http://dx.doi.org/10.1090/conm/218/03010

Contemporary Mathematics
Volume **218**, 1998
B 0-8218-0988-1-03010-7

Spectral/*hp* Methods For Elliptic Problems on Hybrid Grids

Spencer J. Sherwin, Timothy C.E. Warburton,
and George Em Karniadakis

1. Introduction

We review the basic algorithms of spectral/hp element methods on tetrahedral grids and present newer developments on hybrid grids consisting of tetrahedra, hexahedra, prisms, and pyramids. A unified tensor-product trial basis is developed for all elements in terms of non-symmetric Jacobi polynomials. We present in some detail the patching procedure to ensure C^0 continuity and appropriate solution techniques including a multi-level Schur complement algorithm.

In standard low-order methods the quality of the numerical solution of an elliptic problem depends critically on the grid used, especially in three-dimensions. Moreover, the efficiency to obtain this solution depends also on the grid, not only because grid generation may be the most computationally intensive stage of the solution process but also because it may dictate the efficiency of the parallel solver to invert the corresponding algebraic system. It is desirable to employ grids which can handle arbitrary geometric complexity and exploit existing symmetry and structure of the solution and the overall domain.

Tetrahedral grids provide great flexibility in complex geometries but because of their unstructured nature they require more memory compared with structured grids consisting of hexahedra. This extra memory is used to store connectivity information as well as the larger number of tetraheda required to fill a specific domain, i.e. five to six times more tetrahedra than hexahedra. From the parallel solver point of view, large aspect ratio tetrahedra can lead to substantial degradation of convergence rate in iterative solvers, and certain topological constraints need to be imposed to maintain a balanced parallel computation.

The methods we discuss in this paper address both of the aforementioned issues. First, we develop high-order hierarchical expansions with exponential convergence for smooth solutions, which are substantially less sensitive to grid distortions. Second, we employ hybrid grids consisting of tetrahedra, hexahedra, prisms, and pyramids that facilitate great discretisation flexibility and lead to substantial memory savings. An example of the advantage of hybrid grids was reported in [**5**] where only $170K$ tetrahedra in combination with prisms were employed to construct a hybrid grid around the high-speed-civil-transport aircraft instead of an estimated

1991 *Mathematics Subject Classification.* Primary 65N30; Secondary 65N50.

two million if tetrahedra were used everywhere instead of triangular prisms. In general, for elliptic problems with steep boundary layers *hybrid* discretisation is the best approach in accurately resolving the boundary layers while efficiently handling any geometric complexities.

In previous work [**13, 14**] we developed a spectral/hp element method for the numerical solution of the two- and three-dimensional unsteady Navier-Stokes equations. This formulation was implemented in the parallel code $\mathcal{N}\varepsilon\kappa\mathcal{T}\alpha r$ [**11**]. The discretisation was based on arbitrary triangulisations/ tetrahedrisations of (complex-geometry) domains. On each triangle or tetrahedron a spectral expansion basis is employed consisting of Jacobi polynomials of mixed weight that accommodate exact numerical quadrature. The expansion basis is hierarchical of variable order per element and retains the tensor product property (similar to standard spectral expansions), which is key in obtaining computational efficiency via the sum factorisation technique. In addition to employing standard tetrahedral grids for discretisation, the formulation employed is also based on standard finite element concepts. For example, the expansion basis is decomposed into vertex modes, edge modes, face modes and interior modes as in other hexahedral h-p bases [**16, 8**]. With this decomposition, the C^0 continuity requirement for second-order elliptic problems is easily implemented following a direct stiffness assembly procedure.

In this paper, we extend the preliminary work of [**10**] in formulating a unified hierarchical hybrid basis for multiple domains. Specifically, we describe the basis in tensor-product form using a new coordinate system and provide details on how these heterogeneous subdomains can be patched together. We then concentrate on investigating the scaling of the condition number of the Laplacian system and discuss solution techniques, including a multi-level Schur complement algorithm [**15**].

2. Unified Hybrid Expansion Bases

In this section we shall develop a unified hybrid expansion basis suitable for constructing a C^O global expansion using triangular and quadrilateral regions in two-dimensions and tetrahedral, pyramidic, prismatic and hexahedral domains in three-dimensions. This unified approach lends itself naturally to an object orientated implementation as originally developed in [**17**] using C++ in the code $\mathcal{N}\varepsilon\kappa\mathcal{T}\alpha r$. To construct these expansion we must first introduce an appropriate coordinate system as discussed in section 2.1. Having developed the coordinate system the definition of the basis in terms of Jacobi polynomials is outlined in section 2.2.

2.1. Coordinate Systems. We define the standard quadrilateral region as

$$\mathcal{Q}^2 = \{(\xi_1, \xi_2) | -1 \leq \xi_1, \xi_2 \leq 1\},$$

within which we note that the Cartesian coordinates (ξ_1, ξ_2) are bounded by constant limits. This is not, however, the case in the standard triangular region defined as

$$\mathcal{T}^2 = \{(\xi_1, \xi_2) | -1 \leq \xi_1, \xi_2;\ \xi_1 + \xi_2 \leq 0\}.$$

where the bounds of the Cartesian coordinates (ξ_1, ξ_2) are clearly dependent upon each other. To develop a suitable tensorial type basis within unstructured regions, such as the triangle, we need to develop a new coordinate system where the local

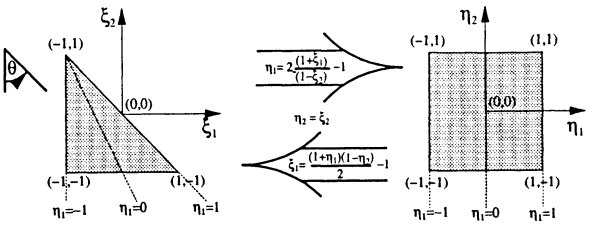


FIGURE 1. Triangle to rectangle transformation.

coordinates have independent bounds. The advantage of such a system is that we can then define one-dimensional functions upon which we can construct our multi-domain tensorial basis. It also defines an appropriate system upon which we can perform important numerical operations such as integration and differentiation [**6**].

2.1.1. *Collapsed Two-Dimensional Coordinate System.* A suitable coordinate system, which describes the triangular region between constant independent limits, is defined by the transformation

$$\eta_1 = 2\frac{(1+\xi_1)}{(1-\xi_2)} - 1 \tag{1}$$
$$\eta_2 = \xi_2,$$

and has the inverse transformation

$$\xi_1 = \frac{(1+\eta_1)(1-\eta_2)}{2} - 1 \tag{2}$$
$$\xi_2 = \eta_2.$$

These new local coordinates (η_1, η_2) define the standard triangular region by

$$\mathcal{T}^2 = \{(\eta_1, \eta_2) | -1 \leq \eta_1, \eta_2 \leq 1\}.$$

The definition of the triangular region in terms of the coordinate system (η_1, η_2) is identical to the definition of the standard quadrilateral region in terms of the Cartesian coordinates (ξ_1, ξ_2). This suggests that we can interpret the transformation (1) as a mapping from the triangular region to a rectangular one as illustrated in figure 1. For this reason, we shall refer to the coordinate system (η_1, η_2) as the *collapsed coordinate system.* Although this transformation introduces a multi-values coordinate (η_1) at $(\xi_1 = -1, \xi_2 = 1)$, we note that singular point of this nature commonly occur in cylindrical and spherical coordinate systems.

2.1.2. *Collapsed Three-Dimensional Coordinate Systems.* The interpretation of a triangle to rectangle mapping of the two-dimensional local coordinate system, as illustrated in figure 1, is helpful in the construction of a new coordinate system for three-dimensional regions. If we consider the local coordinates (η_1, η_2) as independent axes (although they are not orthogonal), then the coordinate system spans a rectangular region. Therefore, if we start with a rectangular region, or hexahedral region in three-dimensions, and apply the inverse transformation (2) we can derive a new local coordinate system in the triangular region $\mathcal{T}^2$, or tetrahedron region $\mathcal{T}^3$ in three-dimensions, where $\mathcal{T}^3$ is defined as:

$$\mathcal{T}^3 = \{-1 \leq \xi_1, \xi_2, \xi_3;\ \xi_1 + \xi_2 + \xi_3 \leq -1\}.$$

To reduce the hexahedron to a tetrahedron requires repeated application of the transformation in (2) as illustrated in figure 2. Initially, we consider a hexahedral

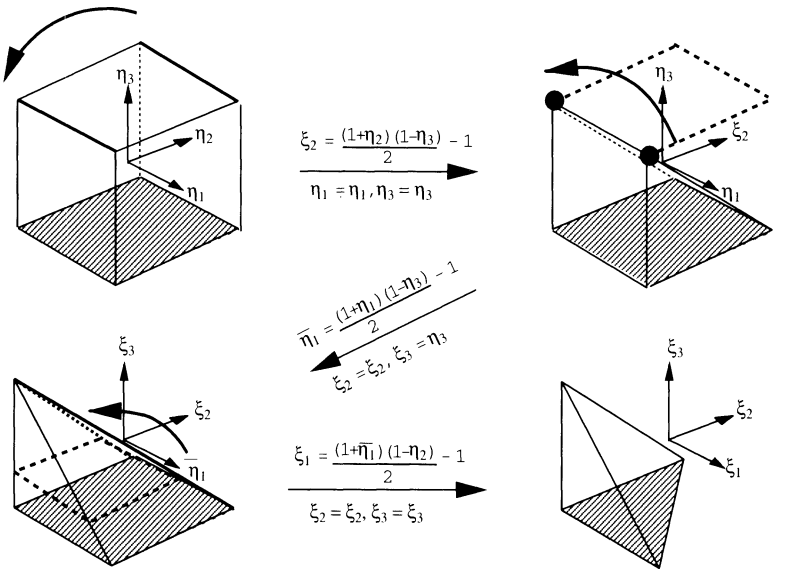


FIGURE 2. Hexahedron to tetrahedron transformation by repeatedly applying the rectangle to triangle mapping (2).

