Geometry Based Framework for *hp* Adaptive FEM

Saikat Dey and Mark S. Shephard
Scientific Computation Research Center
Rensselaer Polytechnic Institute, Troy, NY 12180

Summary

A geometry based framework for *hp*-adaptive finite elements is presented. The framework builds on a two level decomposition of the shape functions which directly support the specification of variable p-order meshes. Numerical integration employs domain shape information as defined in a CAD system. Simple two-dimensional examples demonstrate the effectiveness of the procedures which have been fully implemented in three-dimensions.

1 Introduction

General structures to support *hp*-adaptive finite element methods is an area of active research [8]. In the procedures developed here, the problem input consists of a geometric model, $\overline{\Omega}_G$, describing the problem domain, and a set of attributes tied to the geometric model. The finite element descretization of $\overline{\Omega}_G$ is called the mesh model, $\overline{\Omega}_M$. Figure 1(a) depicts the interactions of the components of an adaptive finite element system. The effective application of *hp* version finite element methods requires a framework that supports adaptively evolving meshes of variable p-order elements. It must also employ appropriate geometry representations for p-elements covering large portions of the problem domain.

This paper presents the development of the two shaded modules of an automatic *hp*-adaptive system for curved three-dimensional domains. Section 2 presents a mesh topological hierarchy tied to the geometric model and decomposed shape functions that allow the specification of variable p-order meshes. Section 3 presents techniques to map the geometry of mesh entities to conform to the exact CAD geometry, thus allowing the direct use of geometry in the construction of element level matrices and for h adaptation of the mesh [3]. The present paper considers only simplices. However, the procedures can as easily be applied to non-simplex and mixed topology meshes [9]. Application examples presented in Section 4 show the efficiency of the decomposed shape functions and superiority of using exact geometry mapping over "fitting"/interpolation techniques.

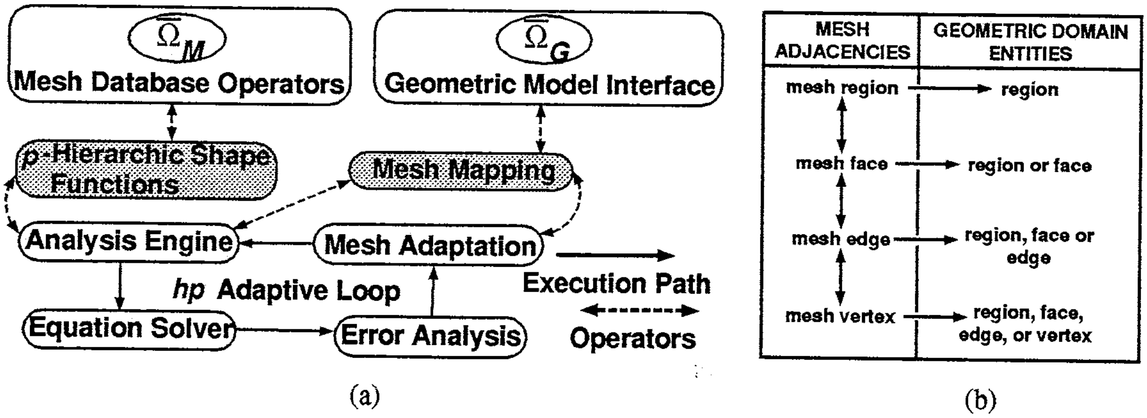

Figure 1. Geometry based (a) *hp*-adaptive framework and (b) mesh topological hierarchy.

2 Geometry Based Mesh Topological Hierarchy for Variable *p*-Order Meshes

A boundary representation for $\overline{\Omega}_M$ consistent with that used for $\overline{\Omega}_G$ provides a natural means to support variable p meshes. Since individual volume finite elements will be limited to simple regions, 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, which for the three-dimensional case (d=3) are:

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

where $M\{M^d\}$, $d = 0, 1, 2, 3$ are respectively the set of vertices, edges, faces and regions defining the primary topological elements of the mesh model $\overline{\Omega}_M$. The association of mesh topological entities with respect to the topological entities of the geometric model [11] is central to obtaining shape information for individual mesh entities directly from the geometric modelling system by various procedures of the adaptive system. Figure 1(b) graphically depicts the mesh topological adjacencies used as well as the association of each mesh entity with the geometric model. The use of the topological hierarchy in the specification of variable p-order meshes consist of the simple assignment of the desired polynomial order information with each topological entity in the mesh. In general that means, each mesh vertex, M_i^0, mesh edge M_i^1, mesh face M_i^2, and mesh region M_i^3, can have its own specification of polynomial order, including accounting for the possibility of different directional polynomial orders for the higher geometric dimension entities of faces and regions.

