PERGAMON

Computers and Structures 77 (2000) 215–232

Computers & Structures

www.elsevier.com/locate/compstruc

Generalized finite element methods for three-dimensional structural mechanics problems

C.A. Duarte[a,*], I. Babuška[b], J.T. Oden

[a]COMCO, Inc., 7800 Shoal Creek Blvd. Suite 290E, Austin, TX 78757, USA
[b]TICAM, The University of Texas at Austin, SHC 310, Austin, TX 78712, USA

Received 26 February 1999; accepted 23 August 1999

Abstract

The present paper summarizes the generalized finite element method formulation and demonstrates some of its advantages over traditional finite element methods to solve complex, three-dimensional (3D) structural mechanics problems. The structure of the stiffness matrix in the GFEM is compared to the corresponding FEM matrix. The performance of the GFEM and FEM in the solution of a 3D elasticity problem is also compared. The construction of *p*-orthotropic approximations on tetrahedral meshes and the use of a-priori knowledge about the solution of elasticity equations in three-dimensions are also presented.

1. Introduction

The analysis of complex three-dimensional (3D) structural components has become a common task in recent years in several manufacturing industries. However, the analysis of this class of problems using traditional finite element methods still poses several difficulties. It is a common practice to use automatic tetrahedral mesh generators to discretize complex 3D structural components. This type of mesh generators can handle very complex geometries with a minimum of human intervention (as compared to, e.g., the manual generation of a mesh of hexahedral elements). The main drawbacks of any automatic mesh generators are:

1. The need of an excessive number of elements in order to keep the aspect ratios of the finite elements within reasonable bounds. This is specially true when the component has transition zones from bulky to slender parts.
2. Polynomial approximations, as used in traditional finite element methods, require the use of a large number of elements in order to capture stress concentrations and singularities at corners and edges of the domain. Meshes that are insufficiently refined at these regions are often used in order to keep the number of degrees of freedom (dof) below a reasonable level or simply because the available mesh generators can not create an appropriate mesh where needed without creating an excessive number of elements everywhere. This practice may lead to results of poor quality even at points in the domain distant from the singularities, due to numerical pollution.
3. Another drawback of automatic mesh generators, especially when tetrahedral elements are used, is that they preclude the use of *p* anisotropic approximations, that is, approximations that have different polynomial orders associated with each direction. Problems where boundary layers occur, such as in

* Corresponding author. Tel.: +1-512-467-0618; Fax: +1-512-467-1382.
E-mail address: armando@comco.com (C.A. Duarte).

0045-7949/00/$ - see front matter
PII: S0045-7949(99)00211-4

the analysis of orthotropic materials or when one of the dimensions of the structural part is much smaller than the others, are examples in which *p*-orthotropic approximations may lead to considerable savings in the number of dof needed to achieve acceptable accuracy.

The generalized finite element method (GFEM) was proposed independently by Babuška and colleagues [2,3,21] (under the names 'special finite element methods', 'generalized finite element method' and 'finite element partition of unity method') and by Duarte and Oden [9–12,25] (under the names '*hp* clouds' and 'cloud-based *hp* finite element method'). Several of the so-called meshless methods proposed in recent years can also be viewed as special cases of the generalized finite element method. Recent surveys on meshless methods can be found in [6,8]. The key feature of these methods is the use of a partition of unity (PU), which is a set of functions whose values sum to the unity at each point $\mathbf{x}$ in a domain Ω. The analysis of the performance and computational cost of *hp* clouds, element free Galerkin [7], diffuse element [22] and reproducing kernel particle [18] methods can be found in [12]. It was found in that study that the integration of the stiffness matrix in these methods can be considerably more expensive than in traditional *hp* finite element methods, depending on the choice of the partition of unity, the Moving Least Squares PU [17], being among the most expensive.

The present paper summarizes the ideas behind the GFEM formulation and demonstrates through numerical examples some of the advantages of GFEM over traditional FEM for solving complex, 3D structural mechanics problems. In Section 5, the structure of the stiffness matrix in the GFEM is compared to the corresponding FEM matrix. In Section 6, we compare the performance of the GFEM and FEM in the solution of a representative 3D elasticity problem. The use of *p*-orthotropic approximations on tetrahedral meshes and the use of a-priori knowledge about the solution of the elasticity equations in three-dimensions is also demonstrated in that section. Finally, in Section 7, major conclusions of the study are given.

2. Formulation of generalized finite element methods

In this section, we review the basic ideas behind the construction of generalized finite element approximations in a one-dimensional (1D) setting using a 1D linear finite element partition of unity. In Section 2.1, the GFE formulation in an *n*-dimensional setting is presented. In Sections 5 and 6, we use generalized tetrahedral finite elements to discretize 3D problems. For a more detailed discussion on the theoretical aspects of GFE approximations, see Refs. [3,10,21,25] and the references therein.

Let u be a function defined on a domain $\Omega \subset \mathbb{R}$. Suppose that we build an open covering

$$\mathscr{T}_N = \{\omega_\alpha\}_{\alpha=1}^N \qquad \bar{\Omega} \subset \bigcup_{\alpha=1}^N \omega_\alpha$$

of Ω consisting of N supports ω_α (often called *clouds*) with centers at x_α, $\alpha = 1,\ldots,N$.

Let u_α be a local approximation of u that belongs to a local space $\chi_\alpha(\omega_\alpha)$ defined on the support ω_α. It is presumed that each space $\chi_\alpha(\omega_\alpha)$, $\alpha = 1,\ldots,N$, can be chosen such that there exists a $u_\alpha \in \chi_\alpha(\omega_\alpha)$ that can approximate well $u|_{\omega_\alpha}$ in some sense. Fig. 1 illustrates the definitions given above. In this case, the supports ω_α are open intervals with centers x_α.

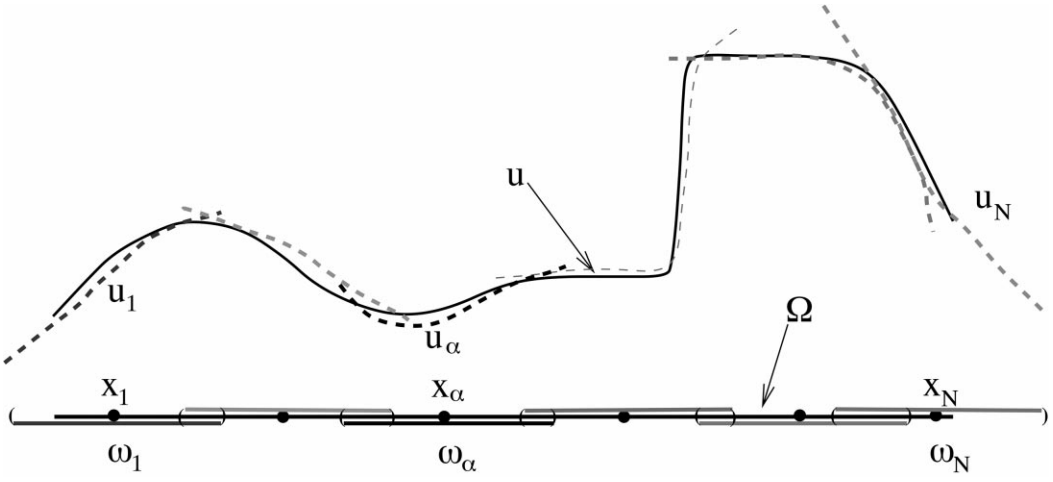

Fig. 1. Local approximations defined on the supports ω_α.

The local approximations u_α, $\alpha = 1,\ldots,N$, have to be somehow combined together to give a global approximation u_{hp} of u. This global approximation has to be built such that the difference between u_{hp} and u, in a given norm, be bounded by the local errors $u - u_\alpha$. In partition of unity methods, this is accomplished using functions defined on the supports ω_α, $\alpha = 1,\ldots,N$, and having the following property

$$\varphi_\alpha \in C^s_0(\omega_\alpha), \quad s \geq 0, \quad 1 \leq \alpha \leq N$$

$$\sum_\alpha \varphi_\alpha(x) = 1 \quad \forall x \in \Omega$$

The functions φ_α are called *a partition of unity subordinate to the open covering* $\mathscr{T}_N$. Examples of partitions of unity are Lagrangian finite elements, the various 'reproducing' functions generated by moving least squares methods and Shepard functions [11,17].

In the case of finite element partitions of unity, the supports (clouds) ω_α are simply the union of the finite elements sharing a vertex node x_α (see, for example Refs. [21,25]). In this case, the implementation of the method is essentially the same as in standard finite element codes, the main difference being the definition of the shape functions as explained below. This choice of partition of unity avoids the problem of integration associated with the use of moving least squares methods or Shepard partitions of unity used in several meshless methods. Here, the integrations are performed with the aid of the so called master elements, as in classical finite elements. Therefore, the GFEM can use existing infrastructure and algorithms developed for the classical finite element method.

Fig. 2 shows a one-dimensional finite element discretization. The partition of unity functions φ_α are the usual global finite element shape functions, the classical 'hat-functions', associated with node x_α. The support ω_α is thus the union of the elements $\tau_{\alpha-1}$ and τ_α.

Consider now the element τ_α with nodes x_α and $x_{\alpha+1}$ as depicted in Fig. 2. Suppose that the following shape functions are used on this element

$$S_\alpha = \{\varphi_\alpha, \varphi_{\alpha-1}\} \times \{1, u_\alpha, u_{\alpha+1}\}$$
$$= \{\varphi_\alpha, \varphi_{\alpha+1}, \varphi_\alpha u_\alpha, \varphi_\alpha u_{\alpha+1}, \varphi_{\alpha+1} u_\alpha, \varphi_{\alpha+1} u_{\alpha+1}\}$$

That is, the element τ_α has a total of six shape functions (three at each node) built from the product of the standard Lagrangian finite element shape functions (a partition of unity), and the local approximations $u_\alpha, u_{\alpha+1}$ that, by assumption, can approximate well the function u over the finite element τ_α. Of course, we can further generalize this idea by increasing the number of functions u_α and $u_{\alpha+1}$, resulting in a space S_α of still larger dimension. This approach is discussed in the next section.