domain defined in terms of the local coordinate system (η_1, η_2, η_3) where all three coordinates are bounded by constant limits, i.e. $(-1 \leq \eta_1, \eta_2, \eta_3 \leq 1)$. Applying the rectangle to triangle transformation (2) in the (η_2, η_3) we obtain a new ordinate (ξ_2) such that

$$\xi_2 = \frac{(1+\eta_2)(1-\eta_3)}{2} - 1 \qquad \eta_3 = \eta_3.$$

Treating the coordinates (η_1, ξ_2, η_3) as independent, the region which originally spanned a hexahedral domain is mapped to a rectangular prism. If we now apply transformation (2) in the (η_1, η_3) plane, introducing the ordinates $\overline{\eta_1}, \xi_3$ defined as

$$\overline{\eta_1} = \frac{(1+\eta_1)(1-\eta_3)}{2} - 1 \qquad \xi_3 = \eta_3,$$

we see that the coordinates $(\overline{\eta_1}, \xi_2, \xi_3)$ span a region of a square based pyramid. The third and final transformation to reach the tetrahedral domain is a little more complicated as to reduce the pyramidic region to a tetrahedron we need to apply the mapping in every square cross section parallel to the $(\overline{\eta_1}, \xi_2)$ plane. This means using the transformation (2) in the $(\overline{\eta_1}, \xi_2)$ plane to define the final ordinate (ξ_1) as

$$\xi_1 = \frac{(1+\overline{\eta_1})(1-\xi_2)}{2} - 1 \qquad \xi_2 = \xi_2.$$

If we choose to define the coordinate of the tetrahedron region (ξ_1, ξ_2, ξ_3) as the orthogonal Cartesian system then, by determining the hexahedral coordinates (η_1, η_2, η_3) in terms of the orthogonal Cartesian system, we obtain

$$\eta_1 = 2\frac{(1+\xi_1)}{(-\xi_2-\xi_3)} - 1, \qquad \eta_2 = 2\frac{(1+\xi_2)}{(1-\xi_3)} - 1, \qquad \eta_3 = \xi_3, \tag{3}$$

TABLE 1. The local Collapsed Cartesian coordinates which have constant bounds within the standard region may be expressed in terms of the Cartesian coordinates ξ_1, ξ_2, ξ_3. Each region may be defined in terms of the local coordinates since having a lower bound of $-1 \leq \xi_1, \xi_2, \xi_3$ and upper bound as indicated in the table. Each region and the planes of constant local coordinate are shown in figure 3.

Region	Upper bound	Local Coordinate		
Hexahedron	$\xi_1, \xi_2, \xi_3 \leq 1$	ξ_1	ξ_2	ξ_3
Prism	$\xi_1, \xi_2 + \xi_3 \leq 1$	ξ_1	$\eta_2 = \frac{2(1+\xi_2)}{(1-\xi_3)} - 1$	ξ_3
Pyramid	$\xi_1 + \xi_3, \xi_2 + \xi_3 \leq 1$	$\overline{\eta_1} = \frac{2(1+\xi_1)}{(1-\xi_3)} - 1$	$\eta_2 = \frac{2(1+\xi_2)}{(1-\xi_3)} - 1$	$\eta_3 = \xi_3$
Tetrahedron	$\xi_1 + \xi_2 + \xi_3 \leq 1$	$\eta_1 = \frac{2(1+\xi_1)}{(-\xi_2-\xi_3)} - 1$	$\eta_2 = \frac{2(1+\xi_2)}{(1-\xi_3)} - 1$	$\eta_3 = \xi_3$

which is a new local coordinate system for the tetrahedral domain which is bounded by constant limits. When $\xi_3 = -1$ this system reduces to the two-dimensional system defined in (1).

In a similar manner, if we had chosen to define the coordinates in either the pyramidic or prismatic region as the orthogonal Cartesian system then evaluating the hexahedral coordinates in terms of these coordinates would generate a new local collapsed system for these domains. Table 1 shows the local collapsed coordinate systems in all the three-dimensional regions. A diagrammatic representation of the local collapsed coordinate system is shown in figure 3.

2.2. C^0 Continuous Unstructured Expansions. Ideally we would like to use an elemental expansion which can be assembled into an globally orthogonal expansion. Although it is possible to derive an orthogonal expansion within an elemental region [**12, 10**], the requirement to easily impose boundary conditions and tessellate the local expansions into multiple domains necessitates some modifications which destroy the orthogonality. To construct a C^0 continuous bases we decompose orthogonal expansion developed in [**12, 10**] into an interior and boundary contribution as is typical of all *hp* finite element methods [**16**]. The interior modes (or bubble functions) are defined as zero on the boundary of the local domain. The completeness of the expansion is then ensured by adding boundary modes which consist of vertex, edge and face contributions. The vertex modes have unit value at one vertex and decay to zero at all other vertices; edge modes have local support along one edge and are zero on all other edges and vertices; face modes have local support on one face and are zero on all other faces, edges and vertices. Using this decomposition C^0 continuity between elements can be enforced by matching similar shaped boundary modes providing some orientation constraints are satisfied as discussed in section 3. To construct the unified hybrid expansions we shall initially define a set of principal functions in section 2.2.1. Using these functions we then define the construction of the expansions in sections 2.2.2 and 2.2.3.

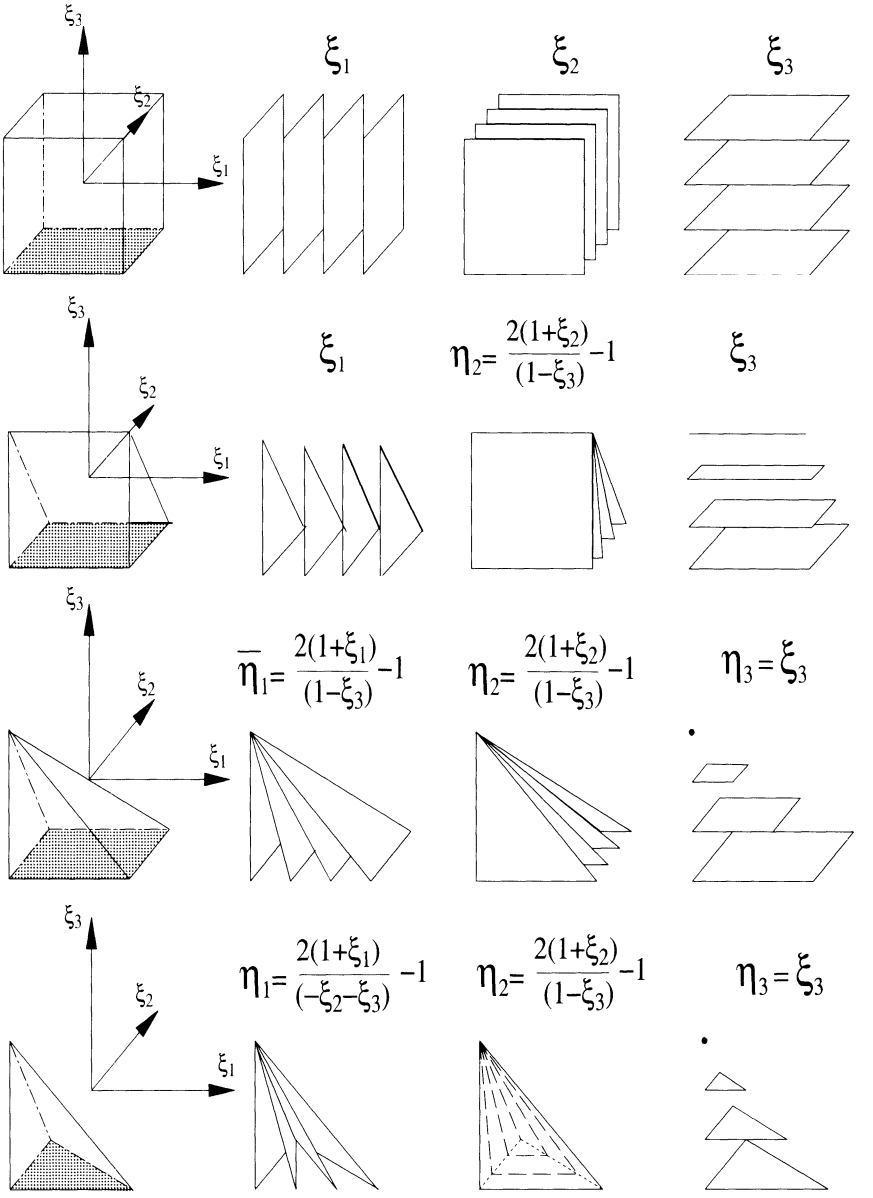


FIGURE 3. Planes of constant value of the local collapsed Cartesian coordinate systems in the hexahedral, prismatic, pyramidic and tetrahedral domains. In all but the hexahedral domain, the standard Cartesian coordinates ξ_1, ξ_2, ξ_3 describing the region have an upper bound which couples the coordinate system as shown in table 1. The local collapsed Cartesian coordinate system $\eta_1, \overline{\eta_1}, \eta_2, \eta_3$ represents a system of non-orthogonal coordinates which are bounded by a constant value within the region.

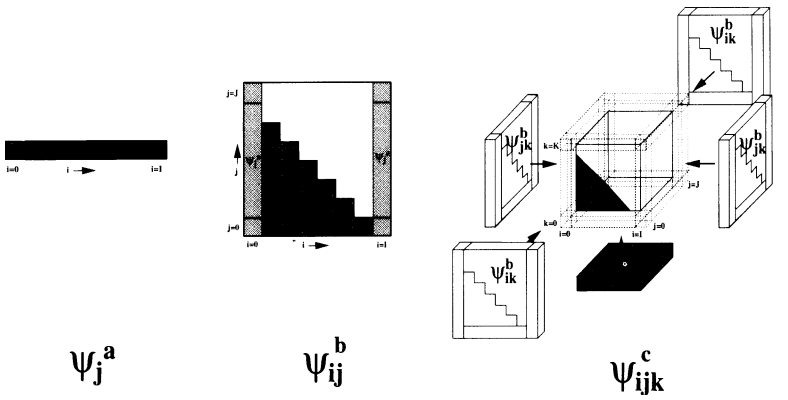


FIGURE 4. Illustration of the structure of the arrays of principal functions $\psi_i^a(z), \psi_{ij}^b(z)$ and $\psi_{ijk}^c(z)$. These arrays are not globally closed packed although any edge, face or interior region of the array may be treated as such. The interior of the arrays $\psi_{ij}^b(z)$ and $\psi_{ijk}^c(z)$ have been shaded to indicate the minimum functions required for a complete triangular and tetrahedral expansion.