The closure of a finite element of dimension d_e, $\overline{\Omega}_e$, can be evaluated as

$$\overline{\Omega}_e = \left\{M_e^{d_e}, \partial\left(M_e^{d_e}\right)\right\} = \left\{M_e^{d_e}, M_e^{d_e}\left\{M_j^{d_e-1}\right\}, \ldots, M_e^{d_e}\{M_j^0\}\right\} \tag{2}$$

where $M_i^{d_i}\{M^{d_j}\}$ defines a first order adjacency of mesh entities [1] given by the set of mesh entities of dimension d_j adjacent to mesh entity $M_i^{d_i}$. For example, the mesh vertices bounding a particular mesh region are denoted $M_i^3\{M^0\}$. The complete basis of an element is formed by shape functions contributed by individual topological entities that belong to closure of the element [12].

This structure helps support a decomposition of the shape functions into the product of a mode shape associated with a bounding mesh entity, $M_i^{d_i}$, and a blending term associated with the element entity, $M_e^{d_e}$. These shape functions can be written as

$$N_a = \psi\left(M_j^{d_j}, M_e^{d_e}\right)\phi\left(M_j^{d_j}\right) \tag{3}$$

where: $\psi\left(M_j^{d_j}, M_e^{d_e}\right)$ is a blending function defined on the mesh entity, $M_e^{d_e}$, written in its parametric coordinate system, ξ_i, which is specific to the mesh entity, $M_j^{d_j}$ (but independent of the polynomial order of the shape function). $\phi\left(M_j^{d_j}\right)$ is a function written in the mesh entity $M_j^{d_j}$ parametric coordinate system, $\hat{\xi}_j$, and is a function of the polynomial order of the desired shape function, but is independent of $\psi\left(M_j^{d_j}, M_e^{d_e}\right)$, and thus is the same for all elements for which $M_j^{d_j}$ is a part of (including elements of different topology and/or dimension).

This decomposition provides complete flexibility of incorporating the variable polynomial order of the shape functions based on the topological hierarchy during element level computations. The construction of the blending and the entity functions is done such that the resulting shape functions meet the following requirements:

1. Satisfy the polynomial order specified on each of the mesh entities in $\overline{\Omega}_e$.
2. Satisfy the C^0 interelement continuity requirement.
3. Use a minimum of operations for evaluation and subsequent integration.

It is important to note that this approach will automatically, and simply[1], satisfy the C^0 interelement continuity requirements for variable p-order meshes. Computational efficiency of this decomposition is

[1] For example it avoids the need to change sign for odd order shape functions on mesh entities common to two elements.

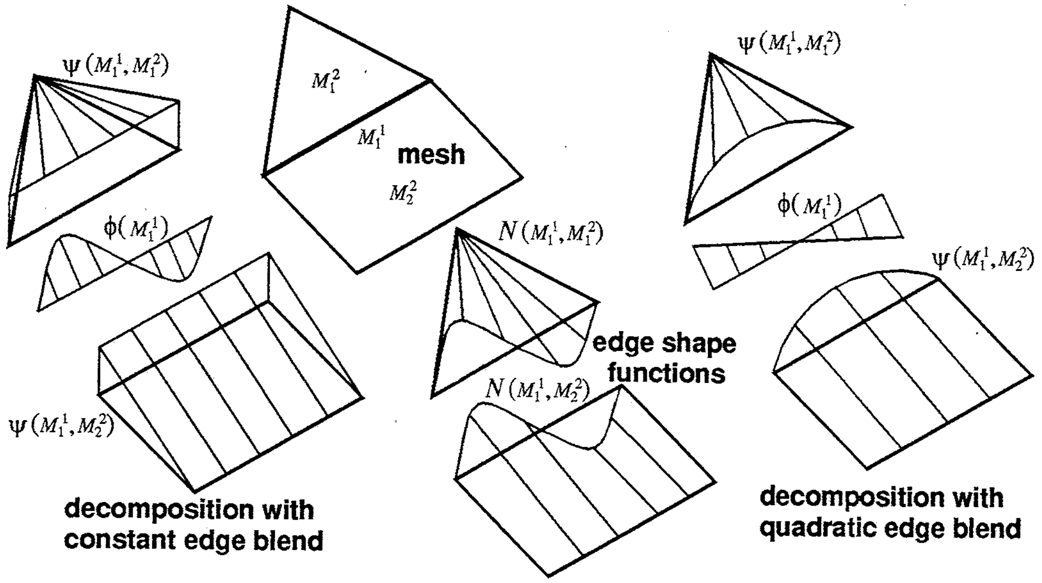

Figure 2. Cubic edge mode decomposition.

