

# Arbitrarily Smooth Generalized Finite Element Approximations

C. A. Duarte\* , D.-J. Kim

*Department of Civil and Environmental Engineering University of Illinois at Urbana-Champaign Newmark Laboratory, 205 North Mathews Avenue Urbana, Illinois 61801, USA*

D. M. Quaresma

*Department of Aerospace and Engineering Mechanics The University of Texas at Austin, Austin, TX, 78712, USA*

---

## Abstract

This paper presents a procedure to build  $C^k$ ,  $k$  arbitrarily large, generalized finite element (FE) shape functions defined on non-structured finite element meshes. The functions have the same support as corresponding global FE Lagrangian shape functions. Meshes with both convex and non-convex clouds (set of elements sharing a vertex node), can be used. The so-called R-functions are used to build  $C^k$  FE-based partition of unity functions with non-convex support. A technique to combine  $C^0$  Lagrangian FE shape functions with the proposed  $C^k$  partition of unity is presented. The technique allows the use of  $C^k$  generalized FE shape functions in parts of the computational domain where their high smoothness is required, as in the case of problems with distributional boundary conditions, and the less computationally demanding  $C^0$  generalized FE shape functions elsewhere in the domain. A linear elasticity problem with a concentrated moment is solved using the proposed  $C^k$  generalized FE method. Higher order distributional boundary conditions can also be handled by the method. A detailed convergence analysis is presented for this class of problems as well as for problems in energy space. The integrability of the functions using standard Gauss-Legendre rules is also investigated.

*Key words:* Generalized finite element method, Meshfree methods, Partition of unity method,  $H_p$ -cloud method, Distributional data, Coupling.

---

\* Corresponding author. Tel.: +1-217-244-2830; Fax: +1-217-333-3821.  
*Email address:* caduarte@uiuc.edu (C. A. Duarte).

# 1 INTRODUCTION

Meshfree methods such as the *hp*-cloud method [21, 22], the method of finite spheres [13], the particle-partition of unity method [28, 30, 31], reproducing kernel particle methods [46–48], the element-free Galerkin method [5–8, 36, 60], the finite point method [57–59], the generalized finite difference method [43, 44, 76, 77], the diffuse element method [54], the modified local Petrov-Galerkin method [1], smooth particle hydrodynamics [27, 73, 74], among others, offer an attractive alternative for the solution of many classes of problems that are difficult or even not feasible to solve using the finite element method. Excellent overviews of meshfree methods and their applications can be found in, for example, [2, 5, 29, 32, 42, 45, 47].

In meshfree methods, the approximation of field variables is constructed in terms of nodes without the aid of a mesh. The actual implementation of some meshfree methods, however, requires the partition of the domain through the use of a background grid for numerical integration. Nonetheless, due to the flexibility in constructing conforming shape functions to meet specific needs for different applications, it has been reported [10, 11, 18, 19, 21, 22, 52] that meshfree methods are particularly suitable for *hp*-adaptivity, simulation of crack propagation, and large deformation problems, among others.

One of the major difficulties encountered in the finite element analysis of tires, elastomeric bearings, vibration isolators and a variety of other products made of rubbery materials, is the excessive element distortion. Distortion of elements is inherent to Lagrangian formulations used to analyze this class of problems. The use of smooth shape functions provided by meshfree methods appears to be particularly effective in dealing with large deformation problems [9–11].

Finite elements based on shear-deformable shell models were dominant over the last decades mostly due to the lower  $C^0$ -continuity requirement of these models compared to a Kirchhoff-Love model which would be often sufficient from the mechanical point of view [62]. Meshfree approximations can be built with arbitrary degree of smoothness and have been successfully used to solve Kirchhoff-Love plate and shell models [37, 38, 51, 66].

High regularity in data approximation can be usefully applied to the ocean dynamic problem. Lekien et al. [40, 41] suggested a flow structure called Lagrangian coherent structure. The use of this structure makes it possible to predict a number of phenomena such as the motion of plankton populations and the evolution of oil spills. The quality of this structure is determined by the smoothness of the velocity field of each fluid particle moving along with the flow. Linear interpolation of the velocity field is not sufficient to create a smooth Lagrangian structure and a smoother interpolation method is essen-

tially required.

Problems with distributional boundary conditions like concentrated forces and moments in linear elasticity can be dealt with in a quite straightforward and mathematically consistent manner if the approximation is sufficiently smooth [3].

From the above, it is clear that there are many circumstances in which the high regularity provided by meshfree approximations can be very effective. One of the main drawbacks of meshfree methods, however, has been the fact that the computational cost is too high in some applications due to the fact that one has to use a great number of integration points in order to integrate the meshfree functions and their products over a computational domain. Further discussion on this issue is presented in Section 7. Another difficulty of meshfree approximations is their implementation into existing finite element data structures. The construction of meshfree shape functions usually requires search for neighboring nodes, special care with the numerical quadrature and imposition of Dirichlet boundary condition, and other requirements that are usually not met by traditional finite element codes.

One approach to reduce the computational cost of meshfree methods is to use these methods only in parts of the domain where they are strictly needed and use a finite element discretization elsewhere [8, 20, 34, 35, 49, 79]. Besides reducing the overall computational cost, this approach, when feasible, has several other appealing features. It facilitates, for example, the implementation of Dirichlet boundary conditions [26, 36] and the coupling with classical structural finite elements, like rods, shells, rigid bars, etc. This topic is further discussed in Section 5.

Another approach to ameliorate the cost of numerical integration of meshfree shape functions, and at the same time to reduce the implementation difficulties of meshfree methods, was proposed by H. C. Edwards [24, 25]. In this approach, a finite element mesh is used to build arbitrarily smooth shape functions that have the same support of corresponding global finite element Lagrangian shape functions defined on the same mesh. As a consequence, the numerical integration of these functions and their products can be efficiently done using the finite element mesh. In this case, the method is no longer meshfree but several of the attractive features of meshfree approximations, like high regularity of the approximation, and the partition of unity property, are still retained. Edwards [24] has shown that this approach can be used in any dimension and for any kind of finite element (triangular, quadrilateral, tetrahedral, hexahedral, etc) while rendering  $C^\infty$  shape functions with the same support as in the finite element method. In Section 7, we discuss and present numerical evidence of the effectiveness of this approach in controlling the cost of numerical integration in a Galerkin method.

Edwards' approach has an important practical limitation—It requires convex support for the functions. However, it is not possible, in general, to build finite element meshes such that the supports of *all* corresponding finite element shape functions are convex. In this paper, Edwards' approach is extended such that it can be applied to any type of finite element mesh with convex and non-convex supports. This is achieved with the aid of the so-called R-functions [63, 64].

The proposed technique is an instance of the partition of unity method like the *hp*-cloud [21, 22] and generalized finite element methods [17, 18, 67, 69]. As such, the proposed method has a well defined mathematical foundation with known a-priori error estimates for the case of mesh refinement (*h*-version), enrichment of the approximation spaces (*p*-version) or their combinations (*hp*-version). A detailed *h*-convergence analysis of the proposed method is presented in Section 6. Problems with smooth as well as singular solutions are solved and the results compared with theoretical estimates. In particular, the problem of a concentrated moment in a linear elasticity setting is solved. The Galerkin method is used in all computations.

The basic idea of the method is to use a finite element mesh to build a  $C^k$  partition of unity (PoU) that has the same support as the corresponding standard  $C^0$  global finite element shape functions. We call this PoU a  $C^k$  *finite element-based partition of unity*. In fact, the proposed PoU functions are  $C^\infty$  functions except at re-entrant corners/edges of their support in two/three dimensions where they are  $k$ -times continuously differentiable, with  $k$  arbitrarily large. This PoU is then used to build shape functions that can reproduce polynomials of arbitrary degree. The technique is quite similar to the generalized finite element method, except that the shape functions are arbitrarily smooth. We call the resulting technique a  $C^k$  *generalized finite element method*.

