The isoparametric maps x_1 (2.2) and x_2, (2.4) are isomorphic. Any pair of parameters (ξ, η) on a boundary of a biunit is uniquely mapped to a point on the corresponding side of the chord element and a point on the corresponding curved side of the eight-node element. The composition of these two maps $x_2 \circ x_1^{-1}$ is an isomorphism which means that there exists a unique map between those two elements.

The chord element defines geometrical properties of the corresponding quadrilateral with curved boundaries.

References

[1] J.T. Oden, A general theory of finite elements. I. Topological considerations, Int. J. Numer. Methods Engrg. 1 (1969) 205–221.
[2] S. Kobayashi and K. Nomizu, Foundations of Differential Geometry (Interscience Publishers, New York, 1963).
[3] Y. Choquet Bruhat, C.de Witt-Morette and M. Dillard-Bleick, Analysis, Manifolds and Physics (North-Holland, New York, 1977).
[4] J.C. Simo, A finite strain beam formulation. The three dimensional dynamic problem. Part I, Comput. Methods Appl. Mech. Engrg. 49 (1985) 55–70.
[5] C. Pacoste and A. Eriksson, Beam elements in instability problems, KTH, Stockholm, Tech. Report, 1996: 8, Struc. Mech.
[6] J.C. Simo and L. Vu-Quoc, A three dimensional finite strain rod model. Part II: Computational aspects, Comput. Methods Appl. Mech. Engrg. 58 (1986) 76–116.
[7] J.C. Simo and L. Vu-Quoc, On the dynamics in space of rods undergoing large motions—A geometrically exact approach, Comput. Methods Appl. Mech. Eng. 66 (1988) 121–161.
[8] J.C. Simo and D.D. Fox, On a stress resultant geometrically exact shell model. Part I: formulation and optimal parametrization, Comput. Methods Appl. Mech. Eng. 72 (1989) 267–304.
[9] J.C. Simo, D.D. Fox and M.S. Rifai, On a stress resultant geometrically exact shell model. Part II: The linear theory; computational aspects, Comput. Methods Appl. Mech. Engrg. 73 (1989) 53–92.
[10] J.C. Simo, A framework for finite strain elastoplasticity based on maximum plastic dissipation and the multiplicative dcomposition: Part I. Continuum formulation, Comput. Methods Appl. Mech. Engrg. 66 (1988) 199–219.
[11] A. Ibrahimbegovic, Equivalent Eulerian and Lagrangian descriptions of finite deformation elastoplasticity in principal axes, LSC, Lausanne, Internal Report, 93/16, Inst. Stat. Struc.
[12] A. Ibrahimbegovic', Finite elastoplastic deformations of space-curved membranes, LSC, Lausanne, Internal Report, 94/1, Inst. Stat. Struc.
[13] K.Y. Yuan and T.T. Pian, The reference coordinates and distortion measures for quadrilateral hybrid stress element, Comput. Mech. 14 (1994) 311–322).
[14] K.Y. Yuan, Y.S. Huang and T.T. Pian, Inverse mapping and distortion measures for quadrilaterals with curved boundaries, Int. J. Numer. Methods Engrg. 37 (1994) 861–875.
[15] K.Y. Yuan, Y.S. Huang, H.T. Yang and T.T. Pian, The inverse mapping and distortion measures for 8-node hexahedral isoparametric elements, Comput. Mech. 14 (1994) 189–199.
[16] N. Lautersztajn and A. Samuelsson, Distortion measures and inverse mapping for isoparametric 8-node plane finite elements with curved boundaries, Com. Numer. Methods Engrg., to be submitted.
[17] R. Abraham and J.E. Marsden, Foundations of Mechanics, 2nd edition (The Benjamin/Cumming Publishing Company, Inc., 1978).
[18] J.E. Marsden and T.J.R. Hughes, Mathematical Foundations of Elasticity, (Prentice-Hall, Inc., Englewood Cliffs, NJ, 1983).

ELSEVIER

Comput. Methods Appl. Mech. Engrg. 150 (1997) 39–55

Computer methods in applied mechanics and engineering

Geometry representation issues associated with p-version finite element computations

Saikat Dey*, Mark S. Shephard, Joseph E. Flaherty

Scientific Computation Research Center, Rensselaer Polytechnic Institute, Troy, NY 12180-3590, USA

Dedicated to Prof. J.T. Oden on the occasion of his 60th birthday

Abstract

This paper addresses issues related to accurate geometry representation for p-version finite elements on curved three-dimensional domains. Specific options to account for domain geometry information during element-level computation are identified. Accuracy requirements on the geometry related approximations to preserve the optimal rate of finite element error convergence for second-order elliptic boundary value problems are given. An element geometric mapping scheme based on blending the exact shape of the domain boundary is described that can either be used directly during element integrations, or used to construct element-level geometric approximations of required accuracy. Smoothness issues of the rational blends on simplex topologies are discussed and a numerical example based on the solution of *Poisson's equation* in three dimensions is presented to illustrate the impact of the rational blends on the optimal rate of finite element error convergence.

0. Notation

Domain representation

Ω_v	Domain associated with the model $v, v = G, M$ where G signifies the geometric model and M signifies the mesh model.
Γ_v	Boundary of Ω_v defined as $\Gamma_v = \partial(\Omega_v)$.
$\overline{\Omega}_v$	Closure of domain associated with the model $v, v = G, M$ given by $(\Omega 4_v \cup \partial(\Omega_v))$.
M_i^d	Mesh topological entity i of dimension d, $d = 0$ is a vertex, $d = 1$ is an edge, $d = 2$ is a face, $d = 3$ is a region.
G_i^d	Geometric model topological entity i of dimension d.
$\partial(M_i^d)$	Boundary of mesh topological entity M_i^d.
$\partial(G_i^d)$	Boundary of geometric model topological entity G_i^d.
$\overline{G}_i^d$	Closure of geometric model topological entity defined as $(G_i^d \cup \partial(G_i^d))$.
$\overline{M}_i^d$	Closure of mesh topological entity defined as $(M_i^d \cup \partial(M_i^d))$.
Ω_i^e	Domain of a finite element associated with a M_i^e.
Γ_i^e	Boundary of a finite element domain defined as $\Gamma_i^e = \partial(\Omega_i^e)$.
$\overline{\Omega}_i^e$	Closure of a finite element domain defined as $(\Omega_i^e \cup \Gamma_i^e)$.
$\sqsubset$	Classification symbol used to indicate the association of one or more entities from the mesh model M with the geometric model, G.

* Corresponding author.

0045-7825/97/$17.00 © 1997 Elsevier Science S.A. All rights reserved
PII S0045-7825(97)00103-5

Coordinate systems

ξ_j jth component of the parametric coordinates associated with M_i^d on Ω_i^e.
ξ_j jth component of the parametric coordinates associated with G_i^d.
x_j jth component of the coordinate system of the partial differential equation.

Finite element approximations

$u^{(h,p)}$ hp finite element approximation of unknown solution u using a mesh of size h and shape functions of polynomial order p.
N_i ith element shape function.
H^m mth *Sobolev* space of functions.
$\|u\|_m$ mth *Sobolev* norm of u.
$\|u\|_e$ *Energy* norm of u.

1. Introduction

The advantages of p- and hp-adaptive finite element computational strategies have been well established [4,18,26,35]. For properly designed meshes, p- and hp-adaptive schemes are capable of delivering exponential rates of convergence of the finite element solution error for domains with planar [3,18] and piecewise smooth curvilinear boundaries [2]. Efficient computer implementation of p- and hp-version finite elements on serial and parallel architecture has been the focus of much research [11,12,16,25,30]. Issues related to element level computations using variable order hp-approximations on curved domains can be summarized as:

(1) Specification and evaluation of variable order shape functions.
(2) Accounting for geometric approximations on elements that often cover large portions of curved domain.
(3) Efficient and effective evaluation of integrals required to compute element matrices and vectors.

Recent work [13,31] addressed the first issue and presented a general and efficient framework for the specification and evaluation of variable p-order shape functions on conforming, unstructured meshes consisting of mixed topology elements. This paper focuses on the second issue. The third issue is addressed in a forthcoming paper [14].

