INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2006; **67**:226–252
Published online 13 January 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme.1625

A three-dimensional finite element method with arbitrary polyhedral elements

M. M. Rashid*,† and M. Selimotic‡

Department of Civil and Environmental Engineering, University of California, Davis, Davis, CA 95616, U.S.A.

SUMMARY

The 'variable-element-topology finite element method' (VETFEM) is a finite-element-like Galerkin approximation method in which the elements may take arbitrary polyhedral form. A complete development of the VETFEM is given here for both two and three dimensions. A kinematic enhancement of the displacement-based formulation is also given, which effectively treats the case of near-incompressibility. Convergence of the method is discussed and then illustrated by way of a 2D problem in elastostatics. Also, the VETFEM's performance is compared to that of the conventional FEM with eight-node hex elements in a 3D finite-deformation elastic–plastic problem. The main attraction of the new method is its freedom from the strict rules of construction of conventional finite element meshes, making automatic mesh generation on complex domains a significantly simpler matter. Copyright © 2006 John Wiley & Sons, Ltd.

KEY WORDS: finite element method; polyhedral elements; non-conforming methods; convergence

1. INTRODUCTION

The displacement-based finite element method is well established as the pre-eminent numerical method for solving problems in solid mechanics. The reasons for its dominant status are many: arbitrarily complex geometries are accommodated with a single algorithmic design, both natural and essential boundary conditions are enforced in a straightforward way, and a high level of solution accuracy can be had for a given computational effort. These and other desirable features, however, come at the cost of fairly strict rules of construction attending mesh design.

*Correspondence to: M. M. Rashid, Department of Civil and Environmental Engineering, University of California, Davis, Davis, CA 95616, U.S.A.
†E-mail: mmrashid@ucdavis.edu
‡Graduate Student Researcher.

Contract/grant sponsor: Idaho National Laboratory; contract/grant number: 19760
Contract/grant sponsor: Lawrence Livermore National Laboratory; contract/grant number: ISCR-04-07

Received 14 January 2005
Revised 18 March 2005
Accepted 5 October 2005

It is these rules that represent the main difficulty in the practical solution of many complex problems in modern solid mechanics. Problems in this category include fracture in 3D domains, in which robust automatic mesh generation is a necessity; and mechanical design of extremely complex components, wherein many weeks of human effort are sometimes required to produce an adequate mesh.

The desire to reduce or eliminate the human effort in the meshing phase constitutes a main motivation for the development of both automatic finite element meshing algorithms, as well as so-called 'meshless' approximation schemes. Automatic mesh generation remains a very active field of research; References [1–5] represent an overview of recent activities with relevance to solid mechanics. At this stage, it is far from clear if truly automatic and robust meshing on arbitrary 3D domains is an attainable goal, particularly if hexahedral elements are desired, as they generally are in solid mechanics applications.

Meshless methods [6, 7], such as the element-free Galerkin method (EFG) [8, 9] and the natural element method (NEM) [10–12], seek to reduce the meshing effort required by reducing or eliminating the logical connectivity among the nodes. The displacement interpolant is based on unknowns associated with the nodes, as it is in the FEM. But, in meshless methods the displacement gradient is formed at the integration points generally without the help of a finite-element-like geometric partition of the domain. In the case of the EFG method, a moving-least-squares formulation is used, resulting in failure of the Kronecker-delta (interpolation) property at the nodes—a complication in the presence of essential boundary conditions and internal interfaces. It bears mention that meshless methods are not truly 'meshless' in the sense that some kind of spatial partition is required on which to evaluate the weak-form integrals. However, this quadrature mesh is generally much less constrained in its topology than is a finite element mesh on the same domain. See Reference [8] for a discussion of weak-form quadrature in meshless methods.

The NEM (see Reference [13] for a recent overview) is a fairly recent development which shares many of the motivations and practical attributes of the methodology presented in this paper. This method employs the concept of *natural-neighbour interpolants* to synthesize an approximation space given a region's geometry, together with a set of interior nodes. The basis functions emerge from a geometric construction based on the Voronoi diagram of the point set, along with a test point inserted at the position where the shape functions are to be evaluated. At a given point, the non-zero basis functions are associated with the nodes that are the point's 'natural neighbours.' Recent developments [14–16] have focused on removing a limitation of the original NEM, whereby some basis functions fail to possess the Kronecker-delta property for non-convex domains. The so-called α-NEM employs a modification to the natural-neighbour determination near concave portions of the boundary, which leads to true interpolants, provided the boundary curvature is finite and the local point density is sufficiently high. In other cases, more *ad hoc* procedures must be employed to modify the nearest-neighbour set. Other recent contributions have focused on effective quadrature rules for the NEM [17, 18]. Comparisons are made between specific elements of the NEM and the present method in subsequent sections.

The subject of this paper—the 'variable-element-topology finite element method (VETFEM)'—was first introduced in References [19, 20] in its 2D form. The VETFEM is a true finite-element method in the sense that the domain is partitioned into subregions—elements—which are then used both to construct basis functions and to effect quadrature on the domain. However, the method differs from the conventional FEM in one important respect: the elements are not subject to a requirement of geometric similarity to a canonical element (e.g. a cube), and so

may take any polyhedral form. This freedom, when put to good use, enormously simplifies the task of automatic mesh generation. This is the main advantage of the method.

The geometric freedom inherent in the VETFEM derives from the manner in which the shape functions are constructed. Instead of employing an isoparametric transformation to a canonical element as in the conventional FEM, VETFEM shape functions are instead given by finite-order polynomials in the *physical* co-ordinates of the problem. The coefficients of these polynomials are set so that, upon composition, the basis functions exhibit certain continuity properties across inter-element interfaces. Due to the finiteness of the polynomial order, strict continuity on inter-element interfaces is not possible in general; the VETFEM is therefore properly regarded as a non-conforming method. The main analytical tool for establishment of convergence of such methods is Stummel's *generalized patch test* [21] and its derivatives. Unlike the *engineering patch test* [22, 23], which refers to systematic numerical experimentation whose results suggest (but do not prove) overall good behaviour, the GPT is applied to the finite element function spaces themselves. In essence, the GPT is a means by which to establish, under mesh refinement, both an approximability property of the non-conforming finite element function space, as well as a closedness property. These properties, in conjunction with a two-sided error bound for non-conforming methods, are the required ingredients to establish convergence of the method. In practice, the GPT is not easy to apply to specific element formulations; however, more readily applicable tests have been derived from it, e.g. the F-E-M test of Shi [24].

The plan of the paper is as follows. First, to provide context, the variational boundary-value problem of Lagrangian continuum mechanics is briefly stated. Next, the VETFEM element formulation is described in detail for both 2D and 3D cases. A brief, descriptive account is then given of convergence issues, followed by description of a kinematic enhancement intended to effectively treat the case of near-incompressibility. The paper concludes with some illustrative linear and non-linear calculations. Notation is standard throughout. Terminology that is not routinely encountered in the engineering literature, even if standard in functional analysis, is explained in the course of the development.

2. VARIATIONAL PROBLEM STATEMENT

Preparatory to describing the VETFEM, the variational boundary-value problem of the purely mechanical theory of solid mechanics is herein stated. Let Ω be a bounded and open polyhedral subset of $\mathbb{R}^d$, $d=2$ or 3, with boundary $\partial\Omega=\partial_t\Omega\cup\partial_u\Omega$, $\partial_t\Omega\cap\partial_u\Omega=\emptyset$. Ω represents the spatial region occupied by the material body in a suitable reference configuration, whereas $\partial_t\Omega$ and $\partial_u\Omega$ are, respectively, the portions of $\partial\Omega$ on which the Piola traction vector is given as $\mathbf{p}=\bar{\mathbf{p}}$ and the displacement vector is given as $\mathbf{u}=\bar{\mathbf{u}}$. The displacement field $\mathbf{u}(\mathbf{X},t), \mathbf{X}\in\Omega, t>0$, with Cartesian components u_i, is governed by

$$\mathbf{u}\in V, \quad \int_\Omega [\nabla\mathbf{w}\cdot\mathbf{P}+\mathbf{w}\cdot\rho\ddot{\mathbf{u}}]\,\mathrm{d}v=\int_\Omega \mathbf{w}\cdot\rho\mathbf{b}\,\mathrm{d}v+\int_{\partial_t\Omega}\mathbf{w}\cdot\bar{\mathbf{p}}\,\mathrm{d}a \quad \forall\mathbf{w}\in V^0 \tag{1}$$

Here, $\mathbf{P}$ is the first Piola–Kirchhoff stress tensor and is a function of the motion through the deformation gradient $\mathbf{F}=\partial\mathbf{x}/\partial\mathbf{X}$, $\mathbf{x}=\mathbf{X}+\mathbf{u}$; $\rho\mathbf{b}$ is the body force per unit reference volume, and ∇ is the gradient operator in $\mathbb{R}^d$. The function space V is given by

$$V=\{\mathbf{u}\,|\,u_i\in H^1(\Omega)\times C^0[0,\infty);\, u_i=\bar{u}_i,\, \mathbf{X}\in\partial_u\Omega\} \tag{2}$$

whereas V^0 is similar but with homogeneous essential boundary condition. Following standard notation, $H^1(\Omega)$ here denotes the Sobolev space of order one on Ω. This function space consists of functions which belong to $L_2(\Omega)$—i.e. which are square-integrable on the open domain Ω—and whose weak derivatives[§] of order one exist and also belong to $L_2(\Omega)$. The displacement $\mathbf{u}(\mathbf{X}, \mathbf{t})$ is additionally subject to the initial conditions

$$\mathbf{u}(\mathbf{X}, 0) = \mathbf{u}^0(\mathbf{X}), \quad \dot{\mathbf{u}}(\mathbf{X}, 0) = \mathbf{v}^0(\mathbf{X}) \tag{3}$$

where $\mathbf{u}^0$ and $\mathbf{v}^0$ are given functions belonging to $H^1(\Omega)$.

It bears mention that the total-Lagrangian discretization envisaged here, whereby the reference configuration Ω is discretized, has the advantage that the shape-function gradients need be computed only once, and then stored for use throughout the course of the calculation. As explained in Section 3, no isoparametric mapping is employed in the synthesis of VET shape functions. Accordingly, even if an updated-Lagrangian formulation is for some reason desired, element distortion does not present conditioning problems for VET elements. However, as with any approximation method, large local distortions imply large gradients in strain, which in turn generally require fine meshes for accurate solution resolution. In this connection, the VETFEM appears to hold promise for convenient local remeshing and refinement to accommodate large local distortions.