Among the proposed FE shape functions with high regularity we can mention: (i) the finite element interpolations for *rectangular* elements in any dimension and with arbitrary smoothness proposed by Surana et al. [72] and (ii) the tricubic  $C^1$  hexahedral finite element of Lekien and Marsden [40]. The so-called Reproducing Kernel Element Method recently proposed by Liu et al. [33, 50, 51, 66] provides interpolating shape functions of arbitrary smoothness, but their supports do not coincide with the finite element mesh.

The outline of the paper is as follows. Partition of unity approximations are briefly reviewed in Section 2. The  $C^\infty$  finite element-based partition of unity for convex clouds proposed by Edwards [24] is discussed in Section 3. A  $C^k$  finite element-based partition of unity for non-convex clouds is proposed in Section 4. An algorithm to seamlessly combine the  $C^k$  generalized finite element method with  $C^0$  generalized or standard finite element methods is described in Section 5. A detailed convergence analysis is presented in Section 6. The

problem of a linear elastic body subjected to a concentrated moment is solved in Section 6. The issue of numerical integration of the proposed finite element-based shape functions is discussed in Section 7. Finally, the conclusions are drawn in Section 8.

## 2 PARTITION OF UNITY APPROXIMATIONS

The  $C^\infty$  finite element (FE) based shape functions proposed by Edwards [24] and the  $C^k$  generalized FE approximations presented in this paper are examples of the so-called partition of unity approximations. In this section, the construction and an a-priori error estimate for this type of approximation are briefly reviewed. Additional examples of methods based on the partition of unity framework are  $hp$ -cloud [21, 22] and generalized finite element methods [17, 18, 67, 69].

Hereafter,  $\mathcal{T}_N = \{\omega_\alpha\}_{\alpha=1}^N$  denotes an open covering of a domain  $\Omega \subset \mathbb{R}^n$ ,  $n = 1, 2, 3$ , i.e.,

$$\bar{\Omega} \subset \bigcup_{\alpha=1}^N \omega_\alpha$$

We call the open sets  $\omega_\alpha$ ,  $\alpha = 1, \dots, N$ , *clouds* and associate with each one of them a node, denoted by  $\mathbf{x}_\alpha$ . In the  $hp$  cloud and in the method of finite spheres  $\omega_\alpha$  is either a circle or rectangle in two dimensions, and a sphere or cube in three dimensions. In the generalized finite element method and in the methods discussed in this paper, a cloud  $\omega_\alpha$  is the union of all finite elements sharing the vertex node  $\mathbf{x}_\alpha$  of a finite element mesh.

A set of functions  $\mathcal{S}_N = \{\varphi_\alpha\}_{\alpha=1}^N$  with the properties

$$\begin{aligned} \varphi_\alpha &\in C_0^k(\omega_\alpha), \quad k \geq 0, \quad 1 \leq \alpha \leq N \\ \sum_{\alpha} \varphi_\alpha(\mathbf{x}) &= 1 \quad \forall \mathbf{x} \in \Omega \end{aligned}$$

is called a *partition of unity (PoU) subordinate to the open covering  $\mathcal{T}_N$* . The first property indicates that the function  $\varphi_\alpha$  is non-zero only over the cloud  $\omega_\alpha$ , i.e., it has a *support* given by  $\omega_\alpha$  and that it is at least  $k$  times continuously differentiable.

Let  $\chi_\alpha^p(\omega_\alpha) = \text{span}\{L_{i\alpha}\}_{i \in \mathcal{I}(\alpha)}$  denote local spaces defined on  $\omega_\alpha$ ,  $\alpha = 1, \dots, N$ , where  $\mathcal{I}(\alpha)$ ,  $\alpha = 1, \dots, N$ , are index sets and  $\{L_{i\alpha}\}_{i \in \mathcal{I}(\alpha)}$  is a set of basis

functions satisfying

$$\mathcal{P}_p(\omega_\alpha) \subset \chi_\alpha^p(\omega_\alpha)$$

where  $\mathcal{P}_p$  denotes the space of polynomials of degree less or equal to  $p$ . Functions  $L_{i\alpha}$  are also denoted by enrichment or local approximation functions.

Examples of enrichment functions at a node  $\mathbf{x}_\alpha = (x_\alpha, y_\alpha)$  in two-dimensions are

$$\begin{aligned} & \{1, (x - x_\alpha), (y - y_\alpha)\} \quad \text{linear basis} \\ & \{1, (x - x_\alpha), (y - y_\alpha), (x - x_\alpha)^2, (x - x_\alpha)(y - y_\alpha), (y - y_\alpha)^2\} \quad \text{quadratic basis} \end{aligned} \tag{1}$$

Using the above two ingredients—a partition of unity and enrichment functions, the partition of unity shape functions associated with a node  $\mathbf{x}_\alpha$  are defined by

$$\phi_{\alpha i} := \varphi_\alpha L_{i\alpha}, \quad i \in \mathcal{I}(\alpha) \quad (\text{no sum on } \alpha) \tag{2}$$

Different choices for the partition of unity functions are possible. Each one of them will lead to a different class of shape functions. The  $C^k$  finite element-based partition of unity proposed in Section 4 has features of both the standard finite element PoU and Shepard PoU (cf. Sections 2.1 and 2.2).

The following theorem provides an a-priori error estimate for the  $h$ -version of partition of unity approximations [23, 52]. This estimate is used in Section 6. Suppose the following conditions hold:

- *Overlapping Condition:* There exists a  $M \in \mathbb{N}$  such that  $\forall \mathbf{x} \in \bar{\Omega} \text{ card}\{\alpha : \mathbf{x} \in \omega_\alpha\} \leq M$ .
- *PoU:*  $\mathcal{S}_N = \{\varphi_\alpha\}_{\alpha=1}^N$  satisfies:
 
$$\|\varphi_\alpha\|_{L^\infty(\mathbb{R}^n)} \leq C_\infty$$

$$\|\nabla \varphi_\alpha\|_{L^\infty(\mathbb{R}^n)} \leq \frac{C_G}{h_\alpha}$$
 where  $C_\infty, C_G > 0$  are two constants and  $h_\alpha = \text{diam } \omega_\alpha$ .

Let

$$\mathbf{X}^{hp}(\Omega) = \text{span}\{\phi_{\alpha i}\}, \quad \alpha = 1, \dots, N, \quad i \in \mathcal{I}(\alpha)$$

where  $\phi_{\alpha i}$  are partition of unity shape functions defined in (2).

Using the above hypothesis, and assuming that the local spaces  $\chi_\alpha^p(\omega_\alpha)$ ,  $\alpha = 1, \dots, N$ , can locally approximate a function  $u \in H^k(\Omega)$ ,  $k > 1$ , (see, for example, [23, 52] for details and proofs), it can be shown that the following a-priori error estimate holds:

**Theorem 1** *Let  $u \in H^k(\Omega)$ ,  $k > 1$ , then for fixed  $h$  and  $p$  there is  $u_{hp} \in \mathbf{X}^{hp}$  such that*

$$\|u - u_{hp}\|_{E(\Omega)} \leq Ch^{\min\{p, k-1\}} \|u\|_{H^k(\Omega)}, \quad (3)$$

where

$$h = \max_{\alpha=1, \dots, N(h)} h_\alpha$$

with  $h_\alpha = \text{diam}\omega_\alpha$  and  $C$  is a constant independent of  $u$  and  $h$ . Here,  $\|\cdot\|_{E(\Omega)}$  denotes the energy norm.

## 2.1 Shepard Partition of Unity

Shepard [39, 65] has proposed a very simple approach to build partition of unity functions which is used in several meshfree methods like  $hp$  clouds [21, 22], the method of finite spheres [13], the particle-partition of unity method [28, 30, 31], among others. The finite element-based partitions of unity presented in Sections 3, 4 and 5 also use Shepard's approach. This technique is briefly reviewed in this section.