Realization of exponential convergence possible with hp-adaptive finite element methods, requires that the approximation of the domain geometry be carefully controlled. Approximate geometry representation leads to inexact representation of boundary and initial conditions and to the approximate evaluation of element level integrals. In order to sustain the exponential rate of convergence with hp-refinement, these perturbation errors must also converge exponentially. Thus, in a p-adaptive environment, the accuracy of geometry approximations and that of geometry related items must be related to p. This paper presents a geometric mapping scheme based on blending the shapes of the domain boundary exactly as described within a geometric modeling system. The scheme is well suited for a p-adaptive environment for two reasons. First, it ensures that the resulting curvilinear finite element discretization (mesh) conforms exactly to the true geometry of the problem domain, thus, eliminating any geometric approximation. Second, unlike schemes that approximate the domain geometry, there is no need to improve the accuracy of geometric approximation as p is enriched.

All the computational constructs presented here rely on a mesh topological hierarchy [6] that relates mesh entities directly to specific model topological entities. This provides direct access to the shape information of the problem domain as defined within a geometric modelling system and makes it possible to provide geometric approximation with the accuracy needed by the specific problem at hand.

A broad outline of the paper follows. The next section identifies specific element level operations that require domain geometry information. Section 3 discusses possible approaches to approximate element level computations on curved domains. Section 4 discusses accuracy requirements that geometry related approximations must satisfy to preserve the optimal rate of convergence of finite element error in a p-adaptive environment. Section 5 describes a technique to obtain element geometric mapping that is exact with respect to the curved domain boundary, as represented within a geometric modelling system. Issues related to the limited smoothness of the rational blends for simplex elements are discussed in Section 6. Numerical example based on the solution of Poisson's equation is presented to illustrate the impact of the rational blends on the optimal convergence rate of

the finite element discretization error and the influence of the efficiency of the geometry modelling system on the analysis time.

2. Domain geometry representation issues for p-adaptive computations

For a typical geometry based specification of a boundary value problem discretized using finite elements (see Fig. 1), the need for a mappigng $\boldsymbol{x}$ arises because of two specific operations involved in the finite element method:

(1) Evaluation of the element level integrals associated with the *weak* form of the problem defined over the domain of the curvilinear element, Ω_i^e, or its boundary, Γ_i^e, and
(2) Enforcement of essential boundary condition(s) specified on curved portion(s) of the boundary of Ω_G.

For $\Omega_G \subset \Re^n$, components of the mapping are given by $x_i(\xi_j)$; $i = 1, 2, \ldots, n$, where the parametric coordinates ξ_j depend on the topology and the dimension of the mesh entity as depicted in Table 1.

In the first operation, $\boldsymbol{x}(\boldsymbol{\xi})$ is used to transform the integrals to the parametric coordinate system where integration is performed. For example, the integral transformations for typical stiffness and mass terms for element k are given by [21,34,24,27]

$$\int_{\Omega_k^e} \frac{\partial N_i}{\partial \boldsymbol{x}} \frac{\partial N_j}{\partial \boldsymbol{x}} \, \mathrm{d}\boldsymbol{x} = \int_{\Omega_k^e} \frac{\partial N_i}{\partial \boldsymbol{\xi}} \left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{x}}\right) \frac{\partial N_j}{\partial \boldsymbol{\xi}} \left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{x}}\right) J(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi} \tag{1}$$

and

$$\int_{\Omega_k^e} N_i(\boldsymbol{x}) N_j(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = \int_{\Omega_k^e} N_i(\boldsymbol{x}(\boldsymbol{\xi})) N_j(\boldsymbol{x}(\boldsymbol{\xi})) J(\boldsymbol{\xi}) \, \mathrm{d}\boldsymbol{\xi} \,, \tag{2}$$

respectively, with $J(\boldsymbol{\xi}) = |\partial x_i / \partial \xi_j|$. Similarly, integrals resulting from the weak enforcement of natural boundary condition(s), $h_n(\boldsymbol{x})$, are given by [21, 34, 24, 27]

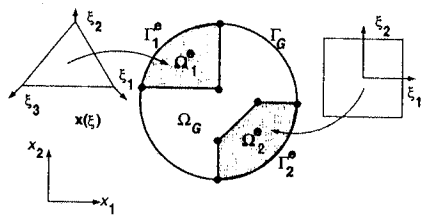

Fig. 1. Mesh entity geometry mapping.

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		$\xi_1, \xi_2, \xi_3 \in [0, 1]$, $\xi_1 + \xi_2 + \xi_3 = 1$, $\xi_4 = 1$, $\xi_4[-1, 1]$
Pyramid		$\xi_2 \in [-1, 1]$, $\frac{-(1-\xi_2)}{2} \leqslant \xi_1, \xi_3 \leqslant \frac{(1-\xi_2)}{2}$

$$\int_{\Gamma_k^e} N_i(x)h_n(x)\,dx = \int_{\Gamma_k^e} N_i(x(\xi))h_n(x(\xi))J(\xi)\,d\xi \tag{3}$$

where Γ_k^e defines the portion of the element boundary where natural boundary conditions are specified.

The second operation is associated with the satisfaction of the essential boundary condition(s) by the finite element solution, i.e.

$$u^{(h,p)}(x) = g(x), \quad \forall x \in \Gamma_k^e, \tag{4}$$

where Γ_k^e defines the portion of the element boundary where $g(x)$ is prescribed. The most straightforward way of satisfying Eq. (4) is to interpolate $g(x)$ on Γ_k^e using the finite element basis as

$$\begin{aligned} &g(x(\xi)) \approx g^*(x(\xi)) = \sum_{l=1}^{n_p} a_l N_l(\xi) \\ &g^*(x(\xi^{(l)})) = g(x(\xi^{(l)})); \quad l = 1, \ldots, n_p, \end{aligned} \tag{5}$$

where $\{\xi^{(l)}\}$ and $\{N_l\}$ respectively define the sets of n_p interpolation points and basis functions for pth order interpolation. The solution of the interpolation problem to determine a_l requires the values of $g(x)$ at the interpolation points $\{\xi^{(l)}\}$, $l = 1, \ldots, n_p$, which requires the pointwise evaluation of the mapping function $x(\xi^{(l)})$.

The next key issue that must be addressed is the accuracy with which geometry must be accounted for during element level computations within a *hp*-adaptive environment. To this end, one needs to first examine the possible option(s) to account for geometry during the element level computations and then study the impact of any approximation(s) made on the convergence properties of *hp*-adaptive finite element approximations.

3. Approximate element level computations

If the element level integral(s) are represented abstractly as

$$I_{\Omega^e} = \int_{\Omega^e} \kappa_{\Omega^e}(x(\xi))\,d\xi, \qquad I_{\Gamma^e} = \int_{\Gamma^e} \kappa_{\Gamma^e}(x(\xi))\,d\xi \tag{6}$$

where κ_{Γ^e} and κ_{Ω^e} respectively represent the integrands associated with interior and the boundary of the element domain, then approximations can be introduced at one or more following basic functional levels:

(1) Approximation of $x(\xi)$.
(2) Approximation of κ_{Ω^e} and κ_{Γ^e}.
(3) Approximation integration method.

Useful operators to compute I_{Ω^e} and I_{Γ^e} denoted by $\int_{\Omega^e}^{*_i}$ and $\int_{\Gamma^e}^{*_i}$, respectively, that use combinations of the basic approximations possible can be stated as:

(1) Exact geometry representation followed by approximate integration yielding

$$\begin{aligned} &\int_{\Omega^e}^{*_1} \kappa_{\Omega^e}\,d\xi \approx \sum_{l=1}^{n_g} \{\kappa_{\Omega^e}(x(\xi^{(l)}))\}w^{(l)} \\ &\int_{\Gamma^e}^{*_1} \kappa_{\Gamma^e}\,d\xi \approx \sum_{l=1}^{n_g} \{\kappa_{\Gamma^e}(x(\xi^{(l)}))\}w^{(l)}. \end{aligned} \tag{7}$$

Using this approach, the integral from Eq. (1) can be approximated as

$$\begin{aligned} \int_{\Omega^e} \frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x}\,dx &= \int_{\Omega^e}^{*_1} \frac{\partial N_i}{\partial \xi}\left(\frac{\partial \xi}{\partial x}\right)\frac{\partial N_j}{\partial \xi}\left(\frac{\partial \xi}{\partial x}\right)J(\xi)\,d\xi \\ &\approx \sum_{l=1}^{n_g}\left\{\left[\frac{\partial N_i}{\partial \xi}(\xi^{(l)})\frac{\partial \xi}{\partial x}(\xi^{(l)})\right]\left[\frac{\partial N_j}{\partial \xi}(\xi^{(l)})\frac{\partial \xi}{\partial x}(\xi^{(l)})\right]J(\xi^{(l)})\right\}w^{(l)} \end{aligned} \tag{8}$$

with

$$J(\xi^{(l)}) = \left|\frac{\partial x_m}{\partial \xi_n}\right|(\xi^{(l)}).$$

Assuming that an exact representation of element geometry $x(\xi)$ is available, the errors with this approach are entirely the result of approximate numerical integration.