Thanks to the partition of unity property of the finite element shape functions, we can easily show that linear combinations of the shape functions defined above can reproduce the local approximations u_α, $u_{\alpha+1}$, that is,

$$\varphi_\alpha u_\alpha + \varphi_{\alpha+1} u_\alpha = u_\alpha(\varphi_\alpha + \varphi_{\alpha+1}) = u_\alpha \quad \text{(no sum on } \alpha)$$

$$\varphi_\alpha u_{\alpha+1} + \varphi_{\alpha+1} u_{\alpha+1} = u_{\alpha+1}(\varphi_\alpha + \varphi_{\alpha+1}) = u_{\alpha+1}$$

In other words,

$$u_\alpha, u_{\alpha+1} \in \text{span}\{S_\alpha\}.$$

The basic idea in partition of unity methods, and in particular, in the GFE method, is to use a partition of unity to paste together local approximation spaces. The shape functions are built such that they can reproduce, through linear combinations, the local approximations defined on each cloud ω_α. The approximation properties of such functions are discussed in the next section.

2.1. Generalized finite element shape functions: the family of functions $\mathscr{F}^p_N$

In this section, we define generalized finite element shape functions in an n-dimensional setting using the same ideas outlined in the previous section.

Let the functions φ_α, $\alpha = 1,\ldots,N$, denote a finite element partition of unity subordinate to the open cov-

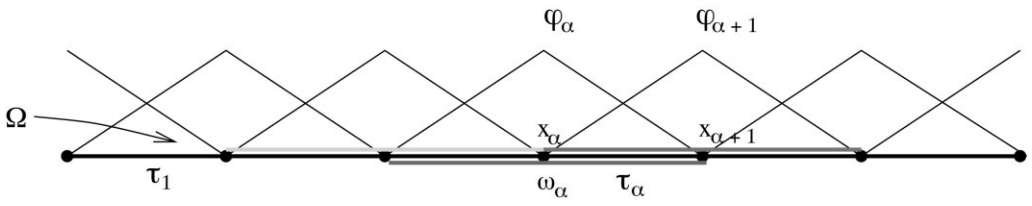

Fig. 2. One-dimensional finite element partition of unity.

ering $\mathscr{T}_N = \{\omega_\alpha\}_{\alpha=1}^N$ of a domain $\Omega \subset \mathbb{R}^n$, $n = 1, 2, 3$. Here, N is the number of vertex nodes in the finite element mesh. The cloud ω_α is the union of the finite elements sharing the vertex node $\mathbf{x}_\alpha$.

Let $\chi_\alpha(\omega_\alpha) = \text{span}\{L_{i\alpha}\}_{i\in\mathscr{I}(\alpha)}$ denote local spaces defined on ω_α, $\alpha = 1,\ldots,N$, where $\mathscr{I}(\alpha)$, $\alpha = 1,\ldots,N$, are index sets and $L_{i\alpha}$ denotes local approximation functions analogous with the functions u_α mentioned in the previous section. Possible choices for these functions are discussed below.

Suppose that the finite element shape functions, φ_α, are linear functions and that

$$\mathscr{P}_{p-1}(\omega_\alpha) \subset \chi_\alpha(\omega_\alpha) \quad \alpha = 1,\ldots,N,$$

where $\mathscr{P}_{p-1}$ denotes the space of polynomials of degree less or equal to $p-1$. The generalized finite element shape functions of degree p are defined by [21,25]

$$\mathscr{F}_N^p = \left\{\phi_i^\alpha = \varphi_\alpha L_{i\alpha}, \alpha = 1,\ldots,N, i \in \mathscr{I}(\alpha)\right\} \tag{1}$$

Note that there is considerable freedom in the choice of the local spaces χ_α. The most obvious choice for a basis of χ_α is polynomial functions which can approximate well smooth functions. In this case, the GFEM is essentially identical to the classical FEM. The implementation of *hp* adaptivity is, however, greatly simplified by the PU framework. Since each basis functions, $\{L_{i\alpha}\}_{i\in\mathscr{I}(\alpha)}$, $\alpha = 1,\ldots,N$, can have a different polynomial order for each α, we can have different polynomial orders associated with each vertex node of the finite element mesh [25]. The approximations can also be non-isotropic (i.e., different polynomial orders in different directions), regardless of the choice of finite element partition of unity (hexahedral, tetrahedral, etc.). Examples of *p*-orthotropic approximations built on a tetrahedral mesh are given in Section 6. The concept of edge and middle nodes, which is used in conventional *p* FEMs, is not needed in the framework of GFEM. The implementation of *h* adaptivity is also simplified in PU methods since it needs to be done only on the partition of unity (linear finite elements in this case). Therefore, the implementation of *h* adaptivity for high-order approximations is the same as for linear approximations (there is no need, for example, of using high-order constraints as done in traditional *hp* finite element methods).

There are many situations in which the solution of a boundary value problem is not a smooth function. In these situations, the use of polynomials to build the approximation space, as in the FEM, may be far from optimal and may lead to poor approximations of the solution u unless carefully designed meshes are used. In the GFEM, we can use any a-priori knowledge about the solution to make better choices for the local spaces χ_α. For example, in Section 6, we solve a boundary value problem in which the solution possesses point or lines of singularities at some parts of the domain. Then we can use the local spaces χ_α to build generalized finite element shape functions that represent these singularities much more effectively than polynomial functions. The construction of these customized shape functions is discussed in details in Section 4.

Detailed convergence analysis of the generalized finite element method can be found in [11,12,20,21]. Several a-priori error estimates along with numerical experiments demonstrating their accuracy can also be found in those works. Below, we summarize the main a-priori error estimate for the generalized finite element method.

Suppose that the partitions of Ω into finite elements which form the partition of unity satisfy the usual regularity assumptions in two and three dimensions, and, in one dimension, the ratio between the lengths of neighboring elements is bounded. We denote

$$h_\alpha = \text{diam}(\omega_\alpha)$$

$$h = \max_{\alpha=1,\ldots,N} h_\alpha$$

and

$$\mathbf{X}^{hp} = \text{span}\left\{\phi_i^\alpha\right\}, \quad \alpha = 1,\ldots,N, \quad i \in \mathscr{I}(\alpha)$$

where the GFE shape functions ϕ_i^α are defined in Eq. (1). In addition, suppose the unity function, 1, belongs to the set of local approximation functions, i.e.,

$$1 \in \chi_\alpha(\omega_\alpha) \quad \alpha = 1,\ldots,N,$$

and that there exists a quantity ϵ depending of α, h, p, and u, such that

$$\|u - u_\alpha\|_{E(\Omega\cap\omega_\alpha)} \leq \epsilon(\alpha, h, p, u) \quad \alpha = 1,\ldots,N$$

Then, it can be proved that [11,21] $\exists\, u_{hp} \in \mathbf{X}^{hp}$ such that

$$\|u - u_{hp}\|_{E(\Omega)} \leq C\left(\sum_{\alpha=1}^{N(h)} \epsilon^2(\alpha, h, p, u)\right)^{1/2}$$

where the constant C is independent of u, h, p.

3. Quadratic GFE shape functions for tetrahedral elements

In this section, quadratic GFE shape functions for tetrahedral elements are defined and analyzed. The issue of linear dependence of GFE shape function and

how to solve the resulting positive semi-definite system of equations are also discussed.

Quadratic GFE shape functions for tetrahedral elements are built from the product of trilinear tetrahedral shape functions and linear monomials as follows

$$\varphi_\alpha \times \left\{1, \frac{x-x_\alpha}{h_\alpha}, \frac{y-y_\alpha}{h_\alpha}, \frac{z-z_\alpha}{h_\alpha}\right\} \quad \alpha = 1,\ldots,N \tag{2}$$

where φ_α, $\alpha = 1,\ldots,N$, are standard trilinear Lagrangian tetrahedral shape functions (e.g., Refs. [4,5,32]), $\mathbf{x}_\alpha = (x_\alpha, y_\alpha, z_\alpha)$ are the coordinates of the node α, h_α is the diameter of the largest finite element sharing the node α and N is the number of nodes in the mesh. This transformation is used to minimize round-off errors. For details see, for example, Refs. [10,11].

In the notation of Section 2.1 we have

$$\chi_\alpha = \operatorname{span}\{L_{i\alpha}\} = \operatorname{span}\left\{1, \frac{x-x_\alpha}{h_\alpha}, \frac{y-y_\alpha}{h_\alpha}, \frac{z-z_\alpha}{h_\alpha}\right\}$$

$$\alpha = 1,\ldots,N$$

Consider now the case of a tetrahedral element τ with, for example, nodes $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, $\mathbf{x}_4$. Then the GFE quadratic shape functions for this element are given by

$$S_\tau = \varphi_\alpha \times \left\{1, \frac{x-x_\alpha}{h_\alpha}, \frac{y-y_\alpha}{h_\alpha}, \frac{z-z_\alpha}{h_\alpha}\right\} \quad \alpha = 1, 2, 3, 4$$

Each quadratic tetrahedral element has therefore sixteen shape functions (instead of ten as in classical FEMs). It should be noted that all shape functions are associated with the vertices of the element as in a linear element and there is no need for the concept of an edge, face or interior node, as in classical high order tri-dimensional finite elements. Also, the support (cloud) of the higher order shape functions (i.e., the domain on which the function is non-zero) is identical to the support of the trilinear shape functions $\{\varphi_1, \varphi_2, \varphi_3, \varphi_4\}$. This fact, as we demonstrate in Section 5, has important implications on the structure of the stiffness matrix.

Generalized tetrahedral shape functions that can reproduce quadratic polynomials can also be built as follows. Suppose that the following basis is used for the local spaces χ_α, $\alpha = 1, 2, 3, 4$

$$\chi_\alpha = \operatorname{span}\{1, \xi^2, \xi\eta, \xi\zeta, \eta^2, \eta\zeta, \zeta^2\}, \quad \alpha = 1, 2, 3, 4$$

where $\xi = (x-x_\alpha)/h_\alpha$, $\eta = (y-y_\alpha)/h_\alpha$, $\zeta = (z-z_\alpha)/h_\alpha$. Then the shape functions for an element τ with nodes $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, $\mathbf{x}_4$, are given by

$$\tilde{S}_\tau = \varphi_\alpha \times \{1, \xi^2, \xi\eta, \xi\zeta, \eta^2, \eta\zeta, \zeta^2\}, \quad \alpha = 1, 2, 3, 4$$

Note that we excluded from the basis of χ_α the elements ξ, η, ζ since they can be reproduced by the partition of unity functions φ_α, $\alpha = 1, 2, 3, 4$. However, as we demonstrate below, this is not sufficient to avoid linear dependencies.