3. THE VARIABLE-ELEMENT-TOPOLOGY FINITE ELEMENT METHOD

As with any Galerkin approximation method, the role of the VETFEM is to provide both a set of suitable basis functions on the domain Ω, and a means of evaluating the weak-form integrals on the domain. The VETFEM is a true finite element method, in the sense that the closure $\bar{\Omega}$ of the open problem domain $\Omega \subset \mathbb{R}^d$ is subdivided into L closed polyhedral elements ω_a such that $\bigcup_{1 \leqslant a \leqslant L} \omega_a = \bar{\Omega}$, and $\underline{\omega}_a \cap \underline{\omega}_b = \emptyset, a \neq b$. (Here, an overbar as $\bar{\omega}$ indicates closure, whereas an underbar indicates an open region, i.e. $\underline{\omega} = \bar{\omega} \backslash \partial\omega$.) The elements are used both to construct basis functions, as well as to effect numerical quadrature. In the VETFEM nodes are placed exclusively at the vertices of the elements.

3.1. Polynomial basis functions

Globally defined basis functions $\Phi_a(\mathbf{X})$ associated with the nodes are composed from shape functions $\phi_a, a = 1, \ldots, n$, defined on a typical element ω, with $\phi_a = 0, \mathbf{X} \notin \omega$. In the conventional FEM, polynomial shape functions are normally defined on a cubic or simplectic canonical element, and are then used to define a transformation between the canonical element and the physical one. In the VETFEM, the shape functions are instead written as low-order polynomials directly, in the physical co-ordinates X_i:

$$\phi_a = \sum_{1 \leqslant \alpha \leqslant M} G^a_\alpha q_\alpha \tag{4}$$

[§]The weak derivative is an extension of the classical derivative to some functions that fail to be differentiable in the classical sense over a set of measure zero. The classical and weak derivatives of a function coincide wherever the former exists. See, e.g. Reference [25, p. 194].

Here, q_α refers to an ordered list of monomials that is complete through a particular order; e.g. in two dimensions, $q_\alpha = \{1, X_1, X_2, X_1^2, X_1X_2, X_2^2, \ldots\}$. If ϕ_a is to be complete through order m, then $M = (m+d)!/(m!d!)$. The coefficients G_α^a are selected so that the shape functions exhibit certain desirable, finite-element-like properties, as described in the following sections.

3.2. Constraints

Writing $\mathbf{X}_b$ for the location of node b, the Kronecker-delta property

$$\phi_a(\mathbf{X}_b) = \sum_{1 \leqslant \alpha \leqslant M} G_\alpha^a q_\alpha(\mathbf{X}_b) = \delta_{ab}, \quad 1 \leqslant a, b \leqslant n \tag{5}$$

is enforced so that trial solutions are true interpolants of the nodal displacements. This property makes the enforcement of essential boundary conditions and interface conditions a straightforward matter, as it is in the conventional FEM. Constraints (5) are always linearly independent, and they are compatible for sufficiently large M.

The coefficients G_α^a are further constrained by a weak continuity condition at inter-element interfaces, as follows. The 2D case, in which VET elements are arbitrary polygons, is considered first, followed by the 3D case. If the shape functions each vary linearly on the (straight) element facets, then strict inter-element compatibility is attained, and the overall method becomes conforming. However, because of the finite order of the polynomial expression (4), strictly linear variation on element facets cannot be obtained for arbitrary polygonal elements. The method is therefore non-conforming in general. However, convergence is nonetheless provable, provided that the basis functions are at least 'weakly continuous,' in a certain sense, across element interfaces. As explained in Section 5.1, the required weak continuity is achieved if the inter-element jump for each basis function vanishes *in the mean* on each element facet. This requirement is satisfied if the mean difference between the shape function and a linear variation that interpolates the nodal values vanishes on each element facet. Although slightly more restrictive than zero mean of the inter-element jump, enforcement of zero mean variation from linearity is used in practice because it leads to an element-by-element decoupling of the calculation for the coefficients G_α^a. The resulting constraint is expressed by

$$\int_{\Gamma_b} \phi_a \, \mathrm{d}s = \frac{1}{2}[|\Gamma_a| \, \delta_{ab} + |\Gamma_c| \, \delta_{cb}](\text{ no sum}), \quad 1 \leqslant a, b \leqslant n \tag{6}$$

Here, Γ_a is the element facet defined by nodes a and $1+\mathrm{mod}(a, n)$, and $c = 1+\mathrm{mod}(n-2+a, n)$. Equation (6) says that the integral of shape function a on a facet containing node a must equal half the length of the facet, and zero for the other facets.

In three dimensions, VETFEM elements are arbitrary polyhedra, and their facets are polygons. The required weak continuity can again be expressed as a condition on the mean of the shape functions on each facet. Whereas the appropriate weak continuity condition on the line-segment facets for 2D elements is readily apparent, the analogous condition on polygonal facets is not as obvious. Specifically, weak continuity requires that the shape functions for adjacent elements must have the same mean value over the common facet. In the 2D case, the mean of a linear function on the edge is clearly an appropriate choice for this mean value. In three dimensions, however, a linear variation on a facet cannot satisfy the Kronecker-delta property, except in the special case of a triangular facet. Accordingly, the 'ideal' linear variation on element facets in 2D is generalized to the 3D case by using the harmonic function that exhibits piecewise-linear

variation on the edges of the facet, consistent with the Kronecker-delta property. The weak continuity condition on a given facet is then given by zero mean difference between the shape function on the facet, and this ideal variation.

Proceeding along these lines, consider an element with K polygonal facets. Let $\Gamma \subset \partial\omega$ be a typical such facet, with edge $\partial\Gamma$. The 'ideal variation' on Γ for shape function ϕ_a is given by the function ψ_a, as follows:

$$\begin{aligned} \bar{\nabla}^2\psi_a = 0, \quad \mathbf{X} \in \Gamma; \quad \psi_a = f_a, \quad \mathbf{X} \in \partial\Gamma; \quad a \in \partial\Gamma \\ \psi_a = 0, \quad a \notin \partial\Gamma \end{aligned} \tag{7}$$

Here, f_a is the piecewise-linear function on $\partial\Gamma$ such that $f_a(\mathbf{X}_b) = \delta_{ab}$, and $\bar{\nabla}^2$ is the 2D Laplacian in the plane of Γ (see Equation (13) and the text following). The weak-continuity constraint on shape function ϕ_a is then

$$\int_\Gamma (\phi_a - \psi_a)\,\mathrm{d}a = 0 \quad \forall\Gamma \subset \partial\omega, \quad 1 \leqslant a \leqslant n \tag{8}$$

In view of (8), it is required to calculate the mean values on each facet Γ of the harmonic functions ψ_a defined in (7). If node a does not lie on $\partial\Gamma$, then obviously the mean of ψ_a on Γ is zero. Otherwise, close approximations to these mean values can be found without explicitly solving the boundary-value problem (7), as described in Appendix A. Using (4), the weak-continuity constraint (8) becomes

$$\sum_{1 \leqslant \alpha \leqslant M} G_\alpha^a \int_\Gamma q_\alpha\,\mathrm{d}a = \int_\Gamma \psi_a\,\mathrm{d}a \quad \forall\Gamma \subset \partial\omega, \quad 1 \leqslant a \leqslant n \tag{9}$$

which is a non-homogeneous system of K linear constraints on the M coefficients G_α^a for each shape function. The right-hand-side values are easily computed as described in Appendix A, whereas evaluation of the facet integrals of the monomials q_α is discussed in Appendix B.

It bears mention that, whereas the formulae developed in Appendix A yield an approximation to the right-hand side of (9) in the general case (but are exact in some special cases, e.g. for rectangular facets), this approximation is without detriment to the quality of the resulting shape functions. This is because the specification (7) for the ideal variation itself engenders a measure of arbitrariness. What *is* important is that the shape-function mean values on the shared facets of two adjoining elements are identical. This is ensured by enforcing the condition that the facet means of an element's shape functions are equal to values which are characteristic of each facet's geometry. That these mean values derive from (7), or merely from an approximation thereto, is immaterial.

An additional three linear constraints on G_α^a (two in 2D) must be enforced whenever the stress-divergence term in (1) is evaluated using numerical quadrature, as is the case in any practical implementation of the finite element method. A necessary condition for convergence of the method is that the stress-divergence term be exactly integrated in the case of a uniform stress field. This condition was termed the 'integration constraint' by Chen *et al.* [18] (see their equation 19), who examined its consequences in the context of moving-least-squares-based Galerkin methods. Enforcing the integration constraint on each element leads to

$$\int_\omega \nabla\phi_a\,\mathrm{d}v = \int_{\partial\omega} \phi_a \mathbf{n}\,\mathrm{d}a \tag{10}$$

where the left-hand side is to be evaluated using the numerical quadrature rule on the element (see Section 3.4). Equation (10) represents three linear constraints (two in 2D) on the shape-function coefficients G^a_α, after substituting (4) into (10). It bears mention that constraints (10), although necessary for convergence, exert only a very minor influence on the final values of G^a_α. This can be seen by noting that (10) is identically satisfied if both integrals are evaluated exactly, whereas use of numerical quadrature on the left-hand side incurs only a small departure from this circumstance for any polynomial integrand.

Equations (5), together with (10) and (6) (in two dimensions) or (8) and (9) (3D case) constitute a total of $n+K+d$ linear constraints on the selection of the coefficients $G^a_1, \ldots, G^a_M$ for each shape function. These constraints can be collected into the matrix form

$$\mathbf{C}\mathbf{G}^a = \mathbf{b}^a \tag{11}$$

in which $\mathbf{C}$ is a $(n+K+d) \times M$ matrix, and $\mathbf{G}^a$ is the $M \times 1$ vector of polynomial coefficients for shape function a. It is observed that the constraint matrix $\mathbf{C}$ and the vectors $\mathbf{b}^a$ depend only on the element geometry, and further that $\mathbf{C}$ is independent of shape-function number. Because the same $\mathbf{C}$ applies to all shape functions of the element, it need only be factored once in the course of solving for the $\mathbf{G}^a$ coefficients.

3.3. Compatibility and smoothness optimization

The linear constraints (11) represent the minimum conditions on the shape-function coefficients $\mathbf{G}^a$ to produce a convergent method that is finite-element-like in all important respects. Normally, the number M of monomial terms is chosen so that the polynomials (4) are complete through a sufficiently high order to accommodate these $n + K + d$ constraints. This condition requires that the polynomial order m be chosen so that

$$M = (m + d)!/(m!d!) \geqslant n + K + d \tag{12}$$

In practice, it is quite acceptable to take m to be the smallest integer that satisfies this condition.

