c, & y_{,\eta} &= \frac{h_y}{2} = \eta_h^{-1}, & y_{,\xi} &= 0. \end{aligned} \quad (26)$$

so that the Jacobian matrix of the transformation \mathbf{J} and its transpose \mathbf{J}^T are,

$$\mathbf{J} = \mathbf{J}^T = \begin{bmatrix} \frac{h_x}{2} & 0 \\ 0 & \frac{h_y}{2} \end{bmatrix} = \begin{bmatrix} \xi_h^{-1} & 0 \\ 0 & \eta_h^{-1} \end{bmatrix}, \quad \Delta = \frac{1}{\xi_h \eta_h}, \quad \Delta^{-1} \mathbf{J}^T = \begin{bmatrix} \eta_h & 0 \\ 0 & \xi_h \end{bmatrix}, \quad (27)$$

or more generally,

$$\mathbf{J} \equiv \begin{bmatrix} x_{,\xi} & y_{,\xi} \\ x_{,\eta} & y_{,\eta} \end{bmatrix}, \quad \mathbf{J}^{-1} = \Delta^{-1} \begin{bmatrix} y_{,\eta} & -y_{,\xi} \\ -x_{,\eta} & x_{,\xi} \end{bmatrix}, \quad \Delta = \det \mathbf{J} = x_{,\xi} y_{,\eta} - y_{,\xi} x_{,\eta}. \quad (28)$$

These matrices are constant and diagonal for rectangular mappings and constant non-diagonal for affine mappings.

A post-multiplying matrix can be factored out of (19) as,

$$\mathbf{g}_i(x, y) = \hat{\mathbf{g}}_i(\xi, \eta) \begin{bmatrix} 1 & 0 & 0 \\ 0 & \xi_h^{-1} & 0 \\ 0 & 0 & \eta_h^{-1} \end{bmatrix}, \quad (29)$$

where $\hat{\mathbf{g}}_i(\xi, \eta)$ is restricted to the reference square. This generalizes to

$$\begin{aligned} \mathbf{g}_i(x, y) &= \hat{\mathbf{g}}_i(\xi, \eta) \bar{\mathbf{T}}_i^{-1}, \\ \mathbf{G}_i(x, y) &= \mathbf{J}^{-1} \hat{\mathbf{G}}_i(\xi, \eta) \bar{\mathbf{T}}_i^{-1}, \\ \bar{\mathbf{T}}_i^{-1} &= \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{J}_i \end{bmatrix}, \end{aligned} \quad (30)$$

and when combined, can be recognized as a semi-discrete similarity-like transformation,

$$\begin{bmatrix} \mathbf{g}_i(x, y) \\ \mathbf{G}_i(x, y) \end{bmatrix} = \bar{\mathbf{T}} \begin{bmatrix} \hat{\mathbf{g}}_i(\xi, \eta) \\ \hat{\mathbf{G}}_i(\xi, \eta) \end{bmatrix} \bar{\mathbf{T}}_i^{-1}, \quad \bar{\mathbf{T}} \equiv \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{J}^{-1} \end{bmatrix}, \quad (31)$$

or,

$$\mathfrak{G}_i(x, y) = \bar{\mathbf{T}} \hat{\mathfrak{G}}_i(\xi, \eta) \bar{\mathbf{T}}_i^{-1}, \quad \hat{\mathfrak{G}}_i(\xi, \eta) \equiv \begin{bmatrix} \hat{\mathbf{g}}_i(\xi, \eta) \\ \hat{\mathbf{G}}_i(\xi, \eta) \end{bmatrix}, \quad (32)$$

where the subscript on $\bar{\mathbf{T}}_i^{-1}$ indicates evaluation at node i . Likewise, the stream function/velocity elements (22) and (23) can be combined as,

$$\begin{bmatrix} \mathbf{s}_i(x, y) \\ \mathbf{S}_i(x, y) \end{bmatrix} = \mathbf{T} \begin{bmatrix} \hat{\mathbf{s}}_i(\xi, \eta) \\ \hat{\mathbf{S}}_i(\xi, \eta) \end{bmatrix} \mathbf{T}_i^{-1}, \quad \mathbf{T} \equiv \begin{bmatrix} 1 & 0 \\ 0 & \Delta^{-1} \mathbf{J}^T \end{bmatrix}. \quad (33)$$