2.2.1. *Principal Functions.* Denoting $P_i^{\alpha,\beta}(z)$ as the i^{th} order Jacobi polynomial which satisfied the orthogonality condition

$$\int_{-1}^{1}(1-z)^{\alpha}(1+z)^{\beta}P_i^{\alpha,\beta}(z)P_j^{\alpha,\beta}(z)dz = C\delta_{ij} \quad where \quad \alpha,\beta > -1,$$

we define three principal functions denoted by $\psi_i^a(z), \psi_{ij}^b(z)$ and $\psi_{ijk}^c(z)$ $(0 \le i \le I,\ 0 \le j \le J,\ 0 \le k \le K)$:

$$\psi_i^a(z) = \begin{cases} \left(\frac{1-z}{2}\right) & i = 0 \\ \left(\frac{1-z}{2}\right)\left(\frac{1+z}{2}\right) P_{i-1}^{1,1}(z) & 1 \le i \le I-1 \\ \left(\frac{1+z}{2}\right) & i = I \end{cases},$$

$$\psi_{ij}^b(z) = \begin{cases} \psi_j^a(z) & i = 0, \quad 0 \le j \le J \\ \left(\frac{1-z}{2}\right)^{i+1} & 1 \le i \le I-1, \quad j = 0 \\ \left(\frac{1-z}{2}\right)^{i+1}\left(\frac{1+z}{2}\right) P_{j-1}^{2i+1,1}(z) & 1 \le i \le I-1, \quad 1 \le j \le J-1 \\ \psi_j^a(z) & i = I, \quad 0 \le j \le J \end{cases},$$

$$\psi_{ijk}^c(z) = \begin{cases} \psi_{jk}^b(z) & i = 0, \quad 0 \le j \le J, \quad 0 \le k \le K \\ \psi_{ik}^b(z) & 0 \le i \le I, \quad j = 0, \quad 0 \le k \le K \\ \left(\frac{1-z}{2}\right)^{i+j+1} & 1 \le i \le I-1, \quad 1 \le j \le J-1, \quad k = 0 \\ \left(\frac{1-z}{2}\right)^{i+j+1}\left(\frac{1+z}{2}\right) P_{k-1}^{2i+2j+1,1}(z) & 1 \le i \le I-1, \quad 1 \le j \le J-1 \quad 1 \le k \le K-1 \\ \psi_{ik}^b(z) & 0 \le i \le I, \quad j = J, \quad 0 \le k \le K \\ \psi_{jk}^b(z) & i = I, \quad 0 \le j \le J, \quad 0 \le k \le K \end{cases}.$$

Figure 4 diagrammatically indicates the structure of the principle functions $\psi_i^a(z), \psi_{ij}^b(z)$ and $\psi_{ijk}^c(z)$ as well as how the function $\psi_i^a(z)$ is incorporated into $\psi_{ij}^b(z)$, and similarly how $\psi_{ij}^b(z)$ is incorporated into $\psi_{ijk}^c(z)$. The function $\psi_i^a(z)$ has been decomposed into two linearly varying components and a function which is zero at the end points. The linearly varying components generate the vertex modes which are identical to the standard linear finite element expansion. The interior contributions of all the base functions (i.e. $1 \le i \le I$–1, $1 \le j \le J$–1, $1 \le k \le K$–1) are similar in form to the orthogonal basis functions defined in [**10**]. However, they

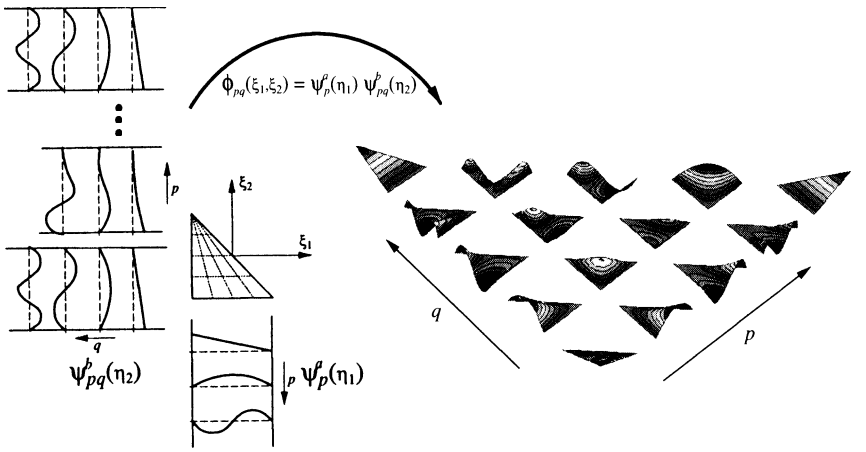


FIGURE 5. Construction of a fourth-order ($P = 4$) triangular expansion using the product of two principal functions $\psi^a_p(\eta_1)$ and $\psi^b_{pq}(\eta_2)$.

are now pre-multiplied by a factor of the form $\left(\frac{1-z}{2}\right)\left(\frac{1+z}{2}\right)$ which ensures that these modes are zero on the boundaries of the domain. The value of α, β in the Jacobi polynomial $P^{\alpha,\beta}_p(x)$ has also been slightly modified to maintain as much orthogonality as possible in the mass and Laplacian systems.

2.2.2. *Hybrid Expansions.* The two-dimensional expansions are defined in terms of the principal functions as:

$$\text{Quadrilateral expansion:} \quad \phi_{pq}(\xi_1,\xi_2) = \psi^a_p(\xi_1)\psi^a_q(\xi_2)$$

$$\text{Triangular expansion:} \quad \phi_{pq}(\xi_1,\xi_2) = \psi^a_p(\eta_1)\psi^b_{pq}(\eta_2)$$

where

$$\eta_1 = \frac{2(1+\xi_1)}{(1-\xi_2)} - 1, \quad \eta_2 = \xi_2,$$

are the two-dimensional collapsed coordinates. In figure 5 we see all of the modified expansion modes for a fourth-order ($P = 4$) modified triangular expansion. From this figure it is immediately evident that the interior modes have zero support on the boundary of the element. This figure also illustrates that the shape of every boundary mode along a single edge is identical to one of the modes along the other two edges and which allows the modal shapes in two regions to be globally assembled into a C^0 continuous expansion. In the three-dimensional expansion an equivalent condition is ensured by the introduction of $\psi^b_{ij}(z)$ into $\psi^c_{ijk}(z)$.

The three-dimensional expansions are defined in terms of the principal functions as:

$$\text{Hexahedral expansion:} \quad \phi_{pqr}(\xi_1,\xi_2,\xi_3) = \psi^a_p(\xi_1)\psi^a_q(\xi_2)\psi^a_r(\xi_3)$$

$$\text{Prismatic expansion:} \quad \phi_{pqr}(\xi_1,\xi_2,\xi_3) = \psi^a_p(\xi_1)\psi^a_q(\eta_2)\psi^b_{qr}(\xi_3)$$

$$\text{Pyramidic expansion:} \quad \phi_{pqr}(\xi_1,\xi_2,\xi_3) = \psi^a_p(\overline{\eta_1})\psi^a_q(\eta_2)\psi^c_{pqr}(\eta_3)$$

$$\text{Tetrahedral expansion:} \quad \phi_{pqr}(\xi_1,\xi_2,\xi_3) = \psi^a_p(\eta_1)\psi^b_{pq}(\eta_2)\psi^c_{pqr}(\eta_3)$$

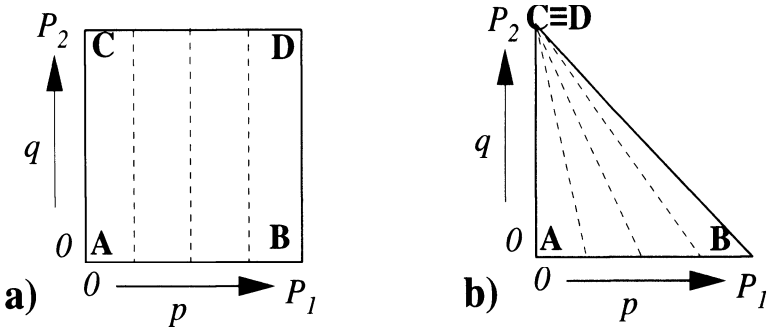


FIGURE 6. The construction of the collapsed Cartesian coordinates system maps vertex D onto vertex C in plot (a). If we consider the quadrilateral region in plot (a) as describing a two-dimensional array in p and q then we can imagine an equivalent array within the triangular region as shown in plot (b).

where

$$\eta_1 = \frac{2(1+\xi_1)}{(-\xi_2-\xi_3)} - 1, \quad \overline{\eta_1} = \frac{2(1+\xi_1)}{(1-\xi_3)} - 1, \quad \eta_2 = \frac{2(1+\xi_2)}{(1-\xi_3)} - 1, \quad \eta_3 = \xi_3,$$

are the three-dimensional collapsed coordinates.

2.2.3. *Construction of Basis From Principal Functions.* As can be appreciated from figure 4 the principal functions for the unstructured regions are not in a closed packed form and so we cannot consecutively loop over the indices p, q and r to arrive at a complete polynomial expansion. Even though these arrays are not closed packed their definition permits an intuitive construction of the expansion basis as discussed below.

Two-Dimensions

The quadrilateral expansion may be constructed by considering the definition of the basis $\phi_{pq}(\xi_1, \xi_2)$ as a two-dimensional array within the standard quadrilateral region with the indices $p = 0, q = 0$ corresponding to the lower left hand corner as indicated in figure 6(a). Using this diagrammatic form of the array it was easy to construct the vertex and edge modes by determining the indices corresponding to the vertex or edge of interest. A similar approach is possible with the modified triangular expansion.

We recall that to construct the local coordinate system we used a collapsed Cartesian system where vertex D in figure 6(a) was collapsed onto vertex C as shown in figure 6(b). Therefore, if we use the equivalent array system in the triangular region we can construct our triangular expansions. For example, the vertices marked A and B in figure 6(b) are defined as

$$\begin{array}{lcll} \text{Vertex A} & = & \phi_{00}(\eta_1, \eta_2) & = \psi^a_0(\eta_1)\psi^b_{00}(\eta_2) \\ \text{Vertex B} & = & \phi_{P_1 0}(\eta_1, \eta_2) & = \psi^a_{P_1}(\eta_1)\psi^b_{P_1 0}(\eta_2). \end{array}$$

The vertex at the position marked CD in figure 6(b) was formed by collapsing the vertex D onto vertex C in figure 6(a). Therefore this mode is generated by adding the contribution from the indices corresponding to the vertices C and D, i.e.

$$\text{Vertex CD} = \phi_{0P_2}(\eta_1, \eta_2) + \phi_{P_1 P_2}(\eta_1, \eta_2) = \psi^a_0(\eta_1)\psi^b_{0P_2}(\eta_2) + \psi^a_{P_1}(\eta_1)\psi^b_{P_1 P_2}(\eta_2).$$

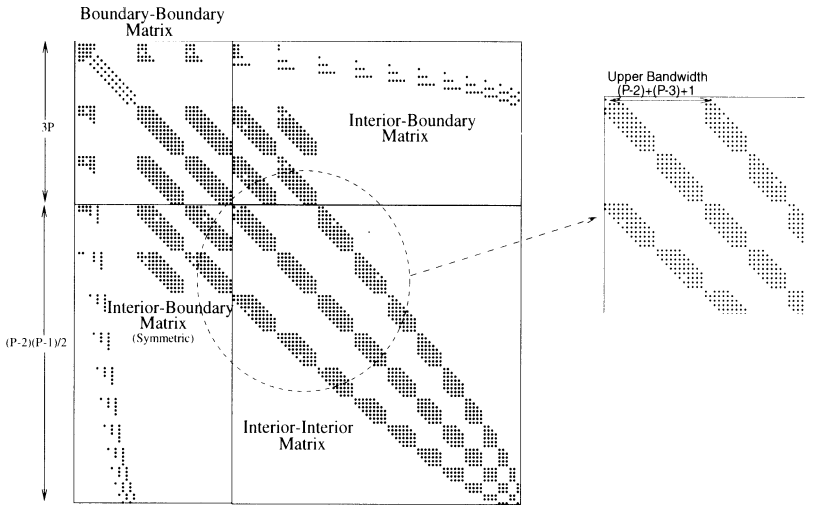


FIGURE 7. The structure of the mass matrix for a triangular expansion $\phi^{pq} = \psi_p^a \psi_{pq}^b$ of order $P_1 = P_2 = 14$ within the standard region $\mathcal{T}^2$. The boundary modes have been ordered first followed by the interior modes. If the q index is allowed to run faster, the interior matrix has a bandwidth of $(P-2)+(P-3)+1$.