Usually, choosing the number of monomial terms in this manner results in a larger number of unknown coefficients G^a_α than constraint equations. In this case, the 'excess freedom' is used to improve the quality, in a certain sense, of the shape functions. Specifically, consistent with constraints (11), the shape-function coefficients are to be chosen so that both inter-element compatibility, and smoothness on the element interior, are optimized. To this end, the positive semi-definite functionals

$$F_a = \frac{\alpha}{|\omega|}\int_\omega (\nabla\cdot\nabla\phi_a)^2\,\mathrm{d}v + \frac{\beta}{|\partial\omega|}\int_{\partial\omega}(\bar{\nabla}\cdot\bar{\nabla}\phi_a)^2\,\mathrm{d}a + \frac{1-\alpha-\beta}{\sum|\partial\Gamma|}\sum_\Gamma\int_{\partial\Gamma}(\bar{\bar{\nabla}}\cdot\bar{\bar{\nabla}}\phi_a)^2\,\mathrm{d}s \tag{13}$$

are defined, in which Γ again represents a typical polygonal facet of the boundary $\partial\omega$. In (13), $\bar{\nabla}$ and $\bar{\bar{\nabla}}$ are surface and line gradient operators, respectively, so that $\bar{\nabla} = \nabla - \mathbf{n}\mathbf{n}\cdot\nabla$ and $\bar{\bar{\nabla}} = \boldsymbol{\lambda}\cdot\nabla$, where $\mathbf{n}$ is the unit normal to the surface and $\boldsymbol{\lambda}$ is the unit tangent to the line. The differential operators in each of the integrands above represent the Laplacian in three, two, and one dimensions, respectively. The three terms characterize the smoothness of the shape function over the element's volume, facets, and edges, respectively. Further, the second term is complementary to constraint (8), (9), which requires that the Laplacian of the shape functions have zero mean on each facet. The parameters $\alpha, \beta \in [0, 1]$, $\alpha+\beta \in [0, 1]$, determine the relative

importance of each of the three measures of smoothness. The last term in (13) is omitted in the 2D case.

The remaining freedom in the shape-function coefficients G_i^a, following enforcement of constraints (11), is employed to minimize F_a. Under the specification (4), the functionals defined in (13) become

$$\begin{aligned} F_a &= \mathbf{G}^{a\mathrm{T}}\mathbf{A}\mathbf{G}^a \\ A_{\alpha\beta} &= \frac{\alpha}{|\omega|}\int_\omega q_{\alpha,ii}q_{\beta,jj}\,\mathrm{d}v + \frac{\beta}{|\partial\omega|}\int_{\partial\omega}(q_{\alpha,ii}-n_in_jq_{\alpha,ij})(q_{\beta,kk}-n_kn_lq_{\beta,kl})\,\mathrm{d}a \\ &\quad + \frac{1-\alpha-\beta}{\sum|\partial\Gamma|}\sum_\Gamma\int_{\partial\Gamma}\lambda_i\lambda_jq_{\alpha,ij}\lambda_k\lambda_lq_{\beta,kl}\,\mathrm{d}s \end{aligned} \tag{14}$$

All integrands in (14) are polynomials in the spatial co-ordinates, and may therefore be reduced to combinations of quadratures on straight line element edges through appropriate use of the divergence and Stokes' theorems (see Appendix B). These integrals, in turn, can be efficiently evaluated by employing Gaussian quadrature in a recursive strategy to build up a list of values corresponding to each required monomial term. The $M \times M$ matrix $\mathbf{A}$ in (14) is symmetric and positive semi-definite. The zero eigenvalue has multiplicity four (three in 2D), corresponding to the constant and linear monomials in $\{q_\alpha\}$. With respect to computational efficiency, it is important to observe that the same matrix $\mathbf{A}$ appears in each F_a; i.e. as with the constraint matrix $\mathbf{C}$ (see Equation (11)), $\mathbf{A}$ is independent of the shape-function number a. These matrices must therefore be factored only once per element, and not once per shape function.

3.4. Integration rule

Application of the VETFEM to linear problems, wherein the stress $\mathbf{P}$ in (1) is replaced by a symmetric rank-four modulus tensor times the small-strain tensor, allows for exact evaluation of the weak-form integrals. In this case, all integrands are polynomials in the physical Cartesian co-ordinates of the problem. The integrals of monomials through a sufficiently high order must, in any case, be calculated on each element in order to form the shape-function coefficients, as described above. In the non-linear case, however, the material state must be sampled at discrete points within each element, and numerical quadrature must then be used to form the stress-divergence term in (1).

In one dimension, the notion of optimality of an integration rule can be clearly defined. Quadrature rules on conventional quadrilateral and hexahedral elements are generally obtained by extension of the optimal 1D Gaussian rules. On general polygons or polyhedra, however, no parallel notion of optimality can be formulated. Instead, the placement of the integration points, and the subsequent assignment of a weight to each point, are, of necessity, regarded as separate issues.

For a number of reasons, among them facilitation of solution remapping following remeshing in large-deformation problems, it is desirable to have nodes and integration points in one-to-one association on each element. Accordingly, the quadrature rule on VET elements is constructed as follows. First, the element is partitioned into n subregions, one for each node. In 2D, these subregions are obtained by connecting the centroid of the element to the midpoints of each element side. In 3D, the subregions are constructed in an analogous way (Figure 1): first, the

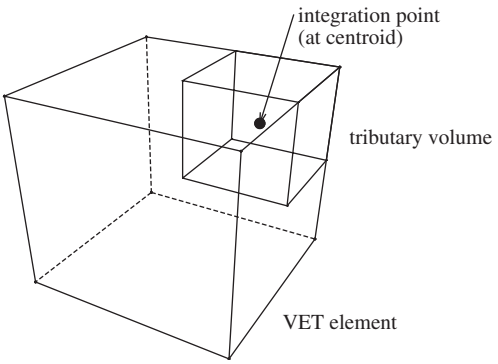

Figure 1. Illustration of the geometric construction of the quadrature rule for 3D VET elements.

centroids of each facet are connected to the midsides of the facet edges, and then the edge midsides as well as the facet centroids are connected to the element centroid. In this manner, the subregion associated with a given node is built up from a collection of pyramids with quadrilateral bases and triangular sides. The area (2D case) or volume (3D), as well as the centroid, of each subregion can be easily computed by reducing these quadratures to integrals along the edges of the subregions. The integration-point locations are then taken to be the centroids of the subregions, and the corresponding weights are the areas (volumes) of the subregions.

Among the desirable features of this simple integration rule is that an arbitrary linear function on the element is exactly integrated. By way of comparison, a conventional bi-unit square Q4 element has its four integration points at $(\pm 0.5774, \pm 0.5774)$, whereas for a VET element with the same geometry, the co-ordinates are $(\pm 0.5, \pm 0.5)$ and the weights are identical to those of the conventional element.

The theoretical formulation of the VET shape functions and quadrature rule is now complete. Before passing to algorithmic aspects, the VET element formulation is compared and contrasted to other recent approaches related to natural-neighbour-based approximation. The NEM differs from the VETFEM in the sense that the former is a *meshless method*, i.e. the approximation space depends only on the location of the nodes and not on an attendant partition of the domain into elements. However, the NEM can be applied to individual polytopes associated with small numbers of neighbouring points, thereby generating finite-element-like basis functions on a polyhedral mesh. This approach was taken by Idelsohn *et al.* [26] to produce a 'meshless finite element method.' Their method is conforming (i.e. the basis functions are continuous), but in order to achieve this property, the point–element associations must be made according to a strict set of rules. Sukumar and Tabarraei [27] took a similar approach by generating conforming shape functions on (2D) regular polygons using a nearest-neighbour interpolant. In that work, shape functions on general convex polygons in the problem's physical space are obtained by invoking a conventional isoparametric mapping. This approach requires that each physical element with n nodes be homeomorphic to the canonical element with n nodes. This is trivially the case in 2D but would require restrictive conditions on the element–facet–node associations in 3D.

A significant distinction between meshless methods and finite element methods exists with respect to numerical quadrature of the weak-form integrals. In finite element methods, the fact

that the shape-function supports are precisely delineated by the element boundaries leads to highly accurate quadrature rules, even when the elements are irregular in shape, as with VET elements. This powerful feature of FEM-like methods was perhaps underappreciated until a substantial body of research in meshless methods had accumulated. Accordingly, much recent work has been devoted to devising accurate quadrature rules for meshless methods. In the case of the NEM, the quadrature rule is usually designed to exploit the Voronoi diagram for the domain, which must be constructed in any case in order to determine nearest-neighbour relationships as well as to construct the shape functions. However, Voronoi-cell boundaries do not coincide with shape-function supports for nearest-neighbour interpolants, which implies the need for large numbers of integration points and carefully chosen locations [12]. The stabilized nodal integration scheme of Chen *et al*. [18], as applied to the NEM by Gonzalez *et al*. [12] and also by Yoo *et al*. [17], represents a promising development, in that high accuracy, satisfaction of the integration constraint (see Section 3.2), and freedom from volumetric locking are reported.

3.5. Algorithm

The algorithmic problem for the determination of the polynomial coefficients may now be stated as follows: minimize F_a as defined in (14), subject to constraints (11). A convenient and efficient method of solution is to first factor the constraint equations (11) using a generalized Gaussian elimination algorithm [19], resulting in the representation

$$\mathbf{G}^a = \mathbf{B}\mathbf{g}^a + \mathbf{d}^a \tag{15}$$

Here, $\mathbf{B}$ is $M \times (M - d - n - K)$ and each $\mathbf{d}^a$ is $M \times 1$; these are the outputs of the factoring operation applied to (11). The $\mathbf{g}^a$ represent the $M - d - n - K$ remaining unknowns among the coefficients for each shape function. In the elimination algorithm referenced above, matrix $\mathbf{B}$ contains a single non-zero entry of unity on each row, so that vector $\mathbf{g}^a$ contains simply the subset of unknowns in $\mathbf{G}^a$ that remain free. These free unknowns are determined by combining (15) with (14) and then minimizing F_a with respect to $\mathbf{g}^a$, resulting in the linear equations

$$\mathbf{B}^{\mathrm{T}}\mathbf{A}\mathbf{B}\mathbf{g}^a = -\mathbf{B}^{\mathrm{T}}\mathbf{A}\mathbf{d}^a \tag{16}$$

The coefficient matrix on the left-hand side of (16) is square, with dimension $M - d - n - K$, and is guaranteed to be of full rank. This matrix must be factored only once for each element, with repeated back-substitution to obtain the vectors $\mathbf{g}^a$. Once these are known, the polynomial coefficients for each shape function are recovered from (15).

In summary, the shape functions on each VET element are given by polynomials in the physical co-ordinates of the problem. The coefficients appearing in these polynomials are set so that the Kronecker-delta property of finite-element-type shape functions is realized, along with inter-element compatibility *in the mean* on each facet, and consistency of the quadrature rule. The remaining freedom in the coefficients is utilized to further refine the inter-element compatibility, as well as to achieve favourable smoothness on the element interior. It is noted that the VET methodology as described herein differs in some of its details from the original development given in Reference [19].

Algorithmically, the calculation proceeds as follows:

(1) Set an appropriate polynomial order m for the shape functions. A simple and effective choice is to simply take m to be the smallest integer that satisfies (12).

(2) Set the integration-point locations and weights as described in Section 3.4.
(3) Exactly evaluate integrals of monomials through the required order on the element interior, on each facet, and on each edge. This step is described in detail in Appendix B, in which it is shown how the interior and facet integrals can be reduced to 1D edge integrals. The required integrals can be efficiently built using a recursive strategy.
(4) Form the constraint matrix $\mathbf{C}$ for the element, and one right-hand-side vector $\mathbf{b}^a$ for each node of the element (see Equations (5), (6), (8)–(11)).
(5) Factor the constraint matrix, and apply back-substitution repeatedly to obtain $\mathbf{B}$ and $\mathbf{d}^a$ (15).
(6) Form the system of equations (16), factor, and then apply back-substitution repeatedly to solve for the $\mathbf{g}^a$.
(7) Finally, recover the shape-function coefficients from (15).
(8) Optionally, the shape-function gradients can be evaluated at the integration points and stored; the shape-function coefficients may then be discarded. This step makes for very efficient element-level calculations in the course of the analysis itself, and generally incurs reduced memory requirements.