Let  $\mathcal{W}_\alpha : \mathbb{R}^n \rightarrow \mathbb{R}$  denote a *weighting function* with compact support  $\omega_\alpha$  that belongs to the space  $C_0^k(\omega_\alpha)$ ,  $k \geq 0$ . Suppose that such a weighting function is built at every cloud  $\omega_\alpha$ ,  $\alpha = 1, \dots, N$ , of an open covering  $\mathcal{T}_N$  of a domain  $\Omega$  in  $\mathbb{R}^n$ .

The Shepard partition of unity subordinate to the covering  $\mathcal{T}_N$  is defined as [39, 65]

$$\varphi_\alpha(\mathbf{x}) = \frac{\mathcal{W}_\alpha(\mathbf{x})}{\sum_{\beta(\mathbf{x})} \mathcal{W}_\beta(\mathbf{x})} \quad \beta(\mathbf{x}) \in \{\gamma \mid \mathcal{W}_\gamma(\mathbf{x}) \neq 0\} \quad \alpha = 1, \dots, N \quad (4)$$

The regularity of this partition of unity depends only on the regularity of the weighting functions. Therefore, Shepard partition of unity functions with arbitrary regularity can easily be built.

## 2.2 $C^0$ Finite Element Partition of Unity

Lagrangian finite element shape functions constitute a partition of unity. In this case, a cloud  $\omega_\alpha$  is simply the union of the finite elements sharing a vertex node  $\mathbf{x}_\alpha$  in the mesh. Each node is associated with its own cloud comprised by the elements surrounding that node. Figure 1 shows examples of such clouds. The cloud  $\omega_1$  of node  $\mathbf{x}_1$  includes elements  $c, d, i, h$  and  $g$  and is an example of a *convex* cloud. The cloud  $\omega_2$  of node  $\mathbf{x}_2$  comprises elements  $a, b, e, d$  and  $c$  and is an example of a *non-convex* cloud.

The partition of unity function  $\varphi_\alpha$  is equal to the usual global finite element shape function associated with node  $\mathbf{x}_\alpha$ . Finite element shape functions are inexpensive to compute and to numerically integrate since they are (mapped) polynomial functions while Shepard functions are, in general, rational polynomials. However, they are, for most practical matters, limited to  $C^0$  regularity in two or higher dimensional spaces. The construction of the so-called  $C^\infty$  finite element-based partition of unity for convex clouds is described in Section 3. Section 4 deals with the case of non-convex clouds.

## 3 A $C^\infty$ FINITE ELEMENT-BASED PARTITION OF UNITY FOR CONVEX CLOUDS

A technique to build  $C^\infty$  partition of unity functions over *convex* finite element clouds was proposed by Edwards [24]. This PoU is then multiplied by enrichment functions as described in Section 2 to create generalized finite element  $C^\infty$  shape functions. These functions can be defined for any standard finite element and in any dimension. One important practical limitation of the technique, however, is that it is limited to convex finite element clouds. As discussed in the previous section, a finite element cloud is not necessarily convex. In this section, we briefly review Edwards' approach. In Section 4, we present an extension of Edwards' approach that can handle the case of non-convex finite element clouds while rendering partition of unity functions with arbitrary smoothness.

### 3.1 $C^\infty$ Finite Element-Based Weighting Function for Convex Clouds

In this section, a technique to build  $C^\infty$  weighting functions over convex finite element clouds is discussed [24]. These weighting functions are said to be finite element-based since they have the same support (cloud) as the classical global finite element shape functions. Therefore, the intersections of the

supports of these weighting functions coincide with the finite elements. As a consequence, the numerical integration of these functions and their products can be efficiently done using the finite element mesh (cf. Section 7).

A  $C^\infty$  finite element-based weighting function with a convex support can be built from the product of the so-called *cloud boundary functions*. Let us consider first the case of a cloud  $\omega_\alpha$  associated with a node not at the boundary of the domain. This cloud is denoted by *interior cloud*. One example is shown in Figure 2. The boundary of a cloud in two dimensions is the polygonal built from the edges of the elements in the cloud that are not connected to its node. This is indicated as side  $j$ ,  $j = 1, \dots, 7$ , in the example of Figure 2.

Associated with each side  $j$  at the boundary of a cloud  $\omega_\alpha$ , there is a parametric coordinate  $\xi_j$  measured in the direction perpendicular to the edge and set to zero at the edge (cf. Figure 2) given by

$$\xi_j = \mathbf{n}_{\alpha,j} \cdot (\mathbf{x} - \mathbf{b}_{\alpha,j})$$

where  $\mathbf{b}_{\alpha,j}$  is a boundary point which is chosen to be the midpoint of each edge, and  $\mathbf{n}_{\alpha,j}$  is a normal vector which is directed towards the inside of the cloud  $\omega_\alpha$ . The definition above can also be used in the three-dimensional case.

A function that vanishes smoothly as the edge is approached and that is greater than zero for points in the cloud  $\omega_\alpha$  is called a *cloud boundary function*. It is defined for the  $j^{\text{th}}$  side of cloud  $\omega_\alpha$  as

$$\widehat{\mathcal{E}}_{\alpha,j}[\xi_j(\mathbf{x})] = \mathcal{E}_{\alpha,j}(\mathbf{x}) := \begin{cases} \frac{1}{e^{(\frac{1}{\xi_j^\gamma})}} = e^{-\xi_j^{-\gamma}}, & 0 < \xi_j \\ 0 & , \text{ otherwise} \end{cases} \quad (5)$$

where  $\gamma$  is a positive constant. The value  $\gamma = 0.6$  is used in the examples presented in Section 6. Cloud boundary functions can also be defined using, e.g., spline functions [14] instead of exponential functions as in (5).

The cloud boundary functions defined above and all of their derivatives are zero on the corresponding edge and on the “negative” side of the edge. They are  $C^\infty$  functions. Figure 3 illustrates a cloud boundary function in two dimensions.

The length of the normal vector  $\mathbf{n}_{\alpha,j}$  is chosen such that all cloud boundary functions have the same value at the cloud node  $\mathbf{x}_\alpha$ . Hence,

$$\mathcal{E}_{\alpha,j}(\mathbf{x}_\alpha) = \mathcal{E}_{\alpha,k}(\mathbf{x}_\alpha), \quad \forall_{j,k}$$

This implies that

$$\mathbf{n}_{\alpha,j} \cdot (\mathbf{x}_\alpha - \mathbf{b}_{\alpha,j}) = \mathbf{n}_{\alpha,k} \cdot (\mathbf{x}_\alpha - \mathbf{b}_{\alpha,k}), \quad \forall_{j,k}$$

This can be accomplished by selecting a parameter  $\beta$  which controls the decay ratio of all cloud boundary functions and is defined by [24]

$$\beta = \frac{\mathcal{E}_{\alpha,j} \left( \frac{1}{2}(\mathbf{x}_\alpha + \mathbf{b}_{\alpha,j}) \right)}{\mathcal{E}_{\alpha,j}(\mathbf{x}_\alpha)}$$

From the above, we have that

$$\begin{aligned} \mathcal{E}_{\alpha,j}(\mathbf{x}_\alpha) &= \frac{1}{\beta} \mathcal{E}_{\alpha,j} \left( \frac{1}{2}(\mathbf{x}_\alpha + \mathbf{b}_{\alpha,j}) \right) \\ (\mathbf{n}_{\alpha,j} \cdot (\mathbf{x}_\alpha - \mathbf{b}_{\alpha,j}))^{-\gamma} &= \left( \mathbf{n}_{\alpha,j} \cdot \left( \frac{1}{2}(\mathbf{x}_\alpha - \mathbf{b}_{\alpha,j}) \right) \right)^{-\gamma} + \log_e \beta \\ (\mathbf{n}_{\alpha,j} \cdot (\mathbf{x}_\alpha - \mathbf{b}_{\alpha,j}))^{-\gamma} &= \left( \frac{1 - 2^\gamma}{\log_e \beta} \right)^{-1} \\ \xi_j(\mathbf{x}_\alpha) = \mathbf{n}_{\alpha,j} \cdot (\mathbf{x}_\alpha - \mathbf{b}_{\alpha,j}) &= \left( \frac{1 - 2^\gamma}{\log_e \beta} \right)^{\frac{1}{\gamma}} = \text{const.} \end{aligned} \quad (6)$$