not obvious and is dependent on the choice of parametric coordinate systems for the mesh entities and the blending functions.

Since shape functions (and their derivatives) evaluations in the parametric domain of the element, ξ_i, requires evaluating functions in the parametric domain of bounding mesh entities, $\hat{\xi}_j$, a mapping transforming ξ_i to $\hat{\xi}_j$ is required. The "^" is used to distinguish the parametric coordinates of the bounding lower order entities. Barycentric coordinates to parameterize edges, triangular faces and tetrahedral regions leads to trivial mapping between the element parametric coordinates and the bounding mesh entity parameters, $\hat{\xi}_i \equiv \xi_j$.

Consider the two possible decompositions of a cubic edge shape function shown in Figure 2 using blending functions that are constant and quadratic over the specific edge. The more intuitive constant blend over the triangle element leads to a rational blend expression, $\psi(M_1^1, M_1^2) = \frac{4\xi_i\xi_j}{1-(\xi_j-\xi_i)^2}$, requiring 2 addition and 7 multiplication operations whereas the less obvious quadratic edge blend leads to a simple polynomial blend expression, $\psi(M_1^1, M_1^2) = -2\xi_i\xi_j$, requiring only 1 addition and 3 multiplication operations.

For simplices, the vertex, edge, face and region blending functions are given by ξ_i, $-2\xi_i\xi_j$, $\xi_i\xi_j\xi_k$ and $\xi_i\xi_j\xi_k\xi_l$ respectively. The indices i, j, k identify the mesh vertices in the element topological hierarchy that bound the particular entity. Details of expressions for blending functions for general element topologies that match those for the simplex elements will be given in [9].

The selection of the entity functions, ϕ, typically considers issues of sparsity and condition number of the resulting matrices [2]. The use of orthogonal basis using integrals of Legendre polynomials are popular [12]. An alternative basis that yields better conditioned element level matrices is given in [2]. Table 3 lists the entity functions required to construct the hierarchic basis for a triangular element with p=5[†]. Detailed expressions for ϕ will be given in [9].

Comparison of operation counts shows that for a variable p-order mesh, the decomposed shape functions can be evaluated as efficiently as their explicitly written form.

3 Construction of Element Matrices based on CAD Geometry

Element level matrices are typically evaluated by numerical integration carried out in the element parametric domain [12]. For example, typical "stiffness" and "mass" matrix calculations can be

[†] The indices i and j are permuted to collect shape functions from the three vertices and three edges.

Entity	p	N= $\psi * \phi$	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	$-2\xi_i\xi_j(\xi_j^2 - 3\xi_i\xi_j + \xi_i^2)$	$-2\xi_i\xi_j$	$\xi_j^2 - 3\xi_i\xi_j + \xi_i^2$
	5	$-2\xi_i\xi_j(\xi_j^3 - 6\xi_i\xi_j^2 + 6\xi_i^2\xi_j - \xi_i^3)$	$-2\xi_i\xi_j$	$\xi_j^3 - 6\xi_i\xi_j^2 + 6\xi_i^2\xi_j - \xi_i^3$
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	$\xi_1\xi_2\xi_3(\xi_1^2 - \frac{3}{4}\xi_1 + \frac{3}{28})$	$\xi_1\xi_2\xi_3$	$\xi_1^2 - \frac{3}{4}\xi_1 + \frac{3}{28}$
		$\xi_1\xi_2\xi_3(\xi_1\xi_2 - \frac{1}{4}(\xi_1 + \xi_2) + \frac{1}{14})$	$\xi_1\xi_2\xi_3$	$\xi_1\xi_2 - \frac{1}{4}(\xi_1 + \xi_2) + \frac{1}{14}$
		$\xi_1\xi_2\xi_3(\xi_2^2 - \frac{3}{4}\xi_2 + \frac{3}{28})$	$\xi_1\xi_2\xi_3$	$\xi_2^2 - \frac{3}{4}\xi_2 + \frac{3}{28}$

Figure 3. Shape functions for a triangle with uniform p=5.

