Generalized Finite Element Methods for Three Dimensional Structural Mechanics Problems

C. A. Duarte

COMCO, Inc., 7800 Shoal Creek Blvd. Suite 290E Austin, Texas, 78757, USA

I. Babuška and J. T. Oden

TICAM, The University of Texas at Austin, Taylor Hall 2.400, Austin, TX, 78712, USA

Summary

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 structural mechanics problems.

Introduction

The analysis of complex three-dimensional structural components has become a common task in recent years at several industries, like, automotive, aerospace, naval, nuclear, etc. However, the analysis of this class of problems using traditional finite element methods still poses several difficulties. It is a common practice in the industry to use automatic tetrahedral mesh generators to discretize complex three-dimensional 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 this approach are:

(i) The need of an excessive number of elements in order to keep the aspect ratio of the finite elements within a reasonable bound. This is specially true when the component has transitions zones from bulky to slender parts. For example, around an oil well, which has a few inches of diameter, in a oil reservoir, which can have several miles.

(ii) Polynomial approximations, as used in traditional finite element methods, requires the use of a large number of elements in order to capture stress concentrations and singularities at corners and edges of the domain.

(iii) One other drawback of automatic tetrahedral mesh generators is that they preclude the use of p anisotropic approximations, that is, approximations that have different polynomial orders associated with each direction.

The generalized finite element method (GFEM) was proposed independently by Babuška and colleagues [1, 2, 12] (under the names special finite element methods, generalized finite element method and finite element partition of unity method) and by Duarte and Oden [6–9, 15] (under the names *hp* clouds and cloud-based *hp* finite element method). Several of the so-called meshless methods proposed in the last years can also be seem as special cases of the generalized finite element method. Recent surveys on meshless methods can be found in [3, 5]. 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 $\boldsymbol{x}$ in the domain Ω. The analysis of the performance and computational cost of *hp* cloud, element free

Galerkin [4], diffuse element [13] and reproducing kernel particle [11] methods can be found in [9]. 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. Moving least square PU [10], being among the most expensive.

The present paper summarizes the GFEM formulation and demonstrates through a numerical example some of the advantages of GFEM over traditional FEM to solve complex, three-dimensional structural mechanics problems.

Formulation of Generalized Finite Element Methods: A Summary

In this section we review the basic formulation of generalized finite element approximations in a one-dimensional setting using a linear finite element partition of unity. However, the procedures are exactly the same in two- and three-dimensions and can be used in combination with any Lagrangian finite element. In the next section, we use generalized tetrahedral finite elements to discretize a three-dimensional problem. For a more detailed discussion we refer the interested reader to [2, 7, 12, 15] and the references therein.

Let $\boldsymbol{u}(\boldsymbol{x})$ be a function defined on a domain $\Omega \subset \mathbb{R}$. Suppose that we build a covering

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

of Ω consisting of N *clouds* ω_α with centers at $\boldsymbol{x}_\alpha$.

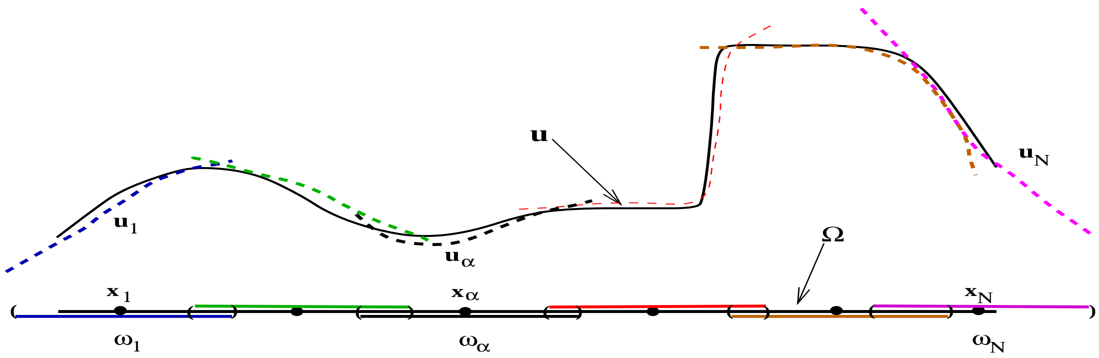

Figure 1: Local Approximations Defined on the Clouds ω_α.

Let $\boldsymbol{u}_\alpha$ be a local approximation of $\boldsymbol{u}$ that belongs to a local space $\chi_\alpha(\omega_\alpha)$ defined on the cloud ω_α. Each space $\chi_\alpha(\omega_\alpha), \alpha = 1, \ldots, N$ can be chosen such that there exists a $\boldsymbol{u}_\alpha \in \chi_\alpha(\omega_\alpha)$ that can approximate well $\boldsymbol{u}|_{\omega_\alpha}$. Figure 1 illustrates the definitions given above. In this case, the clouds ω_α are open intervals with centers $\boldsymbol{x}_\alpha$.

