The isoparametric maps $x_1(2, 2)$ and $x_2(2, 4)$ are isomorphic. Any pair of parameters $(\xi, \eta)$ on a boundary of a brick 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 $x_1 \circ x_2^{-1}$ is an isomorphism which means that there exists a unique map between those two elements.

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

References

Finite element approximations

\[ u^{h,p} \]

\( u^{h,p} \) is the finite element approximation of an unknown solution \( u \) using a mesh of size \( h \) and shape functions of polynomial order \( p \).

\( \phi_i \)

\( \phi_i \) is the \( i \)th element shape function.

\( H^r \)

\( H^r \) is the \( r \)th Sobolev space of functions.

\( \|u\|_p \)

\( \|u\|_p \) is the Sobolev norm of \( u \).

\( \|u\| \)

\( \|u\| \) is the 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 impact 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 of 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 mapping \( \mathbf{x} \) 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 \( \Omega_0 \), or its boundary, \( \Gamma_0 \), and
2. Enforcement of essential boundary condition(s) specified on curved portion(s) of the boundary of \( \Omega_0 \).

For \( \Omega_0 \subseteq \mathbb{R}^n \), components of the mapping are given by \( x_i(\mathbf{xi}) ; i = 1, 2, \ldots, n \) where the parametric coordinates \( \mathbf{xi} \) depend on the topology and the dimension of the mesh entity as depicted in Table 1.

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

\[
\int_{\Omega^k} \frac{\partial n_i}{\partial x} \frac{\partial n_j}{\partial x} \, dx = \int_{\Omega} \frac{\partial n_i}{\partial \xi} \frac{\partial n_j}{\partial \xi} \, J(\mathbf{xi}) \, d\xi
\]

and

\[
\int_{\Omega^k} N_i(x) N_j(x) \, dx = \int_{\Omega} N_i(\mathbf{x}(\mathbf{xi})) N_j(\mathbf{x}(\mathbf{xi})) \, J(\mathbf{xi}) \, d\xi
\]

respectively, with \( J(\mathbf{xi}) = \left| \frac{\partial \mathbf{x}}{\partial \mathbf{xi}} \right| \). Similarly, integrals resulting from the weak enforcement of natural boundary condition(s), \( h_i(x) \), are given by [21, 34, 24, 27]

![Fig. 1. Mesh entity geometry mapping.](image)

Table 1

<table>
<thead>
<tr>
<th>Topology</th>
<th>Parametric domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edge</td>
<td>( \bar{\xi}_1 \in (-1, 1) )</td>
</tr>
<tr>
<td>Triangle</td>
<td>( \bar{\xi}_1, \bar{\xi}_2 \in [0, 1], \bar{\xi}_1 + \bar{\xi}_2 = 1 )</td>
</tr>
<tr>
<td>Quadrilateral</td>
<td>( \bar{\xi}_1, \bar{\xi}_2 \in [0, 1], \bar{\xi}_1 + \bar{\xi}_2 + \bar{\xi}_3 = 1 )</td>
</tr>
<tr>
<td>Tetrahedron</td>
<td>( \bar{\xi}_1, \bar{\xi}_2, \bar{\xi}_3, \bar{\xi}_4 \in [0, 1], \bar{\xi}_1 + \bar{\xi}_2 + \bar{\xi}_3 + \bar{\xi}_4 = 1 )</td>
</tr>
<tr>
<td>Hexahedron</td>
<td>( \bar{\xi}_1, \bar{\xi}_2, \bar{\xi}_3 \in [-1, 1] )</td>
</tr>
<tr>
<td>Pentahedron</td>
<td>( \bar{\xi}_1, \bar{\xi}_2, \bar{\xi}_3 \in [0, 1], \bar{\xi}_1 + \bar{\xi}_2 = 1, \bar{\xi}_3 = 1, \bar{\xi}_4 \in [-1, 1] )</td>
</tr>
<tr>
<td>Pyramid</td>
<td>( \bar{\xi}_1 \in [-1, 1], \bar{\xi}_2 \in (-1, 1) )</td>
</tr>
</tbody>
</table>