(2) Approximate geometry representation, $x(\xi) \approx x^*(\xi)$, followed by approximate numerical evaluation of the resulting integrand to yield

$$\begin{aligned} &\int_{\Omega^e}^{*_2} \kappa_{\Omega^e}\,d\xi \approx \sum_{l=1}^{n_g} \{\kappa_{\Omega^e}(x^*(\xi^{(l)}))\}w^{(l)}, \\ &\int_{\Gamma^e}^{*_2} \kappa_{\Gamma^e}\,d\xi \approx \sum_{l=1}^{n_g} \{\kappa_{\Gamma^e}(x^*(\xi^{(l)}))\}w^{(l)}, \end{aligned} \tag{9}$$

where $\{\xi^{(l)}\}$ and $\{w^{(l)}\}$ define the sets of n_g coordinates and weights associated with the numerical integration scheme [1] chosen. For example, the approximate evaluation of the integral from Eq. (1) would be given by

$$\begin{aligned} \int_{\Omega^e} \frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x}\,dx &\approx \int_{\Omega^e}^{*_2} \frac{\partial N_i}{\partial \xi}\left(\frac{\partial \xi}{\partial x^*}\right)\frac{\partial N_j}{\partial \xi}\left(\frac{\partial \xi}{\partial x^*}\right)J(\xi)\,d\xi \\ &\approx \sum_{l=1}^{n_g}\left\{\left[\frac{\partial N_i}{\partial \xi}(\xi^{(l)})\frac{\partial \xi}{\partial x^*}(\xi^{(l)})\right]\left[\frac{\partial N_j}{\partial \xi}(\xi^{(l)})\frac{\partial \xi}{\partial x^*}(\xi^{(l)})\right]J(\xi^{(l)})\right\}w^{(l)} \end{aligned} \tag{10}$$

with

$$J(\xi^{(l)}) = \left|\frac{\partial x_m^*}{\partial \xi_n}\right|(\xi^{(l)}).$$

This is the commonly used technique in the *isoparametric h*-version finite element method [21,37] where $x^*(\xi)$ is constructed by interpolation identical to that used to construct the unknown solution u. There are two distinct sources of error in this process; the error due to approximation of the geometry and the error due to the approximate numerical computation of the resulting integral.

(3) Direct approximation of the integrand(s) followed by exact integration to yield

$$\int_{\Omega^e}^{*_3} \kappa_{\Omega^e}\,d\xi \approx \int_{\Omega^e} \kappa_{\Omega^e}^*(\xi)\,d\xi, \qquad \int_{\Gamma^e}^{*_3} \kappa_{\Gamma^e}\,d\xi \approx \int_{\Gamma^e} \kappa_{\Gamma^e}^*(\xi)\,d\xi. \tag{11}$$

It is assumed that $\kappa_{\Omega^e}^*(\xi)$ and $\kappa_{\Gamma^e}^*(\xi)$ can be integrated exactly over their respective domains using either closed form formulae or numerical schemes [1]. One possible approach of this type is the construction of $\kappa_{\Omega^e}^*$ and $\kappa_{\Gamma^e}^*$ as linear combinations of polynomials [20], $\phi_{\Omega^e}(\xi)$ and $\phi_{\Gamma^e}(\xi)$, defined over Ω^e and Γ^e, respectively, to obtain

$$\kappa_{\Omega^e}^*(\xi) = \sum_{l=1}^{n} a_{\Omega^e}^{(l)}\phi_{\Omega^e}^{(l)}(\xi), \qquad \kappa_{\Gamma^e}^*(\xi) = \sum_{l=1}^{m} a_{\Gamma^e}^{(l)}\phi_{\Gamma^e}^{(l)}(\xi). \tag{12}$$

The number of polynomial terms used, denoted by n and m, depend on the accuracy of the approximation desired and the topology of Ω^e and Γ^e. The errors associated with this approach result entirely from the approximation of $\kappa_{\Omega^e}^*$ and $\kappa_{\Gamma^e}^*$.

For computational efficiency of the *hp*-adaptive method, accuracy of the approximation(s) should be the minimum required to maintain the rate of convergence of the discretization error. Doing the approximation(s) more accurately than is required does not improve the rate of convergence of the discretization error and hence is not an efficient use of computational resources. To determine the required accuracy of approximation(s), one must relate the rate of convergence of the approximation error(s) to that of the finite element discretization error.

4. Accuracy of approximate element level computations for *hp*-adaptive methods

If $u^{(h,p)}$ represents the *hp* finite element approximation of the exact solution u, then the discretization error can be bounded asymptotically as [3]

$$\|u - u^{(h,p)}\|_m \leqslant \frac{Ch^{\alpha(p,m,r)}\|u\|_r}{p^{r-1}} \tag{13}$$

where, C is a constant independent of u, h and p, where h represents the maximum element diameter, and p represents the maximum degree of complete polynomials used in the basis. For second-order elliptic boundary value problems, $\alpha = \min\{p+1-m, r-m\}$. If u is sufficiently smooth such that $r \geqslant p+1 \Rightarrow \alpha = p+1-m$, then the discretization error will converge exponentially with respect to p-refinement at a rate given by

$$\|u - u^{(h,p)}\|_m \leqslant \frac{Ch^{p+1-m}\|u\|_r}{p^{r-1}}. \tag{14}$$

4.1. Optimal function approximations

The goal of this section is to determine the correct order of approximation, q, in terms of p, such that the error in geometry approximation, $\boldsymbol{x} - \boldsymbol{x}^*$, and integrand approximations, $\kappa_{\Omega^e} - \kappa^*_{\Omega^e}$ and $\kappa_{\Gamma^e} - \kappa^*_{\Gamma^e}$, converge at least as fast as the finite element discretization error given by Eq. (14).

From basic approximation theory, the error in approximating a function $\theta(\boldsymbol{\xi})$ with $\theta^*(\boldsymbol{\xi})$, has the form [29]

$$\|\theta - \theta^*\| \leqslant (1+\lambda)\varphi \tag{15}$$

where φ is the *best approximation error* and λ is the Lebesgue constant of the approximation operator. Since finite element solutions are usually defined as linear combinations of polynomial functions in $H^m(\Omega)$ and their errors measured in appropriate *Sobolev* norms, the order q approximations of $\theta(\xi)$, $\xi \in \bar{\Omega}^e$, will be restricted to those constructed by linear combinations of polynomial functions, $\{\phi_i(\xi) | \phi_i \in H^{q+1}\}$, given by

$$\theta^{(q)}(\boldsymbol{\xi}) = \sum_{i=1}^{n_q} a_i \phi_i(\boldsymbol{\xi}) \tag{16}$$

where the number of polynomial basis functions used, n_q, depends on the topology and the dimension of Ω^e. The number of functions needed to construct a complete qth degree polynomial basis for various element topologies is given in Table 2.

For smooth functions ($\theta \in H^{q+1}$), φ is known to converge exponentially [29] and the overall convergence rate of $\theta - \theta^{(q)}$ is determined by the rate at which λ grows with respect to q [8,9]. When the interpolation points are equally spaced λ can grow exponentially fast causing a loss of exponential convergence of $\|\theta - \theta^{(q)}\|$, even

Table 2
Number of basis functions needed for complete polynomial approximations of degree q

Ω^e	n_q
Line	$q+1$
Triangle	$\frac{(q+1)(q+2)}{2}$
Quadrilateral	$(q+1)^2$
Tetrahedron	$\frac{(q+1)(q+2)(q+3)}{6}$
Hexahedron	$(q+1)^3$
Wedge	$\frac{(q+1)^2(q+2)}{2}$
Pyramid	$(q+1)^3$

when θ is smooth. However, for the non-uniformly spaced interpolation sets of Chen and Babuska [8,9], λ grows approximately as $O(\log(q))$ which ensures exponential convergence of $\|\theta - \theta^{(q)}\|$.