For the triangular expansion the edge modes are similarly defined as:

$$\begin{aligned}
\text{Edge AB}: \quad \phi_{p0}(\eta_1,\eta_2) &= \psi_p^a(\eta_1)\psi_{p0}^b(\eta_2) && (0<p<P_1)\\
\text{Edge AC}: \quad \phi_{0q}(\eta_1,\eta_2) &= \psi_0^a(\eta_1)\psi_{0q}^b(\eta_2) && (0<q<P_2)\\
\text{Edge BD}: \quad \phi_{P_1 q}(\eta_1,\eta_2) &= \psi_{P_1}^a(\eta_1)\psi_{P_1 q}^b(\eta_2) && (0<q<P_2).
\end{aligned}$$

In constructing the triangular region from the quadrilateral region as shown in figure 6 edge CD was eliminated and, as one might expect, it does not contribute to the triangular expansion.

Finally the interior modes of the modified triangular expansion (which become the triangular face modes in the three-dimensional expansions) are defined as

$$\text{Interior}: \phi^{pq}(\eta_1,\eta_2) = \psi_p^a(\eta_1)\psi_{pq}^b(\eta_2) \qquad (0<p,q;\ p<P_1;\ p+q<P_2;\ P_1\le P_2).$$

There is a dependence of the interior modes in the p-direction on the modes in the q-direction which ensures that each mode is a polynomial in terms of the Cartesian coordinates (ξ_1,ξ_2). This dependence requires that there should be as many modes in the q direction as there are in the p direction and hence the restriction that $P_1 \le P_2$. A complete polynomial expansion typically involves all the modes defined above and this expansion is optimal in the sense that it spans the widest possible polynomial space in (ξ_1,ξ_2) with the minimum number of modes. More interior or edge modes could be used but if they are not increased in a consistent manner the polynomial space will not be increased. In figure 7 we see the structure of the mass matrix for a $P_1 = P_2 = 14$ polynomial order triangular expansion within the standard triangular region. The matrix is ordered so the boundary modes are

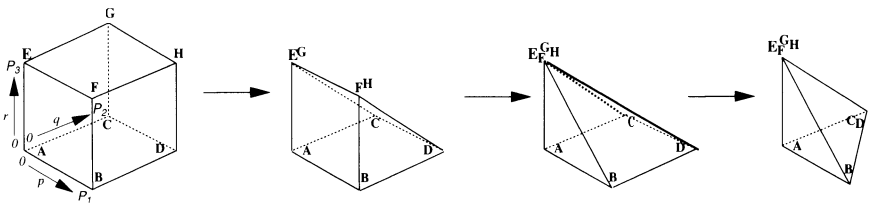


FIGURE 8. Generation of the standard tetrahedral domains from repeated collapsing of a hexahedral region.

first followed by the interior system. It can be shown (see [**13**]) that if we order the interior system so the q index runs fastest then the bandwidth of the interior system is $(P-2)+(P-3)+1$.

Three-Dimensions

As illustrated in figure 8, for the hexahedral domain the indices p, q, r correspond directly to a three-dimensional array where all indices start from zero at the bottom left-hand corner. Therefore, the vertex mode labelled A is described by $\phi_{(000)} = \psi_0^a(\xi_1)\psi_0^a(\xi_2)\psi_0^a(\xi_3)$, similarly the vertex mode labelled H is described by $\phi_{(P_1,P_2,P_3)}$ and the edge modes between C and G correspond to $\phi_{0,P_2,r}$ $(1 < r < P_3)$.

When considering the prismatic domain we use the *equivalent* hexahedral indices. Accordingly, vertex A is now described by $\phi_{(000)} = \psi_0^a(\xi_1)\psi_0^a(\eta_2)\psi_{00}^b(\xi_3)$. In generating the new coordinate system, vertex G was mapped to vertex E and therefore the vertex mode, labelled EG in the prismatic domain, is described by $\phi_{(0,0,P_3)}+\phi_{(0,P_2,P_3)}$ (i.e., adding the two vertices from the hexahedral domain which form the new vertex in the prismatic domain). A similar addition process is necessary for the prismatic edge $EG-FH$ which is constructed by adding the edge modes EF (i.e. $\phi_{(p,0,P_3)}$) to the edge modes GH (i.e., $\phi_{(p,P_2,P_3)}$). In degenerating from the hexahedral domain to the prismatic region the edges EG and FH are removed and therefore do not contribute to the prismatic expansion.

This process can also be extended to construct the expansion for the pyramidic and tetrahedral domains. For both these cases the top vertex is constructed by summing the contribution of E, F, G and H. In the tetrahedral domain edges CG and DH are also added. Although the modified functions ψ_{ij}^b and ψ_{ijk}^c are not closed packed, every individual edge, face and the interior modes may be summed consecutively.

As as final point we note that the use of the collapsed Cartesian coordinate system means that the triangular faces, unlike the quadrilateral faces, are not rotationally symmetric. This means that there is a restriction on how two triangular faces, in a multi-domain expansion, must be aligned. In section 3 we show that this condition can easily be satisfied for all tetrahedral meshes although some care must be taken when using a mixture of different elemental domains.

3. Global Assembly

The elements we have described will be tessellated together to construct a continuous solution domain. We shall only permit elements to connect by sharing common vertices, complete edges and/or complete faces. such a connectivity is

commonly referred to as conforming elements. In this section we shall discuss issues related to the process of globally assembling the elemental bases described in section 2.2.

When two elements share an edge it is important for them to be able to determine if their local coordinate system at that edge are aligned in the same direction. This information is important since it is necessary to ensure that the shape of all edges modes is similar along an edge. If the local coordinate systems are not aligned in the same direction then edges modes of odd polynomial order will have different signs and so one edge mode will need to be multiplied by -1.

This condition becomes more complicated in three dimensions when two elements share a face. In this case it is not automatic that their coordinate systems on the common face will line up. Considering the tetrahedra we see that there is a vertex on each face that the coordinate system for that face radiates from. Similarly for the triangular faces of the prism and pyramid. We will call this vertex the face origin as it is similar to a polar coordinate origin. The alignment constraint necessitates that when two triangular faces meet their origin vertices must coincide. Initally it is not obvious how to satisfy this constraint for a mesh consisting of just tetrahedral elements. We outline two algorithms that will satisfy this constraint. The first is based on the topology of the mesh. We will only use the connections between elements to determine how we should orientate elements. In the second method we will assume that each unique vertex in the mesh will have been given a number. This second method works under some loose conditions but is extremely easy to implement and is very local in its nature.

It is useful to observe that one of the vertices of a tetrahedron is the face origin vertex for the three faces sharing that vertex. We will call this the *local top vertex*. Then there is one more face origin vertex on the remaining face which we call the *local base vertex*.

3.1. Algorithm 1. Given a conforming discretisation we can generate the local orientation of the tetrahedra using the following algorithm. We assume that we have a list of vertices and we know a list of elements which touch each vertex. This list of elements will be called a vertex group and all elements are assumed to have a tag of zero.

For every vertex in the list:

- Orientate all elements with a tag of one in this vertex group so that their *local base vertex* points at this vertex. Then set their tags to two.
- Orientate all elements with a tag of zero in this vertex group so that their *local top vertex* points at this vertex. Then set their tags to one.

This algorithm visits all vertices in the mesh and if this is the first time the elements in the vertex group have been visited the *local top vertex* is orientated at this vertex. If this is the second time the elements in the vertex group have been visited then set the *local base vertex* to this vertex. To see how this works we can consider the example shown in figure 9.

Here we assume that we are given a discretisation of a box using six tetrahedra as shown in figure 9a. Starting our algorithm we begin with vertex A. Since all elements have a tag of zero at this point we go straight to the second part of the algorithm and orientate all elements that touch this vertex so that their *local top vertices* point to A. Therefore tetrahedra HBDA and BHEA are orientated as shown in figure 9b and now have a tag set to one. Continuing to the next vertex B we

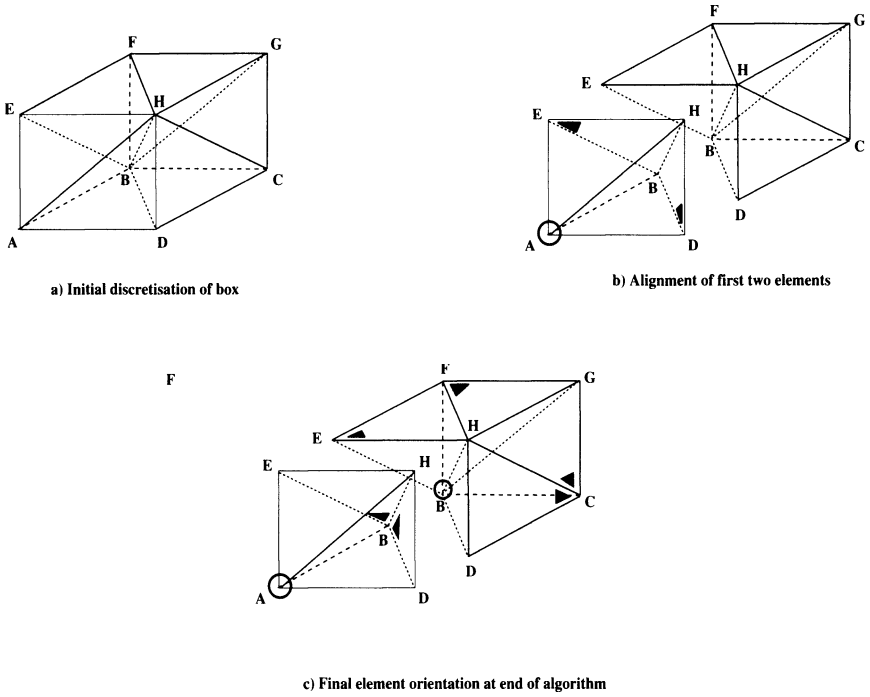


FIGURE 9. Setting up the required connectivity for the discretisation of a box as shown in a). Vertex A is given as the first *local top vertex* as shown in b). In c) vertex B is then given as the *local base vertex* and the *local base vertices* from group one are aligned to satisfy connectivity. The final element orientation is shown in figure d).

see that all elements belong to this vertex group. The first part of the algorithm is to orientate the elements with a tag of one to have their *local base vertex* pointing at B. So the tetrahedra HBDA and BHEA are rotated as shown in figure 9c and their tags are set to two. The second part of the algorithm then orientates all the other tetrahedra to have their *local top vertex* pointing at B. The connectivity is actually satisfied at this point since the orientation the faces have on the boundaries is irrelevant. However, if we continue the algorithm looping through the vertices consecutively we end up with the tetrahedra orientated as shown in figure 9d.