In a similar manner, we can factor out a post-multiplying matrix from (24),

$$\mathbf{s}_i(x, y) = \hat{\mathbf{s}}_i(\xi, \eta) \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \eta_h^{-1} & 0 & 0 & 0 & 0 \\ 0 & 0 & \xi_h^{-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & \xi_h^{-2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \xi_h^{-1}\eta_h^{-1} & 0 \\ 0 & 0 & 0 & 0 & 0 & \eta_h^{-2} \end{bmatrix}, \quad (34)$$

which generalizes to

$$\mathbf{s}_i(x, y) = \hat{\mathbf{s}}_i(\xi, \eta) \mathbf{T}_i^{-1},$$

$$\mathbf{T}_i^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & y_{,\eta} & -x_{,\eta} & 0 & 0 & 0 \\ 0 & -y_{,\xi} & x_{,\xi} & 0 & 0 & 0 \\ 0 & y_{,\xi\xi} & -x_{,\xi\xi} & x_{,\xi}^2 & 2x_{,\xi}y_{,\xi} & y_{,\xi}^2 \\ 0 & y_{,\xi\eta} & -x_{,\xi\eta} & x_{,\xi}x_{,\eta} & (x_{,\xi}y_{,\eta} + x_{,\eta}y_{,\xi}) & y_{,\xi}y_{,\eta} \\ 0 & y_{,\eta\eta} & -x_{,\eta\eta} & x_{,\eta}^2 & 2x_{,\eta}y_{,\eta} & y_{,\eta}^2 \end{bmatrix}, \quad (35)$$

with the matrix elements evaluated at node i , for a more-general transformation. This sparse matrix is diagonal for rectangles, block diagonal for affine elements and lower block triangular for general quadrilaterals. The inverse is easily calculated from the small blocks. Let

$$\mathbf{T}_i^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & A & 0 \\ 0 & C & B \end{bmatrix}, \quad \text{then} \quad \mathbf{T}_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & A^{-1} & 0 \\ 0 & -B^{-1}CA^{-1} & B^{-1} \end{bmatrix}. \quad (36)$$

Two submatrix inverses are given explicitly by,

$$A^{-1} = \Delta^{-1} \mathbf{J}^T, \quad B^{-1} = \Delta^{-2} \begin{bmatrix} y_{,\eta}^2 & -2y_{,\eta}y_{,\xi} & y_{,\xi}^2 \\ -y_{,\eta}x_{,\eta} & y_{,\eta}x_{,\xi} + x_{,\eta}y_{,\xi} & -y_{,\xi}x_{,\xi} \\ x_{,\eta}^2 & -2x_{,\eta}x_{,\xi} & x_{,\xi}^2 \end{bmatrix}. \quad (37)$$

The generalization of $\mathbf{S}_i = \mathbf{curl} \mathbf{s}_i$ adds premultiplication by $\Delta^{-1} \mathbf{J}^T$. This may be summarized by

$$\begin{aligned} \mathbf{s}_i(x, y) &= \hat{\mathbf{s}}_i(\xi, \eta) \mathbf{T}_i^{-1}, \\ \mathbf{S}_i(x, y) &= \Delta^{-1} \mathbf{J}^T \hat{\mathbf{S}}_i(\xi, \eta) \mathbf{T}_i^{-1}, \end{aligned} \quad (38)$$

with $\hat{\mathbf{s}}_i(\xi, \eta)$ and $\hat{\mathbf{S}}_i(\xi, \eta)$ restricted to the reference square, and the matrices \mathbf{T}_i^{-1} given above. Thus, the shape functions can be calculated once for a given quadrature rule and cached. Then only the pre- and post-multiplying matrices need be calculated on a given element, followed by one or two matrix multiplications.