Theorem 1. *Let S_τ and $\tilde{S}_\tau$ be as defined above. Then*

(i)

$$\operatorname{span}\{S_\tau\} = \operatorname{span}\{1, x, y, z, x^2, xy, xz, y^2, yz, z^2\} = \mathscr{P}_2$$

(ii)

$$\operatorname{span}\{\tilde{S}_\tau\} = \operatorname{span}\{1, x, y, z, x^2, xy, xz, y^2, yz,$$
$$z^2, x^3, x^2y, x^2z, y^3, y^2x, y^2z, z^3, z^2x, z^2y, xyz\} = \mathscr{P}_3$$

Proof. In order to simplify the notation, we consider here the case in which the basis functions, $L_{i\alpha}$, of χ_α are given by $\{1, x, y, z\}$. Therefore, in this case,

$$S_\tau = \varphi_\alpha \times \{1, x, y, z\} \quad \alpha = 1, 2, 3, 4$$

Since φ_α, $\alpha = 1, 2, 3, 4$, are trilinear tetrahedral shape functions, there exist constants a_α^x, a_α^y, a_α^z, $\alpha = 1, 2, 3, 4$, such that $\forall \mathbf{x} \in \tau$,

$$\sum_{\alpha=1}^{4} \varphi_\alpha(\mathbf{x}) = 1 \tag{3}$$

$$\sum_{\alpha=1}^{4} a_\alpha^x \varphi_\alpha(\mathbf{x}) = x \tag{4}$$

$$\sum_{\alpha=1}^{4} a_\alpha^y \varphi_\alpha(\mathbf{x}) = y \tag{5}$$

$$\sum_{\alpha=1}^{4} a_\alpha^z \varphi_\alpha(\mathbf{x}) = z \tag{6}$$

Therefore

$$\{1, x, y, z\} \subset \operatorname{span}\{S_\tau\}$$

From the equations above we have

$$\sum_{\alpha=1}^{4} a_\alpha^x (\varphi_\alpha x) = x \sum_{\alpha=1}^{4} a_\alpha^x \varphi_\alpha = x^2$$

$$\sum_{\alpha=1}^{4} a_\alpha^y (\varphi_\alpha x) = x \sum_{\alpha=1}^{4} a_\alpha^y \varphi_\alpha = xy \tag{7}$$

and similarly for the terms xz, y^2, yz, z^2. Therefore

$$\mathscr{P}_2 \subset \text{span}\{S_\tau\} \tag{8}$$

Let us now show that S_τ is a linear dependent set and that the rank deficiency of S_τ is equal to 6. Using the partition of unity property (3) of the functions φ_α

$$\sum_{\alpha=1}^{4}(\varphi_\alpha x) = x\sum_{\alpha=1}^{4}\varphi_\alpha = x$$

Using Eq. (4) and the above

$$\sum_{\alpha=1}^{4}(\varphi_\alpha x) - \sum_{\alpha=1}^{4}a_\alpha^x\varphi_\alpha = 0$$

and similarly for the terms y and z. Therefore, the dimension of a basis of span$\{S_\tau\}$ is less or equal to 13. But

$$\sum_{\alpha=1}^{4}a_\alpha^x(\varphi_\alpha y) = y\sum_{\alpha=1}^{4}a_\alpha^x\varphi_\alpha = yx$$

Using Eq. (7) and the above

$$\sum_{\alpha=1}^{4}a_\alpha^y(\varphi_\alpha x) - \sum_{\alpha=1}^{4}a_\alpha^x(\varphi_\alpha y) = 0$$

and similarly for the terms zx and zy. Therefore, the dimension of a basis of span$\{S_\tau\}$ is less or equal to 10. But from Eq. (8) the dimension of this set is greater or equal to 10. This proves (i). The proof of (ii) follows the same steps.

The linear dependencies that appear above are a consequence of the fact that both, the partition of unity and the basis of the local spaces χ_α, are polynomial functions. Babuška and colleagues [15,29] have proposed two approaches to solve the system of equations

$$\tilde{\mathbf{K}}\tilde{\mathbf{u}} = \tilde{\mathbf{f}} \tag{9}$$

where $\tilde{\mathbf{K}}$ is a positive semi-definite stiffness matrix built from generalized finite element shape functions. The first approach consists simply of using a direct solver for symmetric indefinite systems like Refs. [13,28]. Similar implementations are presently used in several commercial FEM solvers. The second approach consists of the following iterative algorithm [15,19,29]:

Let

$$\mathbf{K} = \mathbf{T}\tilde{\mathbf{K}}\mathbf{T}$$

$$\mathbf{u} = \mathbf{T}^{-1}\tilde{\mathbf{u}}$$

$$\mathbf{f} = \mathbf{T}\tilde{\mathbf{f}}$$

where

$$T_{i,j} = \frac{\delta_{i,j}}{\sqrt{\tilde{K}_{i,j}}}$$

Then

$$\mathbf{K}\mathbf{u} = \mathbf{f} \tag{10}$$

The above transformation leads to a stiffness matrix whose diagonal entries are equal to 1. This scaled matrix is then perturbed as follows

$$\mathbf{K}_\epsilon = \mathbf{K} + \epsilon\mathbf{I}, \quad \epsilon > 0, \quad I_{i,j} = \delta_{i,j}$$

The matrix $\mathbf{K}_\epsilon$ is positive definite and hence non-singular. The solution $\mathbf{u}$ to the system (10) is then computed using the following sequence:

$$\mathbf{u}_0 = \mathbf{K}_\epsilon^{-1}\mathbf{f}$$

$$\mathbf{r}_0 = \mathbf{f} - \mathbf{K}\mathbf{u}_0$$

Let $\mathbf{e}_0 = \mathbf{u} - \mathbf{u}_0$ then

$$\mathbf{K}_\epsilon\mathbf{e}_0 \simeq \mathbf{K}\mathbf{e}_0 = \mathbf{K}\mathbf{u} - \mathbf{K}\mathbf{u}_0 = \mathbf{r}_0$$

therefore

$$\mathbf{e}_0 = \mathbf{K}_\epsilon^{-1}\mathbf{r}_0$$

Compute

$$\mathbf{r}_i = \mathbf{e}_0 - \sum_{j=0}^{i-1}\mathbf{K}\mathbf{e}_j$$

$$\mathbf{e}_i = \mathbf{K}_\epsilon^{-1}\mathbf{r}_i$$

$$\mathbf{u}_i = \mathbf{u}_0 + \sum_{j=0}^{i-1}\mathbf{e}_j$$

for $i \geq 1$ until

$$\left|\frac{\mathbf{e}_i\mathbf{K}\mathbf{e}_i}{\mathbf{u}_i\mathbf{K}\mathbf{u}_i}\right|$$

is sufficiently small. Numerical experiments performed by Strouboulis et al. [15,29] show that in practice, with $\epsilon = 10^{-10}$, a single iteration is sufficient when using double precision accuracy. Each iteration involves a matrix–vector multiplication, a forward- and back-substitutions and lower order operations. Therefore, the

computational cost of each iteration is negligible when compared to the factorization of the stiffness matrix. The solution to the original system (9) is then given by

$$\tilde{\mathbf{u}} = \mathbf{T}\mathbf{u}$$

4. Customized shape functions for an edge in 3D

There are many classes of problems for which the structure of the underlying partial differential equation can be exploited. Oden and Duarte [24,26] have demonstrated how knowledge of the solution of the elasticity equations near a corner in 2D space can be used in a partition of unity method to efficiently model the singularities that occur in this class of problems. This type of singularity is resolved very poorly by polynomial functions such as are used in traditional finite element methods, unless a very refined mesh is used. In this section, the formulation proposed by Oden and Duarte [24,26] is extended to the case of edges in 3D problems. Numerical examples are presented in Section 6.

Consider an straight edge in 3D space as depicted in Fig. 3. In the figure, $2\pi - \alpha$ is the opening angle. Associated with the edge, there is a Cartesian local coordinate system (ξ, η, ζ) and a cylindrical coordinate system (r, θ, ζ') with origins at (O_x, O_y, O_z).

The displacement field $\mathbf{u}(r, \theta, \zeta')$ in the neighborhood of the edge (for points far from its vertices) can be written as [30,31]

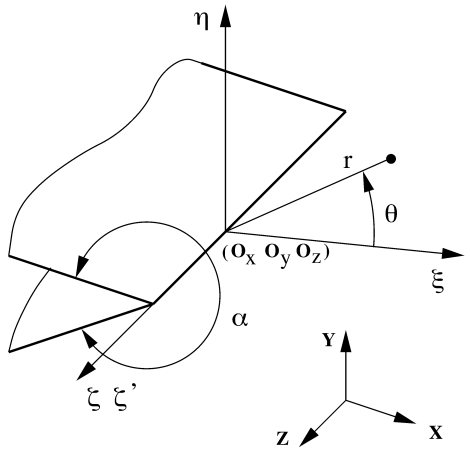

Fig. 3. Coordinate systems associated with an edge in 3D space.

$$\begin{aligned}\mathbf{u}(r,\theta,\zeta') &= \begin{Bmatrix} u_\xi(r,\theta) \\ u_\eta(r,\theta) \\ u_\zeta(r,\theta) \end{Bmatrix} \\ &= \sum_{j=1}^{\infty}\left[A_j^{(1)} \begin{Bmatrix} u_{\xi j}^{(1)}(r,\theta) \\ u_{\eta j}^{(1)}(r,\theta) \\ 0 \end{Bmatrix} + A_j^{(2)} \begin{Bmatrix} u_{\xi j}^{(2)}(r,\theta) \\ u_{\eta j}^{(2)}(r,\theta) \\ 0 \end{Bmatrix} + A_j^{(3)} \begin{Bmatrix} 0 \\ 0 \\ u_{\zeta j}^{(3)}(r,\theta) \end{Bmatrix}\right] \end{aligned} \tag{11}$$

where (r, θ, ζ') are the cilyndrical coordinates relatives to the system shown in Fig. 3, $u_\xi(r, \theta)$, $u_\eta(r, \theta)$ and $u_\zeta(r, \theta)$ are Cartesian components of $\mathbf{u}$ in the ξ-, η- and ζ-directions, respectively.