\( \bar{\xi}_i \) is the parametric coordinate corresponding to the \( i \)th local coordinate axis.
where \( \Gamma^n \) 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.,

\[
n^e(x) = g(x), \quad \forall x \in \Gamma^n,
\]

(4)

where \( \Gamma^n \) 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 \( \Gamma^n \) using the finite element basis as

\[
g(x) = g^*(x) + \sum_{i=1}^{n_e} a_i N_i(x) = g^*(x) + \sum_{i=1}^{n_e} a_i N_i(\xi),
\]

(5)

where \( \{\xi^{(i)}\} \) and \( \{N_i\} \) respectively define the sets of \( n_e \) interpolation points and basis functions for \( p \)-th order interpolation. The solution of the interpolation problem to determine \( a_i \) requires the evaluation of the mapping function \( x(\xi^{(i)}) \).

The next key issue that must be addressed is the accuracy with which geometry must be accounted for during 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

\[
I_{n^e} = \int_{\Omega} \kappa_{n^e}(x) \, dx, \quad I_{r^e} = \int_{\Omega} \kappa_{r^e}(x) \, dx
\]

(6)

where \( \kappa_{n^e} \) and \( \kappa_{r^e} \) 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 \( \kappa_{n^e} \) and \( \kappa_{r^e} \).

(3) Approximation integration method.

Useful operators to compute \( I_{n^e} \) and \( I_{r^e} \), denoted by \( I_{n^e}^{(i)} \) and \( I_{r^e}^{(i)} \), respectively, that use combinations of the basic approximations possible can be stated as:

(1) Exact geometry representation followed by approximate integration yielding

\[
I_{n^e}^{(i)} = \int_{\Omega} \kappa_{n^e}(x) \, dx = \sum_{i=1}^{n_e} \left[ \kappa_{n^e}(x(\xi^{(i)})) \right] w^{(i)}
\]

(7)

\[
I_{r^e}^{(i)} = \int_{\Omega} \kappa_{r^e}(x) \, dx = \sum_{i=1}^{n_e} \left[ \kappa_{r^e}(x(\xi^{(i)})) \right] w^{(i)}
\]

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

\[
\int_{\Omega} \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} \, dx = \sum_{i=1}^{n_e} \left[ \frac{\partial N_i}{\partial x} (\xi^{(i)}) \frac{\partial N_j}{\partial x} (\xi^{(i)}) \right] \left[ \frac{\partial x}{\partial x} (\xi^{(i)}) \frac{\partial x}{\partial x} (\xi^{(i)}) \right] R^{(i)} w^{(i)}
\]

(8)

with

\[
R^{(i)} = \frac{\partial x}{\partial x} (\xi^{(i)}).
\]

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, \( x(\xi) \approx x^{*(\xi)}(\xi) \), followed by approximate numerical evaluation of the resulting integrand to yield

\[
I_{n^e}^{(i)} \approx \sum_{i=1}^{n_e} \left[ \kappa_{n^e}(x^{*(\xi^{(i)})}) \right] w^{(i)},
\]

(9)

\[
I_{r^e}^{(i)} \approx \sum_{i=1}^{n_e} \left[ \kappa_{r^e}(x^{*(\xi^{(i)})}) \right] w^{(i)}
\]

where \( \{\xi^{(i)}\} \) and \( \{w^{(i)}\} \) define the sets of \( n_e \) 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

\[
\int_{\Omega} \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} \, dx = \sum_{i=1}^{n_e} \left[ \frac{\partial N_i}{\partial x} (\xi^{(i)}) \frac{\partial N_j}{\partial x} (\xi^{(i)}) \right] R^{(i)} \, dx
\]

(10)

with

\[
R^{(i)} = \frac{\partial x}{\partial x} (\xi^{(i)}).
\]

This is the commonly used technique in the isoparametric h-version finite element method [21,37] where \( x^{*(\xi)}(\xi) \) 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

\[
I_{n^e}^{(i)} = \int_{\Omega} \kappa_{n^e}(x) \, dx = \int_{\Omega} \kappa_{n^e}(x) \, dx, \quad I_{r^e}^{(i)} = \int_{\Omega} \kappa_{r^e}(x) \, dx.
\]