From (6), we have that the length of the normal vector  $\mathbf{n}_{\alpha,j}$  can be chosen such that all cloud boundary functions have the same value at the corresponding cloud node  $\mathbf{x}_\alpha$ . The normal vector  $\mathbf{n}_{\alpha,j}$  is then set as

$$\mathbf{n}_{\alpha,j} = \frac{\left( \frac{1-2^\gamma}{\log_e \beta} \right)^{\frac{1}{\gamma}}}{|\hat{\mathbf{n}}_{\alpha,j} \cdot (\mathbf{x}_\alpha - \mathbf{b}_{\alpha,j})|} \hat{\mathbf{n}}_{\alpha,j}$$

where  $\hat{\mathbf{n}}_{\alpha,j}$  is the unit vector normal to the cloud edge  $\alpha, j$ . In all computations presented in Section 6,  $\gamma = 0.6$  and  $\beta = 0.3$  is adopted.

The parametric coordinate  $\xi_j$  is then given by

$$\xi_j(\mathbf{x}) = \mathbf{n}_{\alpha,j} \cdot (\mathbf{x} - \mathbf{b}_{\alpha,j}) = \frac{\left( \frac{1-2^\gamma}{\log_e \beta} \right)^{\frac{1}{\gamma}}}{|\hat{\mathbf{n}}_{\alpha,j} \cdot (\mathbf{x}_\alpha - \mathbf{b}_{\alpha,j})|} \hat{\mathbf{n}}_{\alpha,j} \cdot (\mathbf{x} - \mathbf{b}_{\alpha,j})$$

The cloud boundary function defined in (5) can be used to build a  $C^\infty$  weighting function that is zero at the boundary of the cloud  $\omega_\alpha$  and greater than zero inside the cloud as follows

$$\begin{aligned} \mathcal{W}_\alpha(\mathbf{x}) &:= e^{c_\alpha} \prod_{j=1}^{M_\alpha} \mathcal{E}_{\alpha,j}(\mathbf{x}) \\ &= \begin{cases} e^{\left(c_\alpha - \sum_{j=1}^{M_\alpha} \xi_j^{-\gamma}\right)} = e^{\left(c_\alpha - \sum_{j=1}^{M_\alpha} (\mathbf{n}_{\alpha,j} \cdot (\mathbf{x} - \mathbf{b}_{\alpha,j}))^{-\gamma}\right)} & , \mathbf{x} \in \omega_\alpha \\ 0 & , \mathbf{x} \notin \omega_\alpha \end{cases} \quad (7) \end{aligned}$$

where  $M_\alpha$  is the number of cloud boundary functions for the cloud  $\omega_\alpha$  and  $c_\alpha$  is a scaling parameter selected for each node such that  $\mathcal{W}_\alpha(\mathbf{x}_\alpha) = 1$ . It is given by

$$c_\alpha = \sum_{j=1}^{M_\alpha} (\mathbf{n}_{\alpha,j} \cdot (\mathbf{x}_\alpha - \mathbf{b}_{\alpha,j}))^{-\gamma} = M_\alpha \left( \frac{1 - 2^\gamma}{\log_e \beta} \right)^{-1}$$

This scaling is very important since it prevents the value of the weighting function from varying significantly from node to node in one element. This would be the case if the number of edges varies significantly among the clouds. Clouds with a large number of edges would have a much smaller weighting function than those with few edges. Our numerical experience shows that this may lead to detrimental effects on the performance of the resulting approximation.

The finite element-based  $C^\infty$  weighting function defined in (7) can be used in two- or three-dimensions. It is not difficult to show that this is a  $C^\infty$  function.

Further consideration of the weighting functions defined above shows that it is applicable only to convex clouds. If the extension of any edge intersects the cloud, i.e., if the cloud  $\omega_\alpha$  is non-convex, the weighting function will also vanish in the interior of the cloud which is not desirable. This issue is dealt with in Section 4.

The procedure for clouds with nodes located at the boundary of the domain is basically the same as above. The cloud  $\omega_\alpha$  is still the union of the elements sharing the node. However, the node will not be completely surrounded by elements. Consider, for example, cloud  $\omega_\alpha$  shown in Figure 4. The weighting function for cloud  $\omega_\alpha$  is given by

$$\mathcal{W}_\alpha(\mathbf{x}) = e^{c_\alpha} \prod_{j=1}^3 \mathcal{E}_{\alpha,j}(\mathbf{x})$$

Therefore, we use cloud boundary functions only for the edges inside of the domain.

Having the finite element-based weighting functions defined above, we can

now use Shepard's formula (4) to build a partition of unity subordinate to the finite element clouds:

$$\varphi_\alpha(\mathbf{x}) = \frac{\mathcal{W}_\alpha(\mathbf{x})}{\sum_{\beta(\mathbf{x})} \mathcal{W}_\beta(\mathbf{x})} \quad \beta(\mathbf{x}) \in \{\gamma \mid \mathcal{W}_\gamma(\mathbf{x}) \neq 0\} \quad \alpha = 1, \dots, N \quad (8)$$

where  $\mathcal{W}_\alpha(\mathbf{x})$ ,  $\alpha = 1, \dots, N$ , are the  $C^\infty$  finite element-based weighting functions defined in (7).

The resulting partition of unity functions have the same regularity as the weighting functions. In addition, the numerical integration of these functions and their products can be efficiently done using the underlying finite element mesh since their support coincide with the finite elements. This is further investigated in Section 7.

**Lemma 3.1** *The partition of unity functions defined in (8) enjoys the Kronecker-delta property, i.e.,*

$$\varphi_\alpha(\mathbf{x}_\beta) = \delta_{\alpha\beta}, \quad \alpha, \beta = 1, \dots, N$$

**Proof:** From the definition of the  $C^\infty$  finite element-based weighting functions we have that

$$\mathcal{W}_\alpha(\mathbf{x}_\beta) = \delta_{\alpha\beta}, \quad \alpha, \beta = 1, \dots, N$$

Therefore,

$$\varphi_\alpha(\mathbf{x}_\beta) = \frac{\mathcal{W}_\alpha(\mathbf{x}_\beta)}{\sum_{\gamma(\mathbf{x}_\beta)} \mathcal{W}_\gamma(\mathbf{x}_\beta)} = \frac{\delta_{\alpha\beta}}{\sum_{\gamma(\mathbf{x}_\beta)} \delta_{\gamma\beta}} = \frac{\delta_{\alpha\beta}}{\delta_{\beta\beta}} = \delta_{\alpha\beta}$$

■

The partition of unity defined above can be used to build PoU shape functions of any polynomial degree using the technique described in Section 2, i.e.,

$$\phi_{\alpha i} := \varphi_\alpha L_{i\alpha}, \quad i \in \mathcal{I}(\alpha) \quad (\text{no sum on } \alpha) \quad (9)$$

The resulting shape functions are also  $C^\infty$  functions if the enrichment functions,  $L_{i\alpha}$ , have this property. The high regularity of these shape functions can be advantageous to solve several class of problems as discussed in Section 1. Some applications are presented in Section 6.

