A STRAIGHT FORWARD STRUCTURE TO CONSTRUCT SHAPE FUNCTIONS FOR VARIABLE *p*-ORDER MESHES

M. S. Shephard, S. Dey and J. E. Flaherty
Scientific Computation Research Center, and
Rensselaer Rotorcraft Technology Center
Rensselaer Polytechnic Institute, Troy, NY 12180–3590
(518) 276–6795

Summary

This paper describes a method for specifying variable *p*-order meshes based on the topological hierarchy of mesh entities defining the finite elements. A decomposition of *p*-version mesh entity shape functions into a product of a function defined over the mesh entity and a blending function defined over the elements that the mesh entity bounds is presented. The decomposed shape functions allow the effective construction of the shape functions and derivatives needed to construct element matrices for variable *p*-refinement.

1. Introduction

The theory and computational advantages of adaptive *p*- and *hp*-version finite element methods for solving problems of mathematical physics have been well documented [4, 11, 19, 27]. Much research has focused on the development of optimal *p*- and *hp*-adaptive strategies and their efficient implementation [7, 8, 10, 18, 21]. The efficient construction of the element stiffness matrices for *hp*-version finite element meshes, where the polynomial orders can vary from element-to-element or even within the closure of individual elements, requires special consideration to construct appropriate shape functions. Issues associated with element-matrix construction in these cases can be summarized as:

1. Satisfaction of the C^0 continuity requirement.
2. Efficient construction of the shape functions.
3. Effective evaluation of integrals required to compute element matrices.
4. Accounting for geometric approximations on elements that often cover large portions of the domain.

This paper focuses on the first two issues for conforming meshes which can contain combinations of elements of various topologies.

In addition to the polynomial hierarchy of the shape functions, the constructs presented herein employ a hierarchy of topological entities of vertices, edges, faces and regions that define the closure of an element. This hierarchy and the association of shape functions with these entities, provide a convenient means for directly specifying variable p-order approximations which satisfy the appropriate continuity requirements.

The next section presents a geometry-based mesh definition based on the hierarchy of topological mesh entities. The concept of decomposing shape functions associated with a mesh entity into the product of a function of the desired p-order acting over that entity and a blending function that blends it over the element is introduced in Section 3. Sections 4 and 5 address the need to properly match the mesh entity parametric coordinate systems for various element topologies, and demonstrate that p-version shape functions written in these coordinate systems can be decomposed as desired. The computational cost of using the decomposed variable p-order shape functions is examined (Section 6), and a summary of the results follows (Section 7).

2. Geometry-Based Topological Hierarchy for the Specification of Variable *p*-Order Meshes

Standard finite element data structures consisting of node point coordinates and element connectivities are not adequate for variable p-order meshes and should be replaced by data structures that allow the independent assignment of polynomial order over the elements [7, 15]. The basic structure used here for specifying a variable p-order mesh parallels the boundary representations used in geometric modeling systems. The structure assumes a general geometry-based specification of the problem being analyzed and relies on the relationship between the mesh and geometric specification of the domain to support several key functions in the construction of element stiffness matrices.

Although a number of schemes are possible for defining the closure of a geometric domain Ω_G [22], the most advantageous are boundary-based schemes denoted by

$$\overline{\Omega}_G(T_G, S_G) \tag{1}$$

where S_G represents the information defining the entity shapes and T_G represents the topological types and adjacencies[1] of the entities that define the domain. In addition

[1] Adjacencies are the relationships between topological entities which bound each other, e.g., the edges that bound a face.

to being unique, the use of topological entities and their associativities provides a convenient abstraction to define the relationship between different models of the domain. In the present case, the two analysis domain models of concern are the geometric model and mesh model, which is a discretization of $\overline{\Omega}_G$. Boundary representations allow the convenient specification, with respect to the geometric domain, of the physical attributes for that analysis [24, 25]. Current computer aided design systems, furthermore, support a boundary representation of the domains defined within them. This allows the effective combination of these systems with automatic finite element analysis packages. A final advantage of recent boundary representations is their ability to properly represent non-manifold geometric domains commonly used for analysis [12, 17, 29].

Since the representation of the mesh domain, $\overline{\Omega}_M$, may not contain all of the necessary shape information of the geometric domain, $\overline{\Omega}_G$, it is critical to employ a representational scheme which can maintain the relationships between the two models. A boundary representation for the mesh consistent with that used for the geometric model is an ideal means to meet these needs [5].

Since individual finite elements are limited to simple regions that are bounded by simply connected faces, consideration of the topological entities for a mesh model can focus on the basic 0- to d-dimensional topological entities. For the three-dimensional case (d=3) these are:

$$T_M = \{M\{M^0\}, M\{M^1\}, M\{M^2\}, M\{M^3\}\} \tag{2}$$

where $M\{M^d\}$, $d = 0, 1, 2, 3$, are, respectively, the set of vertices, edges, faces and regions defining the primary topological entities of the mesh model M. This structure maintains the relationship between the mesh entity and the geometric model entity to which it belongs [23, 25]. Figure 1 depicts a structure where the topological adjacencies between entities one geometric dimension apart are maintained, as well as the relationship of each mesh entity with respect to the geometric model. Alternative structures [5] that maintain knowledge of each topological entity can also meet the needs of the variable p-order mesh specification described here.

The mesh topological adjacency relations [5] support a variety of operations critical to the effective application of the finite element method, including equation ordering to minimize solution time, associating boundary loads with appropriate elements, and evolving the mesh in adaptive calculations. The use of the topological hierarchy in the specification of variable p-order meshes consists of the simple assignment of the desired polynomial order information with each topological entity in the mesh. Thus, each mesh vertex M_i^0, edge M_i^1, face M_i^2, and region M_i^3, can have its own polynomial order

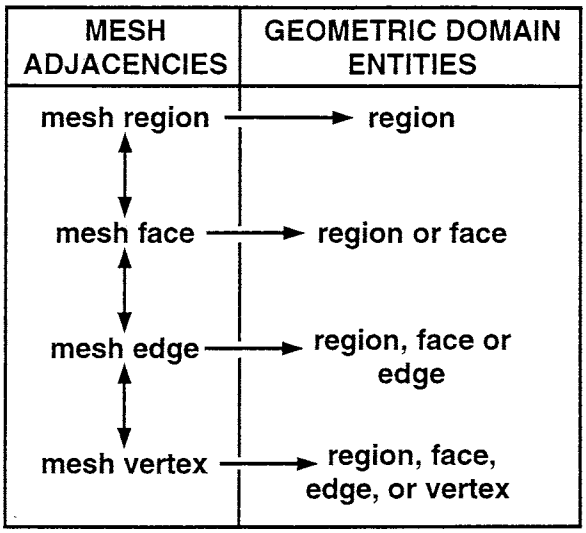

Figure 1. Mesh topological adjacencies and classification information.

specification, including the possibility of different directional polynomial orders for faces and regions.

Although the full set of mesh adjacencies [5] used in defining topologically-based mesh data structures is not used here, it will be useful to specify the first-order adjacency sets of individual mesh entities

$$M_i^{d_i}\left\{M^{d_j}\right\}, \tag{3}$$

which defines the set of mesh entities of dimension d_j adjacent to mesh entity $M_i^{d_i}$. For example, the mesh vertices bounding mesh region i is denoted as $M_i^3\{M^0\}$, while the mesh edges that are bounded by mesh vertex i is denoted as $M_i^0\{M^1\}$. When employing a mesh description in terms of a topological hierarchy, the first-order adjacencies which constitute the closure of a finite element, $\overline{\Omega}_e$, of dimension d_e can be specified and evaluated as

$$\overline{\Omega}_e = \left\{M_e^{d_e}, \partial\left(M_e^{d_e}\right)\right\} = \left\{M_e^{d_e}, M_e^{d_e}\left\{M_j^{d_e-1}\right\}, \ldots, M_e^{d_e}\{M_j^0\}\right\}. \tag{4}$$

3. Shape Function Decomposition Based on the Topological Entities of $\overline{\Omega}_e$

Shape function construction must consider that topological entities in $\overline{\Omega}_e$, can have independent polynomial order. Recalling equation (4), consider a shape function N

associated with mesh entity $M_j^{d_j} \in \overline{\Omega}_e$ in the form

$$N = \psi\left(M_j^{d_j}, M_e^{d_e}\right)\phi\left(M_j^{d_j}\right) \tag{5}$$

where

- $\psi\left(M_j^{d_j}, M_e^{d_e}\right)$ is a blending function defined on $M_e^{d_e}$ specific to $M_j^{d_j}$, written in the parametric coordinate system ξ_i of $M_e^{d_e}$ and independent of the polynomial order of N.
- $\phi\left(M_j^{d_j}\right)$ is a function written in the parametric coordinate system $\hat{\xi}_j$ of $M_j^{d_j}$ that depends on the polynomial order of N, and is independent of $M_e^{d_e}$. Thus, it is the same for all elements (including those of different topologies and/or dimension) connected to $M_j^{d_j}$.

Figure 2 depicts one possibility for this decomposition for a cubic shape function on an edge between a triangular and quadrilateral face. In this example, the cubic edge shape function on the triangular face is defined as $N(M_1^1, M_1^2) = \psi(M_1^1, M_1^2)\phi(M_1^1)$ and on the quadrilateral face as $N(M_1^1, M_2^2) = \psi(M_1^1, M_2^2)\phi(M_1^1)$. Assuming that individual blending functions can be determined, and the complexity is competitive, this approach provides a convenient framework to construct shape functions for variable p-order approximations. It automatically satisfies the C^0 interelement continuity requirements. Most constructions write shape functions such that odd-order shape functions on mesh entities adjacent to two elements require a change of sign on one of the elements because of the opposite use of that entity. With the decomposition (5), there is a single function on the mesh entity, $\phi\left(M_j^{d_j}\right)$, and $\psi\left(M_j^{d_j}, M_e^{d_e}\right)$ on each element ensures its correct behavior on the element.

The flexibility of a topologically-based p-order mesh specification allows consideration of a variety of approaches to construct the shape functions and to perform the operations required to calculate contributions to the element matrices. In any of them, a common feature is identification of the entities over which integrations involving shape functions must be performed to determine the coefficients of the stiffness matrix and load vector. For simplicity and consistency with standard finite element construction methods, these entities are referred to as elements and our focus is the definition of the proper set of shape functions over each element which:

1. Are of the polynomial order specified on each of the mesh entities included in $\overline{\Omega}_e$.
2. Satisfy the C^0 interelement continuity requirement.
3. Require a minimum of numerical operations in the evaluation of the shape functions and subsequent element integrations.

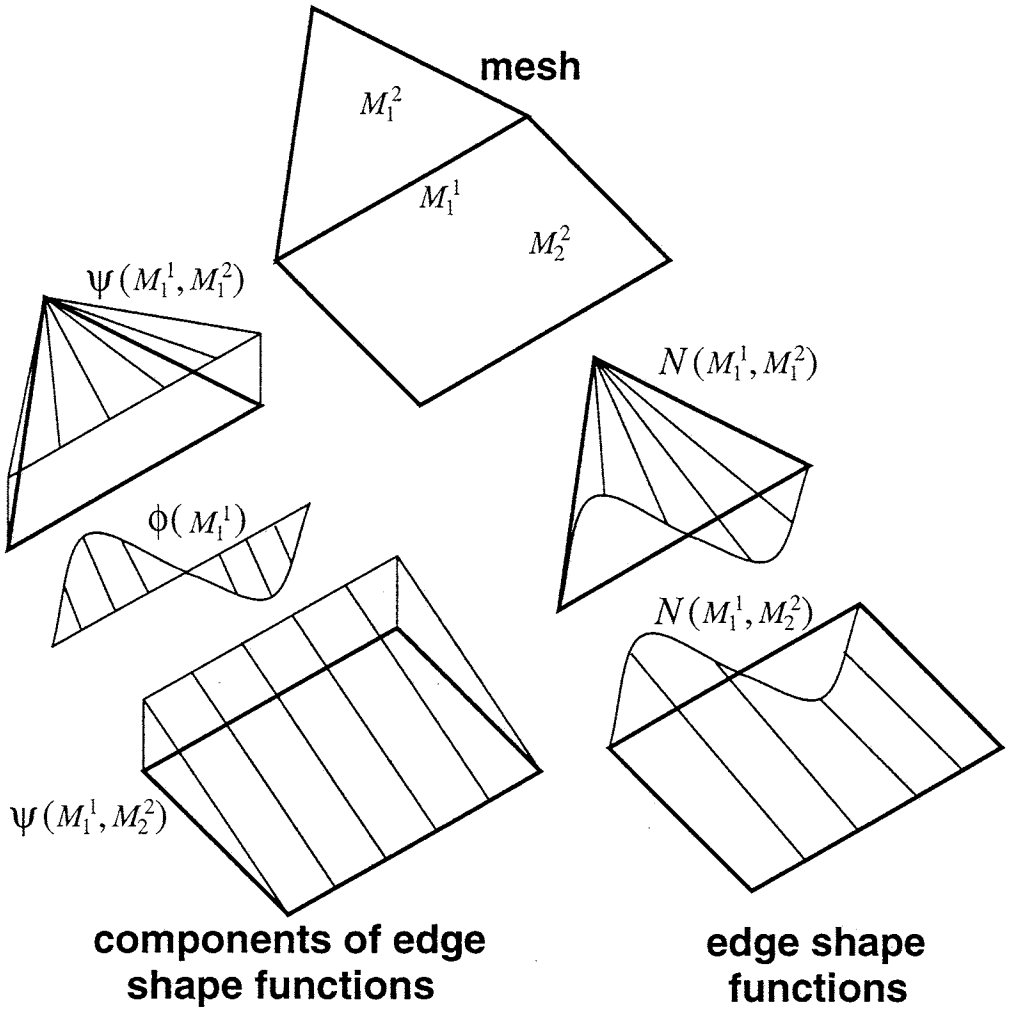

Figure 2. Construction of edge shape functions using a product of blending functions and an edge mode.

The first two requirements are easily met by the shape function decomposition based on topological entities. Although this decomposition is quite natural for some element topologies, specific care is required to ensure that it leads to efficient shape function evaluation over mixed meshes. The two most important issues here are the natural coordinate systems used over the various mesh entities and how the shape functions are decomposed.