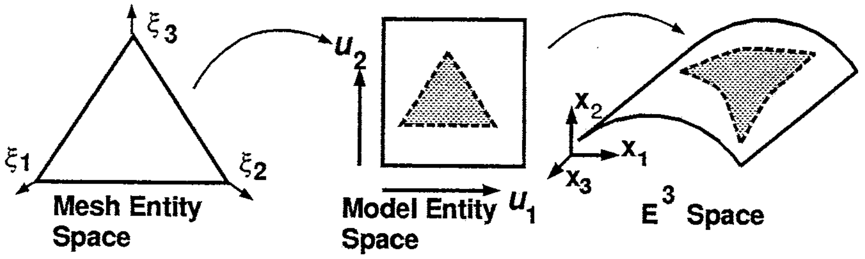

Figure 4. Two step mapping.

represented symbolically as

$$k_{ij} = \int_{\Omega_e} \frac{\partial N_i}{\partial \mathrm{x}} \left(\frac{\partial N_j}{\partial \mathrm{x}} \right)^T |J(\xi_k)| d\Omega; \; m_{ij} = \int_{\Omega_e} N_i N_j |J(\xi_k)| d\Omega \tag{4}$$

respectively where, $\frac{\partial N_i}{\partial \mathrm{x}} = \frac{\partial N_i}{\partial \xi} \frac{\partial \xi}{\partial \mathrm{x}}$ with $J(\xi_k) = \frac{\partial \mathrm{x}}{\partial \xi}$ defining the Jacobian matrix of the mapping, $\mathrm{x}(\xi)$, transforming the element parametric coordinates, ξ_j, to the Cartesian coordinate system. For problems with strong geometry dependence like radiation/scattering problems, the use of low order geometry approximations, using "fitting"/interpolation techniques, for high p elements covering large portions of the domain leads to loss of solution convergence rates and effectivity of estimated errors [4]. Results presented here show the need for mesh geometry conforming to CAD geometry, in general, if coarse elements with high polynomial order are to be used.

The geometry based mesh hierarchy provides an ideal means to construct the mapping $\mathrm{x}(\xi)$ based on the mathematical definition of the domain shape housed within the geometric modeler. For a mesh entity, M_i^d, on a model entity, G_j^d, the desired map can be obtained as

$$\mathrm{x} = \mathrm{x}(u_i(\xi_j)) \tag{5}$$

where u_j represents the parametric coordinate system of G_j^d. The derivative of the mapping can be evaluated as $\frac{\partial \mathrm{x}}{\partial \xi} = \frac{\partial \mathrm{x}}{\partial \mathrm{u}} \frac{\partial u}{\partial \xi}$. Figure 4 depicts the concept for the mapping of a triangular mesh face using the three coordinate systems. $\mathrm{x}(\mathrm{u}), \frac{\partial \mathrm{x}}{\partial \mathrm{u}}$ are queried on a pointwise basis from the geometric modeler. The implicit assumption in this process is that the model entities have an underlying continuous, non-degenerate parametric space. For mesh entities classified on model entities that do not have an underlying parameterization, the shape of the mesh entity is constructed based on blending shapes of bounding lower order entities that conform to the shape of model entities that they are associated with. The use of blends is only required for higher order mesh entities where the blends exactly match the lower order boundary entities, thus eliminating any geometric approximation. The technique used to construct $u_i(\xi_j)$ depends on the association of an $M_k^{d_k}$ with the geometric model.

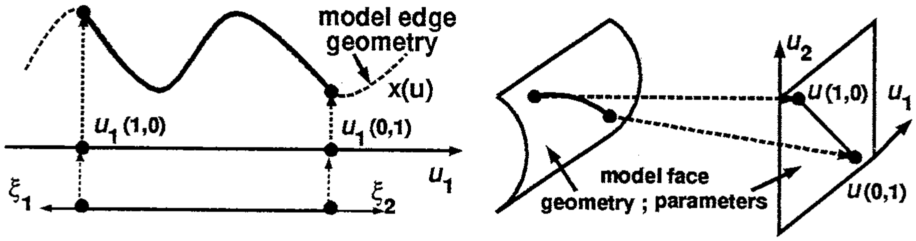

Figure 5. Mesh edge mapping.