This sequence of steps constitutes the element-level calculations required to form the polynomial shape functions. The shape functions and their gradients are subsequently used in a manner identical to conventional isoparametric shape functions, except that no Jacobian matrix or Jacobian determinant is required.

Typical 2D VETFEM shape functions are illustrated in Figure 2. The 2D element contains five nodes. The figure illustrates surface plots of the five shape functions as constructed by the above procedure. As may be seen from the figure, some of the shape functions exhibit slight

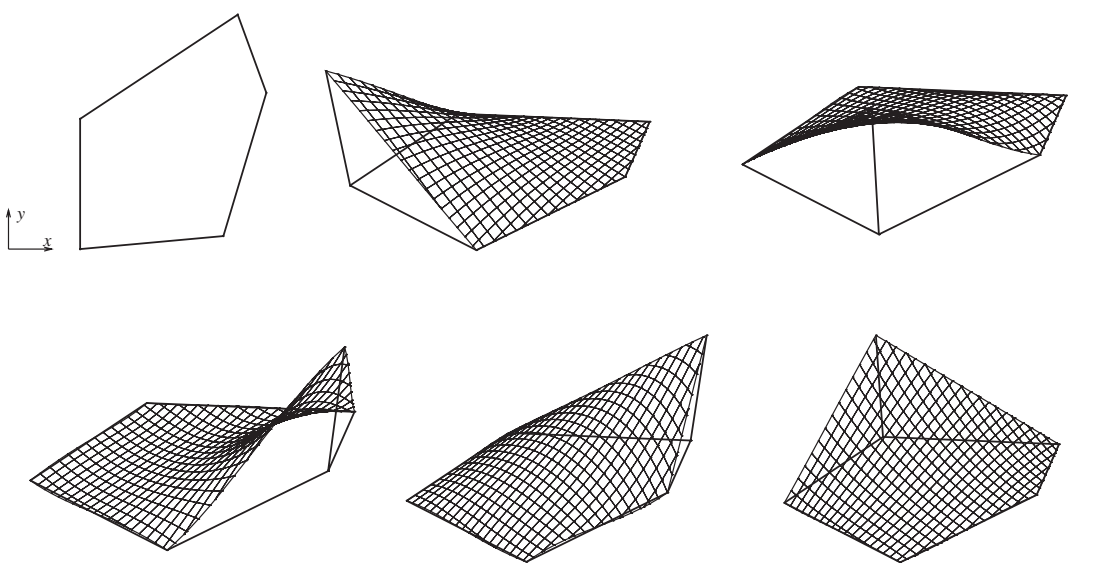

Figure 2. Typical 2D VET element (upper left), and surface plots of its five shape functions. The shape functions shown are fourth-order polynomials containing 15 terms each.

departures from piecewise linearity on all edges, as may be expected in view of the finite order of the shape-function polynomials.

4. BOUNDARY CONDITIONS

4.1. Displacement BCs

The Kronecker-delta property (5), which VET shape functions are constrained to exhibit, ensures that the displacement approximants are true interpolants of the nodal data. This is, of course, the case in the conventional FEM as well, and it is the property which greatly simplifies the enforcement of displacement boundary conditions. As with the conventional method, displacement BCs are enforced simply by constraining the nodal displacements on Γ_u to take their given values. It also bears mention that the Kronecker-delta property renders enforcement of continuity at interior interfaces a trivial matter. Thus, regions of VET elements can be joined with regions of standard four-node quad or eight-node hex elements simply through the expedient of shared nodes on their common interface.

4.2. Traction BCs

With reference to (1), the traction-BC term in the variational BVP requires the evaluation of

$$\int_{\Gamma} \phi_a \bar{\mathbf{p}} \, \mathrm{d}a \tag{17}$$

on element facets $\Gamma \subset \partial_t \Omega$ on which traction is specified. Remaining within the total-Lagrangian setting outlined in Section 2, $\bar{\mathbf{p}}$ is the Piola traction vector. If it is specified to be constant on the facet, then (17) can be exactly evaluated and, by virtue of the weak-continuity constraint (8), is guaranteed to produce consistent nodal forces at boundary nodes only, and zero at interior nodes. On the other hand, if a non-constant Piola traction $\bar{\mathbf{p}}$ is specified, or if a Cauchy traction or pressure is specified, then (17) in general produces small force contributions at interior nodes that share an element with a traction-BC facet. This eventuality does not normally present any algorithmic or practical problems. However, if for some reason it is deemed to be objectionable, the traction-BC term in (17) can be approximated by

$$\int_{\Gamma_t} \bar{\mathbf{p}} \, \mathrm{d}a \times \int_{\Gamma_t} \phi_a \, \mathrm{d}a \tag{18}$$

This expression provides statically equivalent nodal forces on the boundary only. Further, its implementation is economical, as the second term in (18) can be calculated and stored for each traction-BC facet at the time of the initial element-level calculations.

5. CONVERGENCE OF THE VETFEM

Due to the presence, in the general case, of slight discontinuities in the basis functions across inter-element facets, the VETFEM is properly regarded as a non-conforming finite element

method. The convergence of the method, when applied to variational BVPs of type (1), therefore requires verification. A brief, descriptive account of the basis for the convergence proof is given here.

In conventional, conforming Galerkin methods, the basis functions are continuous, and therefore span a function space which is a subspace of the solution space V^0 described in Section 2. In this case, it is easily shown that the Galerkin approximation of the variational BVP amounts to a projection of the exact solution onto the approximating subspace. With this *best-approximation property* in hand, convergence under mesh refinement hinges on the so-called *approximability property*: it is both necessary and sufficient that any function in the space V^0 can be approximated arbitrarily closely, under a suitable norm, from within the approximating subspace associated with a sufficiently fine mesh.

In the case of non-conforming methods, each approximation space contains functions which are discontinuous across inter-element facets. These approximation spaces are not, therefore, subspaces of V^0, and consequently the best-approximation property mentioned above fails to hold. The approximability property remains a requirement for convergence, but an additional consideration now emerges: it is also necessary that the *exact* solution comes ever closer to satisfying the *approximate* problem as the mesh is refined. In essence, the approximability property requires that the expanding (under mesh refinement) approximation space contains the exact solution in the limit, whereas the so-called *closedness property* further requires that the approximate problem itself comes ever closer to the actual problem as the mesh is refined. Closedness is automatic in the case of conforming methods, because the approximate and exact variational problems are in fact essentially identical—the former is merely the latter, restricted to a subspace of the full solution space V^0.

The properties of approximability and closedness, briefly and imprecisely described here, are properties of the function spaces produced by the approximation method—finite element or otherwise—as the mesh is refined. Taken together, these two properties are necessary and sufficient for the convergence of the approximate solution to the exact one [21], under suitable conditions relating to the ellipticity of the underlying BVP. In the context of finite elements, approximability requires, in essence, that arbitrary polynomials of sufficiently high order can be exactly represented within each element. For second-order BVPs, the required polynomial order is one (linear). Closedness, on the other hand, is only in question for non-conforming methods, and is realized so long as the inter-element discontinuities are sufficiently 'weak,' in a certain sense.

Establishment of the closedness property for candidate non-conforming finite elements is the subject of the *generalized patch test* of Stummel [21]. Despite the connotation of the name 'patch test,' Stummel's GPT does not involve numerical experimentation. In fact, the GPT is not trivial to apply for specific elements. Motivated by this fact, beginning with Stummel's GPT. Shi [24] devised a sufficient condition for convergence of non-conforming finite element methods which is easier to apply than the original GPT. Shi's 'F-E-M' test involves two criteria: the F-test which involves element facets only and requires, in essence, that inter-element jumps in the basis functions have zero mean on each facet, and the E-M-test which is applied only if the F-test fails, and which involves the element interiors. By virtue of constraint (6), (8), VETFEM interpolants always have zero mean jump across inter-element facets. The F-test of Shi [24] thereby directly supplies a sufficient condition for the convergence of the VETFEM in all cases in which a conforming Galerkin method would be expected to converge. It is mentioned that the F-test also ensures that basis functions associated with interior nodes vanish, in the

mean, on boundary facets. This, in turn, guarantees that (continuous) displacement boundary conditions are exactly satisfied in the limit under mesh refinement, as required.

6. NEAR-INCOMPRESSIBILITY: KINEMATIC ENHANCEMENT

It has long been recognized that, in the finite element method, strict enforcement of incompressibility at the integration points may so constrain the deformation field that the approximate solution remains far from the exact one, even for very fine meshes. The extent to which this *volumetric locking* phenomenon occurs depends on the order of the displacement interpolant, and on the element's quadrature rule. Bilinear/trilinear quad/hex elements, for which VET elements are, in a sense, a generalization, suffer from volumetric locking when fully integrated.

Remaining within the confines of a displacement-only problem formulation, volumetric locking is remedied by suitably modifying the manner in which the dilatation is sampled. This modification can take one of two forms. In the first, the order of the element's quadrature rule is reduced sufficiently so that enforcement of incompressibility at the remaining integration point(s) does not undesirably constrain the volume-preserving modes of deformation. This approach is popular in the explicit-dynamics community, principally because it drastically reduces the number of constitutive evaluations that must be performed per element: the number of integration points for eight-node hex elements, for example, reduces from eight to one. However, with one-point integration, certain isochoric deformation modes go from overconstrained to completely unresisted, thereby requiring some form of artificial stiffness to prevent mesh entanglement.

In the second approach, the kinematic information at each integration point is modified before it is used to evaluate the stress. Specifically, the description of the deformation at each integration point is altered to reflect an average dilatation for the entire element, while retaining the 'correct,' local deviatoric component. This is the essential idea behind so-called *strain projection* or *assumed-enhanced-strain* methods [28–30]. This approach might be considered a generalization of selective reduced integration, in the sense that a correspondence exists between the two approaches for isotropic linear elasticity. Obviously, the 'enhancement' or modification of the strain-like quantities at the integration points must not be too drastic, or else convergence of the overall method to the exact solution is lost. At a minimum, the modification must disappear in the case of uniform deformation.