Clearly, the connectivity is not unique since any elements that have their *local top vertex* pointing at E can be rotated about E. However, we have demonstrated that it is possible to satisfy the connectivity requirements imposed by the co-ordinate system and thereby imply that the requirement is non-restrictive.

3.2. Algorithm 2. Assuming that every global vertex has a unique number, then for every element we have four vertices with unique global numbers:

- Place the *local top vertex* at the global vertex with the lowest global number.
- Place the *local base vertex* at the global vertex with the second lowest global number
- Orientate the last two vertices to be consistent with the local rotation of the element (typically anti-clockwise).

It has been stated before that since the coordinate systems on the faces of the tetrahedra are not symmetric that it is too difficult to use these coordinate systems. We have shown that it is possible in linear or even constant time to satisfy this constraint for any given tetrahedral mesh. This algorithm is local to each element and should be implemented at a pre-processing stage.

We now extend this approach to include meshes consisting of tetrahedra, prisms, and hexahedra. Unfortunately, in this case we find counter-examples where it is not possible to satisfy the origin alignment constraint. We have isolated the problematic cases and they are unlikely to come up when using a mesh generator.

First we deal with the case when a quadrilateral face is shared by two elements. In this instance it is sufficient to simply make the coordinate directions agree by simply reversing either face coordinate if necessary.

We now investigate over-constrained meshes. These cases can occur when prisms and tetrahedra are used together in a mesh. We will use these examples to motivate the actual algorithm we propose. The cost of this algorithm also depends linearly on the number of elements in the mesh.

It is instructive to construct a chain of prisms. This is simply a long prism, with equilateral triangular faces, divided at intervals along its length into a set of prisms connected at their triangle faces. The connectivity constraint requires that the coordinate origins of the triangular faces must meet at every prism-prism interface. This condition enforces that the collapsed edge must run in a continuous line through the edges of the prism. Now we twist the chain around in a loop and connect its triangle ends. The chain now forms a closed loop of prisms. The orientation of the end faces of the original chain must also satisfy the connectivity constraint when they meet. But we are free to choose the orientation of the faces relative to each other. However, we can make the chain into a Mobius band by twisting it around the axis along its length. In this case the connectivity cannot be satisfied without changing the mesh.

We can construct a second counter-example, this time involving one tetrahedron and two chains of prisms. We construct two chains of prisms as outlined above and we join the tetrahedra into the prism chain by connecting two of its faces to the prism chain triangular end faces. We repeat this operation again connecting the remaining two faces of the tetrahedron to the end faces of the second chain of prisms. We can now repeat the twisting of the prism loops. This over constrain the tetrahedron so that it cannot be oriented to satisfy the connectivity condition.

These two cases indicate that we cannot allow prism chains to reconnect into closed loops and still satisfy our constraints. Also we should not allow a prism chain to connect to a tetrahedron with more than one of its triangular faces. If we only consider meshes that satisfy these two constraints, then the following algorithm will satisfy the connectivity constraints:

3.3. Algorithm For Connecting Prisms And Tetrahedra.

- Find all prism chains in the mesh.
- Create a virtual connection between the faces of the tetrahedra that meet the triangular faces at each end of the chain.
- Proceed with Algorithm 2 to connect the tetrahedral mesh treating the virtual links as real connections.

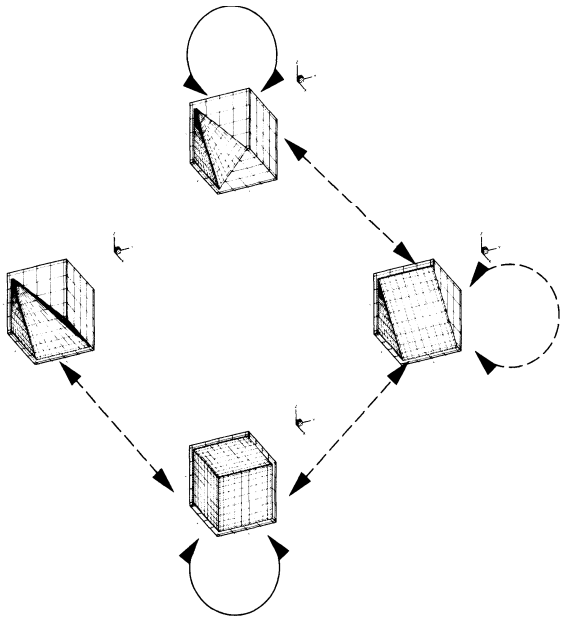


FIGURE 10. Connectivity summary. Solid arrows imply we can connect elements together with no problems, dashed arrows mean that there are some constraints on allowable configurations.

- Orient the prisms in the chains with the same orientation as the triangular faces of the tetrahedra at the ends of the virtual link. This orientation propagate through the chain.

In figure 10 we summarise which elements can be connected using the above algorithms. A solid line between elements means we can connect a given mesh with of the two element types with no problems. If the line is dashed then the mesh has to be changed to meet the connectivity constraints.

4. Global Matrix Properties

As discussed in section 2 and 3 we can construct a global expansions using a tessellated of hybrid domains which are C^0 continuous. These expansions are therefore suitable to solve second order partial differential equations using a Galerkin formulation. Consider the Helmholtz problem

$$\nabla^2 u - \lambda u = f$$

supplemented with appropriate boundary conditions. The Galerkin problem may be stated as find $u^\delta \in \mathcal{X}^\delta$ such that

$$a(v^\delta, u^\delta) = f(v^\delta) \qquad \forall v^\delta \in \mathcal{V}^\delta, \tag{4}$$

where

$$\begin{aligned} a(v,u) &= \int_\Omega \nabla v \cdot \nabla u + \lambda v u \, d\Omega, \\ f(v) &= \int_\Omega v \, f \, d\Omega. \end{aligned}$$

and u^δ is the finite dimensional representation of the solution (i.e. $u^\delta = \sum \hat{u}_{pqr}\phi_{pqr}$) and $\mathcal{X}^\delta, \mathcal{V}^\delta$ are the finite dimensional space of trial and test functions. In the Galerkin approximations we assume that $\mathcal{X}^\delta$ and $\mathcal{V}^\delta$ span the same space.

As is typically of most numerical approaches, equation (4) may be represented as an algebraic system. Although these algebraic systems are typically sparse the number of degrees of freedom of a practical three-dimensional problem requires that we use an iterative solver. We are therefore interested in the conditioning of these algebraic systems. However, before considering the conditioning of this system we shall first review the restructuring of the global matrix using the static condensation technique to take advantage of the matrix structure when using spectral/hp expansions.

4.1. Matrix Solution via Schur Complement. Let us denote the global matrix problem due to the Galerkin problem (4) as

$$\boldsymbol{M}\boldsymbol{x} = \boldsymbol{f}. \tag{5}$$

where $\boldsymbol{x}$ is a vector of global unknowns. The matrix $\boldsymbol{M}$ is typically very sparse although it may have a full bandwidth. We shall assume that the global system $\boldsymbol{M}$ is ordered so that the global boundary degrees of freedom are listed first, followed by the global interior degrees of freedom. In addition, we also assume that the global interior degrees of freedom were numbered consecutively. Adopting this ordering, the global matrix problem (5) can be written as

$$\begin{bmatrix} \boldsymbol{M}_b & \boldsymbol{M}_c \\ \boldsymbol{M}_c^T & \boldsymbol{M}_i \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_b \\ \boldsymbol{x}_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_b \\ \boldsymbol{f}_i \end{bmatrix} \tag{6}$$

where we have distinguished between the boundary and interior components of $\boldsymbol{x}$ and $\boldsymbol{f}$ using $\boldsymbol{x}_b, \boldsymbol{x}_i$ and $\boldsymbol{f}_b, \boldsymbol{f}_i$, respectively.

The matrix $\boldsymbol{M}_b$ corresponds to the global assembly of the elemental boundary-boundary mode contributions and similarly $\boldsymbol{M}_c, \boldsymbol{M}_i$ correspond to the global assembly of the elemental boundary-interior coupling and interior-interior systems. A notable feature of the global system is that the global boundary-boundary, $\boldsymbol{M}_b$, matrix is sparse and may be re-ordered to reduce the bandwidth or re-factored in a multi-level Schur Complement solver as discussed below. The global boundary-interior coupling matrix, $\boldsymbol{M}_c$, is very sparse and as we shall see may be stored in terms of its local elemental contributions. Finally, the natural form of $\boldsymbol{M}_i$ is a block diagonal matrix which is very inexpensive to evaluate since each block may be inverted individually.