Using the above arguments, the specific form of Eq. (15) for polynomial interpolation of θ (Eq. (16)) is given by [27,24]

$$\|\theta - \theta^{(q)}\|_m \leqslant C_1 h^{q+1-m}\|\theta\|_r; \quad r \geqslant q+1, \tag{17}$$

which C_1 is independent of h and θ. C_1 is a function of q and its growth is controlled based on the choice of the interpolation set.

Since θ appears in the integrand, Eq. (17) is the error bound for the integrand approximation. The error bound for the integral approximation will be one order higher [27] given by

$$\left\| \int_{\Omega^e} \theta \, d\Omega - \int_{\Omega^e} \theta^{(q)} \, d\Omega \right\|_m \leqslant \bar{C}_1 h^{q+2-m}\|\theta\|_r; \quad r \geqslant q+1. \tag{18}$$

To maintain the rate of convergence of discretization error given by Eq. (14), Eq. (18) implies that

$$q + 2 - m \geqslant p + 1 - m \tag{19}$$

yielding

$$q \geqslant p - 1. \tag{20}$$

Note that if $r \geqslant p+1$, then Eq. (20) also defines the necessary and sufficient condition for approximation of κ_{Ω^e} and κ_{Γ^e} by Eq. (12) to preserve the optimal convergence rate of the discretization error.

4.2. Optimal numerical integration order

Accuracy requirements for optimal numerical integration in the *h*- and *p*-version of the finite element method have been investigated in detail in [34,10,24,22,5]. Optimal convergence of discretization error for second-order elliptic boundary value problems can be ensured by selecting a numerical integration scheme that accounts for the geometry related terms to order $p-1$ [10,27]. For example, for a given p, $\partial N/\partial \xi$ in Eq. (10) is of order $p-1$ and if the remaining geometry related terms from $\partial \xi_i/\partial x_j$ and $J(\xi)$ are approximated to order $p-1$ then the order of the complete integrand is given by $2(p-1)+(p-1) = 3(p-1)$ and the optimal number of integration points needed in n dimensions using a tensor product rule [1] is given by

$$n_g = \underbrace{\left(\left[\frac{3p-3+\beta}{2}\right]\right) \times \cdots \times \left(\left[\frac{3p-3+\beta}{2}\right]\right)}_{n \text{ times}} \tag{21}$$

where $\beta = 1$ and 2, respectively, for *Gauss–Legendre* and *Gauss–Lobatto* schemes [1].

5. Mesh geometry mapping for higher-order methods

Functions related to geometry that appear in finite element computations can be represented in general by $\partial^n \boldsymbol{x}/\partial \boldsymbol{\xi}^n$, $n \geqslant 0$, with the case of $n = 0$ defining $\boldsymbol{x}(\boldsymbol{\xi})$. The highest-order derivative(s) required depends upon the order of the partial differential equation. For the case of second-order elliptic equations only first-order derivatives are required. Since finite element computations require position and derivatives only at specific points, they can be constructed in a manner that allows for their evaluation at any $\boldsymbol{\xi} \in \bar{\Omega}^e$. For effective use with high-order finite elements, three issues associated with the approximate position mapping $\boldsymbol{x}^*(\boldsymbol{\xi})$, and derivative, $\partial \boldsymbol{x}^*/\partial \boldsymbol{\xi}$ become important:

(1) *Accuracy*: As shown in Section 4.1, $\boldsymbol{x}^*(\boldsymbol{\xi})$ must be accurate to order $p-1$ in order to preserve the optimal convergence rate of the discretization error.
(2) *Smoothness*: The optimal rate of convergence of the finite element error given by Eq. (14) cannot be achieved in practice if $\boldsymbol{x}^*(\boldsymbol{\xi})$ is not smooth enough.
(3) *Cost*: Since the required order of approximation and the number of pointwise evaluations both increase

with increasing p, the construction and the evaluation of the mapping and its derivatives must be done efficiently.

The exact mathematical representation of the shape of Γ_G, $\boldsymbol{x}(\boldsymbol{\zeta})$, is housed within a geometric modelling system in terms of local parametric coordinate systems of model topological entities given by $\boldsymbol{\zeta}$; therefore, a technique to relate $\boldsymbol{\xi}$ to $\boldsymbol{\zeta}$, such that the resulting $\boldsymbol{x}(\boldsymbol{\xi})$ is exact with respect to Γ_G when $\boldsymbol{\xi} \in \Gamma^e \subset \Gamma_G$, is required. Pointwise evaluation of $\boldsymbol{x}(\boldsymbol{\xi})$ can then be used during finite element computations directly, or used in the construction of polynomial approximations of $\boldsymbol{x}^*(\boldsymbol{\xi})$ which have the required accuracy.

5.1. Geometric mapping conforming exactly to Γ_G

This section describes a technique for obtaining $\boldsymbol{x}(\boldsymbol{\xi})$ which is exact with respect to Γ_G when $\boldsymbol{\xi} \in \Gamma^e \subset \Gamma_G$.

The association of mesh topological entities with respect to the topological entities of the geometric model, referred to as *classification* [32], is central to obtaining shape information for individual mesh entities. Fig. 2 depicts the geometry based mesh hierarchy underlying the procedures described in this paper.

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

$$\boldsymbol{x} = \boldsymbol{x}(\zeta_i(\xi_j)) \tag{22}$$

where ζ_i represents the parametric coordinate system associated with G_j^d. The derivative(s) of the mapping can be evaluated by application of the chain-rule,

$$\frac{\partial x}{\partial \xi} = \frac{\partial x}{\partial \zeta}\frac{\partial \zeta}{\partial \xi}.$$

Fig. 3 graphically depicts this mapping for a triangular and quadrilateral mesh face using the three coordinate systems $\boldsymbol{\xi}$, $\boldsymbol{\zeta}$, and $\boldsymbol{x}$. $\boldsymbol{x}(\boldsymbol{\zeta})$, $\partial \boldsymbol{x}/\partial \boldsymbol{\zeta}$ are queried on a pointwise basis from the geometric modeler.

The implicit assumption in this process, similar to the technique outlined in [7], is that the model entities have

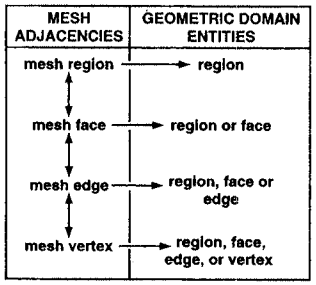

Fig. 2. Mesh topological adjacencies and classification information.

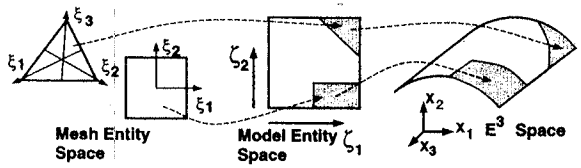

Fig. 3. Mapping concept illustrated for a boundary face.

an underlying continuous, nondegenerate parametric space. The availability of an underlying parameterization for specific G_i^d of Ω_G depends on the representation scheme used within the geometric modelling system.

Most geometric modelers do not have an explicit parametric coordinate system for G_i^3; instead, they are represented implicitly as the portion of three-dimensional space that is bounded by a closed set of G_i^2 [36,15,33]. For example, a solid block is represented implicitly by the volume enclosed by the six faces that bound it. Therefore, for mesh entities $M_i^d \sqsubset G_j^3$, the shape of the mesh entity can be constructed based on blending shapes of the model entities on which the bounding lower order mesh entities are classified. Since the blended shapes match the curved shape of Γ_G and they cover the interior of Ω_G, they introduce no geometric approximation.

The technique used to construct $\zeta_i(\xi_j)$ depends on the classification of a $M_k^{d_k}$ with respect to Ω_G. In addition, the construction of $\zeta_i(\xi_j)$ is also influenced by any *trimming* of the geometric model entities as used by the geometric modeler. A *trimmed* model entity is one which does not span the entire domain of the underlying parametric coordinate system. Fig. 4(a) depicts an *untrimmed* model face whereas, Fig. 4(b) depicts a *trimmed* model face.