The assumed-enhanced-strain approach can be extended to finite deformations in various ways, as described, e.g. by de Souza Neto *et al.* [31]. A formulation which bears some similarity to theirs is adopted here. Let $\mathbf{X}$, $\bar{\mathbf{x}}$, and $\mathbf{x}$ be position vectors of a typical material point in the reference, beginning-step, and end-step configurations, respectively, and define the deformation gradients

$$\hat{\mathbf{F}} = \frac{\partial \mathbf{x}}{\partial \bar{\mathbf{x}}}, \quad \bar{\mathbf{F}} = \frac{\partial \bar{\mathbf{x}}}{\partial \mathbf{X}}, \quad \mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \hat{\mathbf{F}}\bar{\mathbf{F}} \tag{19}$$

In both implicit and explicit calculations, the beginning-step deformation gradient $\bar{\mathbf{F}}$ is known at the outset on each step. The *incremental deformation gradient* $\hat{\mathbf{F}}$, on the other hand, depends on the motion in the step, and contains the kinematic information needed to update the material state at each integration point. $\hat{\mathbf{F}}$ constitutes the input to the 'strongly objective' kinematic

algorithm described in Reference [32], from which there emerges a constant stretch rate $\mathbf{D}$ and an incremental rotation $\hat{\mathbf{R}}$. The constitutive equations are integrated over the step under the assumption of $\mathbf{D}=$constant and zero rotation, followed by an impulsive rotation $\hat{\mathbf{R}}$ with no further strain.

A kinematic enhancement to remedy volumetric locking can be formulated in this setting as follows. At each integration point, before entering the strongly objective kinematic algorithm, $\hat{\mathbf{F}}$ is *replaced* by $\tilde{\mathbf{F}}$, where

$$\tilde{\mathbf{F}}=\left(\frac{A}{\hat{J}}\right)^{1/d}\hat{\mathbf{F}},\quad \hat{J}=\det\hat{\mathbf{F}},\quad A=\det\mathbf{f},\quad \mathbf{f}=\frac{1}{|\bar{\omega}|}\int_{\bar{\omega}}\hat{\mathbf{F}}\,\mathrm{d}v \tag{20}$$

$\tilde{\mathbf{F}}$ is defined by scaling $\hat{\mathbf{F}}$ at each integration point in such a manner that the volume ratio $\det\tilde{\mathbf{F}}$ is a single constant A for the entire element. From the third of (20), this constant value is the volume ratio associated with the *average incremental deformation* $\mathbf{f}$ for the element. In (20), $\bar{\omega}$ is element ω convected to the beginning-step configuration. Also, in (20) the $d=2$ case corresponds to plane strain, in which all deformation gradients are 2D, whereas $d=3$ for all other cases. The integral in (20) can easily be evaluated using the quadrature rule for the element, or, as an alternative, it can be evaluated exactly by using the divergence theorem to convert it to a boundary integral.

Under the replacement (20), the incremental dilatation at all integration points is equal to the incremental dilatation averaged over the element. The deviatoric aspect of the incremental motion, however, retains its local value at each integration point. The element stiffness thereby retains its full rank; at the same time, enforcement of incompressibility at all integration points does not lead to overconstraint of volume-preserving modes of deformation. This formulation has been found in practice to completely obviate volumetric locking in VET elements.

7. ILLUSTRATIVE ANALYSES

7.1. Patch tests

Passage of the engineering (or numerical) patch test [22, 23], in which a 'patch' of elements is subject to boundary conditions which implicate a uniform-stress solution, is generally taken to indicate good behaviour of the approximation method. By way of illustration, numerous 2D and 3D numerical patch tests were performed using the VETFEM (Figure 3). All of the meshes shown in the figure were subjected to both displacement and traction boundary conditions consistent with a uniaxial stress state. In the latter case, minimally sufficient displacement BCs were also applied to prevent rigid-body translation and rotation. The material model was taken to be linear isotropic hypoelasticity which, for the patch-test analyses, resulted in very small strains ($<10^{-5}$). In all cases, integration-point stresses and nodal displacements were found to be consistent with the expected uniform-stress solution, to within machine precision.

7.2. 2D convergence demonstration

The performance of the VETFEM in practice is illustrated here with reference to a simple 2D problem in elastostatics. The anvil-like domain shown in Figure 4 was loaded by a uniform normal traction on the left half of its top edge, and was subject to a roller constraint along its

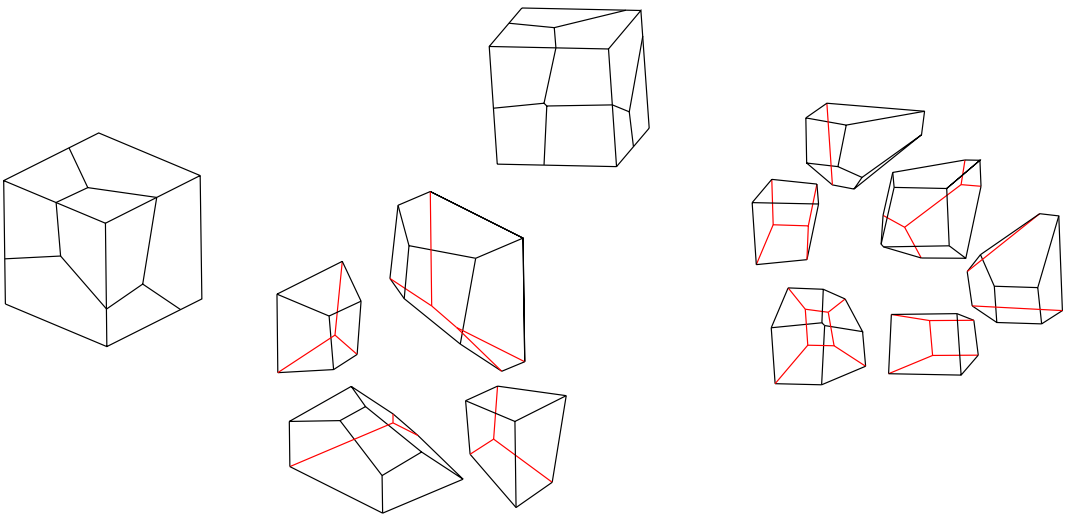

Figure 3. Illustrative 3D VET meshes used for numerical patch tests, with exploded views.

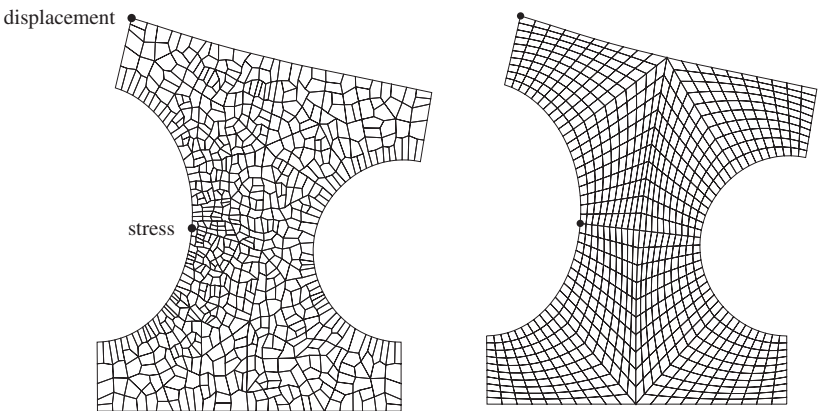

Figure 4. Deformed VET and conventional FEM meshes of a 2D plane-strain linearly elastic domain. The body is subject to a roller constraint along its bottom edge, and a uniform normal traction on the left half of its top edge.

entire bottom edge. The representative meshes shown in the figure consist of 680 VET elements and 800 four-node quadrilateral elements, respectively. The Poisson ratio was 0.3, and plane-strain conditions prevailed. In this study, the displacement magnitude and the vertical-direction normal stress at the indicated locations were compared between the VET and conventional FEM analyses. The stress was recovered at the indicated point by taking the area-weighted average of the stresses at the two neighbouring integration points.

A reference solution was obtained using a very fine (4608 elements) conventional quad mesh. Analyses using conventional quad elements and using VET elements were then performed at four

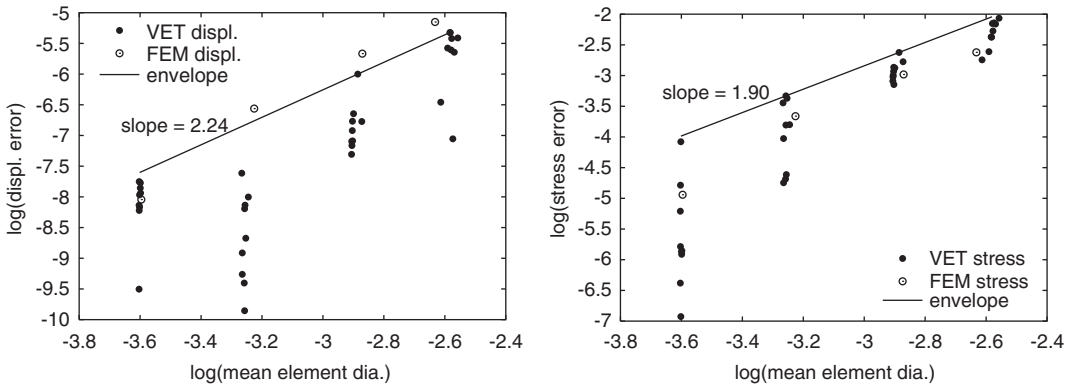

Figure 5. Log–log plots of error in displacement magnitude and in vertical-direction normal stress at the points indicated in Figure 4. At each level of mesh refinement, one conventional FEM analysis and 9 different VET analyses were performed.

levels of mesh refinement. At each level, one conventional FE analysis and nine VET analyses were performed. The quad meshes contained 512, 800, 1568, and 3200 elements at the four refinement levels, respectively, with the VET meshes containing similar numbers of elements. The VET meshes were generated using a simple subdivision algorithm that is applicable to any plane region. The algorithm can be seeded differently to produce any number of mesh realizations at a similar level of refinement.

Another effective method of generating a VET mesh is to fill the region with a distribution of points, and then compute the Voronoi diagram of the region with respect to these points. One slight disadvantage of this approach, as compared to the subdivision strategy used here, is that the very short cell edges that Voronoi diagrams can contain can result in shape functions whose approximating power is low in relation to the number of dof involved. This is easily repaired, though, by collapsing very short edges following construction of the Voronoi diagram. A further slight disadvantage is the need to make interior/exterior determinations for all candidate Voronoi seed points. This can actually be a significant burden for complex regions. In any case, virtually any reasonable method for subdividing planar regions into low-order polygons is suitable for the construction of 2D VET meshes. No single method has been found to possess clear advantages over any other in relation to the quality of the approximation space that results, leaving the choice to be made on the basis of algorithmic or computational expediency.

Figure 5 shows log–log plots of the magnitude of the displacement error and the stress error at the points indicated in Figure 4, for the 36 VET analyses and for the four conventional FE runs. A line corresponding to the envelope of the error in the VET analyses is also shown on the plots. The slopes are 2.24 for the displacement error and 1.90 in the case of the stress. It is clear from the plots that, for the problem studied, the error in the VET results is comparable to that of the conventional FE analyses at similar levels of refinement.