To solve the system (6) we can statically condense out the interior degrees of freedom by performing a block elimination. Pre-multiplying system (6) by the matrix

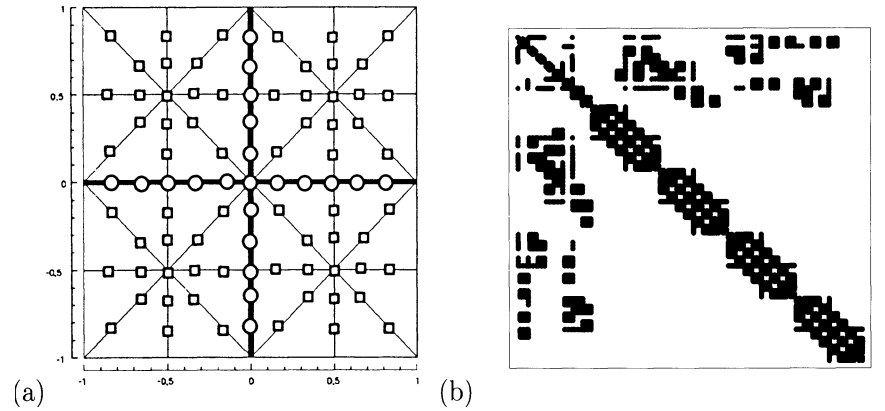
$$\begin{bmatrix} \boldsymbol{I} & -\boldsymbol{M}_c\boldsymbol{M}_i^{-1} \\ 0 & \boldsymbol{I} \end{bmatrix},$$

we arrive at:

$$\begin{bmatrix} \boldsymbol{M}_b - \boldsymbol{M}_c\boldsymbol{M}_i^{-1}\boldsymbol{M}_c^T & 0 \\ \boldsymbol{M}_c^T & \boldsymbol{M}_i \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_b \\ \boldsymbol{x}_i \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_b - \boldsymbol{M}_c\boldsymbol{M}_i^{-1}\boldsymbol{f}_i \\ \boldsymbol{f}_i \end{bmatrix}. \tag{7}$$

The equation for the boundary unknowns is therefore:

$$(\boldsymbol{M}_b - \boldsymbol{M}_c\boldsymbol{M}_i^{-1}\boldsymbol{M}_c^T)\boldsymbol{x}_b = \boldsymbol{f}_b - \boldsymbol{M}_c\boldsymbol{M}_i^{-1}\boldsymbol{f}_i. \tag{8}$$



(a) (b)

FIGURE 11. The boundary degrees of freedom, on the mesh shown in plot (a), are ordered so that the boundary modes indicated by the square symbols are first, followed by the boundary modes indicated by circular symbols within each quadrant. Using this ordering the resulting Schur complement matrix has a block diagonal sub-matrix as shown in figure (b).

Once $\boldsymbol{x}_b$ is known, we can determine $\boldsymbol{x}_i$ from the second row of equation (7) since

$$\boldsymbol{x}_i = \boldsymbol{M}_i^{-1}\boldsymbol{f}_i - \boldsymbol{M}_i^{-1}\boldsymbol{M}_c^T\boldsymbol{x}_b. \tag{9}$$

4.1.1. *Multi-Level Schur Complement.* The motivation behind using Schur complement was the natural decoupling of the interior degrees of freedom within each element leading to a global system which contained a block diagonal sub-matrix. This decoupling can be mathematically attributed to the fact that the interior degrees of freedom in one element are orthogonal to the interior degrees of freedom of another simply because these modes are non-overlapping. To take advantage of this block diagonal sub-matrix we have to construct the Schur complement system

$$\boldsymbol{M}_S = \boldsymbol{M}_b - \boldsymbol{M}_c[\boldsymbol{M}_i]^{-1}(\boldsymbol{M}_c)^T.$$

The effect of constructing each of this system is to orthogonalise the boundary modes from the interior modes. However, the inverse matrix $[\boldsymbol{M}_i]^{-1}$ is typically full, which means that the boundary modes, within an element, become tightly coupled. It is this coupling which dictates the bandwidth of the globally assembled Schur complement system. Nevertheless, an appropriate numbering of the boundary system will lead to a Schur complement matrix which also contains a sub-matrix that is block diagonal and so the static condensation technique can be re-applied. This technique has been more commonly used in the structural mechanics field and is also known as sub-structuring [**15**].

To illustrate this ordering we consider the triangular mesh shown in figure 11(a) using $N_{el} = 32$ elements. The construction of the global Schur complement $\boldsymbol{M}_S$ requires us to globally number all of the boundary degrees of freedom as indicated by the open circles and squares. If we order the numbering of the elemental boundary degrees of freedom so that the vertex and edge modes indicated by the open circles

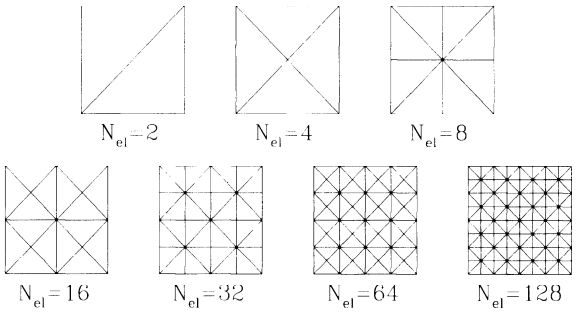


FIGURE 12. Triangulisations used in determining the eigen-spectrum of the Laplacian operator.

are first followed by the vertex and edge modes within each quadrant, indicated by the open squares, then the resulting Schur complement system of the mass matrix for a polynomial expansion of $p = 6$ is shown in figure 11(b). The block diagonal structure of the matrix is due to the fact that even after constructing the elemental Schur complement systems the boundary degrees of freedom in each quadrant do not overlap and so are orthogonal.

We can now construct another Schur complement system to solve for the circle degrees of freedom and decoupling each quadrant of square degrees of freedom. This technique can be repeated providing that there is more than one region of non-overlapping data. The approach is clearly independent of the elemental shape and may equally well be applied to quadrilateral regions or any hybrid shape in three-dimensions.

4.1.2. *Preconditioners.* As mentioned previously, the resolution requirements for problems of practical interest typically require the use of iterative algorithms. The convergence of such algorithms depends on the eigen-spectrum of these matrices as well as the preconditioners used for convergence acceleration.

In the conjugate gradient method we estimate that the number of iterations, N_{iter}, to invert a matrix $\mathbf{M}$ scales with the square root of the condition number, i.e.

$$N_{iter} \propto [\kappa_2(\boldsymbol{M})]^{1/2}.$$

The L^2 condition number of a matrix $\mathbf{M}$ is defined as $\kappa_2(\mathbf{M}) = ||M||_2||M^{-1}||_2$, which for a symmetric matrix is equivalent to the ratio of the largest to the smallest eigenvalue.

In two-dimensions, for the *full* discrete Laplacian $\boldsymbol{L}$ the condition number scales as

$$\kappa_2(\boldsymbol{M}) \propto N_{el}P^3.$$

The required number of iterations can therefore be very high for large problems, and especially high-order P. However, using static condensation we can consider the reduced problem consisting of the element boundary contributions by forming only the Schur complement $\boldsymbol{M}_S$. For a symmetric positive definite system the condition number of the Schur complement matrix $\boldsymbol{M}_S$ can be no larger than the condition number of the complete system $\boldsymbol{M}$ [**15**].

Experimental results indicating the scaling of $\kappa_2(\boldsymbol{M}_S)$ have been obtained in the domains shown in figure 12. The relationship of the condition number with

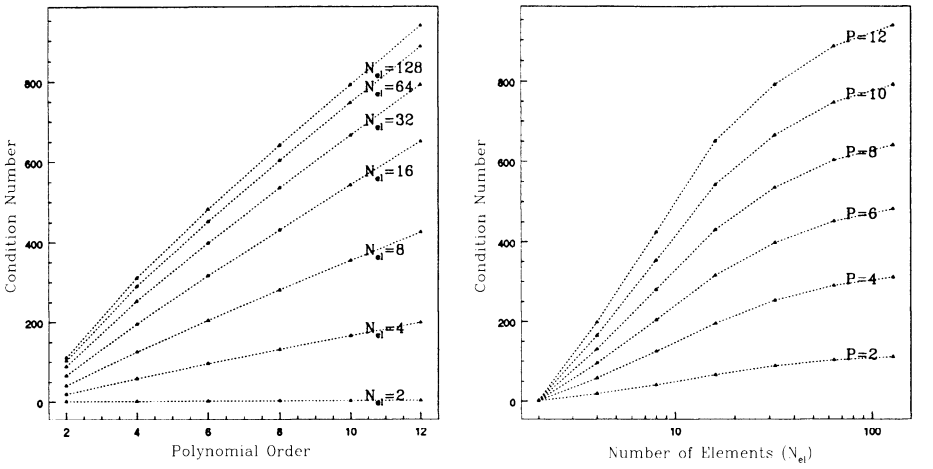


FIGURE 13. Condition number variation of the Schur complement of the discrete Laplacian operator with respect to the order (left) and the number of elements (right).

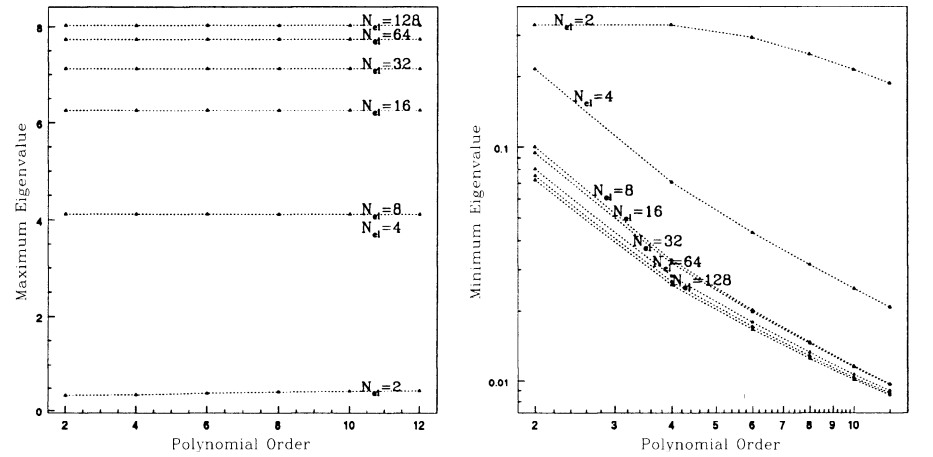
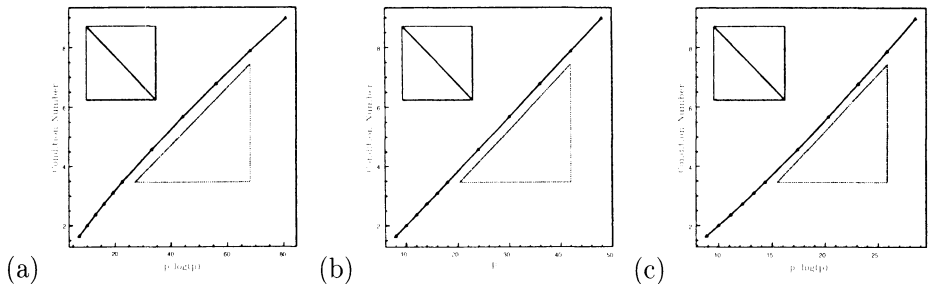


FIGURE 14. Maximum (left) and minimum (right) eigenvalue of the Schur complement of the discrete Laplacian operator with respect to the expansion order.

polynomial order P is demonstrated in figure 13(a), and with the number of elements in figure 13(b). The variation of the maximum and minimum eigenvalue with respect to the expansion order is shown in figure 14. The maximum eigenvalue is independent of the order P in accordance with the estimates in [**3**]. Also, the minimum eigenvalue varies as $\approx 1/P$ which is consistent with the theoretical upper bound estimate in [**3**] of $log(P)/P$. For the range considered, these results also seem to indicate that the condition number grows at most linearly with the order P and slower than logarithmically with the number of elements N_{el}.