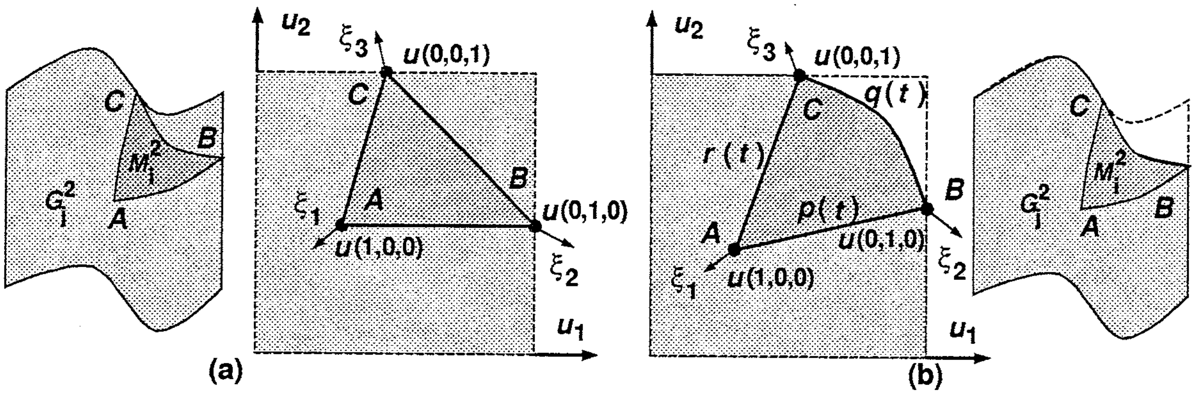

Figure 6. Construction of $U(\xi)$ for (a) untrimmed and (b) trimmed model faces.

Mesh Edges: For a M_i^1 on a $G_i^{d_j}$; $d_j = 1,2$, (Figure 5), the mapping can usually be given by a linear interpolation[3] between $\boldsymbol{u}$ values at the vertices

$$u_i(\xi) = u_i(1,0)\xi_1 + u_i(0,1)\xi_2 \tag{6}$$

For a M_i^1 inside a G_j^3, the shape of the mesh edge is a straight line joining the vertices in the Cartesian coordinate system.

Mesh Faces: The construction of $u_i(\xi_j)$ for M_i^2 on G_j^2 needs to account for the possibility of trimmed model faces which do not span the entire parametric domain of the underlying surface as shown in Figure 6(b). $u_i(\xi_j)$ is given by linear interpolation of the $\boldsymbol{u}$ values at the vertices of the M_i^2 classified on untrimmed G_j^2. For example, for a triangular M_i^2 (Figure 6(a))

$$u_i(\xi) = u(1,0,0)\xi_1 + u(0,1,0)\xi_2 + u(0,0,1)\xi_3. \tag{7}$$

The linear interpolation of vertex values does not work if the G_j^2 is trimmed because the curves defining the mesh edges, in the parametric space of the G_j^2, may not be straight in that space. Consider the situation shown in Figure 6(b) where the construction of $u_i(\xi_j)$ must account for the curved shape of the boundary between B and C. Techniques based on the Boolean sum interpolation theory can be used to blend the boundary curves [5, 6]. Using the scheme described in [6] yields

$$\begin{aligned} u_i(\xi_j) = \frac{1}{2}\{&\left(\frac{\xi_1}{1-\xi_2}\right)p_i(\xi_2) + \left(\frac{\xi_2}{1-\xi_1}\right)p_i(1-\xi_1) + \left(\frac{\xi_3}{1-\xi_2}\right)q_i(1-\xi_2) + \left(\frac{\xi_2}{1-\xi_3}\right)q_i(\xi_3) \\ &+ \left(\frac{\xi_1}{1-\xi_3}\right)r_i(1-\xi_3) + \left(\frac{\xi_3}{1-\xi_1}\right)r_i(\xi_1) - \xi_1 u_i(1,0,0) - \xi_2 u_i(0,1,0) - \xi_3 u_i(0,0,1)\} \end{aligned} \tag{8}$$

where $p(t), q(t), r(t)$; $t \in [0,1]$ define the shape of the curves AB, BC and CD respectively.

[3] In some cases higher order interpolants are required to avoid poorly shaped elements [10].