(11)

It is assumed that \( \kappa_{n^e}(x) \) and \( \kappa_{r^e}(x) \) 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 \( \kappa_{n^e}^{(i)}(x) \) and \( \kappa_{r^e}^{(i)}(x) \) as linear combinations of polynomials [20], \( \phi_{n^e}(x) \) and \( \phi_{r^e}(x) \), defined over \( \Omega^{*} \) and \( \Gamma^{*} \), respectively, to obtain

\[
\kappa_{n^e}^{(i)}(x) = \sum_{i=1}^{m} \alpha_{n^e}^{(i)} \phi_{n^e}^{(i)}(x), \quad \kappa_{r^e}^{(i)}(x) = \sum_{i=1}^{m} \alpha_{r^e}^{(i)} \phi_{r^e}^{(i)}(x).
\]

(12)

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

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 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]

$$
\|u - u^{h,p}\|_{\infty} \leq \frac{C h^{r+\alpha}[u]\|^{1/p}}{p^{r+\alpha}}.
$$

(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 $r \geq p+1 \Rightarrow \alpha = p+1-m$, then the discretization error will converge exponentially with respect to $p$-refinement at a rate given by

$$
\|u - u^{h,p}\|_{\infty} \leq \frac{C h^{r+1}[u]}{p^{r+1}}.
$$

(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, $x \neq x^*$, and integrand approximations, $\kappa_{\gamma^*} - \kappa_{\gamma}^*$ and $\kappa_{\gamma^*} - \kappa_{\gamma}^*$, 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 $\phi(U)$ with $\phi^*(U)$, has the form [29]

$$
\|\phi - \phi^*\| \leq (1 + \lambda) h.
$$

(15)

where $\phi$ is the best approximation error and $\lambda$ is the Lebesgue constant of the approximation operator. Since finite element solutions are usually defined as linear combinations of polynomial functions in $H^r(\Omega)$ and their errors measured in appropriate Sobolev norms, the order $q$ approximations of $\phi(U)$, $\phi \in H^r$, will be restricted to those constructed by linear combinations of polynomial functions, $\phi(U) = \sum_{i=1}^{n} a_i \phi_i(U)$, given by

$$
\phi^*(U) = \sum_{i=1}^{n} a_i \phi_i(U)
$$

(16)

where the number of polynomial basis functions used, $n$, depends on the topology and the dimension of $\Omega$.

The number of functions needed to construct a complete $n_p$ degree polynomial basis for various element topologies is given in Table 2.

For smooth functions ($\phi \in H^{r+\alpha}$), $\phi$ is known to converge exponentially [29] and the overall convergence rate of $\phi - \phi^*(U)$ is determined by the rate at which $\lambda$ grows with respect to $q$ [8,9]. When the interpolation points are equally spaced $a$ can grow exponentially fast causing a loss of exponential convergence of $\|\phi - \phi^*\|$, even when $\beta$ is smooth. However, for the non-uniformly spaced interpolation sets of Chen and Babuška [8,9], $\lambda$ grows approximately as $O(q^p)$ which ensures exponential convergence of $\|\phi - \phi^*\|$.

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

$$
\|\phi - \phi^*\| \leq C h^{r+\alpha-q}[\phi] ; \quad r \geq q + 1,
$$

(17)

where $C_r$ is independent of $h$ and $\phi$; $C_r$ is a function of $q$ and its growth is controlled based on the choice of the interpolation set.