5.1.1. Mesh edge mapping

For a M_i^1 on a $G_i^{d_j}$; $d_j = 1, 2$, (Fig. 5), in the absence of interference with any trimmed boundary entities and with acceptable resulting element shapes, $\zeta_i(\xi_j)$ can be given by a linear interpolation between ζ_i values at the vertices

$$\zeta_i(\xi) = \zeta_i(1,0)\xi_1 + \zeta_i(0,1)\xi_2\,, \qquad \xi_1 + \xi_2 = 1\,. \tag{23}$$

Interference of the linear interpolation of the edge geometry, in the parametric space of the model face, with trimming curve(s) may require the edge to be curved in the parametric space of the model face. For example, $\zeta_i(\xi_j)$ can be assigned general geometric shapes such as a quadratic form given by

$$\zeta_i(\xi_1) = a_i + b_i\xi_1 + c_i\xi_1^2 \tag{24}$$

or, a cubic form given by

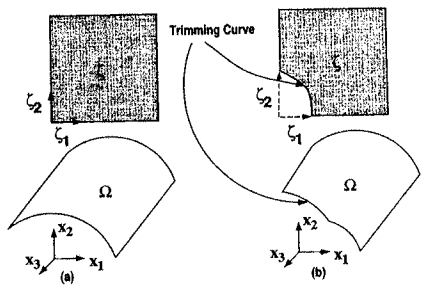

Fig. 4. (a) Untrimmed and (b) trimmed model faces.

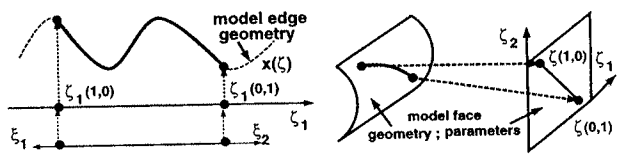

Fig. 5. Construction of mesh edge mapping.

$$\zeta_i(\xi_1) = a_i + b_i\xi_1 + c_i\xi_1^2 + d_i\xi_1^3 . \tag{25}$$

The coefficients a_i, b_i, c_i, d_i are determined by acceptability requirements on the shapes of finite element, in real space, that are bounded by the edge in question. Therefore, the determination of a_i, b_i, c_i, d_i must account for the effect of the mapping $x_i(\zeta_j)$ defining the shape of the model face in the geometric modeling system.

For a M_i^1 inside a G_j^3, the shape of the mesh edge can be represented as a straight line joining the vertices in the Cartesian coordinate system provided the element shapes are acceptable. Interference with trimming boundaries or excessive distortion may require a more general curvilinear representation of the shape of the mesh edges classified in the interior of a model region. For example, the geometry of an interior mesh edge can be assigned a general quadratic form $x_i = a_i + b_i\xi_1 + c_i\xi_1^2$ or a cubic form $x_i = a_i + b_i\xi_1 + c_i\xi_1^2 + d_i\xi_1^3$. The coefficients, a_i, b_i, c_i, d_i, are determined such that the shapes of the elements, in real space, bounded by the edge, are of acceptable quality.

5.1.2. Mesh face mapping

The construction of $\zeta_i(\xi_j)$ 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 Fig. 4(b). If M_i^2 is classified on an untrimmed model face as shown in Fig. 6, then $\zeta_i(\xi_j)$ can be constructed as a linear interpolation of the ζ_i values at the vertices of the face defined by

$$\zeta_i(\xi) = \zeta_i(1,0,0)\xi_1 + \zeta_i(0,1,0)\xi_2 + \zeta_i(0,0,1)\xi_3 . \tag{26}$$

If the shapes of the elements bounded by the face are unacceptable, then $\zeta_i(\xi_j)$ needs to be represented by a curvilinear representation in ζ-space such that the resulting element shapes, in real space, are acceptable.

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 leading to 'spills' or 'gaps' as shown in Fig. 7. In addition, a higher-order interpolation may be required if the resulting element shape is unacceptable.

In such cases the construction of $\zeta_i(\xi_j)$ must account for the curved shape of the boundary between B and C as shown in Fig. 8. Techniques based on the Boolean sum interpolation theory can be used to blend the boundary curves [17,19]. Using the scheme described in [19] yields

$$\begin{aligned}\zeta_i(\xi_j) = \frac{1}{2}\Bigg\{&\left(\frac{\xi_1}{1-\xi_2}\right)p_i(\xi_2) + \left(\frac{\xi_2}{1-\xi_1}\right)p_i(1-\xi_1) + \left(\frac{\xi_3}{1-\xi_2}\right)q_i(1-\zeta_2)\\ &+\left(\frac{\xi_2}{1-\xi_3}\right)q_i(\xi_3) + \left(\frac{\xi_1}{1-\xi_3}\right)r_i(1-\xi_3) + \left(\frac{\xi_3}{1-\xi_1}\right)r_i(\xi_1)\\ &- \xi_1\zeta_i(1,0,0) - \xi_2\zeta_i(0,1,0) - \xi_3\zeta_i(0,0,1)\Bigg\}\end{aligned} \tag{27}$$

where $p(t)$, $q(t)$, $r(t)$; $t \in [0,1]$ define the shape of the curves AB, BC and CD, respectively. The specific expressions for $p(t)$, $q(t)$ and $r(t)$ are derived based on the classification of the mesh edge as described in the previous section.

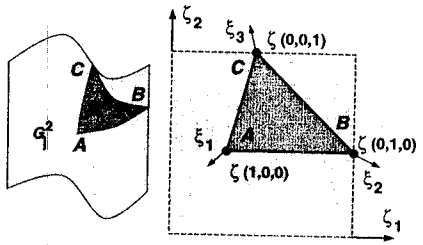

Fig. 6. Construction of $\zeta_i(\xi)$ for untrimmed model faces.

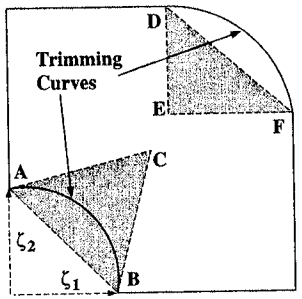

Fig. 7. 'Gaps' and 'spills' with linear interpolation on trimmed model faces.

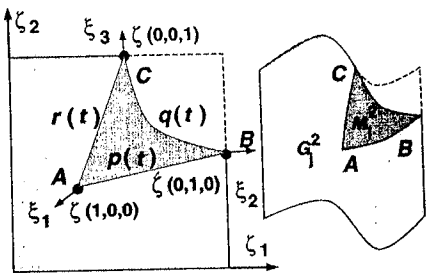

Fig. 8. Construction of $\zeta_i(\xi)$ for trimmed model faces.

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 $x(\xi)$ as

$$\begin{aligned}x_i(\xi_j) = \frac{1}{2}\Bigg\{&\left(\frac{\xi_1}{1-\xi_2}\right)P_i(\xi_2) + \left(\frac{\xi_2}{1-\xi_1}\right)P_i(1-\xi_1) + \left(\frac{\xi_3}{1-\xi_2}\right)Q_i(1-\xi_2) + \left(\frac{\xi_2}{1-\xi_3}\right)Q_i(\xi_3)\\ &+\left(\frac{\xi_1}{1-\xi_3}\right)R_i(1-\xi_3) + \left(\frac{\xi_3}{1-\xi_1}\right)R_i(\xi_1) - \xi_1 x_i(1,0,0) - \xi_2 x_i(0,1,0) - \xi_3 x_i(0,0,1)\Bigg\}\end{aligned} \tag{28}$$

where P, Q, R define the mapped position vectors for the shapes of the bounding mesh edges obtained as defined in the previous section. For mesh faces classified inside model regions with none of the bounding edges classified on a curved model boundary, $x(\xi)$ can be given by linear interpolation of the vertex coordinates

$$x_i(\xi_j) = x_i(1,0,0)\xi_1 + x_i(0,1,0)\xi_2 + x_i(0,0,1)\xi_3 . \tag{29}$$

However, interior mesh faces with no edges on curved boundary may in general be curved to improve mesh quality in terms of distortion measures. For example, quadratic or cubic Lagrangian geometry interpolation may be used with the position of the edge point(s) and the face point varied to obtain the desired curved shape for the face.