If we augment these elements with higher derivatives, defining $\mathfrak{S}_i(\xi, \eta) = [\mathbf{s}_i, \mathbf{S}_i, \psi_{i,\xi\xi}, \psi_{i,\xi\eta}, \psi_{i,\eta\eta}]^T$, then in analogy with (33),

$$\mathfrak{S}_i(x, y) = \mathbf{T}_i \hat{\mathfrak{S}}_i(\xi, \eta) \mathbf{T}_i^{-1}. \quad (39)$$

This provides for computation of higher derivatives of ψ (first derivatives of \mathbf{S}_i) required for the fluid diffusion term.

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

$$\begin{aligned}\mathbf{C}_{ij}^e(\mathbf{U}) &= \int_{-1}^1 d\xi \int_{-1}^1 d\eta \Delta \mathbf{S}_i^T(\xi, \eta) (U_0 \mathbf{S}_{j,x}(\xi, \eta) + V_0 \mathbf{S}_{j,y}(\xi, \eta)), \\ \mathbf{D}_{ij}^e &= - \int_{-1}^1 d\xi \int_{-1}^1 d\eta \Delta (\mathbf{S}_{i,x}^T(\xi, \eta) \mathbf{S}_{j,x}(\xi, \eta) + \mathbf{S}_{i,y}^T(\xi, \eta) \mathbf{S}_{j,y}(\xi, \eta)), \\ \mathbf{B}_{ij}^e &= \int_{-1}^1 d\xi \int_{-1}^1 d\eta \Delta \mathbf{S}_i^T(\xi, \eta) \begin{bmatrix} 0 \\ 1 \end{bmatrix} \mathbf{g}_j(\xi, \eta) \bar{\mathbf{T}}_j^{-1},\end{aligned}\quad (40)$$

where,

$$\begin{aligned}\mathbf{S}_k(\xi, \eta) &= \Delta^{-1} \mathbf{J}^T \hat{\mathbf{S}}_k(\xi, \eta) \mathbf{T}_k^{-1}, \\ \mathbf{S}_{k,x}(\xi, \eta) &= \Delta^{-1} \mathbf{J}^T (J_{11}^{-1} \hat{\mathbf{S}}_{k,\xi}(\xi, \eta) + J_{12}^{-1} \hat{\mathbf{S}}_{k,\eta}(\xi, \eta)) \mathbf{T}_k^{-1}, \quad k = \{i, j\}. \\ \mathbf{S}_{k,y}(\xi, \eta) &= \Delta^{-1} \mathbf{J}^T (J_{21}^{-1} \hat{\mathbf{S}}_{k,\xi}(\xi, \eta) + J_{22}^{-1} \hat{\mathbf{S}}_{k,\eta}(\xi, \eta)) \mathbf{T}_k^{-1}.\end{aligned}\quad (41)$$

Likewise,

$$\begin{aligned}\bar{\mathbf{C}}_{ij}^e(\mathbf{U}) &= \int_{-1}^1 d\xi \int_{-1}^1 d\eta \Delta (\hat{\mathbf{g}}_i(\xi, \eta) \bar{\mathbf{T}}_i^{-1})^T (U \mathbf{g}_{j,x}(\xi, \eta) + V \mathbf{g}_{j,y}(\xi, \eta)), \\ \bar{\mathbf{D}}_{ij}^e &= - \int_{-1}^1 d\xi \int_{-1}^1 d\eta \Delta (\mathbf{g}_{i,x}^T(\xi, \eta) \mathbf{g}_{j,x}(\xi, \eta) + \mathbf{g}_{i,y}^T(\xi, \eta) \mathbf{g}_{j,y}(\xi, \eta)) \bar{\mathbf{T}}_j^{-1},\end{aligned}\quad (42)$$