To get a better indication of the asymptotic behaviour with the polynomial order, P, we can consider the case of two elemental regions for the polynomial



(a) (b) (c)

FIGURE 15. Condition number of the Schur complement of the discrete Laplacian operator for the $N_{el} = 2$ element domain plotted as a function of (a) $P\log(P)$, (b) P and (c) $P/\log(P)$.

range $8 \leq P \leq 48$ as shown in figure 15. In this test case we have imposed Dirichlet boundary conditions on all boundaries and so when we statically condense the system we are only left with the interior edge system. In figure 15 we see the condition number for this problem plotted against the functions (a) $P\log(P)$, (b) P and (c) $P/\log(P)$. As can be seen the condition number clearly grows at a slower rate than $P\log(P)$ but faster than $P/\log(P)$ which is consistent with the results of [**4**]. Although, not formally proven the asymptotic rate is most likely to scale with P. To extend this result to many subdomains the upper and lower bounds on the condition number $P\log(P)$ and $P/\log(P)$ should be scaled by a factor of $\log(P)^2$ [**3**]. Therefore, the asymptotic bound on κ_2 for a large P and N_{el} is

$$P\log(P) \leq \kappa_2 \leq P\log(P)^3.$$

However, when the number of elements, N_{el}, is not large a more conservative bound is

$$P/\log(P) \leq \kappa_2 \leq P\log(P)^3.$$

Since the upper bound is only realized for a large N_{el} the upper bound in these estimates is often observed to be very conservative. Furthermore, one would expect to observe a sharp upper bound of $P\log(P)^2$ based on the numerical behaviour shown in figure 15 for large P. These results may equally well be applied to the quadrilateral region which have similar edge support.

If the diagonal of the Schur complement is used as preconditioner, then the same scaling applies but the absolute magnitude of the condition number is approximately one order of magnitude less compared with the unpreconditioned case. This is not, however, true for the modal basis constructed in [**16**] based on the integrated Legendre polynomials rather than the $P_p^{1,1}(x)$ Jacobi polynomials. The difference between these two formulations in the quadrilateral case is the presence of a factor of P that provides the proper scaling and thus the similar growth in the unpreconditioned and diagonal-preconditioned case. On the other hand, if a block-diagonal preconditioner is used which is constructed based on blocks of edge-edge interactions, then the scaling changes substantially. Numerical experiments suggest a scaling of $(log(P))^2$, in agreement with the estimates reported in [**1**] for a similar construction. The condition number also seems to be independent of the number of elements for $N_{el} \geq 100$, again in agreement with the estimates in [**1**].

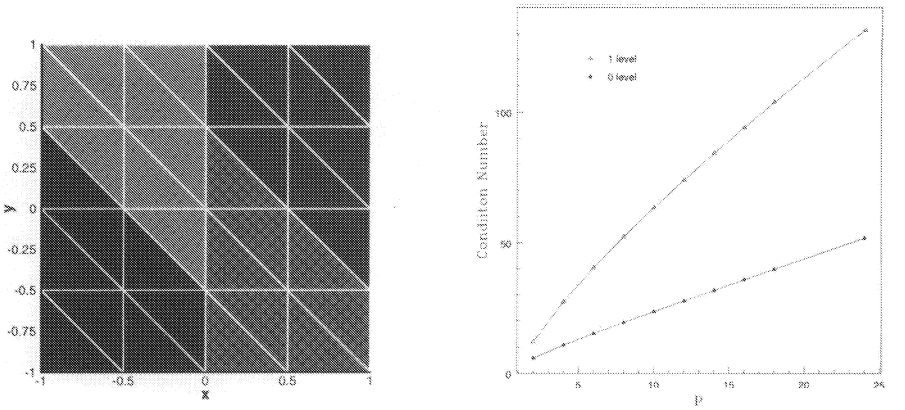


FIGURE 16. Using the $N_{el} = 32$ element mesh shown on the left the condition number of the standard Schur complement and the Schur complement after one level of decomposition were calculated and are shown on the right. The color in the left plot indicates the non-overlapping regions used for the extra level of decomposition.

As mentioned previously it can be shown [**15**] that for a symmetric positive definite matrix the maximum and minimum eigenvalue of the Schur complement are bounded by the maximum and minimum eigenvalue of the original matrix. Therefore, when using the multi-level Schur complement solver we know that the condition number of the inner-most Schur complement must be bounded by the standard Schur complement. This point is illustrated in figure 16 where we consider the condition number of the diagonally preconditioned standard Schur complement as well as the diagonally preconditioned Schur complement after one level of decomposition of a $N_{el} = 32$ elemental domain. The standard Schur complement contains information from all edges interior to the domain whereas in decomposing the system by one level all the edges within a shaded region are blocked together and statically condensed out of the system leaving only the edges along the interfaces between the shaded regions, see section 4.1.1. From the right hand plot in figure 16 we see that the condition number of the Schur complement after one level of decomposition is bounded by the condition number of the standard Schur complement. The effect of the extra level of decomposition is to reduce the slope of the curve although the condition number would appear still to be asymptotically growing with P.

The efficiency by which we can invert the *Helmholtz* matrix depends on the *combined* spectrum of the Laplacian matrix and the mass matrix. The eigen-spectrum of the Schur complement of the mass matrix has not been studied theoretically but numerical experiments with triangular elements suggest a similar dependence as the Laplacian matrix with respect to the number of elements N_{el}. However, with respect to order P its condition number grows much faster. In particular, for no-preconditioning we observed a growth $\kappa_2(\boldsymbol{M}) \propto P^{5/2}$; for diagonal-preconditioning $\kappa_2(\boldsymbol{M}) \propto P^{1.95}$; and for block-diagonal preconditioning as before we obtained $\kappa_2(\boldsymbol{M}) \propto P^{1.6}$. These are probably poly-logarithmic terms but we best-fitted experimental results to obtain these exponents.

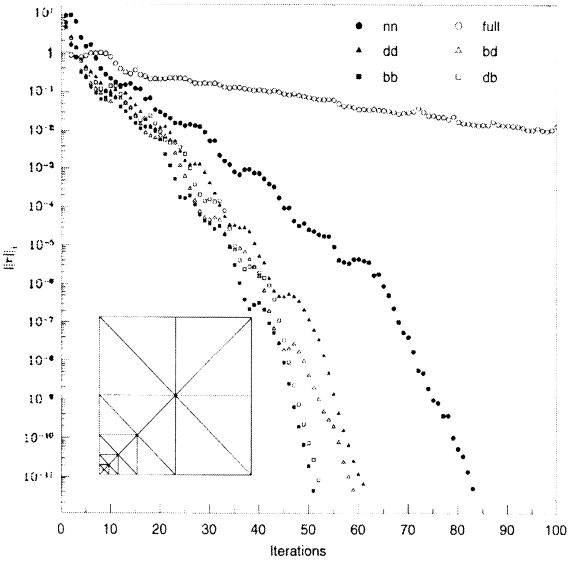


FIGURE 17. Convergence rate of a preconditioned conjugate gradient solver for a Schur complement of the Helmholtz matrix using the following preconditioners: (nn) vertex-non, edge-non; (dd) vertex-diagonal, edge-diagonal; (bd) vertex-diagonal, edge-block; (db) vertex-block, edge-diagonal; (bb) vertex-block, edge-block. Also shown for comparison is the solution for the full Helmholtz matrix (full).

In summary, it is possible to apply preconditioning techniques for inverting the Helmholtz matrix similar to preconditioners for the Laplacian and the mass matrix. The effect on the convergence rate of a preconditioned conjugate gradient solver for the Helmholtz equation with constant $\lambda = 1$ is shown in figure 17, which verifies the fast convergence for the Schur complement in contrast with the full discrete Laplacian.

In three dimensions, it is more difficult to establish estimates of the condition number. Some numerical experiments show that $\kappa_2(\boldsymbol{M}_S) \propto (log(P))^8$ without any preconditioning, but they are inconclusive with respect to the dependence in terms of the number of elements. In [**9**] a polylogarithmic bound was found of the form

$$\kappa_2 \leq C(1 + log(P))^2$$

which is independent on the number of elements. This estimate is valid for hexahedral elements but a similar bound was obtained in [**2**] for tetrahedral elements. The main idea is to use a wire basket preconditioner that is based on a new set of vertex and edge basis functions of "low energy". These low energy functions with highly oscillatory traces on the wire basket decay much faster than the standard basis functions constructed using barycentric coordinates. An alternative approach that employs orthogonalisation of each vertex function with respect to functions of its three faces, and each edge function with respect to functions of its two faces, has been proposed in [**7**].

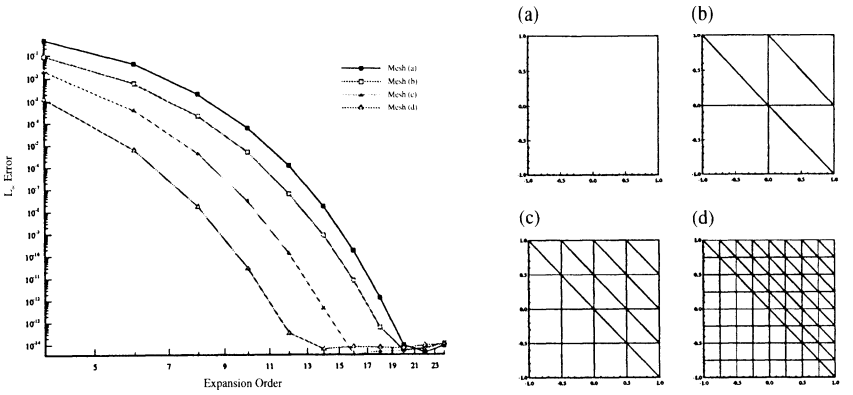


FIGURE 18. Convergence test for the Helmholtz problem, $(\nabla^2 u - \lambda u = f; \lambda = 1)$, using quadrilaterals and triangles, with Dirichlet boundary conditions. The exact solution is $u = sin(\pi x)cos(\pi y)$ and forcing function $f = -(\lambda + 2\pi^2)sin(\pi x)cos(\pi y)$.