The shape functions  $\phi_{\alpha i}$  defined in (9) are used in Section 6 to define trial and test functions for the Galerkin method. A trial function for a scalar value

problem is given by

$$u(\mathbf{x}) = \sum_{\alpha=1}^N \sum_{i \in \mathcal{I}(\alpha)} u_{\alpha i} \phi_{\alpha i}(\mathbf{x}) \quad (10)$$

**Theorem 2** *If the enrichment functions  $L_{i\alpha}$  are such that*

$$\begin{aligned} L_{0\alpha}(\mathbf{x}) &= 1 \\ L_{i\alpha}(\mathbf{x}_\alpha) &= 0 \quad i \neq 0 \end{aligned} \quad (11)$$

*then the trial function defined in (10) has the following property*

$$u(\mathbf{x}_\alpha) = u_{\alpha 0} \quad (12)$$

where  $u_0^\alpha$  is the coefficient of the partition of unity function at node  $\mathbf{x}_\alpha$ .

**Proof:**

$$\begin{aligned} u(\mathbf{x}_\beta) &= \sum_{\alpha=1}^N \sum_{i \in \mathcal{I}(\alpha)} u_{\alpha i} \phi_{\alpha i}(\mathbf{x}_\beta) = \sum_{\alpha=1}^N \sum_{i \in \mathcal{I}(\alpha)} u_{\alpha i} \varphi_\alpha(\mathbf{x}_\beta) L_{i\alpha}(\mathbf{x}_\beta) \\ &= \sum_{\alpha=1}^N \sum_{i \in \mathcal{I}(\alpha)} u_{\alpha i} \delta_{\alpha\beta} L_{i\alpha}(\mathbf{x}_\beta) = \sum_{i \in \mathcal{I}(\beta)} u_{\beta i} L_{i\beta}(\mathbf{x}_\beta) = u_{\beta 0} \cdot 1 \end{aligned}$$

■

The value of the trial function  $u$  at a node  $\mathbf{x}_\alpha$  is then given by the coefficient of the partition of unity function at node  $\mathbf{x}_\alpha$ . This property can be used to impose nodal boundary conditions in a Galerkin method. Examples of enrichment functions with property (11) are given in (1).

Figure 5 shows an example of a finite element-based weighting function and corresponding partition of unity function built on a convex cloud. The functions are  $C^\infty$  in this case. It can be observed that the weighting function is equal to one at the cloud node  $\mathbf{x}_\alpha$  due to the scaling of the weighting functions and zero at all other nodes.

#### 4 A $C^k$ FINITE ELEMENT-BASED POU FOR NON-CONVEX CLOUDS

Consider the finite element cloud  $\omega_\alpha$  depicted in Figure 6 and comprising four elements. The re-entrant corner at the intersection of sides 5 and 6 makes the

cloud non-convex and the procedure outlined in the previous section can not be used because the cloud boundary functions  $\mathcal{E}_{\alpha,5}(\mathbf{x})$  and  $\mathcal{E}_{\alpha,6}(\mathbf{x})$  vanish at points inside the cloud. Consequently, the weighting function  $\mathcal{W}_\alpha(\mathbf{x})$  for cloud  $\omega_\alpha$  will also be zero in the interior of the cloud if they are built using (7). We seek modifications to the approach of Section 3.1 that can handle non-convex finite element clouds.

The proposed procedures are similar to the original one—a cloud boundary function  $\mathcal{E}_{\alpha,j}(\mathbf{x})$  is constructed for each side of a finite element cloud  $\omega_\alpha$  and the weighting function  $\mathcal{W}_\alpha(\mathbf{x})$  for the cloud is again defined using these functions. The non-convexity of the boundary is handled using the concept of the so-called R-functions [63, 64].

Consider again the example of Figure 6. Cloud boundary functions  $\mathcal{E}_{\alpha,j}(\mathbf{x})$ ,  $j = 1, \dots, 6$ , are built using (5). Next, we combine functions  $\mathcal{E}_{\alpha,5}(\mathbf{x})$  and  $\mathcal{E}_{\alpha,6}(\mathbf{x})$  associated with the re-entrant corner into a single cloud boundary function using the notion of R-functions [63, 64]. An R-function is a real-valued function whose sign is completely determined by the signs of its arguments. As an example, the R-function  $f(x, y, z) = xyz$  can be negative only if the number of its negative arguments is odd. Such functions “encode” Boolean logic functions and are called R-functions.

Consider now the R-function “or” with two arguments,  $f_1$  and  $f_2$ , and denoted by  $(f_1 \vee_0^k f_2)$

$$(f_1 \vee_0^k f_2) := \left( f_1 + f_2 + \sqrt{f_1^2 + f_2^2} \right) (f_1^2 + f_2^2)^{\frac{k}{2}} \quad (13)$$

where  $k$  is a positive integer. This function is analytic everywhere except at the origin ( $f_1 = f_2 = 0$ ), where it is at least  $k$  times differentiable, i.e., it belongs to  $C^k(\Omega)$  [64].

If  $f_1 \geq 0$  and  $f_2 \geq 0$  define two regions in  $\mathbb{R}^n$ , then

- $(f_1 \vee_0^k f_2) \geq 0$  and,
- $(f_1 \vee_0^k f_2) > 0$  if  $f_1 > 0$  or  $f_2 > 0$ .

Note that R-functions can be defined in any dimension and the arguments,  $f_1$  and  $f_2$ , can also define regions with curved boundaries.

Let us suppose that sides  $m$  and  $n$  are identified as non-convex sides for a finite element cloud  $\omega_\alpha$  (e.g., sides 5 and 6 for the cloud of Figure 6). A new cloud boundary function combining  $\mathcal{E}_{\alpha,m}$  and  $\mathcal{E}_{\alpha,n}$  is then defined as

$$\mathcal{E}_{\alpha,mn}^{nc}(\mathbf{x}) := \frac{\mathcal{E}_{\alpha,m}(\mathbf{x}) \vee_0^k \mathcal{E}_{\alpha,n}(\mathbf{x})}{\mathcal{E}_{\alpha,m}(\mathbf{x}_\alpha) \vee_0^k \mathcal{E}_{\alpha,n}(\mathbf{x}_\alpha)} \quad (14)$$

where the parameter  $k$  is chosen according to the degree of smoothness desired. In the numerical experiments of Section 6,  $k = 2$  is used. The combined cloud boundary function is scaled by its value at the cloud node  $\mathbf{x}_\alpha$ . This is done, as in the previous section, in order to get a weighting function that has a unit value at the cloud node  $\mathbf{x}_\alpha$ . This combined cloud boundary function together with other boundary functions for the cloud  $\omega_\alpha$  are then used to build a  $C^k$  finite element-based weighting function  $\mathcal{W}_\alpha(\mathbf{x})$  using an expression similar to (7). For the example of Figure 6, the weighting function is given by

$$\mathcal{W}_\alpha(\mathbf{x}) := \left( e^{c_\alpha} \prod_{j=1}^4 \mathcal{E}_{\alpha,j}(\mathbf{x}) \right) \mathcal{E}_{\alpha,56}^{nc}(\mathbf{x}) \quad (15)$$

where

$$c_\alpha = 4 \left( \frac{1 - 2^\gamma}{\log_e \beta} \right)^{-1}$$

The procedure to build cloud boundary functions like,  $\mathcal{E}_{\alpha,mn}^{nc}(\mathbf{x})$ , must be used at all re-entrant corners of a finite element cloud.

It is also possible to define  $C^k$  finite element weighting functions using only R-functions. Consider for example the R-function “and” with two arguments,  $f_1$  and  $f_2$ , and denoted by  $(f_1 \wedge_0^k f_2)$  [64]

$$(f_1 \wedge_0^k f_2) := \left( f_1 + f_2 - \sqrt{f_1^2 + f_2^2} \right) \left( f_1^2 + f_2^2 \right)^{\frac{k}{2}} \quad (16)$$

where  $k$  is a positive integer. Like the R-function “or”, this function is analytic everywhere except at the origin ( $f_1 = f_2 = 0$ ), where it is at least  $k$  times differentiable, i.e., it belongs to  $C^k(\Omega)$  [64].

If  $f_1 \geq 0$  and  $f_2 \geq 0$  define two regions in  $\mathbb{R}^n$ , then

- $(f_1 \wedge_0^k f_2) \geq 0$  and,
- $(f_1 \wedge_0^k f_2) > 0$  if  $f_1 > 0$  and  $f_2 > 0$ .