Consideration of the available tools used to construct elements, and the importance of exercising specific options, makes it advantageous (but not necessary) to assume some specific limitations on the element topologies and polynomial orders specified on the mesh

entities. In what follows a mesh face, M_i^2, is assumed to be either triangular (bounded by three M_j^1), or quadrilateral (bounded by four M_j^1). The motivation for this limitation is the complexity of defining parametric coordinates over faces with more edges. The topology of the region elements considered will be limited to tetrahedra, hexahedra, wedges and pyramids for the same reason. Although it is possible to assign shape functions of any order to any topological entity, it is not desirable to associate higher-order polynomials with mesh vertices when only C^0 continuity is required. Therefore, the mesh vertices will be limited to have linear polynomial shape functions.

With these assumptions a mesh can have polynomials of any order associated with each mesh edge, face and region, and can have any order geometry assigned to it including that of the geometric model being analyzed. For example, Figure 3 shows a variable p-order mesh with boundary mesh entities assigned the shape of the CAD geometry and one curved interior edge.

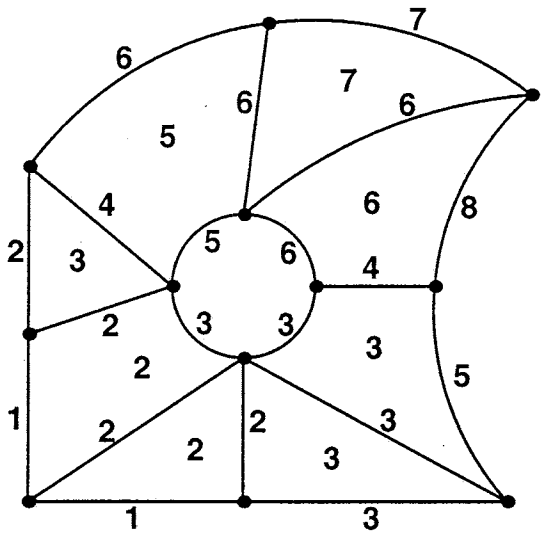

Figure 3. Example of a two-dimensional variable p-order mesh.

In addition, it is possible for the face and region entities to have different p-orders specified in each of the parametric directions of the face or region. Consideration of basic convergence rate expressions would indicate no obvious advantage of different polynomial orders in each direction. However, it is clear that in many problems the gradients are stronger in preferred directions. Therefore, computational efficiency can be improved by employing higher order polynomials in that direction. In general, to provide effective anisotropic p-refinement, the ability to control p-order in selected parametric directions

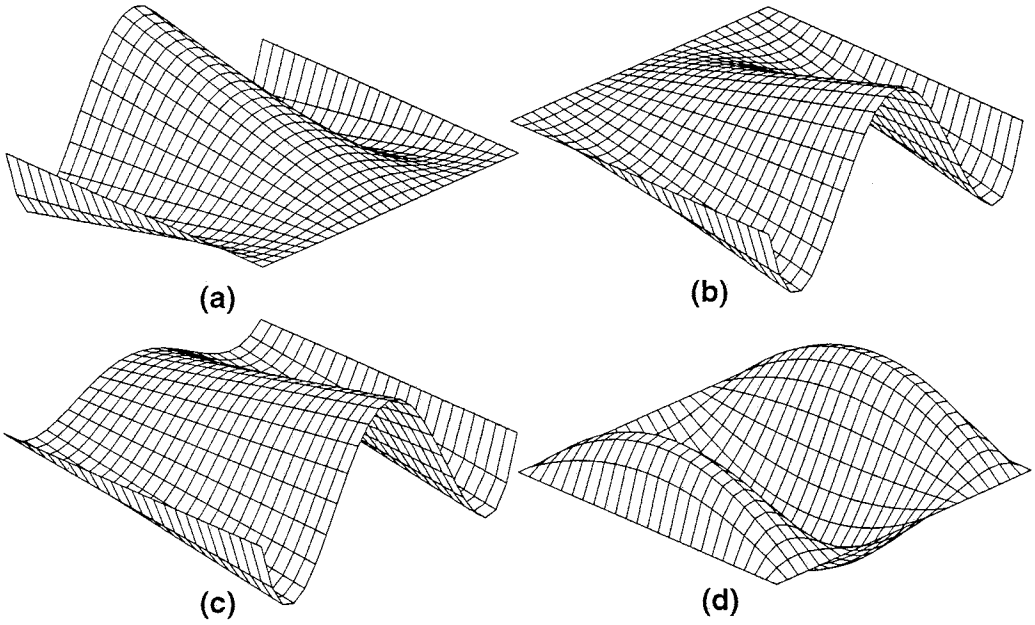

Figure 4. Directionally biased solution fields.

for faces and regions is required in addition to the ability to vary the polynomial order on the bounding mesh entities. As an example, consider a rectangular domain modeled by one quadrilateral element. A linear combination of two cubic edge modes (Figure 4(a,b)) is sufficient to produce a cubic variation on the element in the direction parallel to these edges (Figure 4(c)). However, obtaining a mode which has cubic variation in one parametric direction and a quadratic variation in the other direction, and which is zero on the bounding edges, requires a face mode which has a cubic and quadratic variation in the desired directions (Figure 4(d)). Section 5.2 presents simple geometric constructs to interpret and control the directional behavior of face and region shape functions over the face and region topologies considered here.

4. Parametric Coordinate Systems

Efficient evaluation of mesh-entity-based shape functions depends on the parametric coordinates used on each of the mesh entities on $\overline{\Omega}_e$. Since a mesh entity may bound portions of different elements with different topologies, care must be taken to use the same parametric coordinates for the entity to ensure the ability to decompose the shape functions as described in Section 3.

The primary complications relate to the selection of the parametric coordinate systems for simplices (triangles and tetrahedra) and pyramids, and the matching of those coor-

dinate systems with those of bounding lower order entities having different topologies. The choice of parametric coordinates for the mesh entities is based on the following two considerations:

1. Simplicity and efficiency of the blending functions, ψ.
2. Compatibility and efficiency of use with numerical integration scheme(s).

Parameterization of simplices by barycentric coordinates leads to symmetry of the shape functions with respect to the parametric coordinate components because bounding edges (faces) have equal length (area); simple polynomial forms for the blending functions; and compatibility with most existing integration schemes over triangles and tetrahedra which are defined in terms of area and volume coordinates, respectively [13, 14].

The parametric coordinates for various mesh entities used to obtain the shape function decomposition are shown in Figures 5 and 6. For reasons to be discussed, two edge parameterizations (*I* and *II*) are considered. The number in the circles indicates the identifiers of the bounding mesh vertices in the mesh database and are used in orienting the entity parametric coordinates. The domain of the parametric coordinate component(s) for each entity is listed in Table 1.

Since evaluation of shape functions (and their derivatives) at a point in the element parametric domain involves evaluating functions defined in terms of parametric coordinates of bounding mesh entities, a mapping transforming the element parametric coordinates, ξ_i, to the coordinates of the bounding lower order entities, $\hat{\xi}_j$, is required. The "^" is used to distinguish the parametric coordinates of the bounding lower order entities from the parametric coordinates of the element itself. When a mesh contains both simplex and non-simplex elements, the efficiency of mapping of element parameters, ξ_i, to the corresponding parameters on a bounding edge, $\hat{\xi}_j$, is dependent upon the choice of edge parameterization from Table 1.

Choosing edge parameterization *I* leads to a trivial mapping for quadrilateral and hexahedral elements, $\hat{\xi}_1 \equiv \xi_j$, based on the local edge index in the element topological hierarchy. However, for a simplex, the mapping for an edge between vertices with index i and j (Figures 5(b,d)) is nontrivial and is given by

$$\hat{\xi}_1 = \xi_j - \xi_i. \tag{6}$$

On the other hand, choice of edge parameterization *II* leads to trivial mapping for simplices given the local edge index, $\hat{\xi}_i \equiv \xi_j$. However, for a quadrilateral or a hexahedral

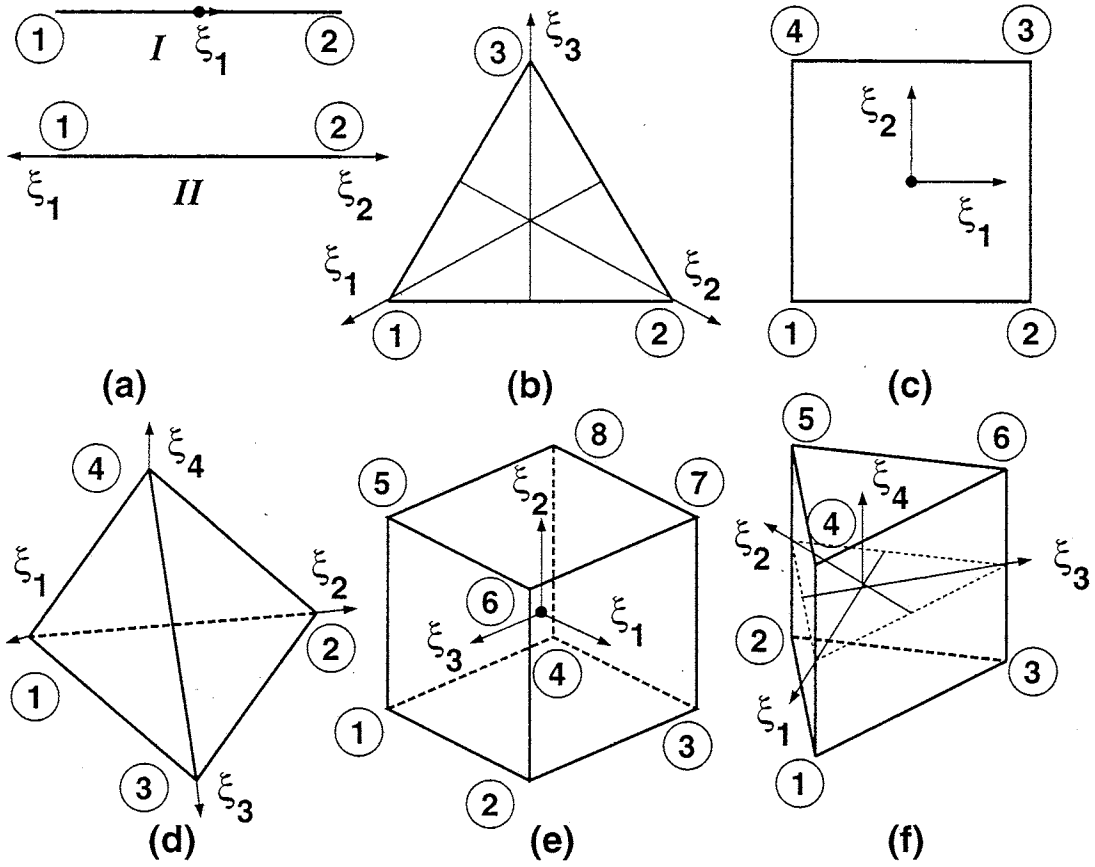

Figure 5. Entity parametric coordinates: (a) edge, (b) triangle, (c) quadrilateral, (d) tetrahedron, (e) hexahedron and (f) wedge form of a pentahedron.

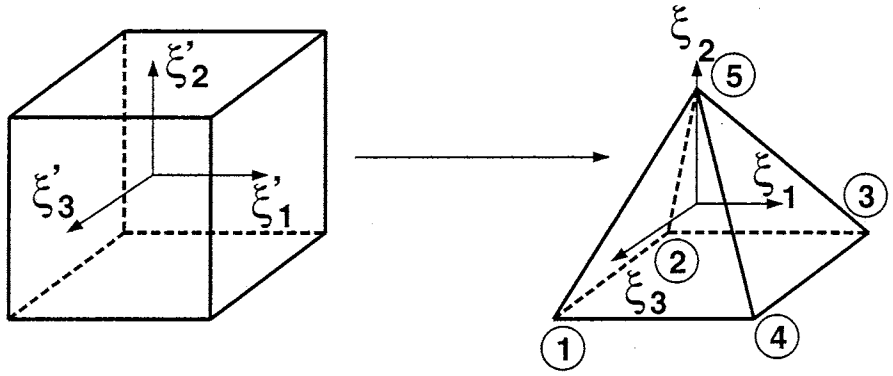

Figure 6. Parametric domain of a pyramid region obtained as a degeneration and scaling of the hexahedral parametric domain.

Table 1. Entity parametric domains.

Topology		Parametric Domain
Edge	*I*	$\xi_1 \in [-1,1]$
	II	$\xi_1, \xi_2 \in [0,1],\ \xi_1 + \xi_2 = 1$
Triangle		$\xi_1, \xi_2, \xi_3 \in [0,1],\ \xi_1 + \xi_2 + \xi_3 = 1$
Quadrilateral		$\xi_1, \xi_2 \in [-1,1]$
Tetrahedron		$\xi_1, \xi_2, \xi_3, \xi_4 \in [0,1],\ \xi_1 + \xi_2 + \xi_3 + \xi_4 = 1$
Hexahedron		$\xi_1, \xi_2, \xi_3 \in [-1,1]$
Pentahedron		$\xi_1, \xi_2, \xi_3 \in [0,1],\ \xi_1 + \xi_2 + \xi_3 = 1,\ \xi_4 \in [-1,1]$
Pyramid		$\xi_2 \in [-1,1],\ \frac{-(1-\xi_2)}{2} \leq \xi_1, \xi_3 \leq \frac{(1-\xi_2)}{2}$

element (Figure 7(a,b)) the mapping is now nontrivial and is given by

$$\begin{aligned} \hat{\xi}_1 &= \frac{1}{2}(1 - \xi_i) \\ \hat{\xi}_2 &= \frac{1}{2}(1 + \xi_i) \end{aligned} \tag{7}$$

for an edge directed along ξ_i parameter of the element. The important point here is that it is not possible to avoid a nontrivial mapping for both simplices and non-simplices when a mesh of mixed element topologies is used. However, for a mesh with all quadrilateral (hexahedral) elements, edge parameterization *I* is more efficient, while for a mesh with all simplex elements, edge parameterization *II* is more efficient.