Since $\theta$ 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

$$
\|\phi - \phi^*\| \leq C h^{r+1-q}[\phi] ; \quad r \geq q + 1.
$$

(18)

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

$$
q + 2 - m \geq p + 1 - m
$$

(19)

yielding

$$
q \geq p - 1.
$$

(20)

Note that if $r \geq p + 1$, then Eq. (20) also defines the necessary and sufficient condition for approximation of $\kappa_{\gamma^*}$ and $\kappa_{\gamma}$ 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 $p$, $n\alpha / h^p$ in Eq. (10) is of order $p - 1$ and if the remaining geometry related terms from $\delta_{x}/h_\alpha$, $\delta_{y}/h_\gamma$, and $\delta_{\gamma}/h_\alpha$ 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 [11] is given by

$$
n = \prod_{i=1}^{n} \frac{3p - 3 + \beta}{2}
$$

(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 $\delta x / \delta \xi^q = n > 0$, with the case of $n = 0$ defining $x(U)$. 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 $U \in \Omega$. For effective use with high-order finite elements, three issues associated with the approximate position mapping $x^*(U)$, and derivative, $\delta x / \delta \xi^q$ become important:

1. Accuracy: As shown in Section 4.1, $x^*(U)$ 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 $x(U)$ 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 \( T_0 \), \( x(\xi) \), is housed within a geometric modelling system in terms of local parametric coordinate systems of model topological entities given by \( \xi \); therefore, a technique to relate \( \eta \) to \( \xi \) such that the resulting \( x(\xi) \) is exact with respect to \( \eta \), when \( \xi \in \Gamma^e \subset \Gamma_0 \), is required. Pointwise evaluation of \( x(\xi) \) can then be used during finite element computations directly, or used in the construction of polynomial approximations of \( x(\xi) \) which have the required accuracy.

5.1. Geometric mapping conforming exactly to \( T_0 \)

This section describes a technique for obtaining \( x(\xi) \) which is exact with respect to \( T_0 \) when \( \xi \in \Gamma^e \subset \Gamma_0 \).

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 \( x(\xi) \) 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, \( G_i^d \), the desired map can be obtained as

\[
x = x(\xi(\xi_i))
\]

(22)

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

\[
\frac{\partial x}{\partial \xi} = \frac{\partial x}{\partial \xi_i} \frac{\partial \xi_i}{\partial \eta}
\]

Fig. 3 graphically depicts this mapping for a triangular and quadrilateral mesh face using the three coordinate systems \( \xi, \xi_i \) and \( x(\xi) \); \( \partial x / \partial \xi \) 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 an underlying continuous, nondegenerate parametric space. The availability of an underlying parameterization for specific \( G_i^d \) of \( \Omega \) depends on the representation scheme used within the geometric modelling system.

Most geometric modelers do not have an explicit parametric coordinate system for \( G_i^d \); instead, they are represented implicitly as the portion of three-dimensional space that is bounded by a closed set of \( G_i^d \) [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 \( M_i^d \subset G_i^d \), 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 \( \Omega \) and they cover the interior of \( \Omega \), they introduce no geometric approximation.

The technique used to construct \( x(\xi) \) depends on the classification of a \( M_i^d \) with respect to \( \Omega \). In addition, the construction of \( x(\xi) \) 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 \( R_j^1 \) on a \( G_i^d \); \( d = 1 \), (Fig. 5), in the absence of interference with any trimmed boundary entities and with acceptable resulting element shapes, \( x(\xi_j) \) can be given by a linear interpolation between \( \xi_j \) values at the vertices

\[
x(\xi_j) = \xi_j(1,0)\xi_1 + \xi_j(0,1)\xi_2, \quad \xi_1 + \xi_2 = 1.
\]

(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, \( x(\xi_j) \) can be assigned general geometric shapes such as a quadratic form given by

\[
x(\xi_j) = a + b\xi_j + c\xi_j^2
\]

(24)

or, a cubic form given by
\[ \zeta_i(\xi_1) = a_i + b_i \xi_1 + c_i \xi_1^2 + d_i \xi_1^3. \]  

(25)

The coefficients \( a, b, c, d \) 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 \( a, b, c, d \) must account for the effect of the mapping \( \zeta(\xi) \) defining the shape of the model face in the geometric modeling system.

For a \( M_i^j \) inside a \( G^j \), 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 \( x = a_i + b_i \xi_1 + c_i \xi_1^2 + d_i \xi_1^3 \). The coefficients \( a, b, c, d \) are determined such that the shapes of the elements, in real space, are bounded by the edge, are acceptable quality.