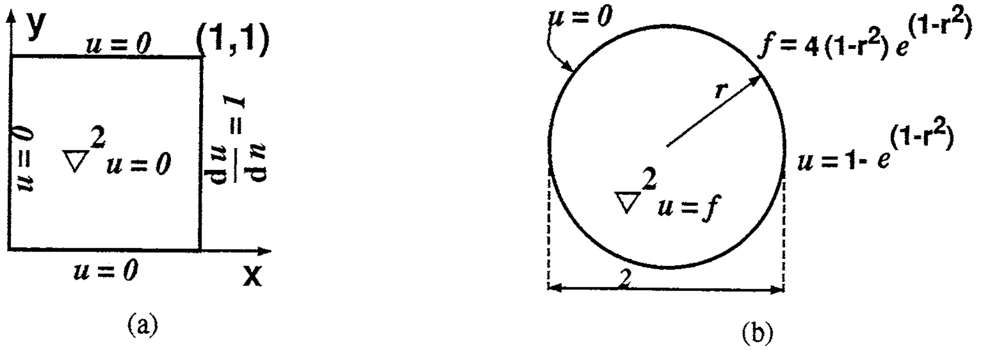

Figure 7. Example problems.

For M_i^2 inside G_j^3 with no underlying parametrization and one or more bounding edges classified on a curved model boundary, the positions of the bounding vertices and the shapes of the bounding edges are blended to obtain $\mathrm{x}(\xi_j)$ as

$$\begin{aligned} x_i(\xi_j) = \frac{1}{2}\{&\left(\frac{\xi_1}{1-\xi_2}\right)P_i(\xi_2) + \left(\frac{\xi_2}{1-\xi_1}\right)P_i(1-\xi_1) + \left(\frac{\xi_3}{1-\xi_2}\right)Q_i(1-\xi_2) + \left(\frac{\xi_2}{1-\xi_3}\right)Q_i(\xi_3) \\ &+ \left(\frac{\xi_1}{1-\xi_3}\right)R_i(1-\xi_3) + \left(\frac{\xi_3}{1-\xi_1}\right)R_i(\xi_1) - \xi_1 x_i(1,0,0) - \xi_2 x_i(0,1,0) - \xi_3 x_i(0,0,1)\} \end{aligned} \tag{9}$$

where P, Q, R define the shapes of the bounding edges. For mesh faces classified inside model regions with none of the bounding edges classified on a curved model boundary, $\mathrm{x}(\xi)$ is given by linear interpolation of the vertex coordinates

$$x_i(\xi_j) = x_i(1,0,0)\xi_1 + x_i(0,1,0)\xi_2 + x_i(0,0,1)\xi_3. \tag{10}$$

Mesh Regions: The volume mapping for a tetrahedron is obtained by blending the boundary faces and edges [7] as

$$\begin{aligned} x_i(\xi_j) = &(1-\xi_1)G_i(\xi') + (1-\xi_2)E_i(\xi') + (1-\xi_3)F_i(\xi') + (1-\xi_4)D_i(\xi') \\ &- (1-\xi_1-\xi_2)W_i(\xi') - (1-\xi_1-\xi_3)T_i(\xi') - (1-\xi_1-\xi_4)Q_i(\xi') \\ &- (1-\xi_2-\xi_3)S_i(\xi') - (1-\xi_2-\xi_4)R_i(\xi') - (1-\xi_3-\xi_4)P_i(\xi') \end{aligned} \tag{11}$$

where P, Q, R, S, T, W and D, E, F, G define the shapes of the bounding edges and faces respectively with ξ' defining the face(edge) parameters normalized such that $\sum_{k=0}^{nvert} \xi_k = 1$. For example, for face $\xi_4 = 0$, $\xi_1' = \frac{\xi_1}{(\xi_1+\xi_3+\xi_3)}, \xi_2' = \frac{\xi_2}{(\xi_1+\xi_3+\xi_3)}, \xi_3' = \frac{\xi_3}{(\xi_1+\xi_3+\xi_3)}$ and for edge $\xi_2 = \xi_3 = 0$, $\xi_1' = \frac{\xi_1}{(\xi_1+\xi_2)}, \xi_2' = \frac{\xi_2}{(\xi_1+\xi_2)}$. For mesh regions with no bounding edges or faces classified on a curved model boundary, the mapping is obtained by linear interpolation of vertex coordinates

$$x_i(\xi_j) = x_i(1,0,0,0)\xi_1 + x_i(0,1,0,0)\xi_2 + x_i(0,0,1,0)\xi_3 + x_i(0,0,0,1)\xi_4 \tag{12}$$

4 Application Example

Two numerical examples based on the solution of Poisson's equation are presented. The first one compares the efficiency of the decomposed shape functions for a variable p mesh presented here with explicitly evaluated shape functions for a uniform p code. The second one compares solution convergence using cubic Lagrangian interpolation and mapping conforming to exact geometry for curved domains.