Assuming that the boundary is traction-free and neglecting body forces, the functions $u_{\xi j}^{(1)}$, $u_{\eta j}^{(1)}$, $u_{\xi j}^{(2)}$, $u_{\eta j}^{(2)}$ are given by [30,31]

$$u_{\xi j}^{(1)}(r,\theta) = \frac{r^{\lambda_j^{(1)}}}{2G}\left\{\left[\kappa - Q_j^{(1)}\left(\lambda_j^{(1)}+1\right)\right]\cos\lambda_j^{(1)}\theta - \lambda_j^{(1)}\cos\left(\lambda_j^{(1)}-2\right)\theta\right\}$$

$$u_{\xi j}^{(2)}(r,\theta) = \frac{r^{\lambda_j^{(2)}}}{2G}\left\{\left[\kappa - Q_j^{(2)}\left(\lambda_j^{(2)}+1\right)\right]\sin\lambda_j^{(2)}\theta - \lambda_j^{(2)}\sin\left(\lambda_j^{(2)}-2\right)\theta\right\}$$

$$u_{\eta j}^{(1)}(r,\theta) = \frac{r^{\lambda_j^{(1)}}}{2G}\left\{\left[\kappa + Q_j^{(1)}\left(\lambda_j^{(1)}+1\right)\right]\sin\lambda_j^{(1)}\theta + \lambda_j^{(1)}\sin\left(\lambda_j^{(1)}-2\right)\theta\right\}$$

$$u_{\eta j}^{(2)}(r,\theta) = -\frac{r^{\lambda_j^{(2)}}}{2G}\left\{\left[\kappa + Q_j^{(2)}\left(\lambda_j^{(2)}+1\right)\right]\cos\lambda_j^{(2)}\theta + \lambda_j^{(2)}\cos\left(\lambda_j^{(2)}-2\right)\theta\right\}$$

where the eigenvalues $\lambda_j^{(1)}$, $\lambda_j^{(2)}$ are found by solving

$$\sin\lambda_j^{(1)}\alpha + \lambda_j^{(1)}\sin\alpha = 0$$

$$\sin\lambda_j^{(2)}\alpha - \lambda_j^{(2)}\sin\alpha = 0$$

In the case of a crack, where $\alpha = 2\pi$, the eigenvalues are $\lambda_j^{(1)} = \lambda_j^{(2)} = \lambda_j$

$$\lambda_1 = \frac{1}{2}, \qquad \lambda_j = \frac{j+1}{2} \quad j \geq 2$$

For the edges in the mechanical part analyzed Section 6, where $\alpha = 4.525\ 264\ 2$,

$$\lambda_1^{(1)} = 0.564\ 349\ 993\ 512\ 569$$

$$\lambda_1^{(2)} = 0.985\ 954\ 537\ 708\ 805$$

The material constant κ and G are

$$\kappa = 3 - 4\nu \qquad G = \frac{E}{2(1+\nu)}$$

where E is the Young's modulus and ν is the Poisson's ratio. This assumes a state of plane strain which is a good approximation for the stress state in the neighborhood of a straight edge in three-dimensions. For points close to the vertex of the edge the stress state is more complex (see for example Refs. [14,23]).

The parameters $Q_j^{(1)}$ and $Q_j^{(2)}$ are given by

$$Q_j^{(1)} = -\frac{\cos\left(\lambda_j^{(1)} - 1\right)\alpha/2}{\cos\left(\lambda_j^{(1)} + 1\right)\alpha/2} = -\Lambda_j^{(1)}\frac{\sin\left(\lambda_j^{(1)} - 1\right)\alpha/2}{\sin\left(\lambda_j^{(1)} + 1\right)\alpha/2}$$

$$Q_j^{(2)} = -\frac{\sin\left(\lambda_j^{(2)} - 1\right)\alpha/2}{\sin\left(\lambda_j^{(2)} + 1\right)\alpha/2} = -\Lambda_j^{(2)}\frac{\cos\left(\lambda_j^{(2)} - 1\right)\alpha/2}{\cos\left(\lambda_j^{(2)} + 1\right)\alpha/2}$$

where

$$\Lambda_j^{(s)} = \frac{\lambda_j^{(s)} - 1}{\lambda_j^{(s)} + 1} \quad s = 1, 2$$

In the case of a crack,

$$Q_j^{(1)} = \begin{cases} -1 & j = 3, 5, 7, \ldots \\ -\Lambda_j^{(1)} & j = 1, 2, 4, 6, \ldots \end{cases}$$

$$Q_j^{(2)} = \begin{cases} -1 & j = 1, 2, 4, 6, \ldots \\ -\Lambda_j^{(2)} & j = 3, 5, 7, \ldots \end{cases}$$

For the edges in the mechanical part analyzed in Section 6,

$$Q_1^{(1)} = 0.599\ 081\ 789 \qquad Q_2^{(2)} = -0.032\ 551\ 169$$

The functions $u_{\zeta j}^{(3)}$ are given by Ref. [30] (assuming that the boundary is traction-free and neglecting body forces)

$$u_{\zeta j}^{(3)} = \begin{cases} \dfrac{r^{\lambda_j^{(3)}}}{2G}\sin\lambda_j^{(3)}\theta & j = 1, 3, 5, \ldots \\ \dfrac{r^{\lambda_j^{(3)}}}{2G}\cos\lambda_j^{(3)}\theta & j = 2, 4, 6, \ldots \end{cases}$$

where

$$\lambda_j^{(3)} = \frac{j\pi}{\alpha} \quad j = 1, 2, \ldots$$

Before using the above functions to build customized GFE shape functions, they have first to be transformed to the physical coordinates (x, y, z) as follows:

Define

$$\hat{u}_{\xi j}^{(s)}(\xi, \eta, \zeta) = u_{\xi j}^{(s)} \circ \mathbf{T}_1^{-1}(\xi, \eta, \zeta)$$
$$\hat{u}_{\eta j}^{(s)}(\xi, \eta, \zeta) = u_{\eta j}^{(s)} \circ \mathbf{T}_1^{-1}(\xi, \eta, \zeta) \quad s = 1, 2 \quad j = 1, \ldots, M$$
$$\hat{u}_{\zeta j}^{(3)}(\xi, \eta, \zeta) = u_{\zeta j}^{(s)} \circ \mathbf{T}_1^{-1}(\xi, \eta, \zeta)$$

$$\mathbf{T}_1^{-1}:(\xi, \eta, \zeta) \rightarrow (r, \theta, \zeta') \tag{12}$$

$$\begin{Bmatrix} r \\ \theta \\ \zeta' \end{Bmatrix} = \begin{Bmatrix} \sqrt{\xi^2 + \eta^2} \\ \arctan\left(\dfrac{\eta}{\xi}\right) \\ \zeta \end{Bmatrix} \tag{13}$$

The coordinate system (ξ, η, ζ) is shown in Fig. 3.

Next define

$$\tilde{u}_{\xi j}^{(s)}(x, y, z) = \hat{u}_{\xi j}^{(s)} \circ \mathbf{T}_2^{-1}(x, y, z)$$
$$\tilde{u}_{\eta j}^{(s)}(x, y, z) = \hat{u}_{\eta j}^{(s)} \circ \mathbf{T}_2^{-1}(x, y, z) \quad s = 1, 2 \quad j = 1, \ldots, M$$
$$\tilde{u}_{\zeta j}^{(3)}(x, y, z) = \hat{u}_{\eta j}^{(3)} \circ \mathbf{T}_2^{-1}(x, y, z)$$

$$\mathbf{T}_2^{-1}:(x, y, z) \rightarrow (\xi, \eta, \zeta) \tag{14}$$

$$\begin{Bmatrix} \xi \\ \eta \\ \zeta \end{Bmatrix} = \mathbf{R}_2^{-1}\begin{Bmatrix} x - O_x \\ y - O_y \\ z - O_z \end{Bmatrix} \tag{15}$$

where $\mathbf{R}_2^{-1} \in \mathbb{R}^3 \times \mathbb{R}^3$, with rows given by the base vectors of the coordinate system (ξ, η, ζ) written with respect of the base vectors of the coordinate system (x, y, z), and $\mathbf{O} = (O_x, O_y, O_z)$ are the coordinates of an arbitrary point along the edge (cf. Fig. 3). In the above, $\tilde{u}_{\xi j}^{(s)}(x, y, z)$, $\tilde{u}_{\eta j}^{(s)}(x, y, z)$, $s = 1, 2$ and $\tilde{u}_{\zeta j}^{(3)}(x, y, z)$ are the components of the displacement vectors in the directions ξ, η and ζ, respectively, written in terms of the physical coordinates x, y and z. To get the components of $\mathbf{u}$ in the directions x, y and z the following transformation needs to be applied:

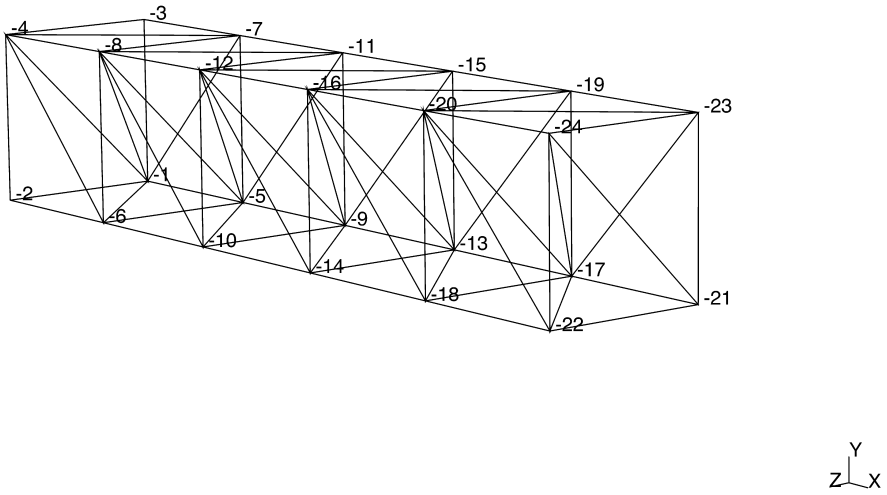

Fig. 4. Tetrahedral mesh used to investigate the structure of the stiffness matrix in GFEMs.

$$\begin{Bmatrix} u^{(s)}_{xj}(\mathbf{x}) \\ u^{(s)}_{yj}(\mathbf{x}) \\ u^{(3)}_{zj}(\mathbf{x}) \end{Bmatrix} = \mathbf{R}_2 \begin{Bmatrix} \tilde{u}^{(s)}_{\xi j}(\mathbf{x}) \\ \tilde{u}^{(s)}_{\eta j}(\mathbf{x}) \\ \tilde{u}^{(3)}_{\zeta j}(\mathbf{x}) \end{Bmatrix} \quad s = 1, 2 \quad j = 1, \ldots, M$$

where $\mathbf{R}_2 = (\mathbf{R}_2^{-1})^{\mathrm{T}}$.