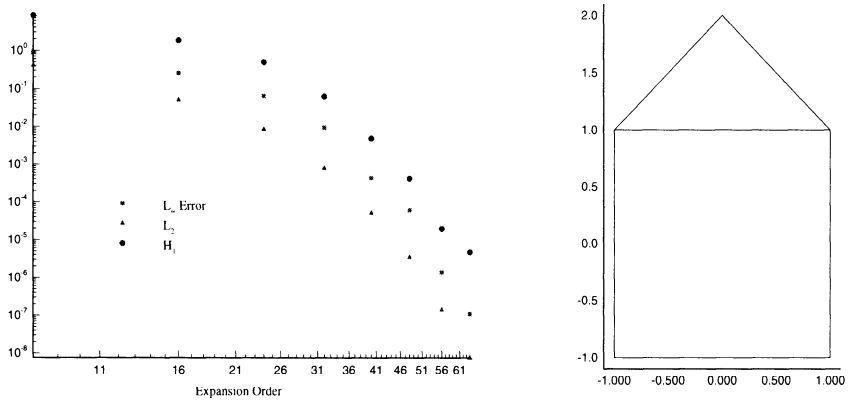


FIGURE 19. Convergence test for the Helmholtz problem, $(\nabla^2 u - \lambda u = f; \lambda = 1)$, using a triangle and a quadrilateral, with Dirichlet boundary conditions. The exact solution is $u = sin(\pi cos(\pi r^2))$ and forcing function $f = -(\lambda + 4\pi^4 r^2 sin(\pi r^2)^2)sin(\pi(cos(\pi r^2))) - 4\pi^2(\pi r^2 cos(\pi r^2) + sin(\pi r^2))cos(\pi cos(\pi r^2))$, where $r^2 = x^2 + y^2$.

5. Results

We now demonstrate that the method is stable up to high polynomial orders and works for complicated combinations of all the element types.

In figure 18 we demonstrate convergence to the exact solution with p-refinement (exponential rate) and h-refinement (algebraic rate) for the Helmholtz equation with $\lambda = 1$ for Dirichlet boundary conditions.

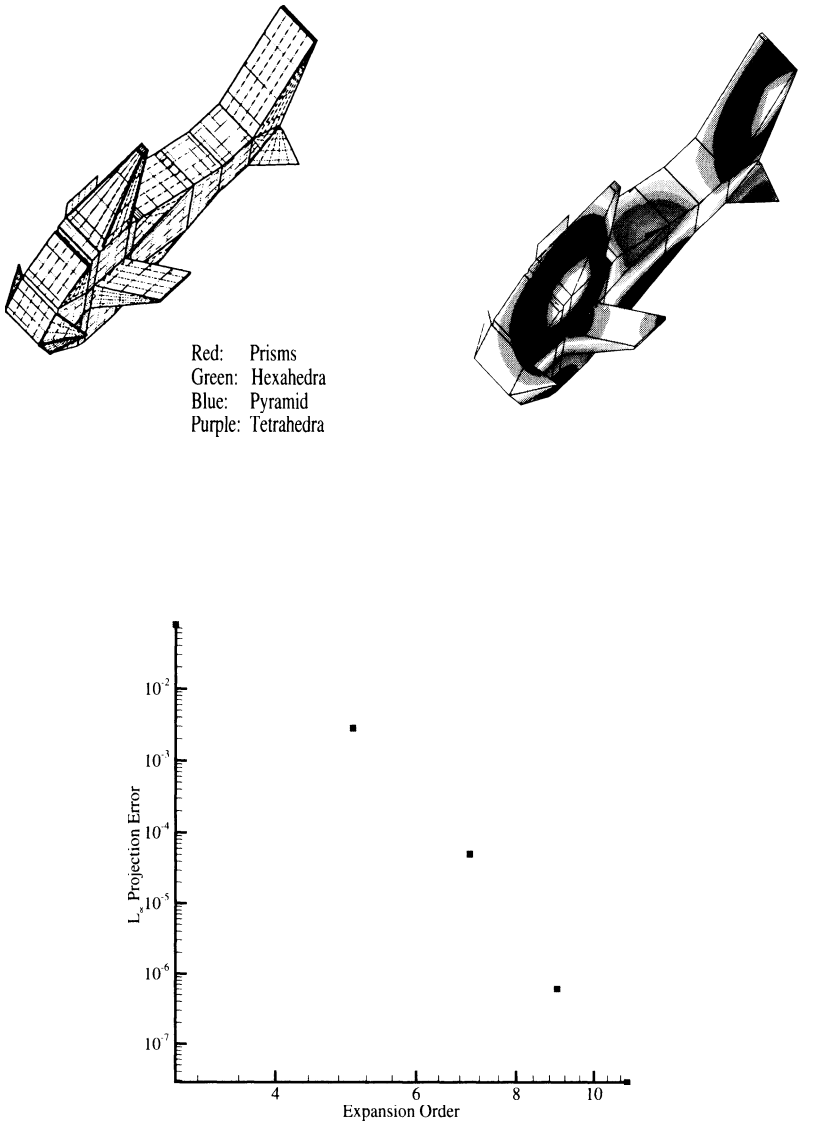


FIGURE 20. Convergence for the Helmholtz problem, $(\nabla^2 u - \lambda u = f; \lambda = 1)$, with Dirichlet boundary conditions on a mesh of twenty six hybrid elements. The exact solution is $sin(x)sin(y)sin(z)$.

In figure 19 we show $p-type$ convergence for a more complicated exact solution. This example demonstrates that the method is stable to at least $P = 64$. This is much higher order than the one used in hp finite element method [**16**].

In figure 20 we solve the Helmholtz problem on a complicated 3D domain discretized using all types of elements, i.e. tetrahedra, hexahedra, prisms and pyramids. Exponential convergence is also verified for a smooth solution.

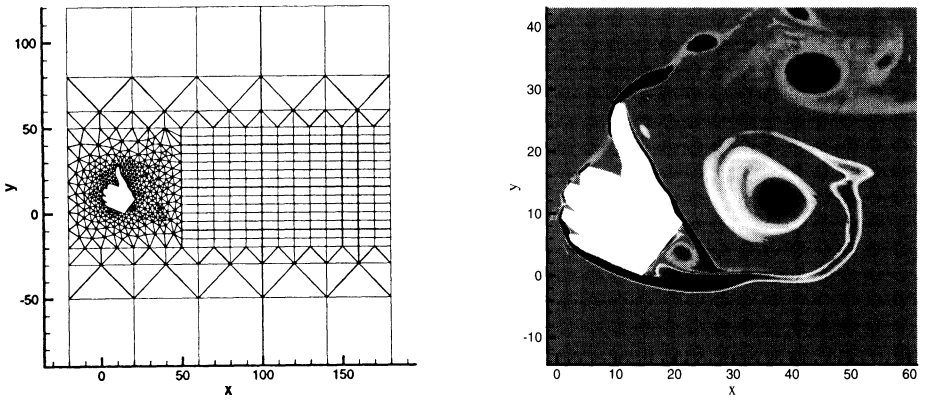


FIGURE 21. $\mathcal{N}\varepsilon\kappa\mathcal{T}\alpha r$ simulations of incompressible flow on the above domain discretized with $N_{el} = 966$ elements (192 quadrilaterals and 774 triangles) and 6th order polynomial expansions on each element. The instantaneous vorticity field is shown on the right plot. The Reynolds number is approximately 1200.

Finally, in figure 21 we show a solution of the incompressible Navier-Stokes equations. The algorithm for triangular elements is described in detail in [**13**] but here we use a mix of triangles and quadrilateral elements, the former to handle the complex geometry and the latter to more efficiently fill the computational domain. The flow computed is start-up from zero initial conditions with uniform inflow at Reynolds number (based on the vertical projected length) approximately 1200.

Acknowledgements

We would like to thank Dr. Mario Casarin for many helpful discussions regarding the scaling of the condition number presented in this paper. This work was partially supported by AFOSR, DOE, ONR and NSF. Computations were performed at the National Center for Supercomputing Aplications, University of Illinois at Urbana-Champaign, and at Maui High Performance Computing Center in Hawaii.

References

1. I. Babuška, A.W. Craig, J. Mandel, and J. Pitkaranta, *Efficient preconditioning for the p-version finite element method in two dimensions*, SIAM J. Numer. Anal. **28** (1991), 624–662.
2. I. Bica, *Iterative substructuring algorithms for the p-version finite element method for elliptic problems*, Ph.D. thesis, New York University, September 1997.
3. M.A. Casarin, *Schwarz preconditioners for spectral and mortar finite element methods with applications to incompressible fluids*, Ph.D. thesis, New York University, March 1996.
4. ———, *Diagonal edge preconditioners in p-version and spectral element methods*, SIAM J. Sci. Comp. **18** (1997), no. 2, 610–620.
5. Y. Kallinderis, A. Khawaja, and H. McMorris, *Hybrid prismatic/tetrahedral grid generation for complex geometries*, Tech. report, University of Texas at Austin, 1995.
6. G.E. Karniadakis and S.J. Sherwin, *Spectral/hp element methods for cfd*, Oxford University Press, New York, 1998.
7. J. Mandel, *Two-level decomposition preconditioning for the p-version finite element method in three dimensions*, Int. J. Num. Meth. Engrg **29** (1991), 1095–1108.

8. J.T. Oden, *Optimal hp-finite element methods*, Tech. Report TICOM Report 92-09, University of Texas at Austin, 1992.
9. L. Pavarino and O. Widlund, *A polylogarithmic bound for an iterative substructuring method for spectral elements in three dimensions*, SIAM J. Numer. Anal. **33** (1996), 1303–1335.
10. S.J. Sherwin, *Hierarchical hp finite elements in hybrid domains*, Finite Elements in Analysis and Design **27** (1997), 109–119.
11. S.J. Sherwin, C. Evangelinos, H. Tufo, and G.E. Karniadakis, *Development of a parallel unstructured spectral/hp method for unsteady fluid dynamics*, Parallel CFD '97, 1997, Manchester, UK.
12. S.J. Sherwin and G.E. Karniadakis, *A new triangular and tetrahedral basis for high-order finite element methods*, Int. J. Num. Meth. Eng. **38** (1995), 3775.
13. ———, *A triangular spectral element method; applications to the incompressible Navier-Stokes equations*, Comp. Meth. Appl. Mech. Eng. **123** (1995), 189.
14. ———, *Tetrahedral hp finite elements: Algorithms and flow simulations*, J. Comp. Phys. **124** (1996), 14.
15. B. Smith, P. Bjorstad, and W. Gropp, *Domain decomposition. parallel mutlilevel methods for elliptic differential equations*, Cambridge University Press, 1996.
16. B. Szabo and I. Babuška, *Finite Element Analysis*, John Wiley & Sons, 1991.
17. T.C.E. Warburton, *Spectral/hp element methods on polymorphic domains*, Ph.D. thesis, Brown University, 1998.

Department of Aeronautics, Imperial College,Prince Consort Road, London SW7 2BY

Center for Fluid Mechanics, Division of Applied Mathematics,Brown University, Providence, R.I. 02912

Center for Fluid Mechanics, Division of Applied Mathematics,Brown University, Providence, R.I. 02912