The weighting function for the cloud  $\omega_\alpha$  of the example of Figure 6 can be built as

$$\mathcal{W}_\alpha(\mathbf{x}) := \frac{((((\mathcal{E}_{\alpha,6}(\mathbf{x}) \vee_0^k \mathcal{E}_{\alpha,5}(\mathbf{x})) \wedge_0^k \mathcal{E}_{\alpha,4}(\mathbf{x})) \wedge_0^k \mathcal{E}_{\alpha,3}(\mathbf{x})) \wedge_0^k \mathcal{E}_{\alpha,2}(\mathbf{x})) \wedge_0^k \mathcal{E}_{\alpha,1}(\mathbf{x}))}{((((\mathcal{E}_{\alpha,6}(\mathbf{x}_\alpha) \vee_0^k \mathcal{E}_{\alpha,5}(\mathbf{x}_\alpha)) \wedge_0^k \mathcal{E}_{\alpha,4}(\mathbf{x}_\alpha)) \wedge_0^k \mathcal{E}_{\alpha,3}(\mathbf{x}_\alpha)) \wedge_0^k \mathcal{E}_{\alpha,2}(\mathbf{x}_\alpha)) \wedge_0^k \mathcal{E}_{\alpha,1}(\mathbf{x}_\alpha))}$$

Shepard’s formula (4) is again used to build a partition of unity using the  $C^k$  finite element-based weighting functions defined above. Partition of unity

shape functions of arbitrary polynomial degree is then built using (2). The resulting shape functions are at least  $k$ -times continuously differentiable. In fact, they are  $C^\infty$  functions except at the re-entrant corners of the clouds where they are  $k$ -times continuously differentiable, with  $k$  arbitrarily large, if the weighting functions are built using the procedure illustrated in (15).

Figure 7 shows an example of a finite element-based weighting function and corresponding partition of unity function built on a non-convex cloud. The functions are  $C^\infty$  everywhere except at the re-entrant corner shown in the figure, where they are  $C^k$ . Like in the case of the  $C^\infty$  weighting function shown in Figure 5, the functions of Figure 7 also have the Kronecker-delta property. However, the maximum value of the partition of unity function is not attained at the cloud node  $\mathbf{x}_\alpha$ .

## 5 CONSTRUCTION OF A COMBINED $C^0$ - $C^k$ FE-BASED PARTITION OF UNITY

The proposed  $C^k$  finite element-based approximation can be used much more efficiently in a Galerkin method than comparable meshfree formulations defined on, for example, circles or rectangles. Nonetheless, its computational cost is still superior to generalized finite element methods based on  $C^0$  Lagrangian partitions of unity as demonstrated in the numerical studies presented in Section 7. Therefore, the proposed  $C^k$  finite element-based approximation should be used only when the high regularity provided by these functions is required or beneficial. In fact, there are situations in which a smooth approximation is necessary only in parts of the domain. Elsewhere, a  $C^0$  approximation suffices. An example is given in Section 6.4. In addition, the use of a standard  $C^0$  partition of unity in parts of the domain, when feasible, can be useful to, for example, impose Dirichlet boundary conditions [26, 36].

Several techniques to couple meshfree and finite element methods have been proposed. Belytschko et al. [8] proposed a coupling technique in which some finite element nodes are replaced by meshfree nodes and a ramp function is used to build the transition between the finite element and meshfree discretizations. Linear consistency is attained with this approach. A generalization of this idea was proposed by Hegen [34] based on the use of Lagrange multipliers.

Huerta and Fernandez-Mendez [35] proposed a technique to couple finite element and meshfree discretizations based on consistency or reproducibility conditions of the resulting shape functions and moving least squares techniques. The support size of the meshfree functions and their distribution must obey some rules in order to be admissible [35]. A related technique was proposed by Liu et al. [49, 79] with the goal of improving a finite element discretization

with meshfree shape functions.

Many of the meshfree shape functions like moving least squares and Shepard functions, constitute a partition of unity. In other words, these functions add to the unity at any point in the domain. Lagrangian finite element shape functions also possess this property. In this section, a technique to combine a  $C^0$  linear finite element-based partition of unity with a  $C^k$  finite element-based partition of unity is presented. The basic idea is to treat linear finite element shape functions as weighting functions and use Shepard's formula (4) to build a partition of unity. We call the resulting PoU a combined  $C^0$ - $C^k$  FE-based partition of unity. The technique, while simple, is quite flexible and generic. It can, in fact, be used to couple finite element approximations with any (meshfree or not) approximation that possess the partition of unity property. Nonetheless, in this section, we concentrate on the case of the partitions of unity discussed in previous sections. Other cases are discussed in [20, 61]. Details of the formulation are presented next.

### 5.1 $C^0$ - $C^k$ Finite Element-Based Weighting Functions

Let  $\Omega$  be an open domain in  $\mathbb{R}^n$ ,  $n = 1, 2, 3$ , covered by a finite element mesh consisting of any type of linear Lagrangian element. Let  $\mathbf{x}_\alpha$  denote a finite element vertex node in the mesh. Associated with each node  $\mathbf{x}_\alpha$  there is a linear finite element shape function  $N_\alpha(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$  with support  $\omega_\alpha = \{\mathbf{x} \in \Omega : N_\alpha(\mathbf{x}) \neq 0\}$ . We also refer to the function  $N_\alpha$  as a  $C^0$  *finite element-based weighting function*.

Suppose that some of the  $C^0$  finite element weighting functions are replaced by the  $C^k$  FE-based weighting functions,  $\mathcal{W}_\beta(\mathbf{x})$ , defined in Section 4. Let  $\mathbf{y}_\beta$ ,  $\beta = 1, \dots, M_{C^k}$ , denote the associated  $C^k$  finite element nodes with  $M_{C^k}$  being the dimension of this set and  $\mathcal{I}_{C^k}$  the index set of these nodes. Let  $\mathcal{I}_{C^0}$  denote the index set of all remaining finite element nodes and  $M_{C^0}$  the dimension of this set, i.e.,  $M_{C^0} = \text{card}\{\mathcal{I}_{C^0}\}$ . Associated with each  $C^k$  finite element node  $\mathbf{y}_\beta$  there is a  $C^k$  FE-based weighting function  $\mathcal{W}_\beta(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$  with the same support as the  $C^0$  FE-based weighting function it replaced and built as described in Section 4.

An example showing a mesh with  $C^0$  and  $C^k$  FE-based weighting functions is shown in Figure 8. Typically,  $C^k$  weighting functions are used when the high regularity of these functions is required or beneficial for a particular problem. An example is provided in Section 6.4.

**Lemma 5.1** *The supports  $\{\omega_\beta\}_{\beta \in \mathcal{I}_{C^k}}$  and  $\{\omega_\alpha\}_{\alpha \in \mathcal{I}_{C^0}}$ , of the  $C^k$  and  $C^0$  finite*

element-based weighting functions are such that

$$\mathcal{T}_{C^k, C^0} := \{\{\omega_\beta\}_{\beta \in \mathcal{I}_{C^k}} \cup \{\omega_\alpha\}_{\alpha \in \mathcal{I}_{C^0}}\}$$

constitutes an open covering for  $\Omega$ . i.e.,

$$\bar{\Omega} \subset \bar{\mathcal{T}}_{C^k, C^0}$$

**Proof:** Immediate from the definition of  $\mathcal{T}_{C^k, C^0}$  ■

## 5.2 $C^0$ - $C^k$ Finite Element-Based Partition of Unity

Shepard's formula (4) is once again used to build a partition of unity using the  $C^0$  and  $C^k$  FE-based weighting functions discussed above. For each  $C^0$  FE node  $\mathbf{x}_\alpha$ ,  $\alpha \in \mathcal{I}_{C^0}$ , and each  $C^k$  FE node node  $\mathbf{y}_\alpha$ ,  $\alpha \in \mathcal{I}_{C^k}$ , a partition of unity function  $\varphi_\alpha$  is defined by