First consider the solution of the problem shown in Figure 7(a). For a uniform 8 by 8 mesh with a total of 128 triangular elements with uniform p=8, the use of decomposed shape functions increases element matrix computation time and the total solution time by approximately 15% and 0.7% respectively. The increase is primarily due to the local searching to determine the local indices for lower order mesh entities in the element topological hierarchy. However, for the same *h* mesh, a series of runs with *p* varying linearly from *p*=1 at *x*=0 to *p*=8 at *x*=1 through *p*=5 at *x*=0 to *p*=8 at *x*=1, the decomposed shape function required an average of one-half the time required by the explicitly evaluated shape functions for element matrix computation using uniform *p*=8. The total solution time required using the decomposed shape functions was one-quater of the total solution time required using explicitly evaluated shape functions with uniform *p*=8. The decrease is primarily due to two reasons. First, for the variable *p* mesh, integration over low order elements require fewer points compared to uniform *p* mesh where all elements must be integrated based on *p*=8. Secondly, variable *p* meshes require fewer global *dof* to achieve the same level of error compared to uniform *p* meshes. The decomposed shape functions directly allow variable *p*-order meshes which can provide significant computational advantages.

Next consider the problem in Figure 7(b). The exact error in the finite element solution, $e = u - u_{hp}$, is measured in the natural norm for this problem [12], $a(e,e) = a(u,u) - a(u_{hp}, u_{hp})$, with $a(u,u) = \int_{\Omega} (u_x^2 + u_y^2) d\Omega$ defining the square of energy in the solution and in the usual L_2 norm [12] defined by $||e||^2_{L_2} = \int_{\Omega} (u - u_{hp})^2 d\Omega$. The error versus the total number of degrees of freedom plotted on the log-log scale (Figure 8(a)) shows the theoretical slopes [12] of -1 and $-\frac{1}{2}$ in L_2 and energy norm respectively, for uniform *h*-refinement of one quarter of the domain. Figure 8(b) plots the error convergence for uniform *p* refinement using one curvilinear triangle to model a quarter of the domain. To account for non constant Jacobian, an integration scheme of order *2p+1* was used for *p* order elements[4] to ensure accuracy of integration using Cubic Lagrangian interpolation of element geometry. Solutions obtained using cubic Lagrangian approximation for geometry results in a significant loss of convergence compared to those obtained with integration done based on CAD geometry for *p*>3. The loss in convergence rate is due to a poor approximation of the true domain geometry by Cubic Lagrangian interpolation. It cannot be attributed to integration errors because in that case the errors should have been larger with the CAD geometry mapping because of the use of rational blends.

5 Closing Remarks

This paper presented flexible and efficient hierarchic shape functions tied to mesh topological hierarchy which are crucial to achieve *p*-adaptivity in a variable *p* environment. The paper also presented a technique to make the domain geometry, as defined within a geometric modeler, available to various components of the *hp* adaptive framework. Examples presented corroborate earlier findings about the importance of domain geometry for *p* refinement using coarse elements [4].

6 Acknowledgements

The authors would like to acknowledge the support of the Office of Naval Research under grants N00014-94-1-0962 and N00014-C-6023.

Bibliography

[1] 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.

[2] 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.*, 26:2607-2622, 1988.

[4] For element of order *p*, the highest polynomial order of stiffness terms is *2p-2*.