For a quadrilateral face of a wedge pentahedron as shown in Figure 7(c), the mapping between the element parametric coordinates, ξ, to the face parametric coordinates, $\hat{\xi}$, is given by

$$\begin{aligned} \hat{\xi}_1 &= \xi_j - \xi_i \\ \hat{\xi}_2 &= \xi_4. \end{aligned} \tag{8}$$

A mesh region with the topology of a pyramid lacks a "natural" parameterization that fits its shape. It is parameterized as a degeneration of a hexahedron, with scaling as shown in Figure 6 using a procedure similar to that used in [26] to parameterize a tetrahedron. The mapping from the hexahedron parameters to the pyramid parameters is given by

$$\begin{aligned} \xi_1 &= \xi_1^{'}\left(1 - \xi_2^{'}\right)/2 \\ \xi_2 &= \xi_2^{'} \\ \xi_3 &= \xi_3^{'}\left(1 - \xi_2^{'}\right)/2 \end{aligned} \tag{9}$$

where $\xi_i^{'}$ define the hexadedral parametric space. The inverse mapping is given by

$$\begin{aligned} \xi_1^{'} &= \frac{2\xi_1}{(1-\xi_2)} \\ \xi_2^{'} &= \xi_2 \\ \xi_3^{'} &= \frac{2\xi_3}{(1-\xi_2)} \end{aligned} \quad , \tag{10}$$

The mapping is degenerate because one face in the parametric domain of the hexahedron maps to a point in the parametric domain of the pyramid. However, the singularity in the parametric domain does not imply that functions written in those coordinates are necessarily singular. The scaling of the parametric bounds for ξ_1 and ξ_3 by $\frac{(1-\xi_2)}{2}$ ensures that after the degeneration, the mapping between ξ_i and the Cartesian coordinates x_j remains one-to-one.

The mapping between the parametric coordinates of a pyramid and its bounding edges that also bound the quadrilateral face is given by $\hat{\xi}_1 \equiv \xi_i$ for edge directed along ξ_i with edge parameterization *I*. Using edge parameterization *II* leads to the mapping given by equation (7). For bounding edges that do not bound the quadrilateral face, the mapping is given by $\hat{\xi}_1 \equiv \xi_3$. The mapping for the parametric coordinates of a bounding triangular face is nontrivial. For example, for the face bounded by local vertices 1, 4 and 5 (Figure 6), the mapping is given by

$$\begin{aligned} \hat{\xi}_1 &= \frac{1}{4}(1-\xi_2-2\xi_1) \\ \hat{\xi}_2 &= \frac{1}{4}(1-\xi_2+2\xi_1) \\ \hat{\xi}_3 &= \frac{1}{2}(1+\xi_2) \end{aligned} \quad , \tag{11}$$

The mapping for the rest of the triangular faces can be obtained by permuting the parametric indices in equation (11) such that ξ_1 is replaced by the parameter along the edge of the triangle that is also shared by the quadrilateral face of the pyramid.

5. Decomposed Shape Functions

5.1 Construction of the Blending Functions ψ

Figure 2 depicted one form of blending function $\psi\left(M_j^{d_j}, M_e^{d_e}\right)$ in which the blending function is a constant on the mesh entity $M_j^{d_j}$ and blends to zero on the "opposite

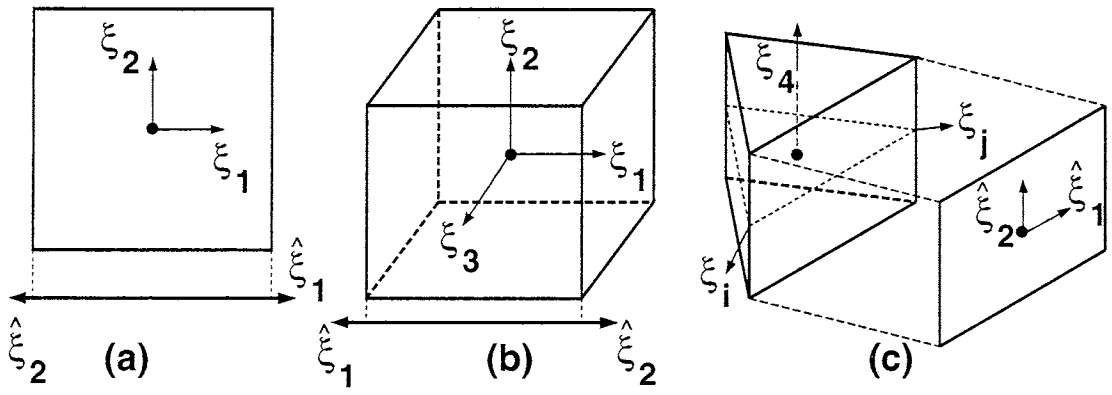

Figure 7. Parametric transformations with edge parameterization *II*.

boundary" of the element $M_e^{d_e}$. Although this form is an obvious choice for edges of a quadrilateral and faces of hexahedron, its construction on edges of triangular faces leads to unnecessary numerical operations since the edge blending function $\psi\left(M_j^1, M_e^{d_e}\right)$ in those cases must be written as a rational function.

Since the minimum polynomial order of any edge, face or region function is at least quadratic, it is not necessary that the decomposition of the shape functions be such that the blending function is a constant over the topological entity for which it blends. Considering again the simple case of an edge shape function for a two-dimensional element, it is possible to decompose each shape function into a product of a quadratic blending function and edge based functions which increase in polynomial order as the polynomial order of the element increases. In this case, the first edge function, $\phi(M_1^1)$, would be a constant, yielding the desired quadratic shape function, the second would be a linear, yielding the desired cubic shape function, etc. Figure 8 reconsiders the construction of cubic edge shape functions in the case where the blending function is quadratic. In this case a product of a linear edge function with the quadratic blends produces the desired cubic functions. The importance of employing the most effective blending functions can be demonstrated by the calculation of the shape function for the two cubic edge mode decompositions of Figures 2 and 8. The number of operations required for the two forms of decomposition for the quadrilateral element is the same. This is not the situation for the triangular element. In the case of producing a constant value over the edge, the blending term is a rational function, $\psi(M_1^1, M_1^2) = \frac{4\xi_i\xi_j}{1-(\xi_j-\xi_i)^2}$, and the edge function is based on the one-dimensional cubic shape function defined over

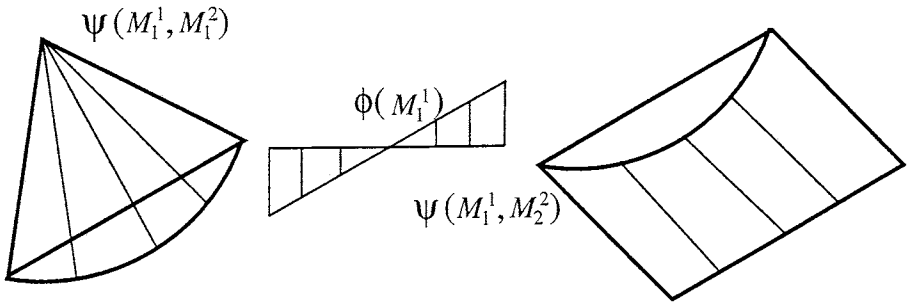

Figure 8. Alternative decomposition of a cubic edge shape function from Figure 2 using quadratic blending functions.

the edge [27], $\phi\left(M_1^1\right) = -\frac{\sqrt{10}}{4}\left[1 - (\xi_j - \xi_i)^2\right](\xi_j - \xi_i)$. The total cost[2] of evaluating the shape function in this case is 2 additions and 7 multiplications. On the other hand, the quadratic blending function is a simple polynomial, $\psi\left(M_1^1, M_1^2\right) = -2\xi_i\xi_j$, and the edge function, $\phi\left(M_1^1\right) = (\xi_j - \xi_i)$, is a linear polynomial giving a total cost of shape function evaluation as 1 addition and 3 multiplications which is the most efficient way to evaluate the cubic edge shape function $N\left(M_1^1, M_1^2\right) = -2\xi_i\xi_j(\xi_j - \xi_i)$. Clearly the decomposition based on the quadratic edge blend is more efficient. The rational blending term adds even more overhead in computing the derivatives of the edge shape functions required for stiffness computation.

The vertex shape functions typically used, and used here, always contribute only to the constant and the linear terms [6, 27, 28]. Since the vertex blending function, $\psi\left(M_j^0, M_i^{d_e}\right)$, $M_j^0 \in \overline{M_i^{d_e}}$, must at least be a linear function, it is the shape function and hence there is no need to perform an explicit product decomposition for vertex modes. Also, if the highest dimension of elements is three, then mesh entities M_i^3 are not shared by multiple element topologies; hence, there is no need to decompose the shape functions contributed by mesh regions.

The blending functions, $\psi\left(M_i^{d_e}, M_j^{d_j}\right)$, and the entity functions, $\phi\left(M_j^{d_j}\right)$, must be constructed such that the resulting shape functions, $N\left(M_j^{d_j}, M_i^{d_e}\right) = \psi\phi$, vanish over all lower order bounding entities except $M_j^{d_j}$ [6]. One possibility is to enforce this

---

[2] The cost is in terms of counting total number of addition(s) and multiplication(s) required. A division is deemed equivalent to a multiplication and a subtraction equivalent to an addition.

end-point property on the blending functions, in which case, the blending functions, $\psi\left(M_j^{d_j}, M_i^{d_e}\right)$, $M_j^{d_j} \in \overline{M_i^{d_e}}$, satisfy:

1. $\psi\left(M_j^{d_j}, M_i^{d_e}\right)$ is nonzero in $M_i^{d_e}$

$$\psi\left(M_j^{d_j}, M_i^{d_e}\right) \neq 0, \quad \forall \xi \in M_i^{d_e} \tag{12}$$

2. $\psi\left(M_j^{d_j}, M_i^{d_e}\right)$ is zero on $\partial\left(M_i^{d_e}\right)$ except on $M_j^{d_j}$

$$\psi\left(M_j^{d_j}, M_i^{d_e}\right) = 0, \quad \forall \xi \in \left[\partial\left(M_i^{d_e}\right) - M_j^{d_j}\right] \tag{13}$$

The higher order edge and face blending functions to follow lead to the most efficient evaluation of decomposed shape functions and satisfy these requirements. The expressions for blending functions are based on the topology of the element and the boundary mesh entity, and the parametric coordinate system of the element. The normalization coefficients used here are based on the orthogonalization described in [6] to improve numerical conditioning of resulting element level matrices.

1. Line Element:

   a. Edge blend: Use of parameterization *I* leads to the following expression for the blending function

$$\psi\left(M_i^1, M_i^1\right) = \frac{\left(\xi_1^2 - 1\right)}{2} \tag{14}$$

   whereas use of parameterization *II* results in an equivalent blend given by

$$\psi\left(M_i^1, M_i^1\right) = -2\xi_1\xi_2. \tag{15}$$

2. Triangle Element:

   a. Edge blend:

$$\psi\left(M_i^1, M_j^2\right) = -2\xi_k\xi_l \tag{16}$$

   where edge M_i^1 is directed from the k-th to the l-th vertex of M_j^2. For example, for the edge between vertices 1 and 2 in Figure 5(b), k=1 and l=2.

   b. Face blend:

$$\psi\left(M_i^2, M_i^2\right) = \xi_1\xi_2\xi_3. \tag{17}$$

3. Quadrilateral Element:

   a. Edge blend:

$$\psi(M_j^1, M_i^2) = \frac{1}{4}[\xi_k^2 - 1](1 \pm \xi_l) \tag{18}$$

   for edges along ξ_k with $k = 1, 2;\ l = 2, 1$. For example, for the edge between vertices 1 and 2 in Figure 5(c), k=1 and l=2.

   b. Face blend:

$$\psi(M_i^2, M_i^2) = \frac{1}{4}[\xi_1^2 - 1][\xi_2^2 - 1]. \tag{19}$$

4. Tetrahedral Element:

   a. Edge blend:

$$\psi(M_i^1, M_j^3) = -2\xi_k\xi_l \tag{20}$$

   where edge M_i^1 is directed from the k-th to the l-th vertex of M_j^3. For example, for the edge between vertices 1 and 2 in Figure 5(d), k=1 and l=2.

   b. Face blend:

$$\psi(M_i^2, M_j^3) = \xi_k\xi_l\xi_m \tag{21}$$

   where face M_i^2 is bounded by vertices of M_j^3 with local index k, l and m. For example, for the face bounded by vertices 1, 2 and 3 in Figure 5(d), k=1, l=2 and m=3.

5. Hexahedral Element:

   a. Edge blend:

$$\psi(M_i^1, M_j^3) = \frac{1}{8}(\xi_k^2 - 1)(1 \pm \xi_l)(1 \pm \xi_m) \tag{22}$$

   for edges along ξ_k with $k = 1, 2, 3;\ l = 2, 3, 1;\ m = 3, 1, 2$. For example, for the edge between vertices 1 and 2 in Figure 5(e), k=1, l=2 and m=3.

   b. Face blend:

$$\psi(M_i^2, M_j^3) = \frac{1}{8}(\xi_l^2 - 1)(\xi_m^2 - 1)(1 \pm \xi_k) \tag{23}$$

   for faces perpendicular to ξ_k with $k = 1, 2, 3;\ l = 2, 3, 1;\ m = 3, 1, 2$. For example, for the face bounded by vertices 1, 2, 3 and 4 in Figure 5(e), k=2, l=1 and m=3.

6. Wedge Element:

   a. Edge blend:

$$\psi(M_i^1, M_j^3) = -\xi_k \xi_l (1 \pm \xi_4) \tag{24}$$

   for edges lying in the plane perpendicular to the ξ_4 axis and defined between vertices at which $\xi_k = 1$ and $\xi_l = 1$, respectively. For example, for the edge between vertices 1 and 2 in Figure 5(f), k=1 and l=2. For edges parallel to the ξ_4 axis

$$\psi(M_i^1, M_j^3) = \frac{1}{2}\xi_k(\xi_4^2 - 1). \tag{25}$$

   For example, for the edge between vertices 1 and 4 in Figure 5(f), k=1.

   b. Face blend:

$$\psi(M_i^2, M_j^3) = \frac{1}{2}\xi_1 \xi_2 \xi_3 (1 \pm \xi_4) \tag{26}$$

   for the two triangular faces in Figure 5(f).

$$\psi(M_i^2, M_j^3) = -\xi_k \xi_l (\xi_4^2 - 1) \tag{27}$$

   for quadrilateral faces bounded by pairs of vertices at which $\xi_k = 1$ and $\xi_l = 1$. For example, for the face bounded by vertices 1, 3, 6, and 4, k=1 and l=3.