A thorough comparison of the computational resources required to complete comparable VET and conventional FE analyses is presented in the next subsection, in connection with the more demanding problem presented there. However, as a point of reference, a brief comparison is given here for the above linearly elastic problem. Two analyses were performed with nearly

identical numbers of dof: a VET analysis with 2439 elements and 3378 nodes, and a conventional quad-element analysis with 3200 elements and 3361 nodes. The code used to perform these analyses is a general-purpose platform, such that all aspects of the two calculations (e.g. the linear solver, equation assembly) except the shape-function formulation were identical. The VET calculation required 6.50 cpu seconds on an Intel Xeon processor running at 2.4 GHz, of which 0.47 s were consumed by the shape-function calculations. The conventional FE analysis, on the other hand, required 5.22 s, with a negligible fraction going to shape-function calculations.

7.3. 3D elastic–plastic problem

This boundary-value problem corresponds to an elastic–plastic cube with an initially spherical void at its centre, loaded by uniform normal tractions acting on one pair of opposing faces. The analysis involved one octant of the cube, with appropriate symmetry boundary conditions on three orthogonal planes. The material response is given by the J_2 flow theory of elastoplasticity, with isotropic hardening of the form

$$Y = Y_0[1 + (e/\bar{e})^n] \tag{21}$$

Here, Y is the current flow stress (with initial value Y_0), e and $\bar{e}$ are the effective plastic strain and a reference value, respectively, and n is the hardening exponent. Values representative of a ductile aluminium alloy were used in the analyses: $Y_0 = 50$ ksi, $\bar{e} = 0.1$, $n = 0.15$, $E = 10\,000$ ksi, $\nu = 0.3$. With these values, the uniaxial stress-strain curve quickly flattens out after initial yield.

Two analyses were performed: one using 1694 VET elements and the other with 1764 conventional eight-node hex elements with full integration (Figure 6). The VET mesh was created by first generating a structured, spherically symmetric hex mesh—a trivially easy task—and then 'chopping down' the mesh to the desired final shape. The end result is a mesh containing hex elements in the interior, and a variety of other polyhedral shapes near the boundary (Table I). As with the 2D problem of the previous subsection, a suitable VET mesh for this problem could have been generated by constructing the Voronoi tesselation of the domain with respect to a distribution of interior points. However, the generation of a strictly interior set of points, as well as the subsequent Voronoi construction, can be algorithmically and computationally intensive tasks, particularly for complicated, non-convex domains and fine meshes. In any case, a central interest here is to explore ways to supplement conventional finite element technology with the greater meshing flexibility afforded by the VET formulation. This greater flexibility is most needed near the boundary of complex shapes.

In both analyses, the uniform-traction boundary condition was applied monotonically over 78 increments, up to a final value of $1.928Y_0$. This is well above the apparent saturation value of the yield stress under uniaxial stress. The solutions are compared in Figure 7, which shows the major- and minor-axis diameters vs load level for both the VET and conventional FE analyses. As can be seen in the figure, these solution parameters are essentially indistinguishable over most of the loading range. Extensive plastic flow occurs near the end of analysis; the final values differ between the two analyses by 0.2% (minor axis) and 1.3% (major axis). In general, the VET analysis exhibited slightly higher levels of deformation at a given load level.

These analyses were carried out on an Intel Xeon processor running at 2.4 GHz. The load was applied over 78 steps for both the VET and conventional-hex meshes. Although the VET mesh consisted of approximately 70% hex elements, the VET element formulation was applied

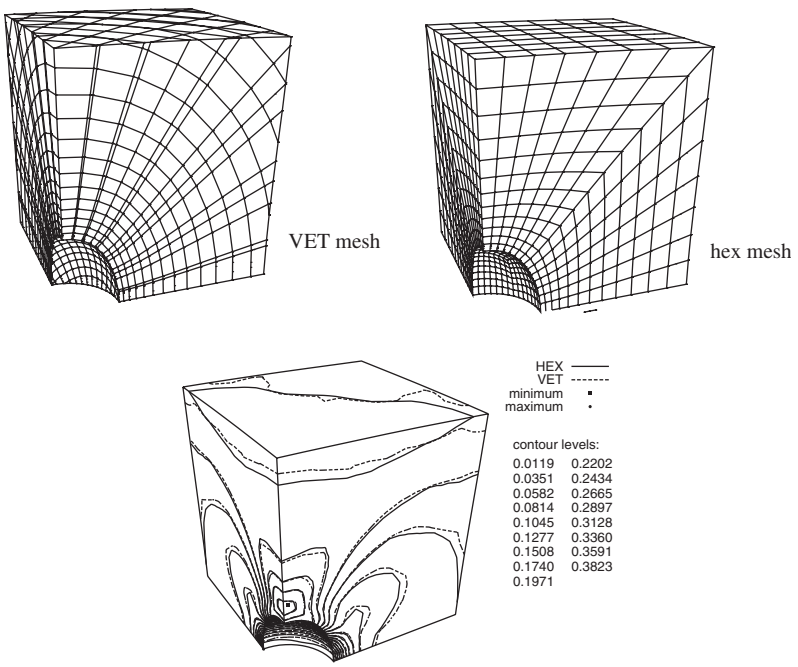

Figure 6. Deformed VET and conventional FE meshes of a single octant of a cube with an initially spherical void at the centre. The cube is loaded by a uniform tensile traction on the top and bottom faces. At bottom, contour lines of equivalent plastic strain at identical levels are superimposed for the VET and FE cases.

Table I. Distribution of element geometries for the VET mesh shown in Figure 6.

Number of elements with:	Nodes	Facets
22 (1.3%)	4	4
232 (13.7%)	6	5
1184 (69.9%)	8	6
246 (14.5%)	10	7
10 (0.6%)	12	8

to all elements in the mesh in order to contrast the computational efficiency of the two methods. Various indices of problem size and computational resources required are compared in Table II. The computer code used for both analyses employs a frontal solver for the solution of the global linear system of equations; the maximum dimension reported in the table refers to the largest size of the active equation system encountered in the course of the solution. It is seen from the table that the hex and VET meshes give rise to computational problems that are very similar in magnitude. A purely Lagrangian approach was used, so that the initial reference configuration is discretized. Among other advantages, in this approach the shape-function gradients need be calculated and stored only once for the entire analysis. Although the shape-function calculations

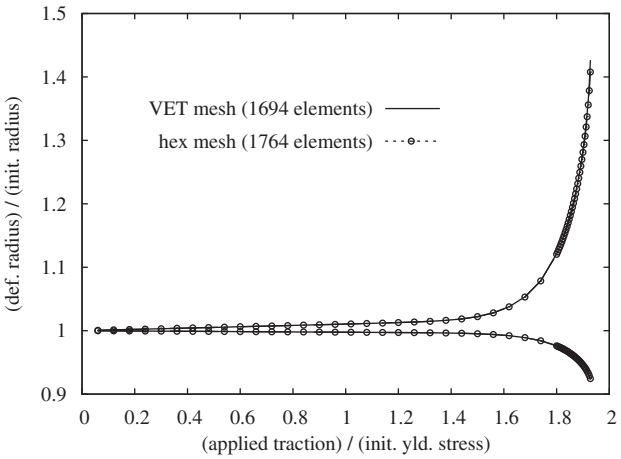

Figure 7. Comparison of the VET and conventional FEM solutions for the major- and minor-axis diameters of a void in an elastic–plastic cube subject to uniaxial tension.

Table II. Comparison of computational problem size and processor usage for the VET and conventional FE problems illustrated in Figure 6.

Quantity	VET analysis	Hex-element analysis
Number of nodes	2257	2197
Number of elements	1694	1764
Maximum dimension	697	687
cpu times (s):		
Shape-function calculations	22.35	0.02
Other element-level	468.37	497.11
Equation assembly	9.09	8.40
Equation solution	3187.74	3778.21
Total cpu time	3712.73	4309.31

are clearly more intensive for VET elements than they are for conventional hex elements, in both cases the cpu time devoted to these calculations is negligible compared to the overall cpu time. This has been found consistently in both linear and non-linear analyses. For this particular problem, the overall cpu time was actually 14% lower for the VET analysis than for the conventional FE run. More generally, the run times for similar-sized problems have been found to be comparable, with no consistent advantage for either method.

8. CLOSURE

The primary attraction of the approximation method described herein is its greater flexibility, over the conventional FEM, in relation to mesh generation. This flexibility is realized

by abandoning the isoparametric transformation of the conventional FEM in favour of a direct polynomial formulation in the physical co-ordinates. The topological restrictions attending the parent-to-physical element mapping thereby disappear, allowing elements to take arbitrary polyhedral form. Others have devised finite-element-like methods with a similar goal in mind [33, 34]. However, the present formulation is unique in that basis functions with the requisite smoothness properties are explicitly formed. The favourable properties of the conventional FEM, specifically in relation to formulation of effective quadrature schemes and enforcement of boundary conditions, are thereby retained. In this connection, the *generalized finite element method* (GFEM) formulation of Strouboulis *et al*. [35] also must be mentioned. In this approach, the local enrichment of the basis functions that is the defining characteristic of the GFEM [36] is further enhanced through the realization that the boundary of the domain need not coincide with element boundaries. Accommodation of 'partial elements'—i.e. those elements whose interiors intersect a part of the domain's boundary—incurs some additional, but tractable, complication related to quadrature and enforcement of boundary conditions.

The primary attraction cited above can also be claimed by meshless methods, among which the NEM (see previously cited references) appears to hold considerable promise for the solution of problems on complex domains. In comparison with the VETFEM, the NEM has in its favour the fact that any distribution of interior points provides a usable approximation, although the Voronoi diagram of the domain must be constructed with respect to these points, and nearest neighbours must be determined for each point. The VETFEM requires that the domain be partitioned into polyhedra from the outset. The VETFEM generally results in cpu times that compare closely to those of corresponding conventional FE analyses, whereas reported comparisons between NEM and FEM runs have shown the processor demand to be about two to four times greater for the NEM for similar problems [16]. This is likely due to the generally larger number of basis functions whose supports overlap at any given point, leading to larger system bandwidths than is typical for finite-element-like methods. Other favourable properties exhibited by the VETFEM—which are more practical in nature—are its algorithmic compatibility with existing finite element platforms; and the applicability, with little or no modification, of finite-element post-processing technology. Ultimately, method selection for real-world technological applications is driven by a great many factors, of which mathematical soundness and accuracy are but two.

The VET element formulation, as presented here, does engender one important geometric constraint, and that is the requirement that the element facets must be planar. The main reason for this restriction relates to the evaluation of monomial integrals over the element and its boundary, as required in Equations (9), (10), (14). In the case of warped facets, these calculations—described in the appendices—would be more complex, and would require the introduction of approximations in order to be tractable. For this reason, it seems more desirable to accept the fairly mild restriction of planar facets, and thereby avoid a more complex—and costly—set of element-level computations.

It also might well be asked if the VET element formulation can be modified to accommodate higher-order shape functions. One or more (or all) element edges can, in fact, be endowed with midside nodes, with shape functions varying in an approximately quadratic fashion along these edges. The requisite modifications to the element-level calculations presented herein are straightforward. It bears mention, however, that fully second-order VET elements would typically contain a fairly large number of nodes, thereby expanding the number of monomial terms in the shape functions. This would increase the element-level computational effort significantly.