where,

$$\begin{bmatrix} \mathbf{g}_{k,x}(\xi, \eta) \\ \mathbf{g}_{k,y}(\xi, \eta) \end{bmatrix} = \mathbf{G}_k(\xi, \eta) = \mathbf{J}^{-1} \hat{\mathbf{G}}_k(\xi, \eta) \bar{\mathbf{T}}_k^{-1},\quad (43)$$

$$\begin{aligned}\mathbf{G}_{k,x}(\xi, \eta) &= \mathbf{J}^{-1} (J_{11}^{-1} \hat{\mathbf{G}}_{k,\xi}(\xi, \eta) + J_{12}^{-1} \hat{\mathbf{G}}_{k,\eta}(\xi, \eta)) \bar{\mathbf{T}}_k^{-1} \\ \mathbf{G}_{k,y}(\xi, \eta) &= \mathbf{J}^{-1} (J_{21}^{-1} \hat{\mathbf{G}}_{k,\xi}(\xi, \eta) + J_{22}^{-1} \hat{\mathbf{G}}_{k,\eta}(\xi, \eta)) \bar{\mathbf{T}}_k^{-1} \quad k = \{i, j\}.\end{aligned}\quad (44)$$

Similar expressions for the pressure matrices in (12) follow from the weak pressure equation (11).

References

- [1] R Temam. *Navier-Stokes Equations, Theory and Numerical Analysis*. AMS Chelsea Pub., Providence, 2001.
- [2] V Girault and P-A Raviart. *Finite Element Methods for Navier-Stokes Equations – Theory and Algorithms*. Springer-Verlag, Berlin, 1986.
- [3] J T Holdeman. A Hermite finite element method for incompressible fluid flow. *To appear in Int J Numer Methods Fluids. Published online 1 September 2009 as DOI: 10.1002/flid.2154*, pages 1–33, 2009.

- [4] M A Christon, P M Gresho, and S B Sutton. Computational predictability of time-dependent natural convection flows in enclosures (including a benchmark solution). *Int J Numer Methods Fluids*, 40:953–980, 2002.
- [5] G de Vahl Davis. Natural convection of air in a square cavity: a benchmark numerical solution. *Int J Numer Methods Fluids*, 3:249–264, 1983.
- [6] P Le Quéré. Accurate solutions to the square thermally driven cavity at high rayleigh number. *Computers Fluids*, 20:29–41, 1991.
- [7] H S Mahdi and R B Kenney. Time-dependent natural convection in a square cavity: application of a new finite volume method. *Int J Numer Methods Fluids*, 11:57–86, 1990.
- [8] C Shu and K H A Wee. Numerical simulation of natural convection in a square cavity by simple-generalized differential quadrature method. *Computers Fluids*, 31:209–226, 2002.
- [9] R J Melosh. Basis of derivation of matrices for the direct stiffness method. *Journal A. I. A. A.*, 1:1631–1637, 1963.
- [10] O C Zienkiewicz. *The Finite Element Method in Engineering Science*. McGraw-Hill, London, 1971.
- [11] S Gopalacharyulu. A higher order conforming rectangular plate element. *International Journal for Numerical Methods in Engineering*, 6:305–308, 1973.
- [12] S Gopalacharyulu. Author’s reply to the discussion by Watkins. *International Journal for Numerical Methods in Engineering*, 10:472–474, 1976.
- [13] D S Watkins. A comment on Gopalacharyulu’s 24 node element. *International Journal for Numerical Methods in Engineering*, 10:471–472, 1976.
- [14] D S Watkins. On the construction of conforming rectangular plate elements. *International Journal for Numerical Methods in Engineering*, 10:925–933, 1976.
- [15] A W Wegmuller. A refined plate bending element. *International Journal Solids Structures*, 10:1173–1178, 1974.