7. Pyramid element:

   a. Edge blend: For edges along ξ_k the blend is

$$\psi(M_i^1, M_j^3) = \frac{1}{8}\left(\left(\frac{2\xi_k}{1-\xi_2}\right)^2 - 1\right)\left(1 \pm \frac{2\xi_l}{1-\xi_2}\right)(1-\xi_2) \tag{28}$$

   with k=1,3 and l=3,1. For example, for the edge between vertices 1 and 4 in Figure 6, k=1 and l=3. For edges along ξ_2 the blend is

$$\psi(M_i^1, M_j^3) = \frac{1}{8}(\xi_2^2 - 1)\left(1 \pm \frac{2\xi_k}{1-\xi_2}\right)\left(1 \pm \frac{2\xi_l}{1-\xi_2}\right) \tag{29}$$

   with k=1,3 and l=3,1. These are the edges bounded by vertex 5 in Figure 6.

   b. Face blend:

$$\psi(M_i^2, M_j^3) = \frac{1}{8}\left(1 - \left(\frac{2\xi_1}{1-\xi_2}\right)^2\right)\left(1 - \left(\frac{2\xi_3}{1-\xi_2}\right)^2\right)(1-\xi_2) \tag{30}$$

   for the quadrilateral face in Figure 6 and

$$\psi(M_i^2, M_j^3) = \frac{1}{8}\left(1 \pm \frac{2\xi_k}{(1-\xi_2)}\right)\left(1 - \left(\frac{2\xi_l}{1-\xi_2}\right)^2\right)(1-\xi_2^2) \tag{31}$$

   for the triangular faces with k=1,3 and l=3,1.

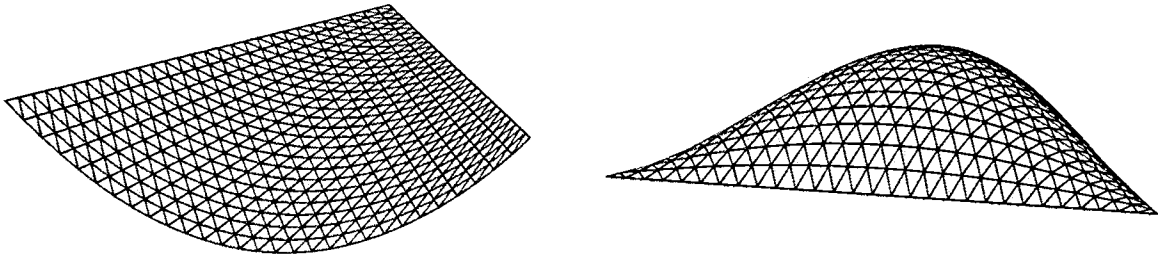

Figure 9. Triangular edge and face blending functions.

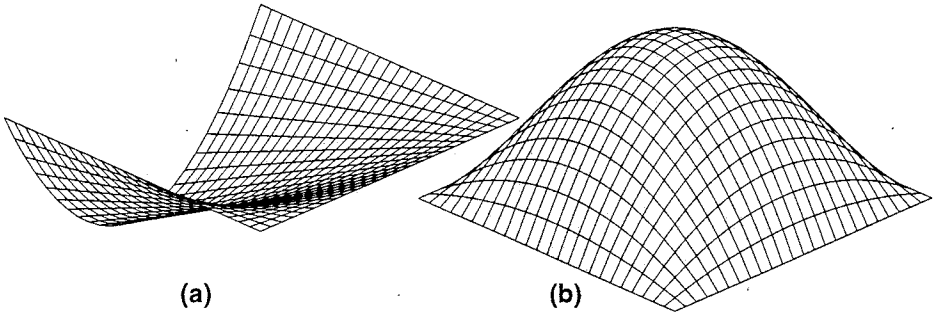

Figure 10. Quadrilateral edge and face blending functions.

The edge and face blends for triangular and quadrilateral elements are shown in Figures 9 and 10.

5.2 Construction of the Mesh Entity Level Functions ϕ

The polynomial order of ϕ for an entity is given by $(p-q)$ where p is the polynomial order of the shape function and q is the polynomial order of the blending function used for that mesh entity. For example, $\phi(M_i^1)$ are of the order $(p-2)$ since $\psi\left(M_i^1, M_j^{d_e}\right)$ is quadratic. The definition of ϕ can be based on any hierarchical polynomial basis. However, practical choices are governed by the numerical conditioning of the resulting element level matrices [2, 6, 27, 28].

Mesh Edge

There is only one edge shape function of a given polynomial order p. The exact expression for the edge functions depends on the choice of the basis and the parameterization used for mesh edges. One possibility is to use the basis defined by the integration

Table 2. Edge functions.

p	ϕ	
	Equation (32)	Equation (33)
2	$\sqrt{\frac{3}{2}}$	1
3	$\sqrt{\frac{5}{2}}(\xi_2 - \xi_1)$	$\xi_2 - \xi_1$
4	$\sqrt{\frac{7}{2}}\left(\frac{5(\xi_2-\xi_2)^2-1}{4}\right)$	$\xi_1^2 - \xi_1\xi_2 + \xi_2^2$
5	$\sqrt{\frac{9}{2}}\left[\left(\frac{\xi_2-\xi_1}{4}\right)\left(7(\xi_2 - \xi_1)^2 - 3\right)\right]$	$\xi_2^2 - 6\xi_1\xi_2^2 + 6\xi_1^2\xi_2 - \xi_1^2$

of the *Legendre* polynomials [1], $P_n(\zeta)$, $\zeta \in [-1, 1]$, as follows

$$\left[\frac{\zeta^2 - 1}{2}\right]\phi(M_1^1) = \sqrt{\frac{2p-1}{2}}\int_{-1}^{\zeta} P_{p-1}(t)dt, \; p \geq 2 \tag{32}$$

where, $\zeta = \xi_1$ and $\zeta = \xi_2 - \xi_1$ for edge parameterization *I* and *II*, respectively. The term $\frac{\zeta^2-1}{2}$ is the edge blend function and can always be factored from the right side of equation (32).

However, when simplices are used, then the following edge functions with parameterization *II* yields a better conditioning of element level matrices [6]

$$\phi(M_1^1) = \sum_{k=0}^{p-2}(-1)^k \frac{1}{k+1}\binom{p-2}{k}\binom{p-1}{k}\xi_1^k\xi_2^{p-2-k}, \; p \geq 2. \tag{33}$$

The expressions for $\phi_{p-2}(M_1^1)$ using edge parameterization *II* for *p=2,3,4,5* based on equations (32) and (33) are given in Table 2. Graphs of these edge functions are given in Figure 11. The shape functions for a triangular and a quadrilateral element obtained from equation (33) are shown in Figures 12 and 13, respectively.

Triangular Mesh Face

There are a total of $(p-2)$ independent face functions for a triangular face that contribute to shape functions of order *p, p≥3*. For shape functions based on *Legendre* polynomials [27], one possible set of triangular face functions is given by

$$\phi(M_i^2) = P_{\alpha_1-1}(\xi_2 - \xi_1)P_{\alpha_2-1}(2\xi_3 - 1) \tag{34}$$

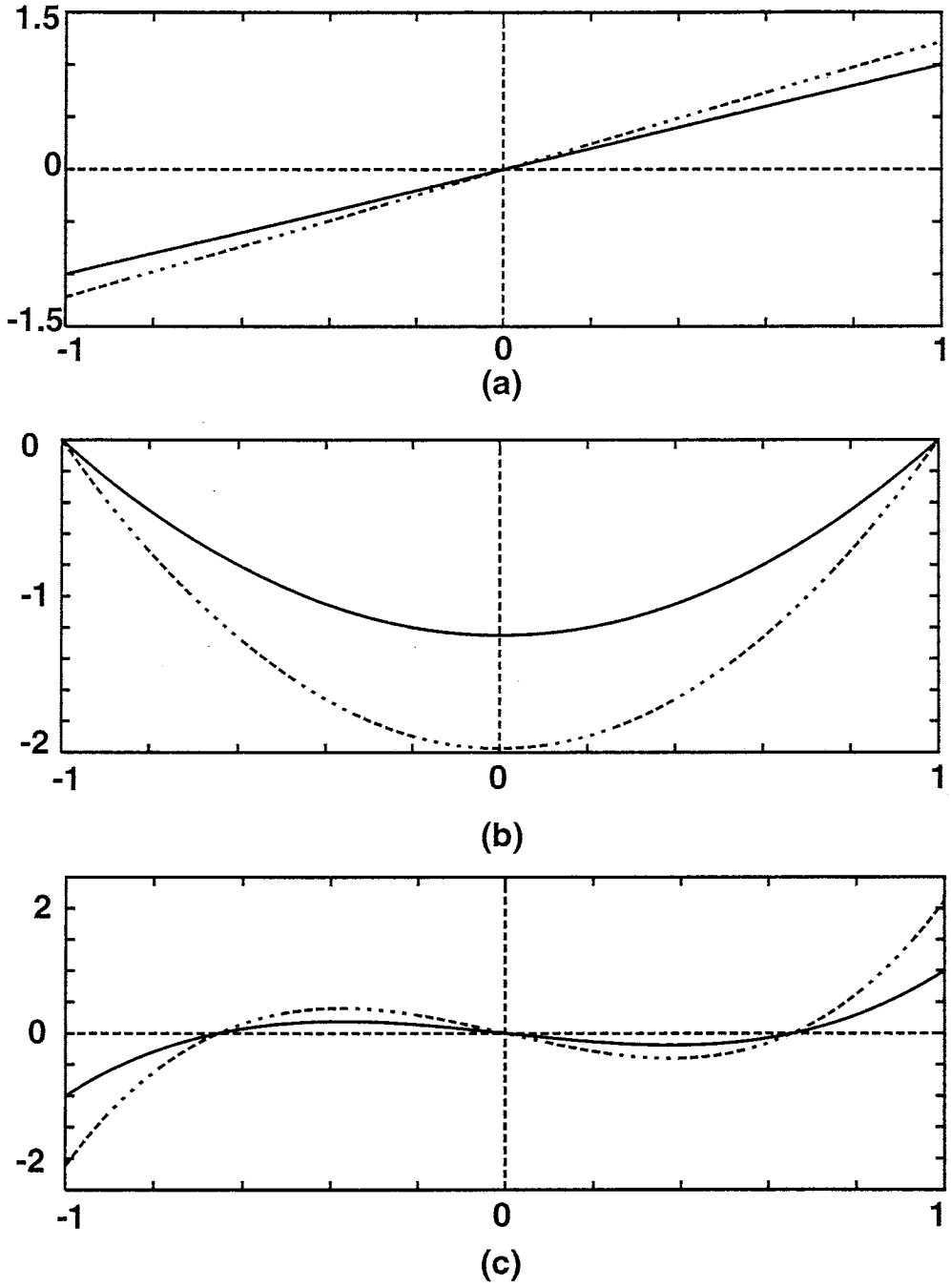

Figure 11. Edge functions $\phi(M_1^1)$ for (a) *p=3*, (b) *p=4* and (c) *p=5*. Solid lines correspond to equation (32) and dashed lines to equation (33).

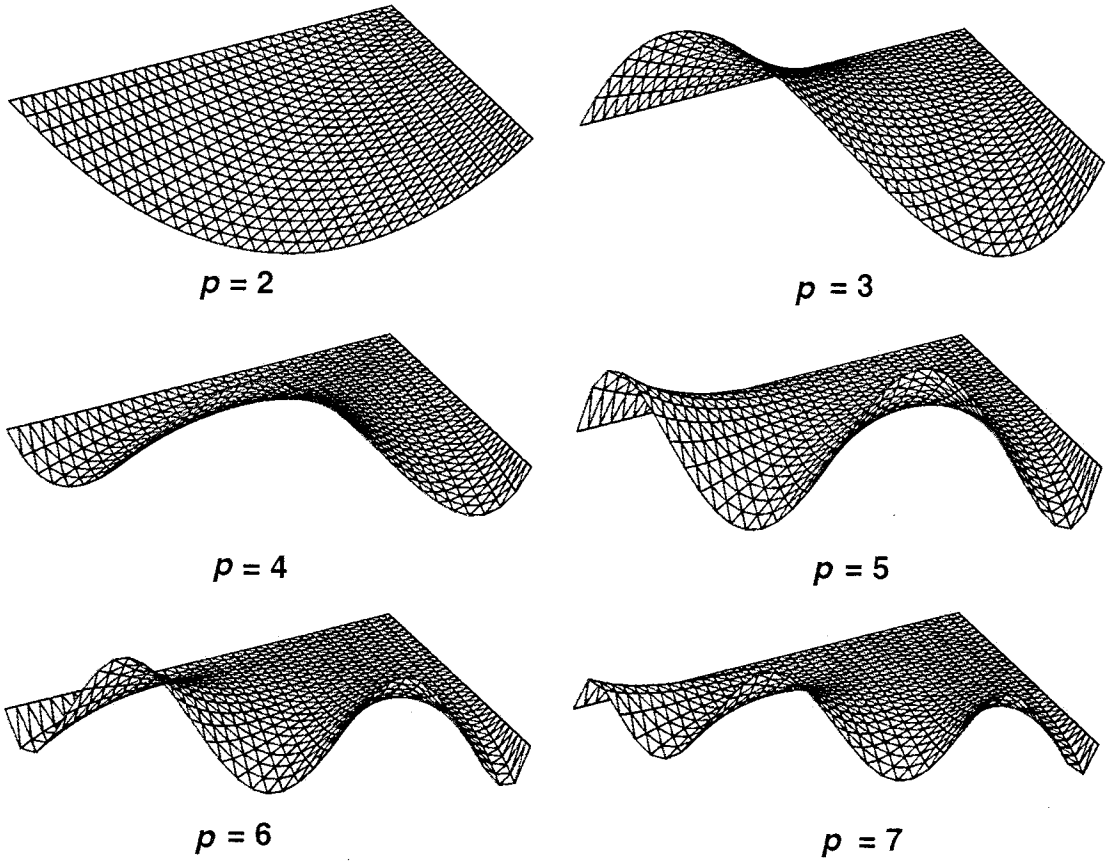

Figure 12. Triangular edge shape functions. The shape function for p=2 is scaled by 0.5 and those for p=5,6,7 are scaled by 2.0 for clarity.

with

$$\begin{aligned} &\alpha_1, \alpha_2 = 1, .., p-2 \\ &\alpha_1 + \alpha_2 + \alpha_3 = p. \end{aligned} \tag{35}$$

An alternative form which yields better conditioned element matrices [6] is given by

$$\begin{aligned} \phi(M_1^2) = &\sum_{i=0}^{\alpha_2-1} \sum_{j=0}^{\alpha_1-1} \left(\frac{-1}{2}\right)^{i+j} i!j!(i+j)! \binom{\alpha_1-1}{j} \binom{\alpha_1}{j} \binom{\alpha_2-1}{i} \binom{\alpha_2}{i} \\ &\times \frac{1}{\prod_{k=1}^{i+j} [k(\alpha_1+\alpha_2) - k(k-1)/2]} (\xi_1)^{\alpha_1-1-j} (\xi_2)^{\alpha_2-1-i} \end{aligned} \tag{36}$$

where $\alpha_i,\ i = 1,2,3$, satisfy equation (35).