5.1.3. Mesh region mapping

The volume mapping for a tetrahedron is obtained by blending the boundary faces and edges [23] as

$$\begin{aligned}x_i(\xi_j) = &(1-\xi_1)G_i(\xi') + (1-\xi_2)E_i(\xi') + (1-\xi_3)F_i(\xi') + (1-\xi_4)D_i(\xi')\\ &-(1-\xi_1-\xi_2)W_i(\xi') - (1-\xi_1-\xi_3)T_i(\xi') - (1-\xi_1-\xi_4)Q_i(\xi')\\ &-(1-\xi_2-\xi_3)S_i(\xi') - (1-\xi_2-\xi_4)R_i(\xi') - (1-\xi_3-\xi_4)P_i(\xi')\\ &+\xi_1 x_i(1,0,0,0) + \xi_2 x_i(0,1,0,0) + \xi_3 x_i(0,0,1,0) + \xi_4 x_i(0,0,0,1)\end{aligned} \tag{30}$$

where P, Q, R, S, T, W and D, E, F, G represent the position vectors defining the shapes of the bounding edges and faces, respectively. The specific shapes of the bounding edges and faces are obtained by mapping process described in Sections 5.1.1 and 5.1.2. ξ' defines the face (edge) parametric coordinates normalized such that $\sum_{k=0}^{n_v}\xi'_k = 1$ where n_v is the number of mesh vertices bounding the face (edge). For example, for face $\xi_4 = 0$, $\xi'_1 = \xi_1/(\xi_1+\xi_3+\xi_3)$, $\xi'_2 = \xi_2/(\xi_1+\xi_3+\xi_3)$, $\xi'_3 = \xi_3/(\xi_1+\xi_3+\xi_3)$ and for edge $\xi_2 = \xi_3 = 0$, $\xi'_1 = \xi_1/(\xi_1+\xi_2)$, $\xi'_2 = \xi_2/(\xi_1+\xi_2)$. For mesh regions with no bounding edges or faces classified on a curved model boundary, the mapping can be obtained by linear interpolation of vertex coordinates

$$x_i(\xi_j) = x_i(1,0,0,0)\xi_1 + x_i(0,1,0,0)\xi_2 + x_i(0,0,1,0)\xi_3 + x_i(0,0,0,1)\xi_4 \tag{31}$$

However, interference with other curved mesh entities in the neighborhood may require curving one or more boundary entities of an interior mesh region. A simple and straightforward option is to use quadratic of cubic Lagrangian interpolation to define the geometry of the curved edges and faces.

5.2. Approximate geometry mapping

The pointwise exact $\boldsymbol{x}(\boldsymbol{\xi})$ can be used to construct $\boldsymbol{x}^*(\boldsymbol{\xi})$ as a linear combination of polynomial basis functions [29,20] defined over $\overline{\Omega}^e$, $\phi_k(\boldsymbol{\xi})$, to give

$$x_i^*(\boldsymbol{\xi}) = \sum_{k=1}^{n_q} \phi_k(\boldsymbol{\xi}) b_i^{(k)} . \tag{32}$$

In the *isoparametric* form of Eq. (32), widely used for low order h-version finite elements [10,34], $q = p$ and $\phi_k = N_k$, where N_k represents the shape functions used to construct the unknown solution u. The required derivatives $\partial x^*/\partial \xi$ are obtained by differentiating Eq. (32) with respect to ξ

$$\frac{\partial x_i^*}{\partial \xi_j} = \sum_{k=1}^{n_q} b_i^{(k)} \frac{\partial \phi_k}{\partial \xi_j} . \tag{33}$$

6. Smoothness of blended geometric mappings

The linear blending functions for simplices are only C^0. To see this, it suffices to examine the continuity of the rational terms from any one face and an edge. Consider for example the blending contribution of face defined between vertices 2, 3 and 4 from Eq. (30) given by

$$x_i(\xi) = (1 - \xi_1) G_i(\xi_1', \xi_2') \tag{34}$$

where

$$\xi_1' = \frac{\xi_2}{\xi_2 + \xi_3 + \xi_4} = \frac{\xi_2}{1 - \xi_1}, \qquad \xi_2' = \frac{\xi_2}{\xi_2 + \xi_3 + \xi_4} = \frac{\xi_3}{1 - \xi_1} . \tag{35}$$

It can be shown by the geometrical construct in Fig. 9(a) that ξ_1' and ξ_2' exist at $\xi_1 = 1$ and correspond to the projection to the centroid of the face implying $\xi_1' = \xi_2' = \frac{1}{3}$. Furthermore, from Eq. (34) it can be seen $G_i(\xi_1', \xi_2')$ contributes nothing at $\xi_1 = 1$ since the multiplying factor $1 - \xi_1$ vanishes. Now consider the edge blend contribution from Eq. (30) for the edge between vertices 1 and 4 given by

$$x_i(\xi) = (1 - \xi_2 - \xi_3) S_i(\xi') \tag{36}$$

where the independent edge coordinate $\xi_1' = \xi_1/(\xi_1 + \xi_2) = \xi_1/(1 - \xi_2 - \xi_3)$. Once again from Fig. 9(b) it is clear that $\xi_2 = \xi_3 = \frac{1}{2}$ corresponds to the midpoint of the edge and hence $\xi_1' = \frac{1}{2}$. Furthermore, at $\xi_2 = \xi_3 = \frac{1}{2}$ the multiplying term $1 - \xi_2 - \xi_3 = 0$, hence S_i contributes zero. This analysis does show that the linear blends are at least C^0. However, the analysis of Section 4.1 assumes that $\boldsymbol{x} \in H^{p+1}$. This implies that for use in p-adaptive environment, continuity of higher-order derivatives of the rational blend terms must also be examined.

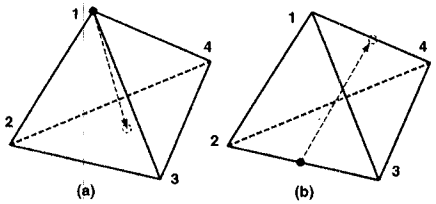

Fig. 9. Coordinate projection for blending.

Differentiating Eq. (34) yields

$$\frac{\partial x_i}{\partial \xi_j} = \frac{\partial (1 - \xi_1)}{\partial \xi_j} G_i + (1 - \xi_1) \frac{\partial G_i}{\partial \xi_k'} \frac{\partial \xi_k'}{\partial \xi_j} . \tag{37}$$

Expanding all the partial derivatives yields

$$\frac{\partial x_i}{\partial \xi_1} = -G_i + \left(\frac{1}{1 - \xi_1}\right)\left\{\xi_2 \frac{\partial G_i}{\partial \xi_1'} + \xi_3 \frac{\partial G_i}{\partial \xi_2'}\right\}$$
$$\frac{\partial x_i}{\partial \xi_2} = \frac{\partial G_i}{\partial \xi_1'}, \qquad \frac{\partial x_i}{\partial \xi_3} = \frac{\partial G_i}{\partial \xi_2'} \tag{38}$$

where the component $\partial x_i/\partial \xi_i$ is unbounded when $\xi_1 = 1$. Similar analysis can be performed to show that the first derivative of the edge blend term from Eq. (36) is unbounded at $\xi_2 = \xi_3 = \frac{1}{2}$. Thus, the linear rational blends are only C^0.