The construction of customized GFE shape functions using singular functions as defined above, follows the same approach as in the case polynomial type shape functions. The singular functions are multiplied by the partition of unity functions φ_α associated with nodes near an edge. In the computations of Section 6, the following singular functions are used in the construction of customized GFE shape functions

$$u^{(1)}_{x1}, u^{(1)}_{y1}, u^{(3)}_{z1}, u^{(2)}_{x1}, u^{(2)}_{y1}, u^{(3)}_{z2}$$

which are given by

$$\begin{bmatrix} u^{(1)}_{x1} & u^{(2)}_{x1} \\ u^{(1)}_{y1} & u^{(2)}_{y1} \\ u^{(3)}_{z1} & u^{(3)}_{z2} \end{bmatrix} = \mathbf{R}_2 \begin{Bmatrix} \tilde{u}^{(1)}_{\xi 1} & \tilde{u}^{(2)}_{\xi 1} \\ \tilde{u}^{(1)}_{\eta 1} & \tilde{u}^{(2)}_{\eta 1} \\ \tilde{u}^{(3)}_{\zeta 1} & \tilde{u}^{(3)}_{\zeta 1} \end{Bmatrix}$$

The customized GFE shape functions are then built as

$$\varphi_\alpha \times \left\{ u^{(1)}_{x1}, u^{(1)}_{y1}, u^{(3)}_{z1}, u^{(2)}_{x1}, u^{(2)}_{y1}, u^{(3)}_{z2} \right\} \tag{16}$$

Here, α in the index of a finite element vertex node on or near an edge in 3D.

5. The structure of the stiffness matrix in GFEMs

In this section, we investigate how the structure of the stiffness matrix in the GFEM compares to that of the corresponding FEM matrix. More specifically, given a mesh, we analyze how the size and sparsity of the two matrices compare. In the case of linear approximations, the two matrices are of course identical. However, for higher degree elements the corresponding matrices can be quite different. To illustrate, consider the mesh of tetrahedral elements shown in Fig. 4. The structures of the stiffness matrices corresponding to ten-node quadratic tetrahedral finite elements and four-node quadratic generalized finite element, as defined in Eq. (2), are compared in this section.

The tetrahedral mesh shown in Fig. 4 is composed

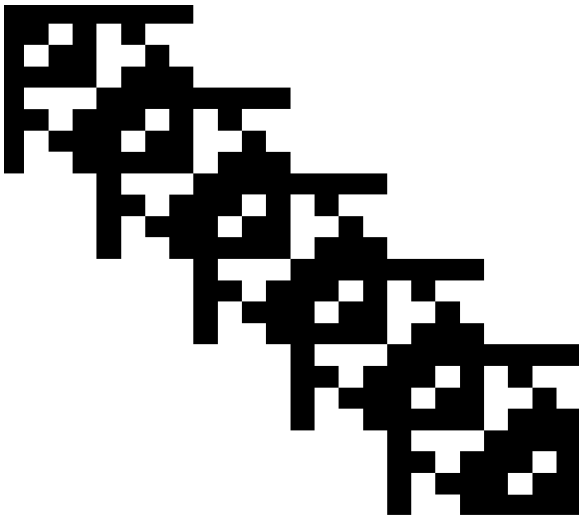

Fig. 5. Non-zero entries of the stiffness matrix for the mesh shown in Fig. 4 with linear elements.

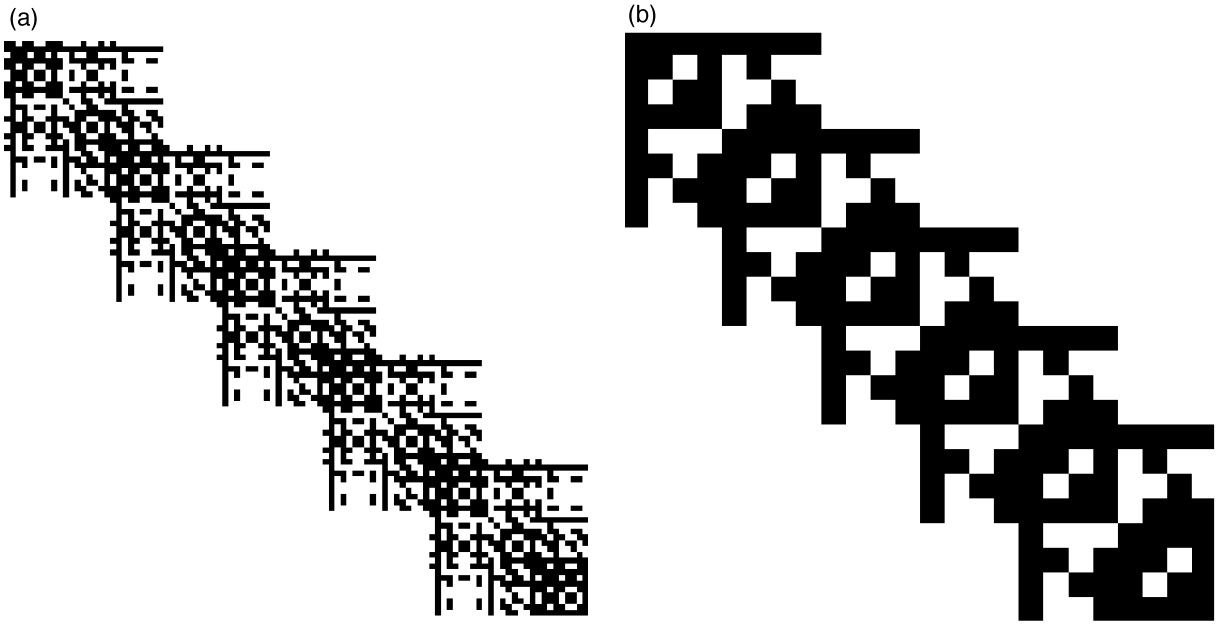

Fig. 6. Non-zero entries of the stiffness matrix for the mesh shown in Fig. 4: (a) matrix structure for ten-node quadratic tetrahedral elements; (b) matrix structure for generalized quadratic tetrahedral elements as defined in Eq. (2).

of 30 elements and 24 vertex nodes. Fig. 5 shows the structure of the stiffness matrix for this mesh with linear tetrahedral elements. The matrix has 72 rows and 819 non-zero entries (represented by the black part of the picture). Fig. 6(a) shows the structure of the stiffness matrix when ten-node finite elements are used. The matrix has 297 rows and 7,857 non-zero entries. If generalized quadratic finite element approximations are used, a stiffness matrix is obtained with the structure shown in Fig. 6(b). The matrix structure for the GFEM is seen to be simpler than that of the FEM. The GFE matrix has 288 rows and 12,672 non-zero entries. These data are shown on Table 1. The table also contains the number of floating point operations required for the numerical factorization of the matrices and the computed strain energy when the model is fixed at the left end and a uniform pressure $\mathbf{T} = (0.0, -1.0, 0.0)$ is applied at the right tip. The material is assumed to be linearly elastic with Young's modulus $E = 1{,}000{,}000$ PA and Poisson's ratio $\nu = 0.3$. The following observations can be made:

1. In the case of classical finite elements, the non-zero pattern of the matrix changes substantially when the approximation is enriched from $p = 1$ to $p = 2$. This is due to the fact that new nodes along the edges of the tetrahedral elements must be created. As discussed in Section 2, the p enrichment of the approximation in the GFEM does not require the creation of new nodes and therefore the structure of the matrix is unchanged with p enrichment, as is observed by comparing Figs. 5 and 6(b). It is also observed that there are several adjacent columns in the matrix shown in Fig. 6(b) with exactly the same number of non-zero entries. Several modern linear solvers can take advantage of this type of structure to improve the performance of the factorization

Table 1
FE and GFE results for the cantilever beam shown in Fig. 4[a]

Method	FEM/GFEM, $p = 1$	FEM, $p = 2$	GFEM, $p = 2$
Number of equations	72	297	288
Number of non-zeros lhs	819	7,857	12,672
Num. Float. Pt. Op.	1.595 1e+04	3.587 7e+05	7.551 4e+05
Strain energy	59.102 124 91	224.998 03	244.978 20

[a] The beam is fixed at the left and a uniform pressure $\mathbf{T} = (0.0, -1.0, 0.0)$ is applied at the right tip.

process. One example is the Boeing sparse linear solver [28] which is used in our implementations.

2. The number of equations, for the same mesh and the same degree of approximation, is smaller in the GFEM than in the FEM (288 versus 297 equations). This seems to be somewhat contradictory with the fact that a quadratic generalized tetrahedral element has 16 shape functions instead of 10 as in the classical FEM. However, in the GFEM all the dof are associated with vertex nodes which are shared by several elements (especially in a tetrahedral mesh) and therefore the assembled system of equations will in general be smaller than in the classical FEM where some nodes are shared by less elements. The ratio between the number of elements and the number of nodes for the mesh shown in Fig. 4 is $30/24 = 1.25$ which is quite low. In more realistic meshes, this ratio can be much larger, and consequently, the difference between the dimensions of the stiffness matrix in the FEM and GFEM is much more pronounced. One illustrative example is given in the next section.
3. The number of non-zeros in the stiffness matrix for the GFEM is substantially larger than in the corresponding matrix for the FEM (12,672 versus 7,857). This is a consequence of the fact that the support of the higher order shape functions in the GFEM is larger than the one in the classical FEM. The support of a shape functions (in the FEM or GFEM) is equal to the union of all elements sharing the node associated with the shape function. In the GFEM, we have only vertex nodes which are, in general, shared by more elements than edge, face and bubble nodes used in high order finite elements. In the next section, we demonstrate that the structure of the GFEM matrices, combined with the fact that for the same mesh and order of approximation their dimensions are smaller in the GFEM than in the FEM, more than compensate for their having more non-zeros *before* factorization. Our numerical experiments show that the factorization time is, for the class of problems and meshes analyzed, much smaller for the GFEM.

6. Analysis of a structural component using GFE methods

In this section, we analyze the structural part shown in Fig. 7 using traditional FE and GFE methods. The mesh used has 15,527 tetrahedral elements and 3,849 vertex nodal points. The mesh and problem size are representative of those used in, e.g., automotive and aerospace industry. The large ratio of the number of elements and number of nodes is typical for this class of problems.