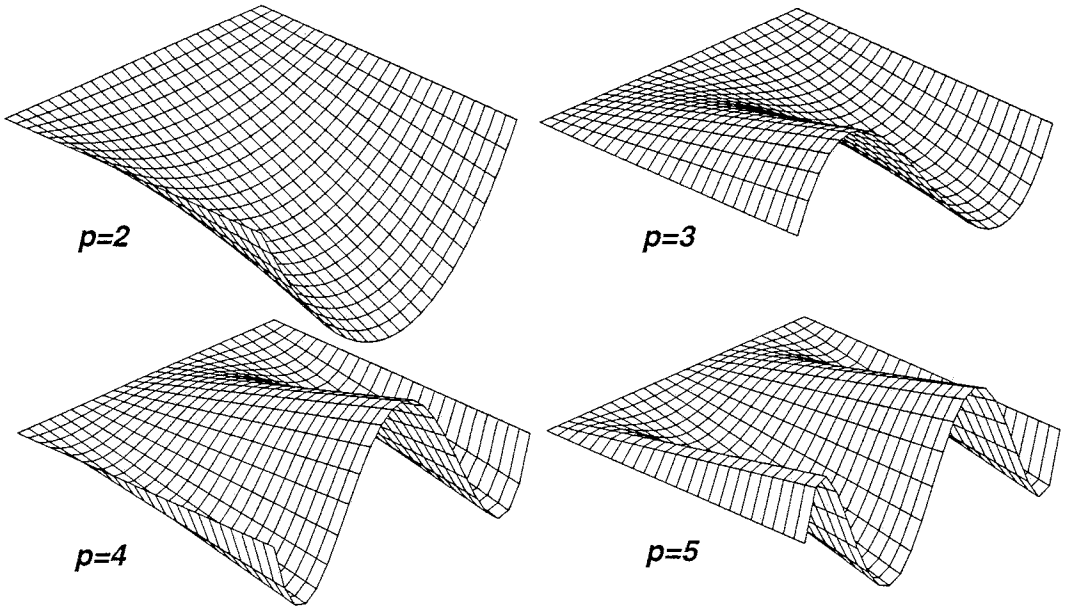

Figure 13. Quadrilateral edge shape functions.

The symmetry and directional bias of the triangular face modes with respect to the barycentric coordinates can be explained by associating the triplet, $(\alpha_1, \alpha_2, \alpha_3)$, with each face function where α_i defines the highest power of ξ_i in the resulting face shape function. The sum of the triplet indices equals the polynomial degree of the shape function. Since only two of the barycentric coordinate components are independent, completeness of the basis can be ensured by varying two of the indices independently while holding the third one fixed. For example, for $\alpha_i = 1,\ i = 1, 2, 3$, the sets of triplets that define the face function combinations based on the three possible choices of the pair of independent indices can be generated using the geometric construct shown in Figure 14. Triplets in each of the triangles ABO, ACO and BCO define the set of linearly independent face functions needed to complete the basis of a given order. Combinations in triangle ABO, BCO and ACO correspond to choosing (ξ_1, ξ_2), (ξ_2, ξ_3) and (ξ_1, ξ_3) as the independent pair of parameter components.

Face functions generated by equation (36) correspond to the triplets in triangle ABO and are listed in Table 3. The shape functions resulting from the face functions from triangle ABO are plotted in Figure 15. Note that the missing symmetric counterpart of (2,1,1) and (1,2,1) for *p*=4, given by (1,1,2), can be obtained as a linear combination of

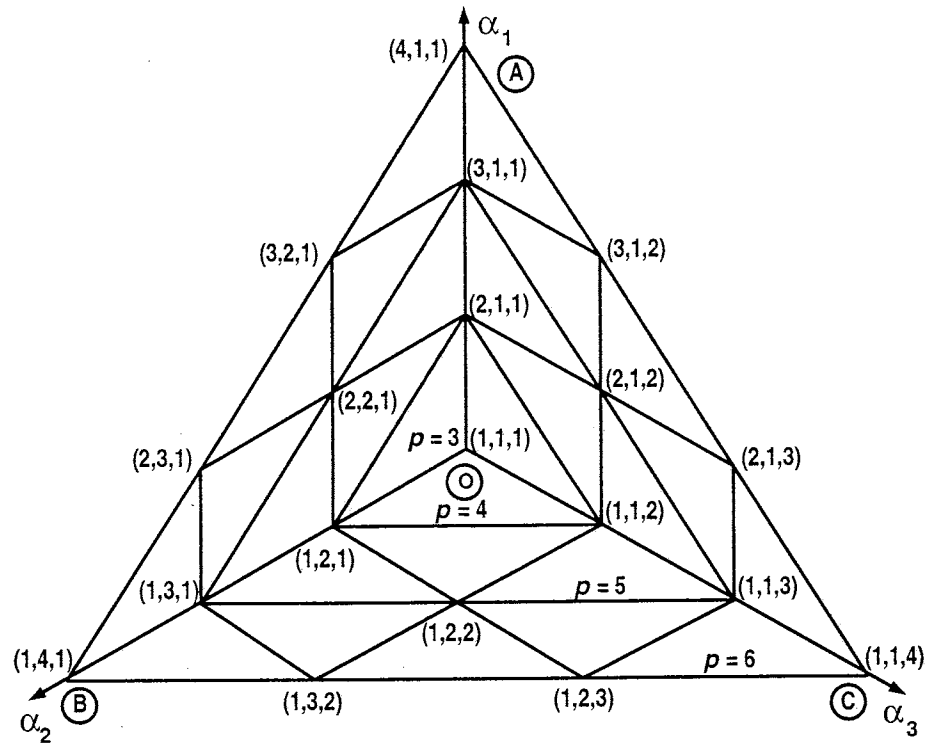

Figure 14. Face function triplet combinations.

(2,1,1) and (1,2,1) modes as follows

$$\left(\xi_3 - \frac{1}{3}\right) = \left(\frac{2}{3} - \xi_1 - \xi_2\right) = -\left[\left(\xi_1 - \frac{1}{3}\right) + \left(\xi_2 - \frac{1}{3}\right)\right] = -[(2,1,1) + (1,2,1)]. \tag{37}$$

In general, the set of face functions defined by any of the triangles ABO, ACO and BCO can be expressed as a linear combination of the set defined by any one of the other triangles. However, in those cases where the error analysis indicates the need for anisotropic p-enrichment by picking only the face shape function defined by (1,1,2), its construction by the linear combination of the (1,2,1) and (2,1,1) functions is not the most effective means to construct it; instead it can be directly obtained by choosing ξ_3 as one of the independent parameters. This is equivalent to switching from triangle ABO to triangle BCO or to triangle ACO.

Quadrilateral Mesh Face

For a quadrilateral face, there are a total of $(p-3)$ face functions that contribute to shape functions of polynomial order p. The face functions are obtained as tensor

Table 3. Triangular face functions.

p	$(\alpha_1, \alpha_2, \alpha_3)$	ϕ	
		Equation (34)	Equation (36)
3	1,1,1	1	1
4	2,1,1	$\xi_2 - \xi_1$	$\xi_1 - \frac{1}{3}$
	1,2,1	$2\xi_3 - 1$	$\xi_2 - \frac{1}{3}$
5	3,1,1	$\frac{1}{2}\left[3(\xi_2 - \xi_2)^2 - 1\right]$	$\xi_1^2 - \frac{3}{4}\xi_1 + \frac{3}{28}$
	2,2,1	$(\xi_2 - \xi_1)(2\xi_3 - 1)$	$\xi_1\xi_2 - \frac{1}{4}(\xi_1 + \xi_2) + \frac{1}{14}$
	1,3,1	$\frac{1}{2}\left[3(2\xi_3 - 1)^2 - 1\right]$	$\xi_2^2 - \frac{3}{4}\xi_2 + \frac{3}{28}$

product polynomials of one-dimensional edge functions [27, 28]. The face functions are derived from

$$\frac{1}{4}\left[\xi_1^2 - 1\right]\left[\xi_2^2 - 1\right]\phi(M_i^2) = \sqrt{\frac{2\alpha_1 - 1}{2}}\int_{-1}^{\xi_1} P_{\alpha_1 - 1}(t)dt\sqrt{\frac{2\alpha_2 - 1}{2}}\int_{-1}^{\xi_2} P_{\alpha_2 - 1}(t)dt \quad (38)$$

with

$$\begin{aligned} &\alpha_1, \alpha_2 \geq 2 \\ &\alpha_1 + \alpha_2 = p \end{aligned} \quad (39)$$

and $p \geq 4$. The term $\frac{1}{4}\left(\xi_1^2 - 1\right)\left(\xi_2^2 - 1\right)$ is the quadrilateral face blend and can always be factored from the right hand side of equation (38). The pair (α_1, α_2) can be used to define the directional behavior of the quadrilateral face functions where α_i is the power of ξ_i in the resulting shape function. The geometrical construct used to generate the possible face shape functions in this case is analogous to the interior of Pascal's triangle [13, 30] and is shown in Figure 16. The expressions for ϕ for *p=4,5,6* are given in Table 4. The corresponding face shape functions for a quadrilateral element are shown in Figure 17.

Tetrahedral Mesh Region

For a tetrahedral region, there are a total of $\frac{(p-2)(p-3)}{2}$ region functions that contribute to shape functions of polynomial order *p, p≥4*. Recall that for efficiency reasons $\phi(M_i^3)$ is

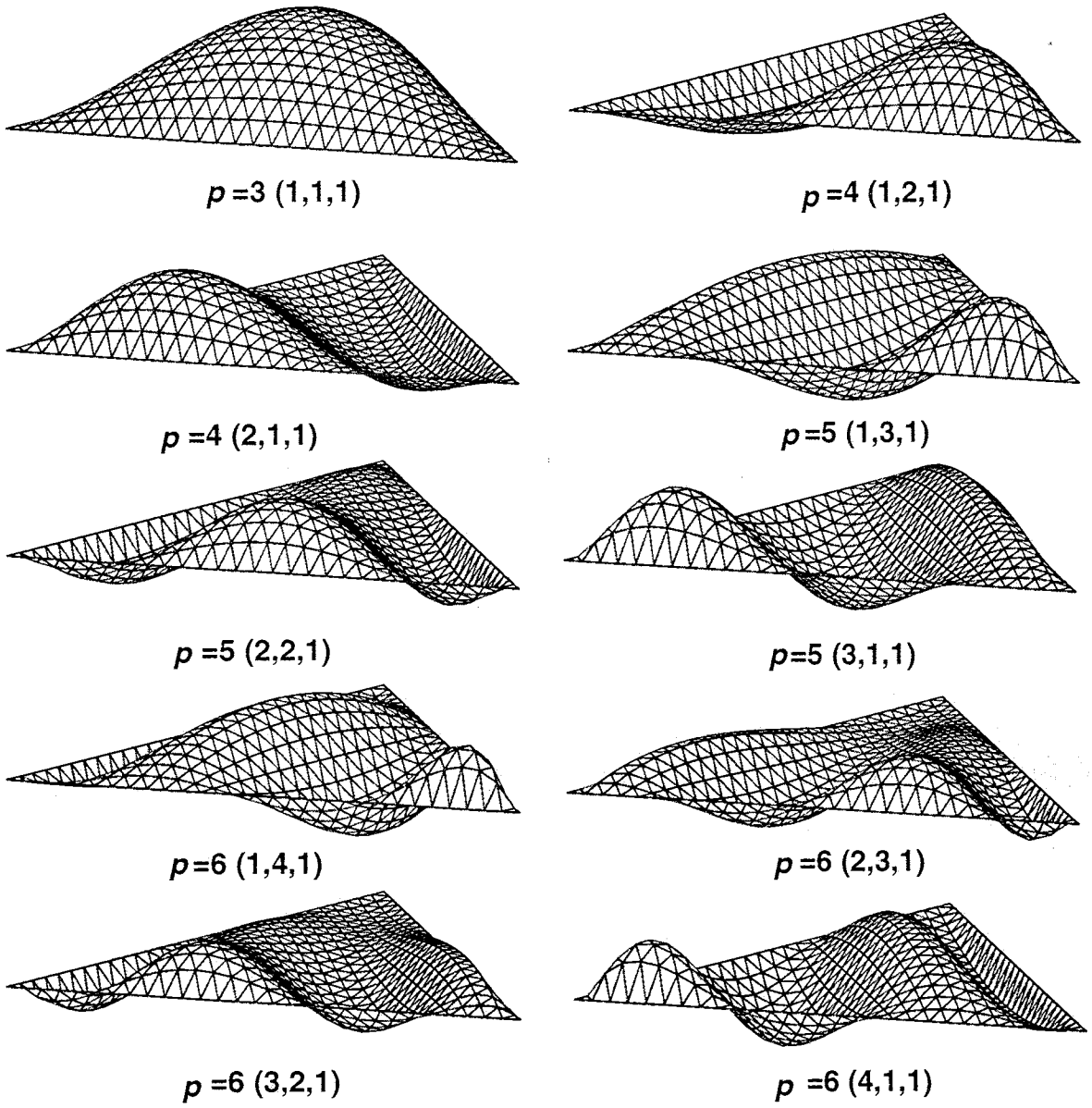

Figure 15. Triangular face shape functions. For clarity the shapes have been scaled by 5, 25, 100 and 200 for p=3, 4, 5 and 6 respectively.

the shape function. One possible set of region functions based on *Legendre* polynomials are given by [28]

$$\phi(M_i^3) = \xi_1\xi_2\xi_3\xi_4 P_{\alpha_1-1}(\xi_2-\xi_1)P_{\alpha_2-1}(2\xi_3-1)P_{\alpha_3-1}(2\xi_4-1) \qquad (40)$$

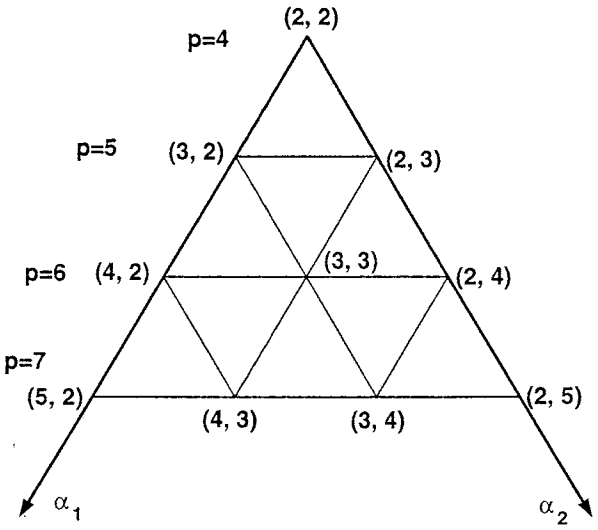

Figure 16. Construct to generate quadrilateral face modes.