Blending schemes for simplex domains, analogous Eq. (30), but with higher order of smoothness can be constructed. One possible tetrahedral blend which is C^k is given by

$$\begin{aligned} x_i(\xi_j) = {} & (1 - \xi_1)^{k+1} G_i(\xi') + (1 - \xi_2)^{k+1} E_i(\xi') + (1 - \xi_3)^{k+1} F_i(\xi') + (1 - \xi_4)^{k+1} D_i(\xi') \\ & - (1 - \xi_1 - \xi_2)^{k+1} W_i(\xi') - (1 - \xi_1 - \xi_3)^{k+1} T_i(\xi') - (1 - \xi_1 - \xi_4)^{k+1} Q_i(\xi') \\ & - (1 - \xi_2 - \xi_3)^{k+1} S_i(\xi') - (1 - \xi_2 - \xi_4)^{k+1} R_i(\xi') - (1 - \xi_3 - \xi_4)^{k+1} P_i(\xi') \\ & + \xi_1^{k+1} x_i(1, 0, 0, 0) + \xi_2^{k+1} x_i(0, 1, 0, 0) + \xi_3^{k+1} x_i(0, 0, 1, 0) + \xi_4^{k+1} x_i(0, 0, 0, 1) . \end{aligned} \tag{39}$$

The face term from Eq. (34) now becomes

$$x_i(\xi) = (1 - \xi_1)^{k+1} G_i(\xi') \tag{40}$$

and the first partial derivatives are now given by

$$\frac{\partial x_i}{\partial \xi_j} = \frac{\partial (1 - \xi_1)^{k+1}}{\partial \xi_j} G_i + (1 - \xi_1)^{k+1} \frac{\partial G_i}{\partial \xi_k'} \frac{\partial \xi_k'}{\partial \xi_j} . \tag{41}$$

which on expanding yield

$$\frac{\partial x_i}{\partial \xi_1} = -(k+1)(1 - \xi_1)^k G_i + (1 - \xi_1)^{k+1} \left\{\frac{\xi_2}{(1 - \xi_1)^2} \frac{\partial G_i}{\partial \xi_1'} + \frac{\xi_3}{(1 - \xi_1)^2} \frac{\partial G_i}{\partial \xi_2'}\right\}$$
$$\frac{\partial x_i}{\partial \xi_2} = (1 - \xi_1)^k \frac{\partial G_i}{\partial \xi_1'} \tag{42}$$
$$\frac{\partial x_i}{\partial \xi_3} = (1 - \xi_1)^k \frac{\partial G_i}{\partial \xi_2'}$$

where $\partial x_i/\partial \xi_1$ can be simplified to

$$\frac{\partial x_i}{\partial \xi_1} = -(k+1)(1 - \xi_1)^k G_i + (1 - \xi_1)^{k-1} \left\{\xi_2 \frac{\partial G_i}{\partial \xi_1'} + \xi_3 \frac{\partial G_i}{\partial \xi_2'}\right\} . \tag{43}$$

$\partial x_i/\partial \xi_1$ exists at $\xi_1 = 1$. In fact, all partial derivatives up to order k are now continuous because the term $1/(1 - \xi_1)^{k+1}$ arising from the derivative of Eq. (35) is now cancelled by the blend term $(1 - \xi_1)^{k+1}$. A similar analysis can be used to demonstrate continuity of the edge blend terms up to k derivatives.

For simplices, higher-order C^k rational blends cost more to evaluate than the linear C^0 rational blends. For element topologies that are not bounded by any simplex entity (quadrilaterals and bricks) the linear blends are simple polynomials [19] and hence C^∞.

The fact that the blends are of limited smoothness implies that when used with p-adaptive finite element methods, the exponential convergence of the discretization error with increasing p will only hold to some critical value of p even for problems with smooth solutions.

6.2. Numerical example

This section demonstrates two issues: (1) the loss of exponential convergence with the use of C^0 rational blends on tetrahedral meshes, and (2) a cost comparison of using geometric mapping described by Eqs. (30) and (32) when the degenerated form of the $n \times n \times n$ tensor-product Gaussian integration is used to evaluate the element level integrals.

Consider the solution of *Poisson's* equation

$$-\Delta u(\boldsymbol{x}) = f(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega_G,$$
$$u(\boldsymbol{x}) = 0, \quad \boldsymbol{x} \in \Gamma_G, \tag{44}$$

where Ω_G is a sphere of unit radius and $f(x)$ is specified such that the exact solution is

$$u = 1000(1-r)r^5, \qquad r = (x_1^2 + x_2^2 + x_3^2)^{1/2}. \tag{45}$$

The exact *energy* of the solution is

$$\|u\|_e = \left\{ \int_{\Omega_G} \left(\frac{\partial u}{\partial \boldsymbol{x}}\right)\left(\frac{\partial u}{\partial \boldsymbol{x}}\right) \mathrm{d}\Omega \right\}^{1/2} = \frac{2000}{143}\sqrt{858\pi}. \tag{46}$$

The problem was solved using uniform p-refinement with $p = 1, \ldots, 9$. A coarse curvilinear mesh shown in Fig. 10 consisting of 64 tetrahedral regions is used to construct $\boldsymbol{x}^*(\boldsymbol{\xi})$ by

(1) The blending scheme described in Eq. (30), and
(2) Polynomial interpolation described in Eq. (32). The interpolation basis is the tetrahedral shape functions described in [31] with the interpolation points given in [8,9]. Although the analysis in Section 4.1 requires the interpolation order to be only $p-1$, a one time interpolation of order 8 was used to avoid having to repeat the solution of the interpolation problem as p was increased from 1 to 9.

In both cases, the accuracy of numerical integration was based on the analysis in Section 4.2. Fig. 10 plots the relative error of the finite element solution energy [34]

$$\|u - u^{(h,p)}\|_e = \frac{\|u\|_e - \|u^{(h,p)}\|_e}{\|u\|_e} \tag{47}$$

versus p. Since the problem has a smooth solution, the logarithm of the error is expected to decrease linearly with p. It is observed that the solution error obtained from Eq. (30) fails to continue to converge exponentially for $p \geq 6$. However, the solution error using Eq. (32) continues to converge exponentially past $p = 6$. However, the solution error using Eq. (32) continues to converge exponentially past $p = 6$. The continued convergence of the polynomial approximation is expected because the mapping defined by Eq. (32) is C^∞.

An explanation of why exponential convergence holds up to $p = 6$ when using C^0 rational blends with tetrahedral elements has not yet been determined. Issues related to the impact of using rational blends for

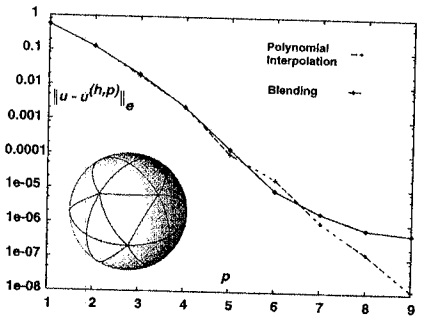

Fig. 10. Relative error in energy norm for smooth problem.

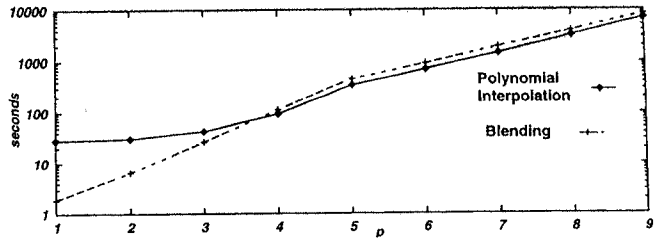

Fig. 11. Elemental computation time using degenerate tensor-product Gaussian integration.

elements with simplex entities in their closure on the convergence rate of the discretization error for a given problem does require further analysis.

Fig. 11 plots the time required to do the element level computations corresponding to the bilinear forms associated with the problem [21] $a(u, v)$ and (f, v). For degenerate tensor-product Gaussian integration, the element computation time grows at the same rate for geometry representation by direct blended mapping and polynomial interpolation of pointwise exact values. However, for $p \geq 4$, element level computations using polynomial approximation of geometry, as done in this experiment, were 1.25 times faster than those using blended mappings. The primary reason for this is that the approximation of the geometry was done once and the resulting coefficients b_k in Eq. (32) were stored for subsequent use. This means that pointwise queries of $\boldsymbol{x}(\boldsymbol{\zeta})$ from the geometric modeling system were only made once. In contrast, mapping using the blending scheme in Eq. (30) must query the modeler for pointwise $\boldsymbol{x}(\boldsymbol{\zeta})$ for each computation with a different p. For the present example, a profiling of the computation time for $p = 4$ showed that modeler queries used about 9% of total computation time when polynomial approximation of geometry was used in comparison to 28% required using direct blended mapping. This example shows that the efficiency of the geometric modelling system in evaluating $\boldsymbol{x}(\boldsymbol{\zeta})$, $\partial \boldsymbol{x} / \partial \boldsymbol{\zeta}$ is an important factor in determining the efficiency of geometry representation during higher-order finite element computations. This example should not be used to infer that polynomial interpolation is more efficient than direct blended mapping because the idea of storing the interpolation coefficient only works if there is no spatial adaptation of the mesh and furthermore, as is shown in a forthcoming paper [14], tensor-product Gaussian integration for simplex elements of general curved shapes is not efficient when compared to alternate schemes and directly approximate the integrand by polynomials followed by the use of precomputed integrals.