The material is assumed to be linearly elastic with Young's modulus $E = 100{,}000$ PA and Poisson's ratio

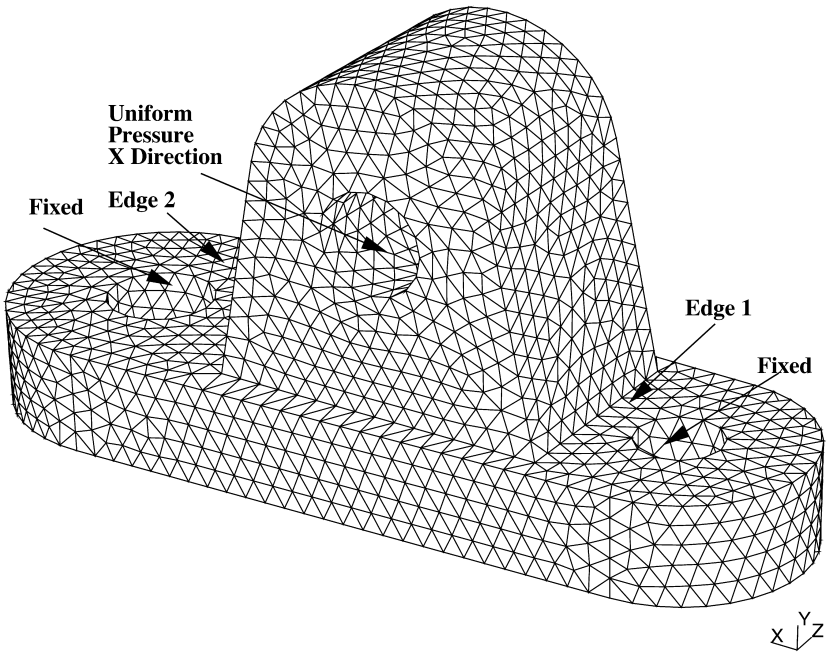

Fig. 7. Boundary conditions and mesh for structural part.

Table 2
Finite elements results using four- and ten-node tetrahedral elements

p	1	2
Build **K**, **F**(CPU, s)	271.30	2,729.71
Number of equations	11,547	76,797
Number of non-zeros **K**	218,844	2,965,077
Num. Fact. **K** (CPU, s)	36.21	2 737.83 (45.6 min)
Num. Float. Pt. Op.	9.422e+08	7.859e+10
MFlops	26.02	28.71
Strain energy	2.375 33	2.699 13
$\|e\|_E/\|u\|_E$	0.367 041	0.129 483

$\nu = 0.33$. The boundary conditions are those represented in Fig. 7. The component is fixed at both supports and there is a uniformly distributed load $p = 1.0$ in the negative x-direction applied at its upper part (see Fig. 7). The aspect ratio of the elements in the mesh are quite good. However, as we demonstrate below, if classical FEMs are used, this mesh can not properly model the singularities present in the model.

Linear and quadratic finite elements results are shown in Table 2. The value of the exact strain energy was estimated using an *hp* adapted finite element mesh with 99,990 dof and the a-posteriori error estimation capabilities of Phlex Solids [27]. In our computations, we adopt the following value for the exact strain energy

$$U(\mathbf{u}) = 2.745\ 15 \tag{17}$$

This value is used to compute the error in the energy norm for all FE and GFE models analyzed in this section. In Tables 2, 3, 5 and 6, 'Build, **K**, **F**(CPU, s)' denotes the CPU time, in seconds, for the numerical integration of stiffness matrix and load vector, 'Number of Non-zeros **K**' denotes the number of non-zero entries in the stiffness matrix, 'Num. Fact. **K**(CPU, s)' denotes the CPU time, in seconds, for the numerical factorization of the stiffness matrix, 'Num. Float. Pt. Op.' denotes the number of floating points operations required for the factorization of the stiffness matrix 'MFlops' is the computational rate attained in megaflops and 'Cond. Number Estim.' is an estimate of the condition number for the stiffness matrx. All the computations were performed on an Hewlett-Packard workstation model 735/125 running HP-UX 10.20.

The linear FE discretization leads to 11,547 dof which can be factorized in 36.21 s, using a sparse linear solver [28]. However, this discretization yields a solution with a relative error of 36.7% in the energy norm which is unacceptable for practical purposes. The quadratic discretization has an error of 12.95%, but it leads to a substantial increase in the number of dof (76,797) and in the solution time (2,737.8 s, which is 75.6 times the factorization time of the linear discretization). Moreover, this particular discretization can not adequately capture the stress singularities that exist in the model and *p*-enrichment will lead only to a small algebraic convergence rate.

In the GFE analysis of this problem, the linear tetrahedral finite element discretization is used as the partition of unity. In the first GFE discretization, the shape functions are as follows

$$\varphi_\alpha \times \left\{1, \frac{x - x_\alpha}{h_\alpha}\right\} \quad \alpha = 1, \ldots, N \tag{18}$$

where, as in Eq. (2), φ_α, $\alpha = 1, \ldots, N$, are standard trilinear Lagrangian tetrahedral shape functions, $\mathbf{x}_\alpha = (x_\alpha, y_\alpha, z_\alpha)$ are the coordinates of the node α, h_α is the diameter of the largest finite element sharing the node α and N is the number of nodes in the mesh. These shape functions are quadratic in the x-direction and linear in the other directions since they are built from the product of a polynomial (linear) partition of unity and the polynomial $x - x_\alpha$. We use the notation $p = 1 + p_x, 1 + p_y, 1 + p_z$ to denote the polynomial order of a GFE approximation, the '1s' indicating the linear order of the functions defining the partition of unity and p_x, p_y, p_z denote the degrees of the polynomials

Table 3
Generalized finite elements results using *p*-orthotropic approximations built on a tetrahedral mesh

p_x, p_y, p_z	1 + 0, 1 + 0, 1 + 0	1 + 1, 1 + 0, 1 + 0	1 + 1, 1 + 1, 1 + 1
Build **K**, **F**(CPU, s)	57.92	141.97	366.10
Number of equations	11,547	23,094	46,188
Number of non-zeros **K**	218,844	863,829	3,432,222
Num. Fact. **K**(CPU,s)	34.20	276.83	2062.13 (34.4 min)
Num. Float. Pt. Op.	9.422e+08	7.501e+09	5.986e+10
MFlops	27.55	27.10	29.03
Strain energy	2.375 35	2.550 13	2.674 91
$\|e\|_{E/}\|u\|_E$	0.367 029	0.266 537	0.159 969
Cond. Number Estim.	6.5270e+04		

used in the construction of the GFE shape functions in the x, y, z directions, respectively. Therefore, if only the partition of unity is used as a basis, we have $p = 1 + 0, 1 + 0, 1 + 0$. For the shape functions defined in Eq. (18) we have $p = 1 + 1, 1 + 0, 1 + 0$. This discretization is used to illustrate that in the GFE it is possible to build p-orthotropic approximations regardless of the underlying finite element mesh used. This choice is also motivated by the fact that the geometry, material properties, and boundary conditions of the problem do not change substantially in the z-direction.

In the second GFE discretization, the shape functions defined in Eq. (2) are used. In our notation, this discretization is of degree $p = 1 + 1, 1 + 1, 1 + 1$, i.e., it is a quadratic approximation in all directions (see Theorem 1). The GFE discretizations have dof only at the vertices of the tetrahedral elements (four dof for each component of the solution in the case of the quadratic discretization). That is, there are no dof along the edges or in the interior of the element. The $p = 1 + 1, 1 + 0, 1 + 0$ discretization has 23,094 dof (exactly twice the linear discretization) and a relative error in the energy norm of 26.6%. The quadratic GFE discretization has 46,188 dof which is only 60% of the number of dof in the finite element discretization. The total CPU time for the factorization of the resulting system of equations is 2,062.1 s, which is about 25% smaller than in the case of quadratic finite elements. The relative error in the energy norm for this discretization is 16.0% which is about 23.5% larger than in the case of quadratic finite elements.

The GFE results are summarized in Table 3. The quadratic GFEM discretization leads to a smaller system of equations than comparable FEM. However, the GFEM matrix has about 16% more non-zero entries than the FEM counterpart *before factorization*. Nonetheless, being a smaller matrix with a more favorable sparse structure, the GFEM matrix can be factorized using about 24% less floating point operations.

6.1. Modeling of singularities using special functions

Polynomial approximations, as used in the FE and GFE discretizations described above, can not efficiently approximate the solution in the neighborhood of geometric edges, such as Edges 1 and 2 shown in Fig. 7. However, asymptotic expansions of the elasticity solution in the neighborhood of such edges are well known and the GFE framework allows a straightforward inclusion of these asymptotic fields in the GFE approximation spaces, as described in Section 4. This approach is demonstrated in the solution of the problem represented in Fig. 7.

The same GFE discretizations described previously are used, but the nodes located along the Edges 1 and 2 are enriched with the GFE shape functions defined in Eq. (16). These functions are built from the product of the trilinear tetrahedral shape functions and the first terms of the mode I, II and III asymptotic expansions of the elasticity solution in the neighborhood of an edge. At the vertex of those edges the elasticity solution is more complex than along the edges (see for e.g., Refs. [14,23]). This feature is ignored in the calculations to be described: the same set of shape functions are used at all nodes located along the Edges 1 and 2. The enrichment of these nodes is justified by the fact that a large fraction the error is near those elements. Fig. 8 shows the error indicators computed for the finite element discretization with $p = 1$. The error indicators are computed using the element residual method (see, for example, Ref. [1] and the references therein) implemented in PhlexSolids [27]. Fig. 9 shows the error indicators on the boundary faces of the elements with error indicators larger than 50% of the largest element error indicator. It is observed that only a small fraction of the elements carries at least 50% of the total error and that the elements on Edges 1 and 2 are among them.

The distribution of the error in the domain can also be analyzed by considering the following sets of finite elements:

Set 0 all elements in the mesh,
Set 1 elements connected to one of the two vertices of Edge 1,
Set 2 elements connected to the Edge 1 but not connected to one of its vertices,
Set 3 elements connected to one of the two vertices of Edge 2,
Set 4 elements connected to the Edge 2 but not connected to one of its vertices,
Set 5 elements connected to one of the rims of the roles where displacement boundary condition is applied,
Set 6 elements not in Sets 1, 2, 3, 4 or 5.