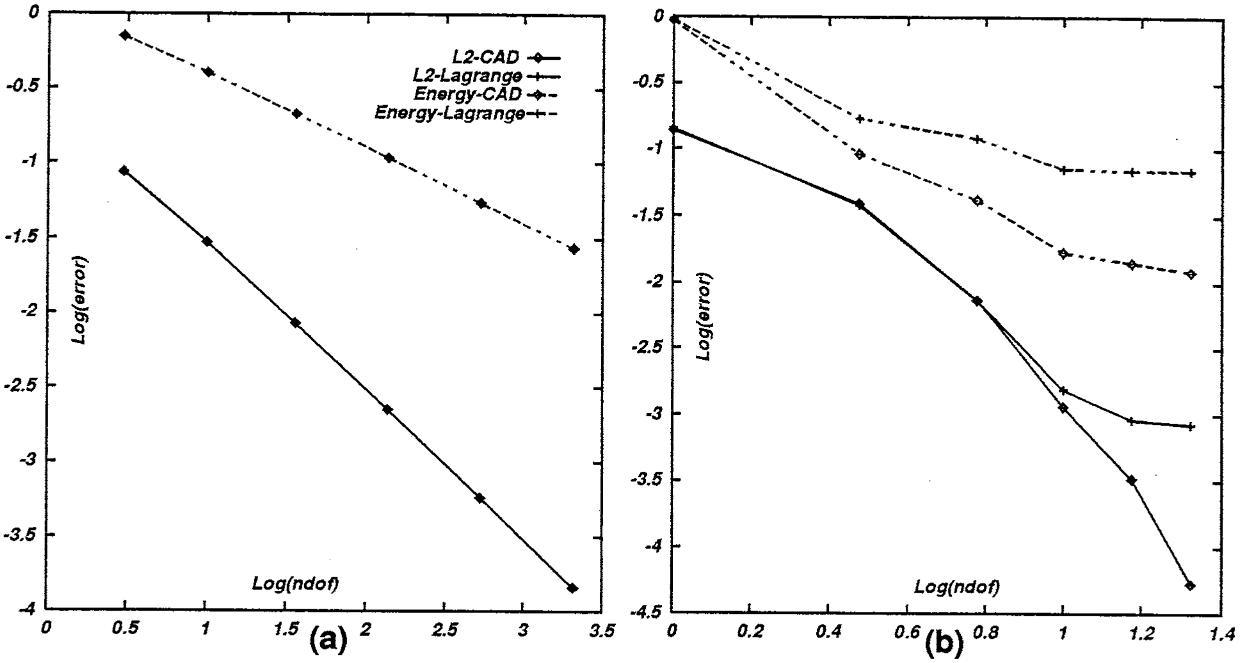

Figure 8. Error convergence for example 2: (a) uniform h refinement with p=1 and (b) uniform p refinement with 1 element.

[3] H. L. de Cougny and M. S. Shephard. Parallel mesh adaptation by local mesh modification. Technical Report 21-1995, Scientific Computation Research Center, Rensselaer Polytechnic Institute, Troy, NY 12180-3590, 1995. submitted to Computer Methods in Applied Mechanics and Engineering.

[4] L. Demkowicz, A. Karafiat, and J. T. Oden. Solution of elastic scattering problems in linear acoustics using h-p boundary element methods. *Computer Methods in Applied Mechanics and Engineering*, 101:251–282, 1991.

[5] W. J. Gordon and C. A. Hall. Construction of curvilinear co-ordinate systems and applications to mesh generation. *Int. J. Numer. Meth. Engng.*, 7:461–477, 1973.

[6] R. B. Haber, M. S. Shephard, J. F. Abel, R. H. Gallagher, and D. P. Greenberg. A generalized two-dimensional finite element preprocessor utilizing discrete transfinite mappings. *Int. J. Numer. Meth. Engng.*, 17:1015–1044, 1981.

[7] C. Lacombe and C. Bedard. Face-apex projectors for the interpolation function of a general tetrahedral mid-edge finite element. *Comp. Meth. Appl. Mech. Engng.*, 68:177–188, 1988.

[8] 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.

[9] M. S. Shephard and S. Dey. A two level heirarchy for defining p-version shape functions for meshes with variable p-order. Technical Report In preparation for submission, Scientific Computation Research Center, Rensselaer Polytechnic Institute, Troy, NY 12180-3590, 1996.

[10] M. S. Shephard, S. Dey, and M. K. Georges. Automatic meshing of curved three-dimensional domains: Curving finite elements and curvature-based mesh control. 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*, pages 67–98. Springer-Verlag, New York, 1995.

[11] M. S. Shephard and M. K. Georges. Reliability of automatic 3-D mesh generation. *Comp. Meth. Appl. Mech. Engng.*, 101:443–462, 1992.

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