$$\varphi_\alpha(\mathbf{x}) := \begin{cases} \frac{N_\alpha(\mathbf{x})}{\mathcal{S}_\alpha(\mathbf{x})} & \text{if } \alpha \in \mathcal{I}_{C^0} \\ \frac{\mathcal{W}_\alpha(\mathbf{x})}{\mathcal{S}_\alpha(\mathbf{x})} & \text{if } \alpha \in \mathcal{I}_{C^k} \end{cases} \quad (17)$$

where

$$\mathcal{S}_\alpha(\mathbf{x}) = \sum_{\beta \in \mathcal{I}_{C^0}(\mathbf{x})} N_\beta(\mathbf{x}) + \sum_{\gamma \in \mathcal{I}_{C^k}(\mathbf{x})} \mathcal{W}_\gamma(\mathbf{x})$$

$$\mathcal{I}_{C^0}(\mathbf{x}) = \{\beta \in \mathcal{I}_{C^0} : N_\beta(\mathbf{x}) \neq 0\}$$

$$\mathcal{I}_{C^k}(\mathbf{x}) = \{\gamma \in \mathcal{I}_{C^k} : \mathcal{W}_\gamma(\mathbf{x}) \neq 0\}$$

Let  $\mathcal{I}_{C^k, C^0} = \mathcal{I}_{C^k} \cup \mathcal{I}_{C^0}$  denote the index set of all nodes in the domain  $\Omega$ . Then, it is straightforward to show that the set

$$\{\varphi_\alpha\}_{\alpha \in \mathcal{I}_{C^k, C^0}}$$

constitutes a partition of unity subordinate to the open covering  $\mathcal{T}_{C^k, C^0}$ , i.e.,

$$\sum_{\alpha \in \mathcal{I}_{C^k, C^0}} \varphi_\alpha(\mathbf{x}) = 1 \quad \forall \mathbf{x} \in \Omega$$

The denominator in (17) is equal to the sum of  $C^k$  FE-based and  $C^0$  (linear) FE weighting functions that are not zero at  $\mathbf{x}$ . It scales the numerator such

that the resulting function,  $\varphi_\alpha$ , constitutes a partition of unity. Consider now element  $\tau_1$  or  $\tau_2$  indicated in Figure 8. All  $C^k$  FE-based weighting functions are zero inside those elements. Then, there is no need to use (17) to build the partition of unity over these elements since their finite element shape functions already constitutes a partition of unity. Therefore, elements like  $\tau_1$  or  $\tau_2$  are standard Lagrangian finite elements.

The procedure above makes the transition between a finite element PoU and any other (meshfree or not) partition of unity quite natural and transparent. It can be used in any dimension and with any type of element as well.

An example of a combined  $C^0$ - $C^k$  FE-based partition of unity is shown in Figure 8. Some elements have  $C^0$  (linear) FE weighting functions at their nodes while others have  $C^k$  FE-based weighting functions. At the transition between  $C^0$  and  $C^k$  approximations, there are elements which have both  $C^0$  and  $C^k$  weighting functions. The weighting and PoU functions associated with nodes  $\mathbf{x}_\alpha$  and  $\mathbf{x}_\beta$  shown in Figure 8 are depicted in Figures 9 and 10, respectively.

Figure 9 depicts two- and three-dimensional representations of the weighting function and PoU function at node  $\mathbf{x}_\alpha$  of the discretization shown in Figure 8. In this case the weighting function is the linear finite element shape function of conventional finite elements and has a very simple shape. However, the partition of unity at this same node has a slightly more complex shape due to the influence of weighting functions at other nodes of the element.

Figure 10 depicts two- and three-dimensional representations of the weighting function and PoU function at node  $\mathbf{x}_\beta$  of the discretization shown in Figure 8. The weighting function in this case is a  $C^k$  FE-based function. The corresponding PoU is only a  $C^0$  function due to the use  $C^0$  weighting functions at other nodes of the element. Interestingly, in the case of node  $\mathbf{x}_\beta$  the weighting function does not attain its maximum value at the node. Nonetheless the weighting function is equal to the unity at the node as a result of the scaling introduced in (7) and (15)

The partition of unity defined in (17) can be used to build shape functions able to reproduce polynomials of any degree using the technique described in Section 2. These functions are like generalized finite element shape functions in regions covered only by finite element shape functions. The transition between the  $C^0$  linear FE PoU and  $C^k$  FE-based PoU is handled quite naturally using the partition of unity defined in (17). The procedure is essentially the same in any dimension, only the construction of the weighting functions changes. The approximation properties of these functions follow from Theorem 1.

## 6 NUMERICAL EXPERIMENTS

Two-dimensional linear elasticity problems are used in this section to investigate the performance of the proposed  $C^k$  GFEM. Consider a domain  $\bar{\Omega} = \Omega \cup \partial\Omega$  in  $\mathbb{R}^2$ . The boundary is decomposed as  $\partial\Omega = \partial\Omega_u \cup \partial\Omega_\sigma$  with  $\partial\Omega_u \cap \partial\Omega_\sigma = \emptyset$ . The strong form of the equilibrium equations is given by

$$\nabla \cdot \boldsymbol{\sigma} = \mathbf{0} \quad \boldsymbol{\sigma} = \mathbf{C} : \boldsymbol{\varepsilon} \quad \text{in } \Omega, \quad (18)$$

where  $\mathbf{C}$  is Hooke's tensor. The following boundary conditions are prescribed on  $\partial\Omega$

$$\mathbf{u} = \bar{\mathbf{u}} \text{ on } \partial\Omega_u \quad \boldsymbol{\sigma} \cdot \mathbf{n} = \bar{\mathbf{t}} \text{ on } \partial\Omega_\sigma, \quad (19)$$

where  $\mathbf{n}$  is the outward unit normal vector to  $\Omega$  and  $\bar{\mathbf{t}}$  and  $\bar{\mathbf{u}}$  are prescribed tractions and displacements, respectively.

The weak formulation of the problem described above is given by: Find  $\mathbf{u} \in H^1(\Omega)$  such that

$$\int_{\Omega} \boldsymbol{\sigma}(\mathbf{u}) : \boldsymbol{\varepsilon}(\mathbf{v}) d\mathbf{x} + \eta \int_{\partial\Omega_u} \mathbf{u} \cdot \mathbf{v} ds = \int_{\partial\Omega_\sigma} \bar{\mathbf{t}} \cdot \mathbf{v} ds + \eta \int_{\partial\Omega_u} \bar{\mathbf{u}} \cdot \mathbf{v} ds \quad \forall \mathbf{v} \in H^1(\Omega) \quad (20)$$

where,  $\eta$  is a penalty parameter. The discretization of the weak form (20) using  $C^k$  GFEM shape functions leads to a system of linear equations for the unknown degrees of freedom.

In all examples of this section, the parameter  $k$  that controls the regularity of the R-functions is taken equal to 2 (cf. Section 4). Therefore, the GFEM shape functions with non-convex support belong to the space  $C^2(\Omega)$ . The parameters  $\beta$  and  $\gamma$  used in the definition of cloud boundary functions (cf. Section 3.1) are taken equal to 0.3 and 0.6, respectively.

### 6.1 Single edge-notch test specimen

As a first example, we consider the single-edge notch test specimen shown in Figure 11. The following parameters are assumed: A state of plane strain; Poisson's ratio  $\nu = 0.3$ ; Young's modulus  $E = 1.0$ ; In-plane dimensions  $a = 1.0$ ,  $d = b = 2.0$ ; Domain thickness  $t = 1.0$ ; Neumann boundary conditions

$$\begin{aligned} t_x = 0.0, t_y = 1.0 & \quad \text{on the boundary CD} \\ t_x = 0.0, t_y = -1.0 & \quad \text{on the boundary GF} \end{aligned}$$

In addition, point Dirichlet boundary conditions  $u_x = u_y = 0.0$  at point A and  $u_y = 0.0$  at point E are imposed to prevent rigid body motions. The exact strain energy for this problem is taken as [23]  $U = 9.198\ 545\ 583$ .