The following quantities were computed using the classical finite element discretization with $p = 1$ and $p = 2$. The sum of the error,

$$\mathbf{e}^{\alpha} =: \left(\sum_{i \in \mathscr{I}^{\alpha}} \|e_i\|_E^2 \right)^{1/2} \qquad \text{Set } \alpha = 0, \ldots, 6 \tag{19}$$

the average error,

$$\bar{\mathbf{e}}^{\alpha} =: \left(\frac{\sum_{i \in \mathscr{I}^{\alpha}} \|e_i\|_E^2}{\operatorname{card} \mathscr{I}^{\alpha}} \right)^{1/2} \qquad \text{Set } \alpha = 0, \ldots, 6 \tag{20}$$

and the normalized average error,

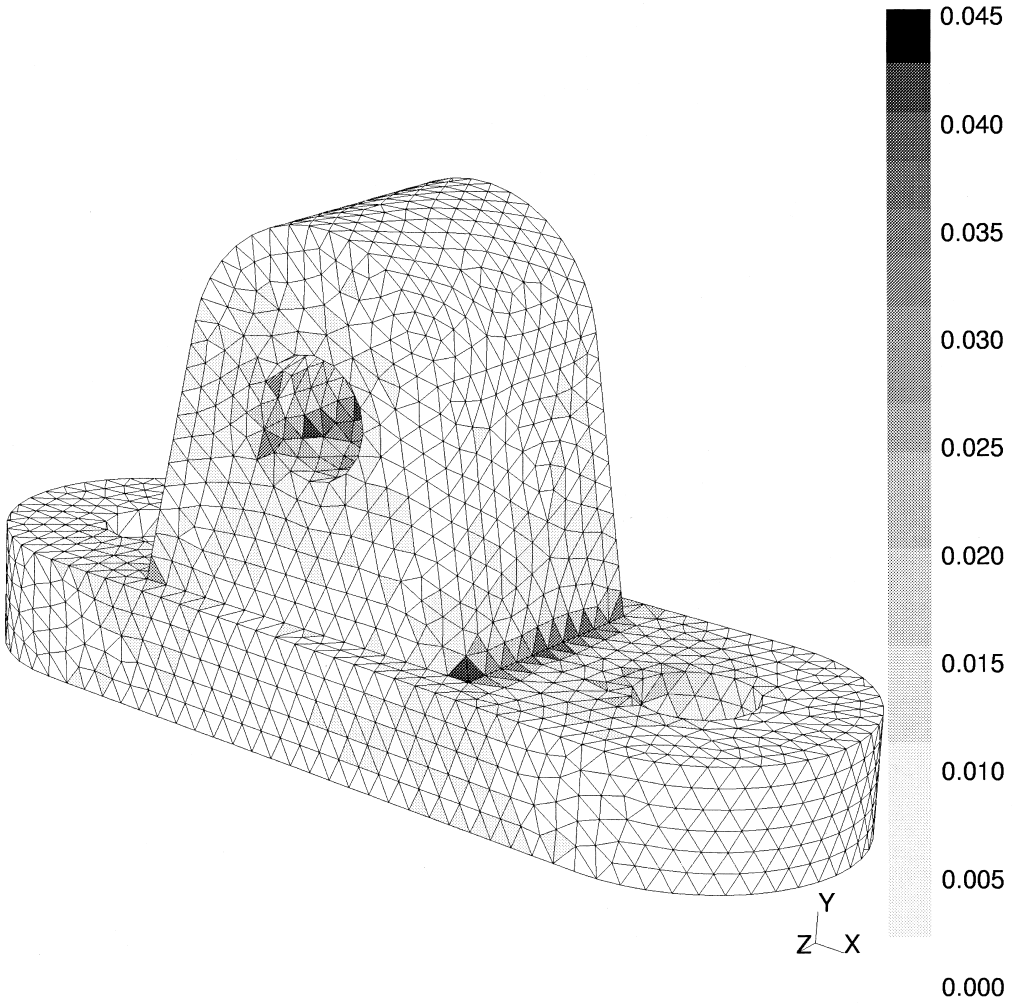

Fig. 8. Error indicators computed for the finite element discretization with $p = 1$.

Table 4
Average error indicators for several sets of elements when classical finite elements with $p = 1$ and 2 are used[a]

Set (α)	0	1	2	3	4	5	6
$\mathbf{e}^{\alpha}$ ($p = 1$)	1.567	1.592e−01	4.518e−01	1.298e−01	4.429e−01	3.298e−01	1.380e+00
$\bar{\mathbf{e}}^{\alpha}$ ($p = 1$)	1.258e−02	3.475e−02	4.541e−02	2.411e−02	4.204e−02	2.045e−02	1.126e−02
$\bar{\bar{\mathbf{e}}}^{\alpha}$ ($p = 1$)	1.0	2.763	3.611	1.917	3.343	1.626	0.896
$\mathbf{e}^{\alpha}$ ($p = 2$)	2.716e−01	3.507e−02	9.788e−02	2.723e−02	9.248e−02	6.872e−02	2.212e−01
$\bar{\mathbf{e}}^{\alpha}$ ($p = 2$)	2.180e−03	7.652e−03	9.837e−03	5.057e−03	8.778e−03	4.262e−03	1.806e−03
$\bar{\bar{\mathbf{e}}}^{\alpha}$ ($p = 2$)	1.0	3.511	4.513	2.320	4.027	1.955	0.829

[a] The quantities $\mathbf{e}^{\alpha}$, $\bar{\mathbf{e}}^{\alpha}$ and $\bar{\bar{\mathbf{e}}}^{\alpha}$ are defined in Eqs. (19), (20) and (21), respectively.

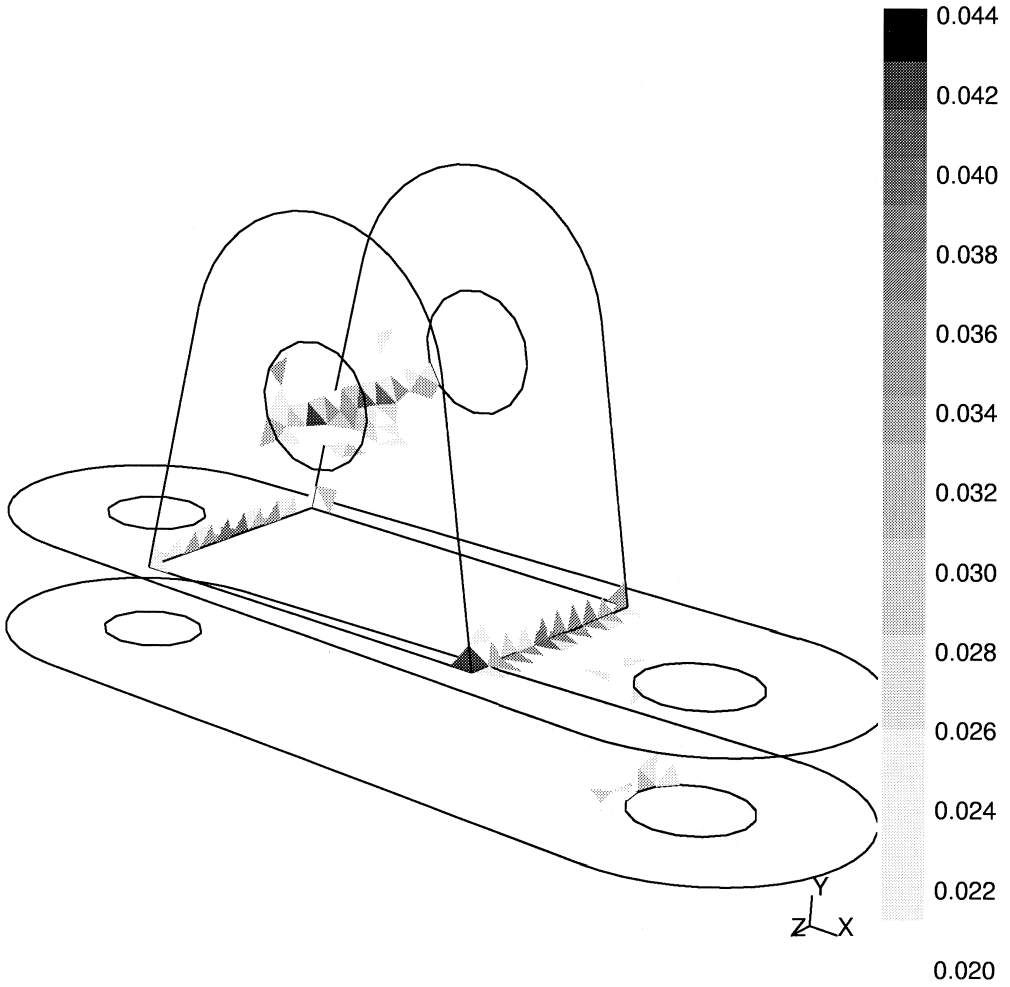

Fig. 9. Error indicators on the boundary faces of the elements with error indicators larger than 50% of the largest element error indicator.

Table 5
Generalized finite elements results using p-orthotropic approximations and singular shape functions at nodes along Edges 1 and 2

p_x, p_y, p_z	1 + 0, 1 + 0, 1 + 0	1 + 1, 1 + 0, 1 + 0	1 + 1, 1 + 1, 1 + 1
Build **K**, **F**(CPU,s)	201.39	428.39	936.32
Number of equations	11,943	23,490	46,584
Number of non-zeros **K**	244,368	904,635	3,503,592
Num. Fact. **K**(CPU,s)	41.22	316.23	2,156.57 (35.9min)
Num. Float. Pt. Op.	9.476e+08	7.355e+09	6.260e+10
MFlops	22.99	23.26	29.03
Strain energy	2.435 777	2.586 049	2.711 088
$\|e\|_{E/}\|u\|_E$	0.335 707	0.240 746	0.111 399
Cond. Number Estim.	7.3942e+04		

$$\bar{\bar{\mathbf{e}}}^{\alpha} = :\frac{\bar{\mathbf{e}}^{\alpha}}{\bar{\mathbf{e}}^{\alpha=0}} \quad \text{Set } \alpha = 0,\ldots,6 \tag{21}$$

where e_i is the error on the element i, $\mathscr{I}^{\alpha}$ denotes an index set for the Set α and $\|\cdot\|_E$ denotes the energy norm. The results are shown on Table 4.

It is observed that the average error $\bar{\mathbf{e}}^{\alpha}$ for the elements along Edges 1 and 2 (Sets 1, 2, 3, 4) is considerably larger than the average error on the whole domain (Set 0). In addition, those elements represent only a small fraction of the total number of elements in the mesh, which justifies our choice of using customized singular shape functions only along those edges.

The results for the enriched GFE discretizations are shown in Table 5. It is observed that the addition of the singular functions along the Edges 1 and 2 increases the total number of dof by only 396. This is only 1% more dof in the case of the GFE discretization with $p = 1 + 1, 1 + 1, 1 + 1$ which increases the solution time by only 4.5%. The effect of this enriched shape functions on the discretization error, however, is quite noticeable. They lead to a decrease of about 30% in the discretization error for GFE with $p = 1 + 1, 1 + 1, 1 + 1$. We also investigate the effect of additionally enriching the nodes connected to Edges 1 and 2 by a finite element *and* located at the boundary of the part. The results for this discretization are shown in Table 6. It is observed that the effect of this additional enrichment is not substantial.