7. Concluding remarks

This paper presented the requirements of geometric approximations made during element level computations for second order elliptic boundary value problems. The order of direct approximation of the geometry and the non-polynomial parts of the integrands using polynomial basis functions must be $p-1$ to preserve the optimal rate of convergence of the discretization error. A mesh geometry mapping scheme, based on using model entity parametric coordinates and the boolean sum interpolation theory, which conforms exactly to the shape of the domain boundary as defined in the geometric modelling system was presented. On simplex domains, the linear blending schemes have limited smoothness and hence theoretically only guarantee optimal convergence of the discretization error up to a critical p. For the example presented here with tetrahedral elements, exponential convergence of the discretization error for smooth analytic solutions was preserved up to $p = 6$. Extension of the linear rational blend on a tetrahedron with higher order of smoothness was also presented. The evaluation of the higher-order rational blends does require more computational effort than the linear rational blends. A complete analysis of the effect on the rational blends on simplex domains on the convergence rate of the discretization error of finite element solution is required.

The idea of directly approximating the geometry related terms appearing in element level integrals can be used in developing efficient element level integration schemes for high order finite elements in general curved domains. The primary steps of such a scheme are:

(1) Approximate portions of the integrand, arising from geometry related or material data quantities which cannot be integrated exactly, by linear combinations of polynomial basis functions based on accuracy requirements described in Section 4.1.
(2) Exactly precompute and store values of the polynomial integrands resulting from step 1 for fast lookup [28] during actual computation.

It should be emphasized that for general curved elements, the integrand approximation is required even if element geometry is approximated by polynomials. Both direct blending or polynomial approximation, are viable options for geometry representation for use in element level integrations using polynomial approximations of integrands. The determination of the best scheme must factor in the cost of polynomial approximations on specific element topologies and the relative cost of doing geometric modeler queries for pointwise position and derivative data. The issue of efficient element level integrations on three-dimensional curvilinear domains is addressed in [14].

Acknowledgements

The work described here is part of the first author's doctoral research which was supported in part by the Office of Naval Research through grant No. N00014-94-1-0962 and the Naval Research Laboratory through ONR grant No. N00014-C-6023. Support of the U.S. Army Research Office through contract No. DAAH04-95-1-0091 is also acknowledged.

References

[1] M. Abramowitz and I.A. Stegun, eds., Handbook of Mathematical Functions (Dover Publications Inc., New York, 1972).
[2] I. Babuska and B.Q. Guo, The *h-p* version of the finite element method for domains with curved boundaries. SIAM J. Numer. Anal. 25(4) (1988) 837–861.
[3] I. Babuska and B.Q. Guo, Approximation properties of the *h-p* version of the finite element method. Comput Methods Appl. Mech. Engrg. 133 (1996) 319–346.
[4] I. Babuska, B. Szabo and I.N. Katz, The *p*-version of the finite element method, SIAM J. Numer. Anal. 18(3) (1981) 515–541.
[5] U. Banerjee and M. Suri, The effect of numerical quadrature in the *p*-version of the finite element method, Math. Comput. 59(199) (1992) 1–20.
[6] M.W. Beall and M.S. Shephard, A general topology-based mesh data structure, Technical Report 19–1996, Scientific Computation Research Center, Rensselaer Polytechnic Institute, Troy, NY 12180-3590, 1996, Int. J. Numer. Methods Engrg., submitted.
[7] C. Bernard, Optimal finite-element interpolation on curved domains, SIAM J. Numer. Anal. 26(5) (1989) 1212–1240.
[8] Q. Chen and I. Babuska, Approximate optimal points for polynomial interpolation of real functions in an interval and in a triangle, Comput. Methods Appl. Mech. Engrg. 128 (1995) 405–417.
[9] Q. Chen and I. Babuska, Approximate optimal polynomial interpolation points in the tetrahedron, 1995, preprint.
[10] P.G. Ciarlet, The Finite Element Method for Elliptic Problems (North-Holland Publishing Company, Amsterdam, 1978).
[11] 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, Comput. Methods Appl. Mech. Engrg. 77 (1989) 79–112.
[12] 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.
[13] S. Dey, Geometry-based three dimensional *hp* finite element modelling and computations, Ph.D. Thesis, Civil Engineering, Rensselaer Polytechnic Institute, Scientific Computation Research Center, RPI, Troy, NY 12180-3590, January 1997.
[14] S. Dey, J.E. Flaherty and M.S. Shephard, Efficient integration of *p*-version finite elements of general geometric shape, in preparation for submission, 1996.
[15] Electronic Data Systems Corporation, Parker's House, 46 Regent Street, Cambridge CB2 1DB England, Parasolid Version 6 KI Programming Reference Manual, October 1994.
[16] J. Fish and R. Guttal, Recent advances in the *p*-version of the finite element method for shells, Comput. Syst. Engrg. 6(3) (1995) 195–211.
[17] W.J. Gordon and C.A. Hall, Construction of curvilinear co-ordinate systems and applications to mesh generation, Int. J. Numer. Methods Engrg. 7 (1973) 461–477.
[18] 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, Numer. Math. 49 (1986) 577–612; 613–657; 659–683.

[19] 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. Methods Engrg. 17 (1981) 1015–1044.
[20] F.B. Hildebrand, Introduction to Numerical Analysis (Dover Publications, Inc., New York, NY, 1987).
[21] T.J.R. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis (Prentice-Hall, Englewood Cliffs, NJ, 1987).
[22] C.G. Kim and M. Suri, On the *p* version of the finite element method in the presence of numerical integration, Numer. Methods for Partial Diff. Eqn. 9 (1993) 593–629.
[23] C. Lacombe and C. Bedard, Face-apex projectors for the interpolation function of a general tetrahedral mid-edge finite element, Comput. Methods Appl. Mech. Engrg. 68 (1988) 177–188.
[24] A.R. Mitchell and R. Wait, The Finite Element Method in Partial Differential Equations (John Wiley and Sons, New York, 1977).
[25] 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, eds., Recent Developments in Finite Element Analysis. International Center for Numerical Methods in Engineering (Barcelona, Spain, 1994) 29–35.
[26] J.T. Oden and L. Demkowicz, *h-p* adaptive finite element methods in computational fluid dynamics, Comput. Methods Appl. Mech. Engrg. 89 (1991) 11–40.
[27] J.T. Oden and J.N. Reddy, An Introduction to the Mathematical Theory of Finite Elements (John Wiley and Sons, New York, 1976).
[28] T.K. Ohsumi, S. Dey, J.E. Flaherty and M.S. Shephard, Table look-up: High order integration on tetrahedra region. 1996, in preparation.
[29] M.J.D. Powell, Approximation Theory and Methods (Cambridge University Press, New York, 1981).
[30] W. Rachowicz, J.T. Oden and L. Demkowicz, Toward a universal *h-p* adaptive finite element strategy, Part 3. Design of *h-p* meshes, Comput. Methods Appl. Mech. Engrg. 77 (1989) 181–212.
[31] M.S. Shephard, S. Dey and J.E. Flaherty, A straightforward structure to construct shape functions for variable *p*-order meshes, Comput. Methods Appl. Mech. Engrg. (1996) to appear.
[32] M.S. Shephard and M.K. Georges, Reliability of automatic 3-D mesh generation, Comput. Methods Appl. Mech. Engrg. 101 (1992) 443–462.
[33] Spatial Technology, Inc., 2425 25th St., Building A, Boulder, Colorado, ACIS Interface Guide and ACIS API Guide, December 1992.
[34] G. Strang and G.J. Fix, An Analysis of the Finite Element Method (Prentice-Hall, Englewood Cliffs, NJ, 1973).
[35] B.A. Szabo and I. Babuska, Finite Element Analysis (Wiley Interscience, New York, 1991).
[36] XOX Corporation, Two Appletree Square, Suite 334, Minneapolis, Minnesota 55425. SHAPES Reference Manual, Release 2.0.8, July 20, 1993.
[37] O.C. Zienkiewicz and R.L. Taylor, The Finite Element Method—Volume 1, 4th edition (McGraw-Hill Book Co., New York, 1987).