Table 4. Quadrilateral face functions.

p	(α_1, α_2)	ϕ
4	2, 2	1
5	2, 3	$\frac{\sqrt{12}}{4}\xi_2$
	3, 2	$\frac{\sqrt{12}}{4}\xi_1$
6	2, 4	$\frac{\sqrt{5}}{8}\left[5\xi_2^2 - 1\right]$
	3, 3	$\frac{3}{2}\xi_1\xi_2$
	4, 2	$\frac{\sqrt{5}}{8}\left[5\xi_1^2 - 1\right]$

with

$$\begin{aligned} &\alpha_1, \alpha_2, \alpha_3 = 1, .., p-3 \\ &\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = p \end{aligned} \tag{41}$$

An alternate basis that results in better conditioning of element level matrices [6] is

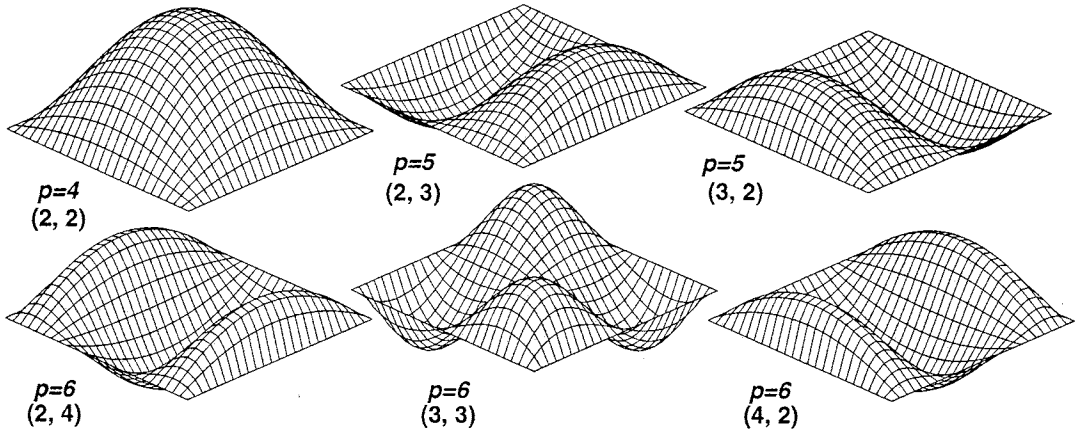

Figure 17. Quadrilateral face shape functions.

given by

$$\phi(M_i^3) = \xi_1\xi_2\xi_3\xi_4(A \times B \times C)$$

$$A = \sum_{i=0}^{\alpha_1-1} (-1)^i i! \binom{\alpha_1 - 1}{i} \binom{\alpha_1}{i} \frac{(2m+5-i)!}{(2m+5)} (\xi_1)^{\alpha_1-1-i}$$

$$B = \sum_{i=0}^{\alpha_2-1} i! \binom{\alpha_2 - 1}{i} \binom{\alpha_2}{i} \frac{(2n+3-i)!}{(2n+3)} (\xi_2)^{\alpha_2-1-i} (\xi_1 - 1)^i \tag{42}$$

$$C = \sum_{i=0}^{\alpha_3-1} i! \binom{\alpha_3 - 1}{i} \binom{\alpha_3}{i} \frac{(2\alpha_3 - i)!}{2\alpha_3} (\xi_3)^{\alpha_3-1-i} (\xi_1 + \xi_2 - 1)^i$$

with α_i satisfying equation (41) and where $m = \alpha_1 + \alpha_2 + \alpha_3 - 3$, $n = \alpha_2 + \alpha_3 - 2$. Analogous to the triangular face functions, the directional behavior of the tetrahedral region functions can be interpreted by associating a quadruplet, $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$, with each region function where, α_i represents the power of ξ_i in the resulting shape function. Since only three of the four barycentric coordinates can be varied independently, the possible region functions are defined by the variation of any three of α_i based on equation (41). Equation (42) uses ξ_1, ξ_2, ξ_3 as the independent coordinates with $\alpha_4 = 1$. The corresponding geometric construct to generate the possible region functions is a set of four tetrahedra each of which allow for the independent variation of three of the four indices in the quadruplet. Figure 18 shows one of the tetrahedra that varies $\alpha_1, \alpha_2, \alpha_3$. A similar construction can be used to define the three tetrahedra that define the variation of $\alpha_1, \alpha_2, \alpha_4$ or $\alpha_1, \alpha_3, \alpha_4$ or $\alpha_2, \alpha_3, \alpha_4$. The expressions for the tetrahedral region functions for *p=4,5,6* based on equations (40) and (42) are listed in Table 5. Similar expressions are also given in [6, 28].

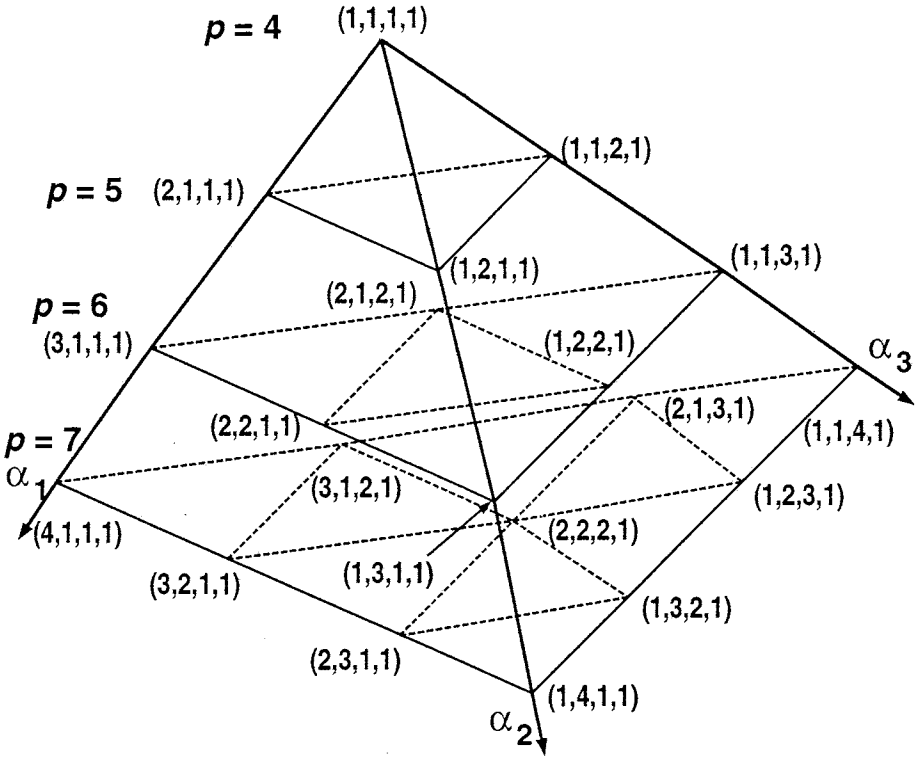

Figure 18. Geometric construct to generate tetrahedral region functions.

Hexahedral Mesh Region

For a hexahedral region, there are a total of $\frac{(p-4)(p-5)}{2}$ shape functions of polynomial order p, $p \geq 6$. The entity function for the regions are derived based on the tensor product polynomial basis using the *Legendre* polynomials [27] and are given by

$$\begin{aligned}
\phi(M_i^3) &= A(\xi_1) \times B(\xi_2) \times C(\xi_3) \\
A(\xi_1) &= \sqrt{\frac{2\alpha_1 - 1}{2}} \int_{-1}^{\xi_1} P_{\alpha_1 - 1}(t)dt \\
B(\xi_2) &= \sqrt{\frac{2\alpha_2 - 1}{2}} \int_{-1}^{\xi_2} P_{\alpha_2 - 1}(t)dt \\
C(\xi_3) &= \sqrt{\frac{2\alpha_3 - 1}{2}} \int_{-1}^{\xi_2} P_{\alpha_3 - 1}(t)dt
\end{aligned} \tag{43}$$

Table 5. Tetrahedral region functions.

p	$(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$	ϕ	
		Equation (40)	Equation (42)
4	(1,1,1,1)	1	1
5	(2,1,1,1)	$\xi_2 - \xi_1$	$\xi_1 - \frac{2}{7}$
	(1,2,1,1)	$2\xi_3 - 1$	$\xi_2 + \frac{2}{5}(\xi_1 - 1)$
	(1,1,2,1)	$1 - 2(\xi_1 + \xi_2 + \xi_3)$	$\xi_3 + \frac{2}{3}(\xi_1 + \xi_2 - 1)$
6	(3,1,1,1)	$\frac{3}{2}(\xi_2 - \xi_1)^2 - \frac{1}{2}$	$\xi_1^2 - \frac{2}{3}\xi_1 + \frac{1}{12}$
	(2,2,1,1)	$(\xi_2 - \xi_1)(2\xi_3 - 1)$	$\frac{-1}{45}(9\xi_1 - 2)(2\xi_1 + 5\xi_2 - 2)$
	(2,1,2,1)	$(\xi_2 - \xi_1)(1 - 2(\xi_1 + \xi_2 + \xi_3))$	$\frac{-1}{27}(9\xi_1 - 2)(2(\xi_1 + \xi_2 - 1) + 3\xi_3)$
	(1,3,1,1)	$\frac{3}{2}(2\xi_3 - 1)^2 - \frac{1}{2}$	$\xi_2^2 + \frac{6}{7}\xi_2(\xi_1 - 1) + \frac{1}{7}(\xi_1 - 1)^2$
	(1,2,2,1)	$(2\xi_3 - 1)(1 - 2(\xi_1 + \xi_2 + \xi_3))$	$\frac{-1}{27}(2\xi_1 + 7\xi_2 - 2)(2(\xi_1 + \xi_2 - 1) + 3\xi_3)$
	(1,1,3,1)	$\frac{3}{2}(1 - 2(\xi_1 + \xi_2 + \xi_3))^2 - \frac{1}{2}$	$\xi_3^2 + \frac{6}{5}\xi_3(\xi_1 + \xi_2 - 1) + \frac{3}{10}(\xi_1 + \xi_2 - 1)^2$

with

$$\begin{aligned} &\alpha_1, \alpha_2, \alpha_3 \geq 2 \\ &\alpha_1 + \alpha_2 + \alpha_3 = p. \end{aligned} \tag{44}$$

Analogous to quadrilateral face functions, the directional behavior of the hexahedral region functions can be understood by associating a triplet, $(\alpha_1, \alpha_2, \alpha_3)$, with each region function where α_i is the power of ξ_i in the resulting shape function. Similar to the quadrilateral face, the geometric construct to generate the possible triplet combinations in this case is analogous to Pascal's tetrahedron used with equation (44). For example, at the apex (p=6) would be (2,2,2), followed by (3,2,2), (2,3,2) and (2,2,3) in the next layer (p=7). The expressions for hexahedral region functions generated from equation (43) are listed in Table 6 for p=6,7.

Wedge Mesh Region

There are a total of $\frac{(p-3)(p-4)}{2}$ wedge region functions with polynomial order p, $p \geq 5$. The region modes of polynomial order p are products of the triangular face functions and

Table 6. Hexahedral region functions.

p	$(\alpha_1, \alpha_2, \alpha_3)$	ϕ
6	2, 2, 2	$\frac{3\sqrt{6}}{8}\left[\left(3\xi_1^2-1\right)\left(3\xi_2^2-1\right)\left(3\xi_3^2-1\right)\right]$
7	3, 2, 2	$\frac{3\sqrt{10}}{8}\left[\left(5\xi_1^3-3\xi_1\right)\left(3\xi_2^2-1\right)\left(3\xi_3^2-1\right)\right]$
	2, 3, 2	$\frac{3\sqrt{10}}{8}\left[\left(3\xi_1^2-1\right)\left(5\xi_2^3-3\xi_2\right)\left(3\xi_3^2-1\right)\right]$
	2, 2, 3	$\frac{3\sqrt{10}}{8}\left[\left(3\xi_1^2-1\right)\left(3\xi_2^2-1\right)\left(5\xi_3^3-3\xi_3\right)\right]$

the one-dimensional edge function along ξ_4 given by

$$\begin{aligned} \phi\left(M_i^3\right) &= A(\xi_1,\xi_2,\xi_3)\times B(\xi_4) \\ A(\xi_1,\xi_2,\xi_3) &= \xi_1\xi_2\xi_3\phi\left(M_j^2\right) \\ B\left(\xi_4^3\right) &= -\left(\frac{1-\xi_4^2}{2}\right)\phi\left(M_k^1\right) \end{aligned} \tag{45}$$

where $\phi\left(M_j^2\right)$ is given by equation (36) with $\hat{\xi}_1 = \xi_1$, $\hat{\xi}_2 = \xi_2$ and $\hat{\xi}_3 = \xi_3$. $\phi\left(M_k^1\right)$ is given by equation (33) with $\hat{\xi}_1 = \xi_4$ for parameterization *I* or $\hat{\xi}_1 = \frac{1-\xi_4}{2}$ and $\hat{\xi}_2 = \frac{1+\xi_4}{2}$ for parameterization *II*, and $p = \alpha_4$. The directional properties of the region functions can be interpreted by associating a quadruplet, $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ with each mode. As in the case of triangles, since only two of ξ_1, ξ_2, ξ_3 can be varied independently. Therefore, two of $\alpha_1, \alpha_2, \alpha_3$, and α_4 are used in the generation of a possible set of region functions. For example, if ξ_1, ξ_2, ξ_4 are chosen as the independent coordinate parameters, the following rules generate the set of possible region functions,

$$\begin{aligned} &\alpha_1, \alpha_2 \geq 1 \\ &\alpha_4 \geq 2 \\ &\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = p. \end{aligned} \tag{46}$$

The region functions for *p=5,6* are listed in Table 7.