A central issue not directly addressed herein is that of making the best use of the flexibility afforded by the VETFEM. Discretizing a domain into arbitrary polyhedra, while generally much simpler than meshing the same domain with hexahedra, is not necessarily a trivial task. In 2D, a fairly simple algorithm based on recursive subdivision has been found to be effective on arbitrarily complex domains. This scheme was used to generate the meshes for the convergence study of Section 7.2. In this approach, each element is given a score based on its number of nodes and its size relative to some target size. The element with the highest score is subdivided into either two or three 'child' elements, based on a simple set of rules. The process is applied recursively, beginning with the entire domain. Any number of mesh realizations can be generated by perturbing the first subdivision 'by hand.' Further, complex patterns of mesh gradation and shape anisotropy can be easily accommodated by the algorithm. On the other hand, a conceptually similar 3D VET meshing algorithm would be considerably more complicated, particularly if the same high standard of robustness is to be achieved. Another, probably simpler meshing construct is suggested by the 3D example problem of Section 7.3. In this problem, a space-filling structured hex mesh is reduced to the domain of interest through a series of 'cutting' operations. Work is ongoing to generalize this procedure to more general Boolean operations, with the goal of automatically producing VET meshes from a trivial-to-generate structured hex mesh along with a CAD representation of the domain.

APPENDIX A

The purpose of this appendix is to derive an estimate for the right-hand side of (9). In view of the fact that a typical function of the type defined by (7) is considered, the subscript $()_a$ is suppressed. As specified in (7), ψ satisfies the 2D Dirichlet problem on a typical polygonal facet Γ

$$\bar{\nabla} \cdot \bar{\nabla}\psi = 0 \text{ on } \Gamma, \quad \psi = f \text{ on } \partial\Gamma \tag{A1}$$

where $\bar{\nabla} = \nabla - \mathbf{nn} \cdot \nabla$ is the surface gradient operator with respect to Γ with $\mathbf{n}$ the (constant) unit normal to this facet; and f is the piecewise-linear function on $\partial\Gamma$ with value unity at one vertex and zero at the others.

Let $\boldsymbol{\xi}$ be the position vector measured from the centroid of Γ, and $\boldsymbol{\lambda}$ the (piecewise constant) unit tangent to $\partial\Gamma$. By Stokes' theorem,

$$\int_{\partial\Gamma} (\psi \mathbf{n} \times \boldsymbol{\xi}) \cdot \boldsymbol{\lambda} \, \mathrm{d}s = \int_{\Gamma} \nabla \times (\psi \mathbf{n} \times \boldsymbol{\xi}) \cdot \mathbf{n} \, \mathrm{d}a \tag{A2}$$

After some manipulation, (A2) reads

$$\int_{\partial\Gamma} (f \mathbf{n} \times \boldsymbol{\xi}) \cdot \boldsymbol{\lambda} \, \mathrm{d}s = \int_{\Gamma} [\nabla\psi \times (\mathbf{n} \times \boldsymbol{\xi}) \cdot \mathbf{n} + 2\psi] \, \mathrm{d}a \tag{A3}$$

where ψ has been replaced by the piecewise-linear boundary function f on the left-hand side. Next, the vector- and scalar-triple-product formulas are used, along with the fact that $\boldsymbol{\xi} \cdot \mathbf{n} = 0$ on Γ to yield, after rearranging,

$$\int_{\Gamma} \psi \, \mathrm{d}a = \frac{1}{2} \int_{\partial\Gamma} f \boldsymbol{\xi} \cdot \boldsymbol{\lambda} \times \mathbf{n} \, \mathrm{d}s - \frac{1}{2} \int_{\Gamma} \nabla\psi \cdot \boldsymbol{\xi} \, \mathrm{d}a \tag{A4}$$

The first term on the right-hand side of (A4) is an integral of a quadratic function on the polygonal boundary, and is trivial to evaluate. If the second term can be estimated, then this equation will provide the facet mean value of ψ that is required for the consistency constraint (9). To this end, it is supposed that Γ contains its centroid. The facet is partitioned into J triangles Δ_j, where J is the number of sides of Γ. The immediate objective is to approximate ψ on Γ for the purposes of computing the right-hand side of (A4). A simple and quite accurate estimate is obtained by taking ψ to vary linearly in each Δ_j, consistent with the piecewise-linear function f on $\partial\Gamma$. Under this specification, the only remaining freedom in ψ is the centroidal value $\psi|_{\boldsymbol{\xi}=\mathbf{0}}$. This value can be estimated by noting that the Dirichlet problem governing ψ is the Euler–Lagrange equation of the functional

$$\int_\Gamma \bar{\nabla}\psi \cdot \bar{\nabla}\psi \, \mathrm{d}a \tag{A5}$$

together with the boundary condition $\psi = f$ on $\partial\Gamma$. Approximating ψ as a piecewise-linear function on Γ as described above, and then minimizing (A5) with respect to $\psi|_{\boldsymbol{\xi}=\mathbf{0}}$ yields, after some calculation,

$$\psi|_{\boldsymbol{\xi}=\mathbf{0}} \approx \bar{\psi} = \sum_j \frac{|\boldsymbol{\xi}^{j+} - \boldsymbol{\xi}^j|^2}{|\boldsymbol{\xi}^j \times \boldsymbol{\xi}^{j+}|} + \frac{\boldsymbol{\xi}^{a+} \cdot (\boldsymbol{\xi}^{a+} - \boldsymbol{\xi}^a)}{|\boldsymbol{\xi}^a \times \boldsymbol{\xi}^{a+}|} + \frac{\boldsymbol{\xi}^{a-} \cdot (\boldsymbol{\xi}^{a-} - \boldsymbol{\xi}^a)}{|\boldsymbol{\xi}^{a-} \times \boldsymbol{\xi}^a|} \tag{A6}$$

where $\boldsymbol{\xi}^j$ is the position, with respect to the facet centroid, of vertex j; and $j-$ and $j+$ superscripts refer to the vertex immediately before and after j when the vertices are taken in anticlockwise sequence. In (A6), the sum on the left-hand side is taken over all vertices, whereas the index a on the right-hand side refers to the vertex at which $f = 1$. The estimate $\bar{\psi}$ given in (A6) is easily calculated from the locations of the vertices.

With the estimate (A6) in hand, the piecewise-linear approximation for ψ on Γ is again invoked to approximately evaluate the second integral on the right-hand side of (A4). Carrying out this calculation and combining with (A4) yields the final result

$$\begin{aligned}\int_\Gamma \psi \, \mathrm{d}a \approx \frac{1}{2}\int_{\partial\Gamma} f\boldsymbol{\xi} \cdot \boldsymbol{\lambda} \times \mathbf{n} \, \mathrm{d}s - \frac{-\bar{\psi}}{6}\sum_j \frac{1}{|\boldsymbol{\xi}^j \times \boldsymbol{\xi}^{j+}|}[\boldsymbol{\xi}^j \cdot \boldsymbol{\xi}^j\boldsymbol{\xi}^{j+} \cdot \boldsymbol{\xi}^{j+} - (\boldsymbol{\xi}^j \cdot \boldsymbol{\xi}^{j+})^2] \\ + \frac{1}{12|\boldsymbol{\xi}^{a-} \times \boldsymbol{\xi}^a|}[\boldsymbol{\xi}^{a-} \cdot \boldsymbol{\xi}^{a-}\boldsymbol{\xi}^a \cdot \boldsymbol{\xi}^a - (\boldsymbol{\xi}^{a-} \cdot \boldsymbol{\xi}^a)^2] \\ + \frac{1}{12|\boldsymbol{\xi}^a \times \boldsymbol{\xi}^{a+}|}[\boldsymbol{\xi}^a \cdot \boldsymbol{\xi}^a\boldsymbol{\xi}^{a+} \cdot \boldsymbol{\xi}^{a+} - (\boldsymbol{\xi}^a \cdot \boldsymbol{\xi}^{a+})^2]\end{aligned} \tag{A7}$$

Here, $\bar{\psi}$ is the estimate of $\psi|_{\boldsymbol{\xi}=\mathbf{0}}$ given in (A6), and again the subscript a refers to the vertex at which $f = 1$. (A7), together with (A6), furnish the required right-hand-side values for the weak-continuity constraint (9). As is evident from (A6), (A7), these values are computed in a straightforward manner from the co-ordinates of the vertices of the facet. In closing, it is mentioned that the approximation (A7) is in fact exact in many cases; e.g. a rectangle.

APPENDIX B

In this appendix, a method is presented for evaluating the monomial integrals required in the VET element formulation. Following the notation of Section 3, n is the number of nodes for the element under consideration, m is the polynomial order of the shape functions (see (4)), M is the resulting number of monomial terms (see (2)), and d is the physical dimensionality (2 or 3).

Integrals of monomials are needed for a number of purposes in the VET element formulation: the weak-continuity constraint (9) and the quadrature-rule-consistency constraint (10) both require monomial integrals through order m on each facet, whereas the compatibility and smoothness optimization (14) requires integrals through order $\max(2m-4, 0)$ on the element interior, on each facet, and, if $d=3$, on the edges of the facets in turn. In what follows, the divergence theorem and Stokes' theorem will be used to reduce the element-interior integrals to facet integrals, and the facet integrals to edge integrals. With each reduction of the dimensionality of the integration domain by one, the polynomial order of the integrand increases by one. Accordingly, the overall requirement is for all monomials through order $\max(m+1, 2m+d-5)$ to be integrated on the line segments that make up the element edges. For a typical VET element, $m=4$ is an appropriate choice; this incurs a requirement for integration of all monomials through order 6 (5 in 2D) on each edge, or 84 terms (21 in 2D). These values are needed only on an element-by-element basis, for the purpose of the initial element formulation. They may be stored in scratch space, and then discarded before moving on to the next element.

Integrals of monomials over the element interior may be converted to boundary integrals through a straightforward application of the divergence theorem, as

$$\int_{\omega} q \, \mathrm{d}v = \frac{1}{k+d} \sum_{\Gamma} \int_{\Gamma} q \mathbf{X} \cdot \mathbf{n} \, \mathrm{d}a \tag{B1}$$

where q is a typical monomial term, $k \geqslant 0$ is the order of q, and the sum is carried out over each facet $\Gamma \subset \partial\omega$.

In 2D, (B1) is the only result required, as the element facets are simply straight line segments in this case. These 1D integrals can be easily and efficiently evaluated using Gaussian quadrature of sufficiently high order, in conjunction with a recursive strategy. If $d=3$, on the other hand, (B1) serves to convert element-interior integrals to integrals over polygonal facets. It then remains to evaluate these integrals, given the values of the monomial integrals on the facet edges. This step is not at all obvious, and is treated in some detail in what follows.