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

$$\begin{array}{ll} \varphi_\alpha \in C_0^s(\omega_\alpha), \; s \geq 0, & 1 \leq \alpha \leq N \\ \sum_\alpha \varphi_\alpha(\boldsymbol{x}) = 1 & \forall \boldsymbol{x} \in \Omega \end{array}$$

The functions φ_α are called a partition of unity subordinated to the open covering $\mathcal{T}_N$. Examples of partitions of unity are Lagrangian finite elements, moving least squares and Shepard functions [8, 10].

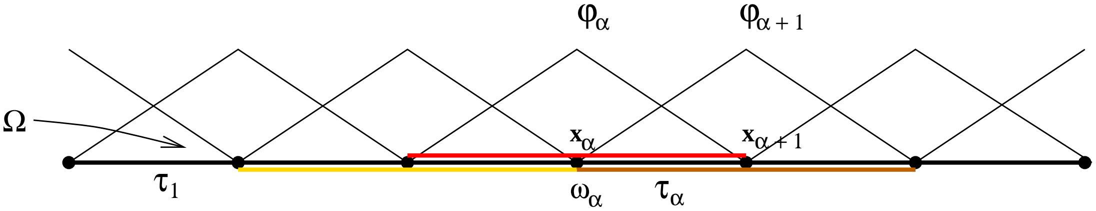

Figure 2: One dimensional finite element partition of unity.

In the case of finite element partitions of unity, the clouds ω_α are simply the union of the finite elements sharing a vertex node (see, for example, [12, 15]). 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. Figure 2 shows a one dimensional finite element discretization. The partition of unity function φ_α is the global finite element shape function associated with node $\boldsymbol{x}_\alpha$. The cloud ω_α is the union of the elements $\tau_{\alpha-1}$ and τ_α.

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

$$S_\alpha = \{\varphi_\alpha, \varphi_{\alpha+1}\} \times \{1, \boldsymbol{u}_\alpha, \boldsymbol{u}_{\alpha+1}\} = \{\varphi_\alpha, \varphi_{\alpha+1}, \varphi_\alpha \boldsymbol{u}_\alpha, \varphi_\alpha \boldsymbol{u}_{\alpha+1}, \varphi_{\alpha+1} \boldsymbol{u}_\alpha, \varphi_{\alpha+1} \boldsymbol{u}_{\alpha+1}\}$$

That is, the element τ_α has a total of six shape functions built from the product of the standard Lagrangian finite element shape functions (a partition of unity), and the local approximations $\boldsymbol{u}_\alpha, \boldsymbol{u}_{\alpha+1}$ that, by assumption, can approximate well the function $\boldsymbol{u}$ over the finite element τ_α.

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 $\boldsymbol{u}_\alpha, \boldsymbol{u}_{\alpha+1}$, that is,

$$\begin{aligned} \varphi_\alpha \boldsymbol{u}_\alpha + \varphi_{\alpha+1} \boldsymbol{u}_\alpha &= \boldsymbol{u}_\alpha(\varphi_\alpha + \varphi_{\alpha+1}) = \boldsymbol{u}_\alpha \qquad \text{(no sum on } \alpha) \\ \varphi_\alpha \boldsymbol{u}_{\alpha+1} + \varphi_{\alpha+1} \boldsymbol{u}_{\alpha+1} &= \boldsymbol{u}_{\alpha+1}(\varphi_\alpha + \varphi_{\alpha+1}) = \boldsymbol{u}_{\alpha+1} \end{aligned}$$

In other words, $\boldsymbol{u}_\alpha, \boldsymbol{u}_{\alpha+1} \in < S_\alpha >$. It can also be proved that [8,12] if $\|\boldsymbol{u} - \boldsymbol{u}_\alpha\|_{E(\Omega \cap \omega_\alpha)} \leq \epsilon(\alpha, h, p, \boldsymbol{u})$ then there $\exists\ \boldsymbol{u}_{hp} \in \boldsymbol{X}^{hp}$ such that $\|\boldsymbol{u} - \boldsymbol{u}_{hp}\|_{E(\Omega)} \leq C \left(\sum_{\alpha=1}^{N(h)} \epsilon^2(\alpha, h, p, \boldsymbol{u})\right)^{1/2}$ where the space $\boldsymbol{X}^{hp}$ is the span of the generalized finite element shape functions as defined above and the constant C is independent of $\boldsymbol{u}, h, p$.