Pyramid Mesh Region

One possible way to construct region shape functions, similar to the scheme for hexahedral elements, is given by the following expression

$$\phi\left(M_i^3\right) = P_{\alpha_1}\left(\frac{2\xi_1}{1-\xi_2}\right)P_{\alpha_2}(\xi_2)P_{\alpha_3}\left(\frac{2\xi_3}{1-\xi_2}\right) \tag{47}$$

Table 7. Region functions for a wedge.

p	$(\alpha_1\alpha_2, \alpha_3, \alpha_4)$	ϕ
5	(1,1,1,2)	$\phi(M_i^3) = -\frac{1}{2}\xi_1\xi_2\xi_3(1-\xi_4^2)$
6	(1,1,1,3)	$\phi(M_i^3) = -\frac{1}{2}\xi_1\xi_2\xi_3\xi_4(1-\xi_4^2)$
	(1,2,1,2)	$\phi(M_i^3) = -\frac{1}{2}\xi_1\xi_2\xi_3(\xi_2-\frac{1}{3})(1-\xi_4^2)$
	(2,1,1,2)	$\phi(M_i^3) = -\frac{1}{2}\xi_1\xi_2\xi_3(\xi_1-\frac{1}{3})(1-\xi_4^2)$

Table 8. Pyramid region functions.

p	$(\alpha_1, \alpha_2, \alpha_3)$	ϕ
6	2, 2, 2	$\left(\frac{6\xi_1^2}{(1-\xi_2)^2}-\frac{1}{2}\right)\left(\frac{3}{2}\xi_2^2-\frac{1}{2}\right)\left(\frac{6\xi_3^2}{(1-\xi_2)^2}-\frac{1}{2}\right)$
7	3, 2, 2	$\left(\frac{20\xi_1^3}{(1-\xi_2)^3}-\frac{3\xi_1}{1-\xi_2}\right)\left(\frac{3}{2}\xi_2^2-\frac{1}{2}\right)\left(\frac{6\xi_3^2}{(1-\xi_2)^2}-\frac{1}{2}\right)$
	2, 3, 2	$\left(\frac{6\xi_1^2}{(1-\xi_2)^2}-\frac{1}{2}\right)\left(\frac{5}{2}\xi_2^3-\frac{3}{2}\xi_2\right)\left(\frac{6\xi_3^2}{(1-\xi_2)^2}-\frac{1}{2}\right)$
	2, 2, 3	$\left(\frac{6\xi_1^2}{(1-\xi_2)^2}-\frac{1}{2}\right)\left(\frac{3}{2}\xi_2^2-\frac{1}{2}\right)\left(\frac{20\xi_3^3}{(1-\xi_2)^3}-\frac{3\xi_3}{1-\xi_2}\right)$

where the parametric coordinates are scaled based on equation (10) resulting from the degeneration described in Figure 6 with

$$\begin{aligned} &\alpha_1, \alpha_2, \alpha_3 \geq 2 \\ &\alpha_1 + \alpha_2 + \alpha_3 = p. \end{aligned} \tag{48}$$

This results in a total of $\frac{(p-4)(p-5)}{2}$ shape functions of polynomials order p, $p \geq 6$. The region shape functions from equation (47) can be generated by using the same triplet scheme as that for the hexahedron entity using Pascal's tetrahedron as the geometric construct. The region functions for a pyramid using equation (47) and p=6,7 are listed in Table 8.

6. Computational Cost for Constructing Variable *p*-Order Meshes Using Decomposed Shape Functions and their Application in 3D

Comparisons of computational cost of shape-function evaluation using the decomposed and explicit uniform p forms are presented in terms of the operation counts required

to evaluate a complete set of shape functions for a given p and the CPU times to evaluate element "stiffness" matrix

$$k_{ij} = \int_{\Omega^e} \frac{\partial N_i}{\partial \mathrm{x}} \left(\frac{\partial N_j}{\partial \mathrm{x}} \right)^T d\Omega. \tag{49}$$

This comparison is based on two-dimensional simplex elements. Counts for other element topologies can be performed in a similar manner. A third comparison illustrates the application of the methodology to the solution of *Poisson*'s equation in three dimensions showing the computational advantage of using the structures of the variable p environment.

The additional overhead of entity based p-specification is associated with the queries of each entity for its polynomial order which cannot be avoided in a variable p-refinement framework. The decomposition of the shape function requires that the index of the bounding lower-order mesh entity in the element topological hierarchy be determined. This requires a constant time search of the list of the bounding faces, edges and vertices. For example, the determination of the local index of a mesh vertex when dealing with a tetrahedral element requires a traversal through the list of four vertices that belong to the closure of that element. The overhead in CPU time of this process compared to the explicit evaluation of the shape functions as a percent of the total CPU time required to solve a given problem is shown to be minimal in numerical example that follows.

Consider the set of all shape functions for a triangular element with uniform polynomial order 6 on all entities of its closure as shown in Table 9. With permutation of indices i and j, there are a total of 28 shape functions (3 from the vertices, 15 from the edges and 10 from the face). The number of multiplication and addition operations required to evaluate each of the shape functions is listed in the columns marked with "*" and "+", respectively, in Table 10. The total cost of evaluating all of the shape functions explicitly is 202 multiplications and 35 additions. The total cost of evaluating all of the shape functions in the decomposed form is 212 multiplications and 35 additions. The additional 10 multiplications in the decomposed form result from the redundant multiplication $\psi * \phi$ when $\phi = 1$. This can be eliminated at the cost of checking for the p values, and is never performed at the three vertices since they are known *a-priori* to always have $\phi = 1$.

A variable p-refinement scheme solver for *Poisson*'s equation was implemented to compare the cost of decomposed shape-function evaluations versus the explicit uniform-p evaluation and its impact on the total solution time. To compute the element level matrices for a variable p mesh the following functionalities are required

1. Access to the topological hierarchy of mesh entities underlying an element. This functionality is available in the SCOREC mesh database [5].

Table 9. Shape functions for a triangle with uniform p=6.

Entity	p	Explicit	Decomposed	
			ψ	ϕ
vertex	0-1	ξ_i	ξ_i	1
edge	2	$-2\xi_i\xi_j$	$-2\xi_i\xi_j$	1
	3	$-2\xi_i\xi_j(\xi_j-\xi_i)$	$-2\xi_i\xi_j$	$\xi_j-\xi_i$
	4	$-2\xi_i\xi_j\left(\xi_j^2-3\xi_i\xi_j+\xi_i^2\right)$	$-2\xi_i\xi_j$	$\xi_j^2-3\xi_i\xi_j+\xi_i^2$
	5	$-2\xi_i\xi_j\left(\xi_j^3-6\xi_i\xi_j^2+6\xi_i^2\xi_j-\xi_i^3\right)$	$-2\xi_i\xi_j$	$\xi_j^3-6\xi_i\xi_j^2+6\xi_i^2\xi_j-\xi_i^3$
	6	$-2\xi_i\xi_j\left(\xi_j^4-10\left(\xi_i\xi_j^3-2\xi_i^2\xi_j^2+\xi_i^3\xi_j\right)+\xi_i^4\right)$	$-2\xi_i\xi_j$	$\xi_j^4-10\left(\xi_i\xi_j^3-2\xi_i^2\xi_j^2+\xi_i^3\xi_j\right)+\xi_i^4$
face	3	$\xi_1\xi_2\xi_3$	$\xi_1\xi_2\xi_3$	1
	4	$\xi_1\xi_2\xi_3(\xi_1-\frac{1}{3})$	$\xi_1\xi_2\xi_3$	$\xi_1-\frac{1}{3}$
		$\xi_1\xi_2\xi_3(\xi_2-\frac{1}{3})$	$\xi_1\xi_2\xi_3$	$\xi_2-\frac{1}{3}$
	5	$\xi_1\xi_2\xi_3(\xi_1^2-\frac{3}{4}\xi_1+\frac{3}{28})$	$\xi_1\xi_2\xi_3$	$\xi_1^2-\frac{3}{4}\xi_1+\frac{3}{28}$
		$\xi_1\xi_2\xi_3(\xi_1\xi_2-\frac{1}{4}(\xi_1+\xi_2)+\frac{1}{14})$	$\xi_1\xi_2\xi_3$	$\xi_1\xi_2-\frac{1}{4}(\xi_1+\xi_2)+\frac{1}{14}$
		$\xi_1\xi_2\xi_3(\xi_2^2-\frac{3}{4}\xi_2+\frac{3}{28})$	$\xi_1\xi_2\xi_3$	$\xi_2^2-\frac{3}{4}\xi_2+\frac{3}{28}$
	6	$\xi_1\xi_2\xi_3(\xi_2^3-\frac{1}{5}(6\xi_2^2-2\xi_2+\frac{1}{6}))$	$\xi_1\xi_2\xi_3$	$\xi_2^3-\frac{1}{5}(6\xi_2^2-2\xi_2+\frac{1}{6})$
		$\xi_1\xi_2\xi_3\left(\xi_1\xi_2(\xi_2-\frac{3}{5})+\frac{1}{5}\left(\frac{\xi_1}{3}-\xi_2^2+\frac{2\xi_2}{3}-\frac{1}{12}\right)\right)$	$\xi_1\xi_2\xi_3$	$\xi_1\xi_2(\xi_2-\frac{3}{5})+\frac{1}{5}\left(\frac{\xi_1}{3}-\xi_2^2+\frac{2\xi_2}{3}-\frac{1}{12}\right)$
		$\xi_1\xi_2\xi_3\left(\xi_1\xi_2(\xi_1-\frac{3}{5})+\frac{1}{5}\left(\frac{\xi_2}{3}-\xi_1^2+\frac{2\xi_1}{3}-\frac{1}{12}\right)\right)$	$\xi_1\xi_2\xi_3$	$\xi_1\xi_2(\xi_1-\frac{3}{5})+\frac{1}{5}\left(\frac{\xi_2}{3}-\xi_1^2+\frac{2\xi_1}{3}-\frac{1}{12}\right)$
		$\xi_1\xi_2\xi_3(\xi_1^3-\frac{1}{5}(6\xi_1^2-2\xi_1+\frac{1}{6}))$	$\xi_1\xi_2\xi_3$	$\xi_1^3-\frac{1}{5}(6\xi_1^2-2\xi_1+\frac{1}{6})$

2. Ability to query (and set) the polynomial order of interpolation for a given field variable over a given mesh entity.
3. Ability to query which of the specific shape functions of a given polynomial order are desired for a mesh entity.
4. Ability to evaluate shape function(s) of a given polynomial order from a mesh entity at a point in the domain of the element.
5. Ability to evaluate derivatives of shape function(s) of a given polynomial order from a mesh entity at a point in the domain of the element.

Table 10. Operation count for shape functions for a triangle with uniform p=6.

Entity	p	Explicit		Decomposed					
				ψ		ϕ		$\psi * \phi$	
		*	+	*	+	*	+	*	+
vertex	0-1	0	0	0	0	0	0	1	0
edge	2	2	0	2	0	0	0	1	0
	3	3	1	2	0	0	1	1	0
	4	9	2	2	0	6	2	1	0
	5	13	3	2	0	10	3	1	0
	6	20	4	2	0	17	4	1	0
face	3	2	0	2	0	0	0	1	0
	4	3	1	2	0	0	1	1	0
		3	1	2	0	0	1	1	0
	5	5	2	2	0	2	2	1	0
		5	3	2	0	2	3	1	0
		5	2	2	0	2	2	1	0
	6	10	3	2	0	7	3	1	0
		9	5	2	0	6	5	1	0
		9	5	2	0	6	5	1	0
		10	3	2	0	7	3	1	0

A set of generic operators provide these functionalities.

Consider the solution of *Laplace*'s equation on the unit square $[0,1] \times [0,1]$ with $u = 0$ on $x = 0$, $y = 0$, and $y = 1$ and $\frac{\partial u}{\partial x} = 1$ on $x = 1$. The exact solution of this problem is [16]

$$u(x,y) = \frac{4}{\pi^2} \sum_{n=1,3,5}^{\infty} \frac{\sinh(n\pi x)\sin(n\pi y)}{n^2 \cosh(n\pi)}. \tag{50}$$

The CPU time required to compute the element level matrices for a uniform 8 by 8 mesh of 128 triangular element mesh for p=1 to 8 with the explicit uniform p and the decomposed form of the shape functions are shown in Figure 19(a). Use of decomposed shape functions increases element matrix computation time by 10% relative to explicit

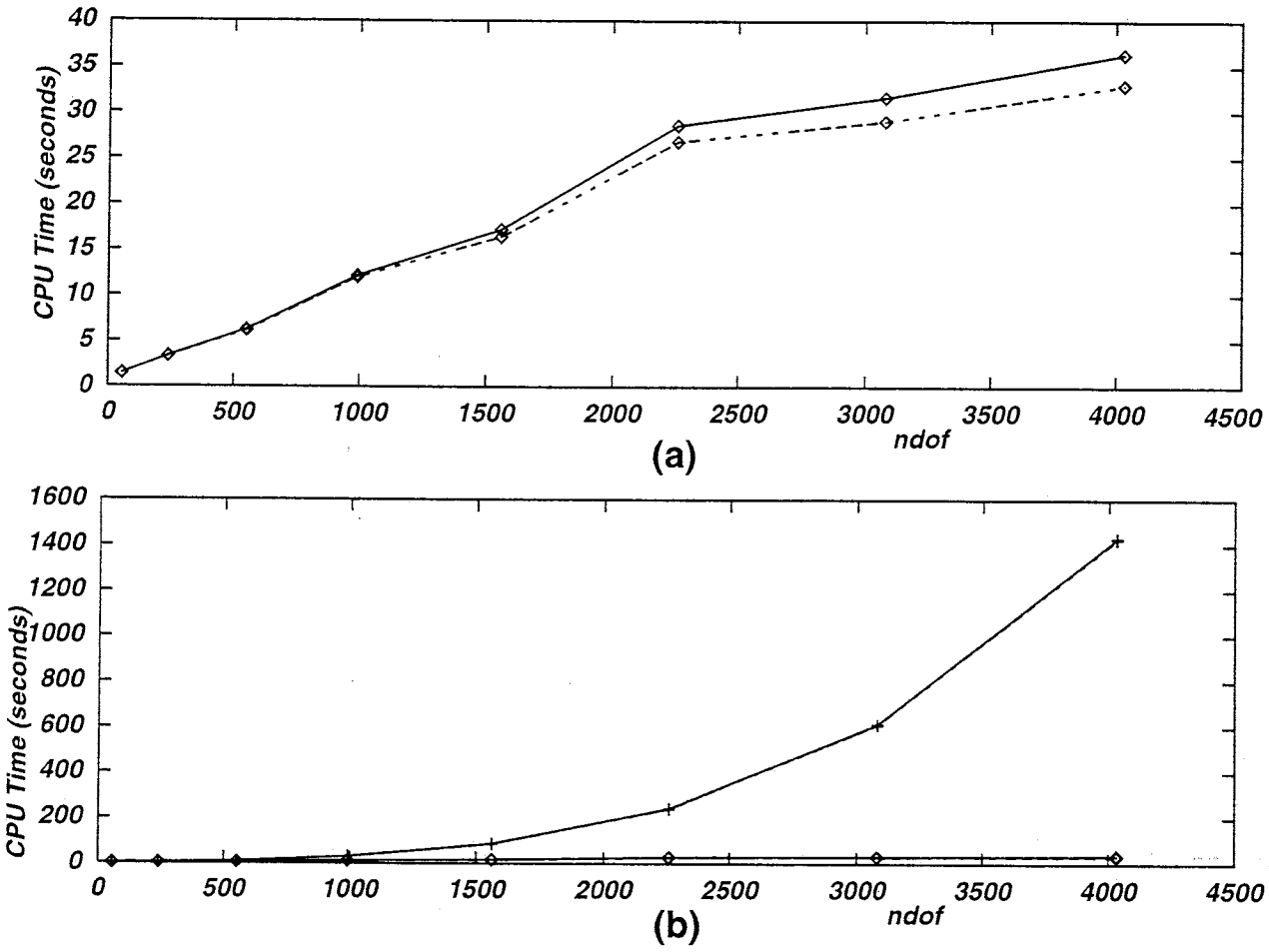

Figure 19. CPU time to compute (a) element level matrices and (b) total solution for 128 elements with uniform p from 1 to 8. Solid and broken lines represent decomposed and explicitly evaluated shape functions, respectively; ◇ and + represent times to compute element matrices and total solution time, respectively.