The problem at hand is to integrate q over a polygonal facet Γ, given the values of the monomial integrals on the edges of Γ through order $k+1$. The (constant) normal to Γ will again be given by $\mathbf{n}$. Let $\mathbf{v} = q\boldsymbol{\xi} \times \mathbf{n}$, where $\boldsymbol{\xi} = \mathbf{X} - \mathbf{c}$, $\mathbf{X}$ the position vector relative to an arbitrary origin and $\mathbf{c}$ an arbitrary point in the plane of Γ (e.g. a vertex). Note that $\boldsymbol{\xi}$ lies in the plane of Γ. Applying Stokes' theorem (A2) to $\mathbf{v}$ results in, after a little manipulation,

$$\int_{\Gamma} [(\mathbf{n} \cdot \boldsymbol{\xi})\mathbf{n} \cdot \nabla q - 2q - \boldsymbol{\xi} \cdot \nabla q] \, \mathrm{d}a = \int_{\partial\Gamma} \mathbf{v} \cdot \boldsymbol{\lambda} \, \mathrm{d}s \tag{B2}$$

The first term in the left-hand integrand is zero, because $\mathbf{n} \cdot \boldsymbol{\xi} = 0$. The third term, on the other hand, is $kq - \mathbf{c} \cdot \nabla q$. Making these substitutions and using the definition of $\mathbf{v}$, (B2) becomes

$$\int_\Gamma q \, \mathrm{d}a = \frac{1}{2+k} \int_\Gamma \mathbf{c} \cdot \nabla q \, \mathrm{d}a + \frac{1}{2+k} \int_{\partial\Gamma} \boldsymbol{\lambda} \times \mathbf{n} \cdot \boldsymbol{\xi} \, q \, \mathrm{d}s \tag{B3}$$

We now focus on the first term of the right-hand side. Defining a vector $\mathbf{p} = q\mathbf{c}$, the integrand of this term is $\nabla \cdot \mathbf{p}$, which can be written $\bar{\nabla} \cdot \mathbf{p} + \mathbf{n} \cdot \nabla(\mathbf{p} \cdot \mathbf{n})$. Here, $\bar{\nabla}$ is the in-plane gradient operator, as defined in Appendix A and in Section 3. The 2D divergence theorem in the plane of Γ can be applied to the $\bar{\nabla} \cdot \mathbf{p}$ term, so that

$$\int_\Gamma \mathbf{c} \cdot \nabla q \, \mathrm{d}a = \int_{\partial\Gamma} \mathbf{p} \cdot \boldsymbol{\eta} \, \mathrm{d}s + \int_\Gamma \mathbf{n} \cdot \nabla(\mathbf{p} \cdot \mathbf{n}) \, \mathrm{d}a \tag{B4}$$

where $\boldsymbol{\eta} = \boldsymbol{\lambda} \times \mathbf{n}$ is the in-plane normal to Γ along its boundary. The integrand of the second term on the right-hand side of (B4) can, in turn, be written $h\mathbf{n} \cdot \nabla q$, where $h = \mathbf{c} \cdot \mathbf{n}$ is the perpendicular distance from the origin to the plane containing Γ, and is conveniently computed by taking $\mathbf{c}$ to be the position of a vertex on $\partial\Gamma$. Applying these developments to (B3) yields the expression of interest:

$$\int_\Gamma q \, \mathrm{d}a = \frac{h}{2+k} \int_\Gamma \mathbf{n} \cdot \nabla q \, \mathrm{d}a + \frac{1}{2+k} \int_{\partial\Gamma} \boldsymbol{\eta} \cdot \boldsymbol{\xi} \, q \, \mathrm{d}s \tag{B5}$$

The second term on the right-hand side involves a sum of integrals along straight line segments of monomials of order $k+1$. The first term, while an integral over the facet itself, is easily computed, because its polynomial integrand is of order $k-1$. Therefore, by adopting a recursive strategy whereby all monomials of a given order are evaluated before proceeding to the next-highest order, (B5) yields a complete list of all required integrals with very little computational expense. Note that the first term on the right-hand side is zero for $k = 0$, i.e. for the initial monomial term unity.

In summary, and with reference to step 3 of the algorithm presented in Section 3.5, integrals of all monomials through order $\max(m+1, 2m+d-5)$ are evaluated on all line-segment element edges, using Gaussian quadrature of sufficiently high order to ensure exact values. Here, m is the polynomial order of the shape functions; an appropriate choice is given by (12). Then, if $d = 2$, (B1) is used to generate integrals of all monomials over the element interior through order $\max(2m-4, 0)$. If $d = 3$, (B5) is first used recursively to integrate all monomials through order $\max(m, 2m-3)$ on each facet, and finally (B1) is employed to integrate monomials through order $\max(2m-4, 0)$ on the element interior. These calculations obviously require tables of node–edge and edge–facet association; this information, along with the lists of integrals themselves, may be discarded once the shape-function coefficients have been computed per step (7) in Section 3.5.

ACKNOWLEDGEMENTS

The support of the Idaho National Laboratory under contract 19760, and of Lawrence Livermore National Laboratory under grant ISCR-04-07, with the University of California, Davis is gratefully acknowledged.

REFERENCES

1. Canann SA, Saigal S, Owen SJ (eds). Special edition on unstructured mesh generation. *International Journal for Numerical Methods in Engineering* 2000; **49**(1):1–351.
2. Field DA. The legacy of automatic mesh generation from solid modeling. *Computer Aided Geometric Design* 1995; **12**:651–673.
3. Owen SJ. A survey of unstructured mesh generation technology. *Proceedings of the 7th International Meshing Roundtable*, Sandia National Laboratories, Albuquerque, NM, 1998; 239–267.
4. Shimada K (ed.). The 8th international meshing roundtable special issue: advances in mesh generation. *Computer-Aided Design* 2001; **33**(3):197–277.
5. Viceconti M, Taddei F. Automatic generation of finite element meshes from computed tomography data. *Critical Reviews in Biomedical Engineering* 2003; **31**:27–72.
6. Chen JS, Liu WK. Meshfree methods: recent advances and new applications. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**(12–14):933–1321.
7. Babuska I, Banerjee U, Osborn JE. Survey of meshless and generalized finite element methods: a unified approach. *Acta Numerica* 2003; **12**:1–125.
8. Dolbow J, Belytschko T. Volumetric locking in the element free Galerkin method. *International Journal for Numerical Methods in Engineering* 1999; **46**:925–942.
9. Krysl P, Belytschko T. The element free Galerkin method for dynamic propagation of arbitrary 3-D cracks. *International Journal for Numerical Methods in Engineering* 1999; **44**:767–800.
10. Sukumar N, Moran B, Belytschko T. The natural element method in solid mechanics. *International Journal for Numerical Methods in Engineering* 1998; **43**:839–887.
11. Sukumar N, Moran B, Semenov AY, Belikov VV. Natural neighbour Galerkin methods. *International Journal for Numerical Methods in Engineering* 2001; **50**:1–27.
12. Gonzalez D, Cueto E, Martinez MA, Doblare M. Numerical integration in natural neighbour Galerkin methods. *International Journal for Numerical Methods in Engineering* 2004; **60**:2077–2104.
13. Cueto E, Sukumar N, Calvo B, Martinez MA, Cegonino J, Doblare M. Overview of recent advances in natural neighbour Galerkin methods. *Archives of Computational Methods in Engineering* 2003; **10**:307–384.
14. Cueto E, Doblare M, Gracia L. Imposing essential boundary conditions in the natural element method by means of density-scaled α-shapes. *International Journal for Numerical Methods in Engineering* 2000; **49**: 519–546.
15. Cueto E, Calvo B, Doblare M. Modelling three-dimensional piece-wise homogeneous domains using the α-shape-based natural element method. *International Journal for Numerical Methods in Engineering* 2002; **54**:871–897.
16. Calvo B, Martinez MA, Doblare M. On solving large strain hyperelastic problems with the natural element method. *International Journal for Numerical Methods in Engineering* 2005; **62**:159–185.
17. Yoo JW, Moran B, Chen JS. Stabilized conforming nodal integration in the natural-element method. *International Journal for Numerical Methods in Engineering* 2004; **60**:861–890.
18. Chen JS, Wu CT, Yoon S, You Y. A stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering* 2001; **50**:435–466.
19. Rashid MM, Gullett PM. On a finite element method with variable element topology. *Computer Methods in Applied Mechanics and Engineering* 2000; **190**:1509–1527.
20. Rashid MM, Gullett PM. Formulation of a finite element method with arbitrary element geometry. In *Modeling and Simulation-Based Life Cycle Engineering*, Chong KP, Saigal S, Thynell S, Morgan HS (eds). Spon: London, 2002; 76–90.
21. Stummel F. The generalized patch test. *SIAM Journal on Numerical Analysis* 1979; **16**:449–471.
22. Taylor RL, Simo JC, Zienkiewicz OC, Chan AC. The patch test: a condition for assessing finite element convergence. *International Journal for Numerical Methods in Engineering* 1986; **22**:39–62.
23. Zienkiewicz OC, Taylor RL. The finite element patch test revisited: a computer test for convergence, validation, and error estimates. *Computer Methods in Applied Mechanics and Engineering* 1997; **149**: 223–254.
24. Shi ZC. The F-E-M test for convergence of nonconforming finite element. *Mathematics of Computation* 1987; **49**:391–405.
25. Atkinson KE, Han W. *Theoretical Numerical Analysis*: *A Functional Analysis Framework*. Springer: New York, 2001.

26. Idelsohn SR, Onate E, Calvo N, Del Pin F. The meshless finite element method. *International Journal for Numerical Methods in Engineering* 2003; **58**:893–912.
27. Sukumar N, Tabarraei A. Conforming polygonal finite elements. *International Journal for Numerical Methods in Engineering* 2004; **61**:2045–2066.
28. Simo JC, Rifai MS. A class of mixed assumed strain methods and the method of incompatible modes. *International Journal for Numerical Methods in Engineering* 1990; **29**:1595–1638.
29. Simo JC, Armero F, Taylor RL. Improved versions of assumed enhanced strain tri-linear elements for 3D finite deformation problems. *Computer Methods in Applied Mechanics and Engineering* 1993; **110**:359–386.
30. Hughes TJR. Generalization of selective integration procedures to anisotropic and nonlinear media. *International Journal for Numerical Methods in Engineering* 1980; **15**:1413–1418.
31. de Souza Neto EA, Peric D, Dutko M, Owen DRJ. Design of simple low order finite elements for large strain analysis of nearly incompressible solids. *International Journal of Solids and Structures* 1996; **33**:3277–3296.
32. Rashid MM. Incremental kinematics for finite element applications. *International Journal for Numerical Methods in Engineering* 1993; **36**:3937–3956.
33. Peters JF, Heymsfield E. Application of the 2-D constant strain assumption to FEM elements consisting of an arbitrary number of nodes. *International Journal of Solids and Structures* 2003; **40**:143–159.
34. Dohrmann CR, Rashid MM. Polynomial approximation of shape function gradients from element geometries. *International Journal for Numerical Methods in Engineering* 2002; **53**:945–958.
35. Strouboulis T, Copps K, Babuska I. The generalized finite element method. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:4081–4193.
36. Babuska I, Strouboulis T, Copps K. The design and analysis of the generalized finite element method. *Computer Methods in Applied Mechanics and Engineering* 2000; **181**:43–69.