5.1.2. Mesh face mapping

The construction of \( \zeta(\xi) \) for \( M_i^j \) on \( G^j \) 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 \( M_i^j \) is classified on an untrimmed model face as shown in Fig. 6, then \( \zeta(\xi) \) can be constructed as a linear interpolation of the \( \zeta \) values at the vertices of the face defined by

\[ \zeta_i(\xi) = \zeta(1, 0, 0) \xi_1 + \zeta(0, 1, 0) \xi_2 + \zeta(0, 0, 1) \xi_3. \]  

(26)

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

The linear interpolation of vertex values does not work if the \( G^j \) is trimmed because the curves defining the mesh edges, in the parametric space of the \( G^j \), may not be straight in that space leading to ‘spills’ or ‘gaps’ as unacceptable.

In such cases the construction of \( \zeta(\xi) \) 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

\[ \zeta_i(\xi) = \frac{1}{2} \left( \left( \frac{\xi_1}{1-\xi_1} \right) p(\xi_1) + \left( \frac{\xi_2}{1-\xi_2} \right) p(1-\xi_2) + \left( \frac{\xi_3}{1-\xi_3} \right) q(\xi_3) + \left( \frac{\xi_4}{1-\xi_4} \right) r(\xi_4) \right) \]

\[ + \left( \frac{\xi_5}{1-\xi_5} \right) q(\xi_5) + \left( \frac{\xi_6}{1-\xi_6} \right) r(1-\xi_6) + \left( \frac{\xi_7}{1-\xi_7} \right) q(1-\xi_7) + \left( \frac{\xi_8}{1-\xi_8} \right) r(1-\xi_8) \]

\[ - \xi_1 \xi_2 \xi_3 \xi_4 + \xi_1 \xi_2 \xi_3 \xi_4 \]

(27)

where \( p(\xi), q(\xi), r(\xi) \) are defined as the shape of the curves \( AB, BC \) and \( CD \), respectively. The specific expressions for \( p(\xi), q(\xi) \) and \( r(\xi) \) are derived based on the classification of the mesh edge as described in the previous section.

5.1.3. Mesh region mapping

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

\[ x_i(\xi) = (1-\xi_1)Q_i(\xi_1) - (1-\xi_2)R_i(\xi_2) + (1-\xi_3)E_i(\xi_3) + (1-\xi_4)D_i(\xi_4) \]

\[ - (1-\xi_1)Q_i(\xi_1) - (1-\xi_2)R_i(\xi_2) - (1-\xi_3)E_i(\xi_3) - (1-\xi_4)D_i(\xi_4) \]

\[ + (1-\xi_1)Q_i(\xi_1) + (1-\xi_2)R_i(\xi_2) + (1-\xi_3)E_i(\xi_3) + (1-\xi_4)D_i(\xi_4) \]

\[ + x_i(1,0,0,0) + x_i(0,1,0,0) + x_i(0,0,1,0) + x_i(0,0,0,1) \]

(28)

where \( Q, R, S, T, W \) and \( D, E, F, G \) represent the position vectors defining the shapes of the bounding edges and faces, respectively. The specific shapes of the boundary edges and faces are obtained by mapping process described in Sections 5.1.1 and 5.1.2. \( \xi \) defines the face (edge) parameteric coordinates normalized such that \( x_{\xi_i} = 1 \) when \( \xi_i = 0 \), \( \xi_i = \xi_i(\xi_i + \xi_i) \), \( \xi_i = \xi_i(\xi_i + \xi_i) \) and for edge \( \xi_i = 0 \), \( \xi_i = \xi_i(\xi_i + \xi_i) \). For mesh regions with no boundary edges or faces classified on a curved model boundary, the mapping can be obtained by linear interpolation of vertex coordinates

\[ x_i(\xi) = x_i(1,0,0,0)\xi_1 + x_i(0,1,0,0)\xi_2 + x_i(0,0,1,0)\xi_3 + x_i(0,0,0,1)\xi_4 \]