evaluation when p=8. The increase is due to the local searching to determine the indices of lower-order mesh entities in the element topological hierarchy. However, the increase in the CPU time required to solve the problem completely (Figure 19(b)), is minimal (0.3%).

Optimality of the *hp*-adaptive procedure lies in simultaneous variation of both the mesh size and the order of interpolation to attain solutions with prescribed accuracy using minimal computational resources [3, 20, 27]. To demonstrate the advantages of varying polynomial order over the domain of three-dimensional problems, consider the solution

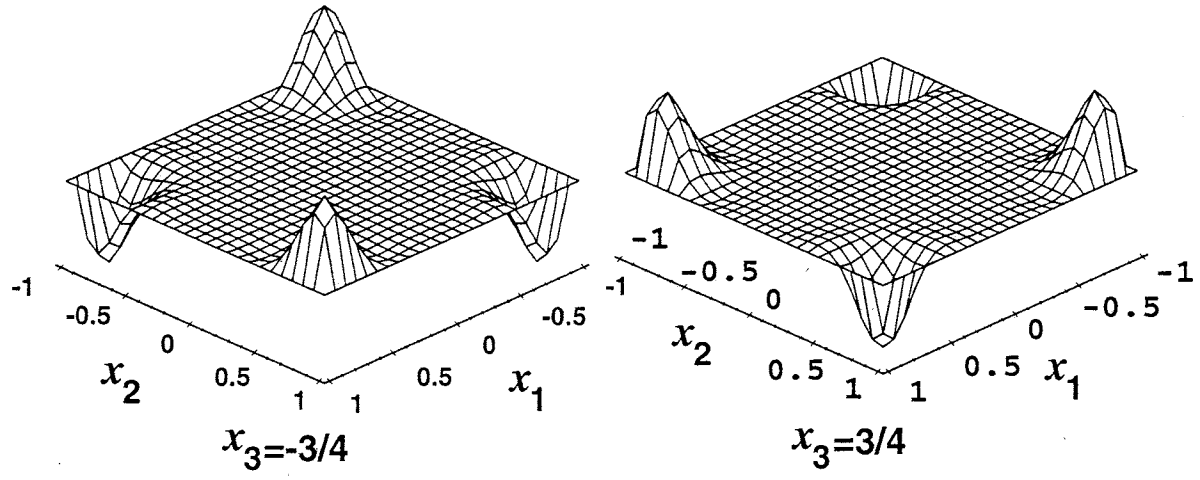

Figure 20. Solution profile at $x_3 = \pm\frac{3}{4}$ for n=4.

of *Poisson*'s equation

$$\begin{aligned} \nabla^2 u(\mathrm{x}) &= f(\mathrm{x}),\ \mathrm{x} \in \overline{\Omega} \\ u(\mathrm{x}) &= 0,\ \mathrm{x} \in \partial(\Omega) \end{aligned} \tag{51}$$

where Ω is defined by a cube with side length 2 units and $f(x)$ is such that the exact solution is

$$u(\mathrm{x}) = x_1^n x_2^n x_3^n sin\Big(\frac{n\pi x_1}{2}\Big) sin\Big(\frac{n\pi x_2}{2}\Big) sin\Big(\frac{n\pi x_3}{2}\Big). \tag{52}$$

This solution is smooth within the domain but has highly localized gradients near the corners of Ω. Solution projections at $x_3 = \pm\frac{3}{4}$ for n=4 are shown in Figure 20.

Since the solution gradients are localized, a variable p-refinement strategy offers significant computational advantages over a uniform p-refinement when the mesh shown in Figure 21 is used to solve the problem. For example, if one considers the percentage global relative error, ρ, in the finite element solution, u^*, given by

$$\rho = 100 \frac{\|u - u^*\|_2}{\|u\|_2} \tag{53}$$

then the use of a simple p-adaptation scheme based on equidistribution of regionwise error uses 25% fewer global *dof*'s compared to uniform p specification to achieve a numerical solution with 1% global error. The total CPU time needed by the adaptive procedure was 43% of that needed by the uniform p procedure.

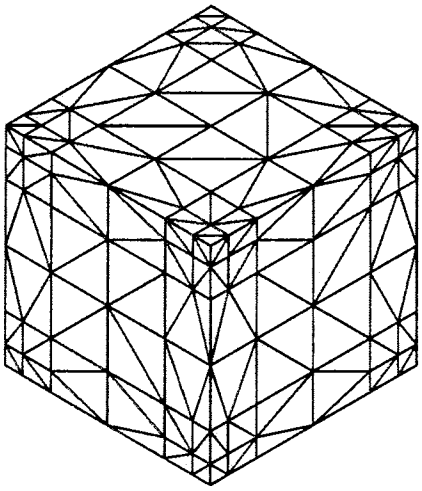

Figure 21. Mesh used for the three dimensional example problem with 943 mesh regions, 2078 mesh faces, 1411 mesh edges and 277 mesh vertices.

7. Closing Remarks

This paper has presented a geometry-based methodology to support the construction of variable p-order meshes employing elements of various topological form. Key to the effectiveness of this approach is the decomposition of higher-order mesh entity shape functions into a function over the mesh entity multiplied by a blending function over the domain of the element. The geometry-based structures also support a variety of functions needed by mesh-based solution methods, including integration over the curved domain of p-version finite elements [9].

The shape function decomposition process is efficient and easily accomplished when an appropriate set of mesh entity parametric coordinates are used. In fact, all the shape functions given here for the tetrahedral and the hexahedral elements are decomposed versions of previously presented p-version shape functions. With the specific blending functions used here, one has the option to employ a variety of entity level functions in the construction of shape functions for that entity. Such a structure may be advantageous for the construction of specialized shape functions, such as those that capture singularities that may be present over that mesh entity.

A final advantage of the structures presented here is their central role in a generalized hp-adaptive finite element analysis program.

8. Acknowledgment

The authors would like to acknowledge the support of the Office of Naval Research through grant No. N00014-94-1-0962 and Naval Research Laboratory through ONR grant No. N00014-C-6023.

9. References

[1] M. Abramowitz and I. A. Stegun, editors. *Handbook of Mathematical Functions.* Dover Pulications Inc., New York, 1972.

[2] I. Babuska, M. Griebel, and J. Pitkaranta. The problem of selecting the shape functions for p-type finite elements. *Int. J. Numer. Meth. Engng.*, 28:1891–1908, 1989.

[3] I. Babuska and M. Suri. The p- and hp-versions of the finite element method: An overview. *Comp. Meth. Appl. Mech. Engng.*, pages 5–26, 1990.

[4] I. Babuska, B. Szabo, and I. N. Katz. The p-version of the finite element method. *SIAM J. Numer. Anal.*, 18(3):515–541, 1981.

[5] M. W. Beall and M. S. Shephard. Mesh data structures for advanced finite element computations. Technical Report 19-1995, Scientific Computation Research Center, Rensselaer Polytechnic Institute, Troy, NY 12180-3590, 1995. submitted to Int. J. Num. Meth. Engng.

[6] P. Carnevali, R. B. Morris, Y. Tsuji, and G. Taylor. New basis functions and computational procedures for p-version finite element analysis. *Int. J. Numer. Meth. Engng.*, 36:3759–3779, 1993.

[7] L. Demkowicz, J. T. Oden, W. Rachowicz, and O. Hardy. Toward a universal h-p adaptive finite element strategy, part 1: Constrained approximation and data structure. *Computer Methods in Applied Mechanics and Engineering*, 77:79–112, 1989.

[8] K. D. Devine and J. E. Flaherty. A parallel adaptive hp-refinement finite element method with dynamic load balancing for the solution of hyperbolic conservation laws. Technical Report SCOREC Report # 14-1995, Troy, NY 12180-3590, 1995. Submitted to: Special issue of Applied Numerical Mathematics.

[9] S. Dey and M. S. Shephard. Geometry-based entity level integrations in higher order p- and hp-adaptive fem. Technical Report In preparation for submission, Scientific Computation Research Center, Rensselaer Polytechnic Institute, Troy, NY 12180-3590, 1996.

[10] J. Fish and R. Guttal. Recent advances in the p-version of the finite element method for shells. *Computing Systems in Engineering*, 6(3):195–211, 1995.

[11] W. Gui and I. Babuska. The h-, p- and hp-version of the finite element method in one dimension. part 1: The analysis of the p-version. part 2: The error analysis of the h- and hp-version. part 3: The adaptive hp-version. *Numerische Mathematik*, 49:577–612; 613–657; 659–683, 1986.

[12] E. L. Gursoz, Y. Choi, and F. B. Prinz. Vertex-based representation of non-manifold boundaries. In M. J. Wozny, J. U. Turner, and K. Priess, editors, *Geometric Modeling Product Engineering*, pages 107–130. North Holland, 1990.

[13] T. J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Prentice Hall, Englewood Cliffs, NJ, 1987.

[14] Y. Jinyun. Symmetric gaussian quadrature formulae for tetrahedral regions. *Comp. Meth. Appl. Mech. Engng.*, 43(3):349–353, 1984.

[15] T. J. Liszka, C. W. Berry, and O. P. Hardy. Constrained hp-approximation in anisotropic hex-based finite element code. In *Third US National Congress on Computational Mechanics*, page 232, 1995.

[16] C. R. MacCluer. *Boundary Value Problems and Orthogonal Expansions Physical Problems from a Sobolev Viewpoint*. IEEE Press, 1994.

[17] M. Mäntylä. *Introduction to Solid Modeling*. Computer Science Press, Rockville, Maryland, 1988.

[18] J. T. Oden. Parallel adaptive hp-finite element methods for problems in fluid and solid mechanics. In T. J. R. Hughes, E. Onate, and O. C. Zienkiewicz, editors, *Recent developments in finite element analysis*, pages 29–35. International Center for Numerical Methods in Engineering, Barcelona, Spain, 1994.

[19] J. T. Oden and L. Demkowicz. h-p adaptive finite element methods in computational fluid dynamics. *Comp. Meth. Appl. Mech. Engng.*, 89:11–40, 1991.

[20] J. T. Oden, W. Wu, and M. Ainsworth. Three-step h-p adaptive strategy for the incompressible Navier-Stokes equations. In I. Babuska, J. E. Flaherty, J. E. Hopcroft, W. D. Henshaw, J. E. Oliger, and T. Tezduyar, editors, *Modeling, Mesh Generation, and Adaptive Numerical Methods for Partial Differential Equations*, volume 75. Springer-Verlog, 1995. IMA Volumes in Mathematics and its Applications.

[21] W. Rachowicz, J. T. Oden, and L. Demkowicz. Toward a universal h-p adaptive finite element strategy, part 3. design of h-p meshes. *Computer Methods in Applied Mechanics and Engineering*, 77:181–212, 1989.

[22] A. A. G. Requicha and H. B. Voelcker. Solid modeling: Current status and research directions. *IEEE Computer Graphics and Applications*, 3(7):25–37, 1983.

[23] W. J. Schroeder and M. S. Shephard. On rigorous conditions for automatically generated finite element meshes. In J. Turner, J. Pegna, and M. Wozny, editors, *Product Modeling for Computer-Aided Design and Manufacturing*, pages 267–281. North Holland, 1991.

[24] M. S. Shephard. The specification of physical attribute information for engineering analysis. *Engineering with Computers*, 4:145–155, 1988.

[25] M. S. Shephard and P. M. Finnigan. Toward automatic model generation. In A. K. Noor and J. T. Oden, editors, *State-of-the-Art Surveys on Computational Mechanics*, pages 335–366. ASME, 1989.

[26] S. J. Sherwin and G. E. Karniadakis. A new triangular and tetrahedral basis for high-order (hp) finite element methods. *Int. J. Numer. Meth. Engng.*, 38:3775–3802, 1995.

[27] B. A. Szabo and I. Babuska. *Finite Element Analysis*. Wiley Interscience, New York, 1991.

[28] B. A. Szabo and A. G. Peano. Hierarchic finite elements. In H. Kardestuncer and D. H. Norrie, editors, *Finite Element Handbook*, pages 2.227–2.233. McGraw-Hill, New York, 1987.

[29] K. J. Weiler. The radial-edge structure: A topological representation for non-manifold geometric boundary representations. In M. J. Wozny, H. W. McLaughlin, and J. L. Encarnacao, editors, *Geometric Modeling for CAD Applications*, pages 3–36. North Holland, 1988.

[30] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method Volume 1 Basic Formulation and Linear Problems*. McGraw-Hill Book Company, London, 1989.