The enrichment of the GFE discretization with singular functions leads to the issue of numerical integration of these functions. In this work, our main goal is to investigate the benefits of adding these functions in terms of controling the discretization error. A sufficiently high quadrature rule was used on the elements with nodes carrying singular functions. The order of the quadrature rule was chosen so that the numerical integration errors were small enough to not affect the computed discretization errors. Table 7 shows the computed strain energy and relative discretization error $\|e|_{E/}\|u\|_E$ for the GFE discretization $p = 1 + 0, 1 + 0, 1 + 0$ enriched with singular functions along Edges 1 and 2 as a function of the number of integration points used in the elements with nodes carrying singular functions. The rules with 4, 11, 24 and 45 points are Keast [16] quadrature rules and the others are tensor product Gaussian quadrature. The discretization error is computed using the reference value (17) for the exact strain energy. Based on these computations, and the fact that we use discretizations with an error of more than 5%, we employ tensor product Gaussian quadrature with 1000 points on the elements with nodes carrying singular functions. A more computationally efficient approach is, of course, to use adaptive integration on those elements and set the tolerance of the numerical integration according to the required accuracy of the approximate solution.

Table 6
Generalized finite elements results using singular functions at nodes along Edges 1 and 2 and at nodes connected to these edges by a finite element and located at the boundary of the part

p_x, p_y, p_z	1 + 1, 1 + 1, 1 + 1
Build **K**, **F**(CPU, s)	1,276.36
Number of equations	47,448
Number of non-zeros **K**	3,646,476
Num. Fact. **K**(CPU, s)	2,072.94 (34.5 min)
Num. Float. Pt. Op.	6.043e+10
MFlops	29.15
Strain energy	2.712 563 9
$\|e\|_{E/}\|u\|_E$	0.108 959 8

Table 7
Computed strain energy and discretization error for various choices of quadrature. The rules with 4,11,24 and 45 points are Keast [16] quadrature rule). (the others are tensor product Gaussian quadrature rule. The discretization used is GFE with $p = 1 + 0, 1 + 0, 1 + 0$ and singular functions at nodes along Edges 1 and 2

Num. Int. Pts.	Strain energy	$\|e\|_{E/}\|u\|_E$
4	2.458 221 212	0.323 301 264
11	2.437 851 048	0.334 580 503
24	2.436 257 160	0.335 447 063
45	2.436 024 790	0.335 573 210
216	2.436 094 921	0.335 535 142
343	2.435 938 998	0.335 619 772
512	2.435 846 523	0.335 669 954
729	2.435 777 180	0.335 707 578
1000	2.435 739 019	0.335 728 282
1331	2.435 710 261	0.335 743 883
1728	2.435 684 414	0.335 757 904

Another important issue is the effect of the enrichment with singular functions on the condition number of the stiffness matrix. Estimates of the condition number are shown in Tables 5 and 3. It can be observed that the enrichment with singular functions of the nodes along Edges 1 and 2 has no detrimental effect on the conditioning of the stiffness matrix. The enrichment with singular functions does not lead to linear dependences as in the case of polynomial enrichment. This last case is discussed in Section 3.

7. Conclusions

The key feature of the Generalized Finite Element Method is the use of a partition of unity to build the approximation spaces. This partition of unity

framework has several powerful properties such as the ability to produce seamless *hp* finite element approximations with nonuniform *h* and *p*, the ability to develop customized approximations for specific applications, the capability to build *p*-orthotropic approximations on, e.g., 3D tetrahedral meshes, etc. Several of the so-called meshless methods proposed in recent years also make use, explicitly or implicitly, of a partition of unity to build the approximation spaces [7,11,18,22]. The fundamental difference between these methods and the GFEM is in the choice of the partition of unity. In the GFEM, the partition of unity is provided by conventional finite element methods. In this case, the implementation of the method is essentially the same as in standard finite element codes, the main difference being the definition of the shape functions. This choice of partition of unity avoids the problem of numerical integration associated with the use of moving least squares or Shepard partitions of unity common to several meshless methods. In addition, the use of a finite element partition of unity allows easy implementation of essential boundary conditions, which may not be a straightforward proposition in other techniques using moving least squares partition of unity.

In this paper, we investigate several important practical aspects of the generalized finite element method in a 3D setting. The structure of the stiffness matrix in the GFEM is compared with the finite element counterpart when the same mesh and polynomial degree of the approximation is used. We show that the structure of the matrix in the GFEM does not change with *p*-enrichment, in contrast with the classical FEM case. Modern linear solvers can take advantage of the type of structure in the GFE matrices to improve the performance of the factorization process. It is also demonstrated that for the same mesh and degree of approximation, the GFEM leads to a smaller system of equations than in the classical FEM. This difference in the number of equations is specially pronounced in meshes where the ratio between the number of elements and nodes in the mesh is high, as in the case of tetrahedral meshes of complex structures. This is translated into faster solution times for the GFEM.

The GFEM framework allows for straightforward construction of special solution spaces using a-priori knowledge of properties of the solution of the problem. In this paper, this is demonstrated for the case of elasticity equations in three dimensions. This procedure does not require the modification of the finite element mesh. In general, these properties allows us to obtain the solution of 3D problems with higher accuracy and less computational effort than the classical finite element method.

Acknowledgements

The work of J.T. Oden on this project was supported by Army Research Office under grant DAAH04-96-1-0062.

References

[1] Ainsworth M, Oden JT. A posteriori error estimation in finite element analysis. Computational Mechanics Advances Special Issue of Computer Methods in Applied Mechanics and Engineering 1997;142:1–88.
[2] Babuska I, Caloz G, Osborn JE. Special finite element methods for a class of second order elliptic problems with rough coefficients. SIAM J Numerical Analysis 1994;31(4):745–981.
[3] Babuška I, Melenk JM. The partition of unity finite element method. International Journal for Numerical Methods in Engineering 1997;40:727–58.
[4] Bathe KJ. Finite element procedures in engineering analysis. Englewood Cliffs, NJ: Prentice-Hall, 1982.
[5] Becker EB, Carey GF, Oden JT. Finite elements: an introduction. Englewood Cliffs: Prentice-Hall, 1981 Volume I in Texas finite element series.
[6] Belytschko T, Krongauz Y, Organ D, Fleming M. Meshless methods: An overview and recent developments. Computer Methods in Applied Mechanics and Engineering 1996;139:3–47.
[7] Belytschko T, Lu YY, Gu L. Element-free Galerkin methods. International Journal for Numerical Methods in Engineering 1994;37:229–56.
[8] Duarte CAM. A review of some meshless methods to solve partial differential equations. Technical Report 95-06, TICAM, University of Texas at Austin, 1995.
[9] Duarte CAM, Oden JT. *Hp* clouds — a meshless method to solve boundary-value problems. Technical Report 95-05, TICAM, University of Texas at Austin, May 1995.
[10] Duarte CAM, Oden JT. An *hp* adaptive method using clouds. Computer Methods in Applied Mechanics and Engineering 1996;139:237–62.
[11] Duarte CAM, Oden JT. *Hp* clouds — an *hp* meshless method. Numerical Methods for Partial Differential Equations 1996;12:673–705.
[12] Duarte CA. The *hp* cloud method. PhD dissertation, University of Texas at Austin, Austin, TX, USA, December 1996.
[13] Duff I, Reid J. The multifrontal solution of indefinite sparse symmetric linear systems. ACM Trans Math Softw 1983;9:302–25.
[14] Grisvard P. Singularities in boundary value problems. New York: Springer-Verlag, 1982 Research notes in Appl. Math.
[15] Babuška I, Strouboulis T, Copps K, Gangara SK, Upadhyay CS. A-posteriori error estimation for finite element and generalized finite element method. Available from http://yoyodyne.tamu.edu/research/error/gfem_france.pdf.
[16] Keast P. Moderate-degree tetrahedral quadrature for-

mulas. Computer Methods in Applied Mechanics and Engineering 1986;55:339–48.
[17] Lancaster P, Salkauskas K. Surfaces generated by moving least squares methods. Mathematics of Computation 1981;37(155):141–58.
[18] Liu WK, Jun S, Zhang YF. Reproducing kernel particle methods. International Journal for Numerical Methods in Engineering 1995;20:1081–106.
[19] Melenk JM. Finite element methods with harmonic shape functions for solving laplace's equation. Master's thesis, University of Maryland, 1992.
[20] Melenk JM. On generalized finite element methods. Ph.D. thesis, University of Maryland, 1995.
[21] Melenk JM, Babuška I. The partition of unity finite element method: basic theory and applications. Computer Methods in Applied Mechanics and Engineering 1996;139:289–314.
[22] Nayroles B, Touzot G, Villon P. Generalizing the finite element method: diffuse approximation and diffuse elements. Computational Mechanics 1992;10:307–18.
[23] Nazarov SA, Plamenevsky BA. Elliptic problems in domains with piecewise smooth boundaries. Berlin: Walter de Gruyter, 1994 volume 13 of De Gruyter Expositions in Mathematics.
[24] Oden JT, Duarte CA. Clouds, cracks and fem's. In: Reddy BD, editor. Recent developments in computational and applied mechanics. Barcelona, Spain: International Center for Numerical Methods in Engineering, CIMNE, 1997. p. 302–21.
[25] Oden JT, Duarte CA, Zienkiewicz OC. A new cloud-based *hp* finite element method. Computer Methods in Applied Mechanics and Engineering 1998;153:117–26.
[26] Oden JT, Duarte CAM. Solution of singular problems using *hp* clouds. In: Whiteman JR, editor. The mathematics of finite elements and applications 96. New York, NY: Wiley, 1982. p. 35–54.
[27] PHLEXSolids. Computational Mechanics Company, www.comco.com.
[28] The Boeing extended mathematical subprogram library. Boeing Computer Services, Seattle, WA, USA.
[29] Strouboulis T, Babuška I, Copps K. The design and analysis of the generalized finite element method. Computer Methods in Applied Mechanics and Engineering 2000;81(1–3):43–69.
[30] Szabo BA, Babuska I. Computation of the amplitude of stress singular terms for cracks and reentrant corners. In: Cruse TA, editor. Fracture Mechanics: Nineteenth Symposium, ASTM STP 969, 1988, pp. 101–124.
[31] Barna Szabo, Babuska Ivo. Finite element analysis. New York: Wiley, 1991.
[32] Zienkiewicz OC, Taylor RL. In: 4th ed. The finite element mehtod, vol. I. New York: McGraw-Hill, 1981.