Note that there is a great freedom in the choice of the local spaces χ_α (and consequently of the local approximations $\boldsymbol{u}_\alpha$). 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 local approximation $\boldsymbol{u}_\alpha, \alpha = 1, \ldots, N$ can have a different polynomial order, we can have different polynomial orders associated with each vertex node of the finite element mesh. The approximations can also be non-isotropic, regardless of the choice of FE PU (hexahedral, tetrahedral, etc.). The concept of edge and middle nodes, which are 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 functions and the use of polynomials to build the approximation space, as in the FEM, is far from optimal and leads to a poor approximation of the solution $\boldsymbol{u}$ unless carefully designed meshes are used. In the GFEM we can use any a-priori knowledge we have about the solution to make a wise choice for the local spaces χ_α. For example, in the next section we solve a boundary value problem in which the solution is singular at some parts of the domain and we use local spaces χ_α to build our generalized finite element shape functions space that can represent these singularities much more effectively than polynomial functions.

Analysis of a Structural Component Using GFE Methods

In this section, we analyze the structural part shown on Fig. 3 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 aspect ratio of the elements in the mesh and the approximation of the geometry of the structural part are quite good. However, this mesh is not adequate to properly model the singularities present in the model. The material is assumed to be linearly elastic with Young's modulus $E = 100000.0$ and Poisson's ratio $\nu = 0.33$. The boundary conditions are those represented in Fig. 3. 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 (Cf. Fig. 3).

Linear and quadratic (Serendipity) finite elements results are shown in Table 1. The value of the exact strain energy was estimated using the quadratic finite element discretization and the a-posteriori error estimation capabilities of Phlex Solids [16]. The linear discretization leads to 11,547 degrees of freedom (dofs) which can be factorized in 36.21 seconds using the sparse linear solver option of Phlex Solids [16]. However this discretization has a relative error of 36.3% in the energy norm which is unacceptable for all practical purposes. The quadratic discretization has an error of 11.61 % and it leads to a substantial increase in the number of dofs (76,797) and in the solution time (2,737.8 seconds). This discretization is not adequate to capture the stress singularities that exist in the model and p-enrichment will lead only to algebraic convergence rates.

In the GFE analysis of this problem the linear tetrahedral discretization is used as the partition of unity. Two discretizations are then build using this partition of unity and the technique outlined in the previous section. In the first GFE discretization the local spaces $\chi_\alpha, \alpha = 1, \ldots, N$ are the span of the polynomials $1, x - x_\alpha, y - y_\alpha, z - z_\alpha$ where $\boldsymbol{x}_\alpha = (x_\alpha, y_\alpha, z_\alpha)$. In other words, the partition of unity is multiplied by linear monomials, $1, x - x_\alpha, y - y_\alpha, z - z_\alpha$, to create quadratic GFE shape functions. The resulting approximation is quadratic because it is built from the product of a polynomial (linear) PU and linear monomials. This discretization has degrees of freedom only at the vertices of the tetrahedral elements (four dofs for each component of the solution). That is, there are no degrees of freedom along the edges or in the interior of the element. This quadratic GFE discretization has 46,188 dofs which is about 40% less dofs than the finite element counterpart. The total CPU time for the factorization of the resulting system of equations is 2,062.1 seconds, which is about 25% smaller than in the case of quadratic finite elements. The relative error in the energy norm for this discretization is 14.94% which is about 22.3 % larger than in the case of quadratic finite elements. The GFE results are summarized in Table 1.

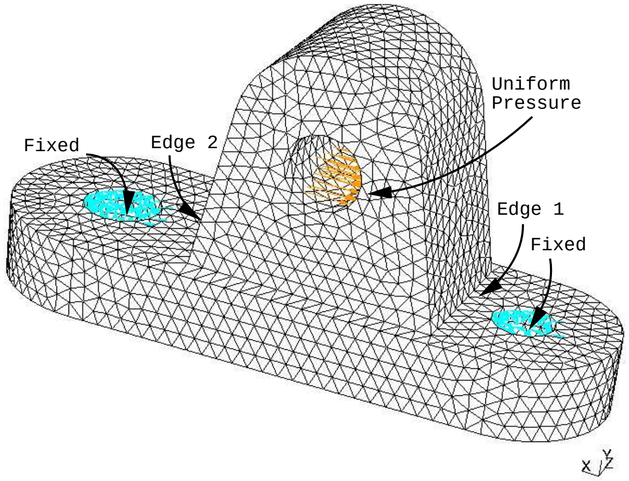

Figure 3: Boundary conditions and mesh for structural part.