(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 $x_i(\xi)$ can be used to construct $x^*(\xi)$ as a linear combination of polynomial basis functions [29,30] defined over $\Gamma_i$, $\phi_i(\xi)$, to give

$$
x_i(\xi) = \sum_{i=1}^{n} \phi_i(\xi) x_i^0.
$$

In the isoparametric form of Eq. (32), widely used for low order $h$-version finite elements [10,34], $q = p$ and $\phi_i = N_i$, where $N_i$ 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 $\xi$

$$
\frac{\partial x^*_i}{\partial \xi_j} = \sum_{i=1}^{n} \frac{\partial \phi_i}{\partial \xi_j} x_i^0.
$$

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 1 between vertices 2, 3 and 4 from Eq. (30) given by

$$
x_i(\xi) = (1 - \xi_i) G_i(\xi_1, \xi_2)
$$

where

$$
\xi_1 = \frac{\xi_1}{\xi_1 + \xi_2 + \xi_3} = 1 - \xi_2, \quad \xi_2 = \frac{\xi_2}{\xi_1 + \xi_2 + \xi_3} = \frac{\xi_1}{1 - \xi_2}.
$$

It can be shown by the geometrical constraints in Fig. 9(a) that $\xi_1$ and $\xi_2$ exist at $\xi_i = 1$ and correspond to the projection to the centroid of the face implying $\xi_1 = \frac{1}{2}$. Furthermore, from Eq. (34) it can be seen that $G_i(\xi_1, \xi_2)$ contributes nothing at $\xi_i = 1$ since the multiplying factor $1 - \xi_i$ vanishes. Now consider the edge blend contribution from Eq. (30) for the edge between vertices 1 and 4 given by

$$
x_i(\xi) = (1 - \xi_i) S_i(\xi_1, \xi_2)
$$

where the independent edge coordinate $\xi_1 = \xi_i/(1 - \xi_i)$ and $\xi_2 = \xi_i/(1 - \xi_i - \xi_i)$. Once again from Fig. 9(b) it is clear that $\xi_i = \xi_i = \frac{1}{2}$ corresponds to the midpoint of the edge and hence $\xi_1 = \frac{1}{2}$. Furthermore, at $\xi_1 = \xi_2 = \frac{1}{2}$ the multiplying term $1 - \xi_i = 0$, 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 $x \in H^{r+1}$. 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.](image-url)
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) where 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

\[
-\Delta u(x) = f(x), \quad x \in \Omega_0,
\]
\[
u_0 = 0, \quad x \in \Gamma_0,
\]  
(44)

where \( \Omega_0 \) is a sphere of unit radius and \( f(x) \) is specified such that the exact solution is

\[
u = 1000(1-r)^3, \quad r = (x_1^2 + x_2^2 + x_3^2)^{1/3}.
\]  
(45)

The exact energy of the solution is

\[
\|v_0\|_{L_2} = \left( \int_{\Omega_0} \left( \frac{\partial v}{\partial x_i} \right)^2 \, dx \right)^{1/2} = \frac{2000}{43} \sqrt{858 \pi}.
\]  
(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 \( x(\xi) \) 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 having at least two of the solution of the interpolation problem as \( p \) 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]

\[
\|v - u^{(n)}\|_{L_2} = \frac{\|v_0 - u^{(n)}\|_{L_2}}{\|v_0\|_{L_2}}
\]  
(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 \( p \leq 6 \). However, the solution error using Eq. (32) continues to converge exponentially past \( p = 6 \). However, the continued convergence of the polynomial approximation is expected because the mapping defined by Eq. (32) is \( C^0 \).

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 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 \( f(u) \). 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 \( p > 4 \), element level computations using polynomial approximation of geometry, as done in the experiment, were 4.25 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 \( b_i \) in Eq. (32) were stored for subsequent use. This means that pointwise queries of \( x(\xi) \) 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 \( x(\xi) \) for each computation with a different \( p \). For the present example, a profiling of the computation time for \( p = 4 \) showed that the 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 modeling system in evaluating \( x(\xi) \) 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 adaptivity of the mesh and furthermore, as 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 modeling 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 of 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 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-0623. Support of the U.S. Army Research Office through contract No. DAAH04-95-1-0091 is also acknowledged.

References