Polynomial approximations as used in the FE and GFE discretizations described above can not approximate well the solution in the neighborhood of, e.g., the edges 1 and 2 shown in Fig. 3. However, asymptotic expansions of the elasticity solution in the neighborhood of those edges are well known and the GFE framework allows straightforward inclusion of these asymptotic fields in the GFE approximation spaces (for details see for example [9,14]). We demonstrate this approach in the solution of the problem represented in Fig. 3. The same quadratic GFE discretization as described above is used but the nodes located along the edges 1 and 2 are enriched with functions built from the product of the linear 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. The results for this discretization are shown in Table 1. It can be observed that the addition of the singular functions along the edges 1 and 2 increases the number of dofs by only 396 dofs (less than 1% more dofs) and increases the solution time by only 4.5 %. The effect of this enriched functions on the discretization error, however, is quite noticeable. They lead to a decrease of about 36% in the discretization error.

Table 1: Tetrahedral FE and Generalized FE Results. All reported timings are in seconds

Method	FEM p=1	FEM p=2	GFEM p=2	GFEM p=2+ sing. fn.
Integration of LHS, RHS	271.30	2,729.71	366.10	936.32
Number of Equations	11,547	76,797	46,188	46,584
Numerical Factorization	36.21	2,737.83	2,062.13	2,156.57
Num. Float. Pt. Operat.	9.422E+08	7.859E+10	5.986E+10	6.260E+10
Strain Energy	2.37533	2.69913	2.67491	2.7110881
$\|e\|_E/\|u\|_E$	0.363081	0.116106	0.14944	0.0954462

Table 1 reports the CPU time for the numerical integration and assembly of stiffness matrix and load vector, the number of equations, the CPU time for the numerical factorization, the number of floating point operations for the factorization of the stiffness matrix, the computed strain energy and the estimated relative error in the energy norm. All the computations were performed on an Hewlett-Packard workstation model 735/125 running HP-UX 10.20.

Conclusions

In the GFEM the PU is provided by conventional finite element methods. While such an approach bypasses some of the advantages of the meshless formulation based on, e.g., Shepard PU, most other useful properties of the cloud/PU methods are retained, 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., three-dimensional tetrahedral meshes, etc. On top of that, the use of a finite element PU allows ease implementation of essential boundary conditions and efficient integration of the stiffness matrix.

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

References

[1] I. Babuska, G. Caloz, and Osborn J. E. Special finite element methods for a class of second order elliptic problems with rough coefficients. *SIAM J. Numerical Analysis*, 31(4):745–981, 1994.

[2] I. Babuška and J. M. Melenk. The partition of unity finite element method. *International Journal for Numerical Methods in Engineering*, 40:727–758, 1997.

[3] T. Belytschko, Y. Krongauz, D. Organ, and M. Fleming. Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering*, 139:3–47, 1996.

[4] T. Belytschko, Y. Y. Lu, and L. Gu. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 37:229–256, 1994.

[5] C. A. M. Duarte. A review of some meshless methods to solve partial differential equations. Technical Report 95-06, TICAM, The University of Texas at Austin, 1995.

[6] C. A. M. Duarte and J. T. Oden. Hp clouds–a meshless method to solve boundary-value problems. Technical Report 95-05, TICAM, The University of Texas at Austin, May 1995.

[7] C. A. M. Duarte and J. T. Oden. An *hp* adaptive method using clouds. *Computer Methods in Applied Mechanics and Engineering*, 139:237–262, 1996.

[8] C. A. M. Duarte and J. T. Oden. *Hp* clouds—an *hp* meshless method. *Numerical Methods for Partial Differential Equations*, 12:673–705, 1996.

[9] C. Armando Duarte. *The* hp *Cloud Method*. PhD thesis, The University of Texas at Austin, Austin, TX, USA, December 1996.

[10] P. Lancaster and K. Salkauskas. Surfaces generated by moving least squares methods. *Mathematics of Computation*, 37(155):141–158, 1981.

[11] W. K. Liu, S. Jun, and Y. F. Zhang. Reproducing kernel particle methods. *International Journal for Numerical Methods in Engineering*, 20:1081–1106, 1995.

[12] J. M. Melenk and I. Babuška. The partition of unity finite element method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 139:289–314, 1996.

[13] B. Nayroles, G. Touzot, and P. Villon. Generalizing the finite element method: Diffuse approximation and diffuse elements. *Computational Mechanics*, 10:307–318, 1992.

[14] J. T. Oden and C. A. Duarte. *Solution of Singular Problems Using* h-p *Clouds*, chapter 2, pages 35–54. John Wiley & Sons, New York, NY, 1997.

[15] J. T. Oden, C. A. Duarte, and O. C. Zienkiewicz. A new cloud-based *hp* finite element method. *Computer Methods in Applied Mechanics and Engineering*, 153:117–126, 1998.

[16] Phlex solids. Computational Mechanics Company Inc., www.comco.com.