This problem will be used to investigate the performance of the proposed  $C^k$  GFEM and its sensitivity to the convexity of the support of the shape functions. Two sequences of uniform meshes are used. In the first, all the supports of the shape functions are convex leading to the  $C^\infty$  partition of unity proposed by Edwards [24]. Figures 12(a) and 12(b) show examples of members of this sequence. Members of the second sequence of meshes are shown in Figures 12(c) and 12(d). In this case, not all clouds are convex and the technique proposed in Section 4 is used to build  $C^{k=2}$  FE-based partitions of unity. The convergence rates of these two sequences of discretizations are compared with those given by Theorem 1. Linear as well as quadratic enrichment functions as defined in (1) are used. The rate of convergence is controlled by the singularity at the crack tip and its asymptotic value, according to Theorem 1, is equal to 0.5 with respect to element size  $h$ . This corresponds to a rate of only 0.25 with respect to number of degrees of freedom in two-dimensional uniform meshes.

The computed strain energy,  $U_h$ , and convergence rates,  $\beta_h$ , for the sequence of meshes with convex clouds are listed in Table 1 for both linear and quadratic enrichments. In both cases the convergence rates quickly approach the asymptotic value of 0.25 predicted by Theorem 1. Figure 13 depicts the data from Table 1. The quadratic approximation provides a slightly smaller error than the linear case when the same number of degrees of freedom is used.

Table 2 lists the results for the  $C^{k=2}$  GFEM. The computations were done on the sequence of meshes with convex and non-convex supports. The performance of the  $C^{k=2}$  GFEM is essentially the same as the  $C^\infty$  GFEM, attesting the effectiveness of the proposed approach to deal with non-convex clouds. The convergence rate also approaches the asymptotic value predicted by Theorem 1 even when relatively coarse meshes are used. Figure 14 depicts the results of Table 2.

## 6.2 Beam-Like Problem

The problem illustrated in Figure 15 is analyzed in this section. The boundary conditions applied are such that the solution far from the left and right edges of the domain is the same as the cantilever beam illustrated in Figure 16.

A state of plane stress is assumed and the following parameters are adopted:

Poisson's ratio  $\nu = 0.3$ ; Young's modulus  $E = 1.0$ ; Domain dimensions  $d = 1.0$  and  $L = 8.0$ ; Thickness  $t = 1.0$ .

The boundary conditions for the problem of Figure 15 are given by

$$\begin{aligned} t_x &= 0.0, \text{ and } t_y = -\frac{P}{2I}(d^2 - y^2) & \text{at } x = 0 \\ t_x &= \frac{PL}{I}y, \text{ and } t_y = \frac{P}{2I}(d^2 - y^2) & \text{at } x = L \end{aligned}$$

where  $P$  is the resultant vertical force applied at  $x = 0$  and  $x = L$  and  $I$  is the moment of inertia given by  $I = 2td^3/3 = 2.0/3.0$ . The value  $P = 1.0$  is used in the computations. Additionally, point constraints  $u_x = u_y = 0.0$  at point B and  $u_x = 0.0$  at point C (cf. Figure 15) are prescribed in order to preclude rigid body motions.

The solution of this problem is given by [75, 78]

$$\sigma_x = \frac{Pxy}{I}, \quad \sigma_y = 0, \quad \tau_{xy} = \frac{P}{2I}(d^2 - y^2) \quad (21)$$

The exact strain energy is given by

$$\begin{aligned} U &= \int_V \left[ \frac{1}{2E}(\sigma_x^2 + \sigma_y^2 + \sigma_z^2) - \frac{\nu}{E}(\sigma_x\sigma_y + \sigma_y\sigma_z + \sigma_z\sigma_x) + \frac{1}{2G}(\tau_{xy}^2 + \tau_{yz}^2 + \tau_{zx}^2) \right] dV \\ &= \frac{1}{2} \int_0^L \int_{-d}^d \left( \frac{\sigma_x^2}{E} + \frac{\tau_{xy}^2}{G} \right) dydx = \frac{1}{9} \frac{P^2 d^3 L^3}{EI^2} + \frac{2}{15} \frac{P^2 d^5 L}{GI^2} \end{aligned} \quad (22)$$

If the assumed parameters are used in (22), we get  $U = 134.24$ .

The rate of convergence of  $h$ -extensions for this problem is controlled by the polynomial order of the approximation since the solution is  $C^\infty$ . In this section, we investigate the performance of the proposed  $C^k$  and  $C^\infty$  GFEM on this class of problem. Like in the previous example, two sequences of uniform meshes are used. In the first one, all the supports of the shape functions are convex while in the second one not all clouds are convex and the technique proposed in Section 4 is used to build  $C^{k=2}$  FE-based partitions of unity. Representative examples of these meshes are shown in Figure 17. Again, the linear and quadratic enrichments given in (1) are used. The asymptotic convergence rates for these two case are, according to Theorem 1, equal to 0.5 and 1.0, respectively.

The computed strain energy,  $U_h$ , and convergence rates,  $\beta_h$ , for the sequence of meshes with convex clouds are listed in Table 3 for both linear and quadratic

enrichments. In both cases, the convergence rates quickly approach the optimal asymptotic values. Figure 18 depicts the data from Table 3. The relative error in energy norm is much smaller than in the example of Section 6.1. This is due to the smoothness of the solution.

Table 4 and Figure 19 show the  $h$ -convergence data for the sequence of meshes with convex and non-convex supports. The non-convex supports require the use of the approach proposed in Section 4. As in the previous example,  $C^{k=2}$  FE-based partitions of unity are built on the non-convex supports using the concept of R-functions. The convergence rates approach the asymptotic values predicted by Theorem 1 even when relatively coarse meshes are used. The performance of the proposed  $C^{k=2}$  GFEM is again very similar to the  $C^\infty$  GFEM proposed by Edwards [24].

### 6.3 Concentrated moment in a linear elasticity problem

In this section, the problem of a concentrated moment  $M$  acting on a straight boundary of an infinitely large plate is analyzed. The problem is illustrated in Figure 20. The thickness of the panel is unity and the distribution of the load is uniform along its thickness.

The computation of approximations to the solution of this problem using, for example, the Principle of Virtual Work, requires shape functions that are smoother than  $C^0$  finite element shape functions. This is demonstrated below. A detailed mathematical analysis of problems with distributional data, like this one, including a-priori error estimates for the generalized finite element method, can be found in [3].

The exact stress distribution in the panel can be computed using the stress function approach which is available in many texts on the theory of elasticity [75]. The stress function for the problem illustrated in Figure 20 can be written as [75]

$$\psi = \frac{M}{\pi}(\theta + \sin \theta \cos \theta) \quad (23)$$

where  $M$  is the concentrated moment and  $\theta$  is the polar coordinate indicated in Figure 20. The stress components in polar coordinates are then given by

$$\begin{aligned} \sigma_r &= \frac{1}{r} \frac{\partial \psi}{\partial r} + \frac{1}{r^2} \frac{\partial^2 \psi}{\partial r^2} = -\frac{2M}{\pi r^2} \sin 2\theta \\ \sigma_\theta &= \frac{\partial^2 \psi}{\partial r^2} = 0 \end{aligned} \quad (24)$$

$$\tau_{r\theta} = -\frac{\partial}{\partial r}\left(\frac{1}{r}\frac{\partial\psi}{\partial\theta}\right) = \frac{M}{\pi r^2}(1 + \cos 2\theta)$$

Alternatively, the stress components in Cartesian coordinates can be written as follows

$$\begin{aligned}\sigma_x &= \sigma_r \cos^2 \theta - 2\tau_{r\theta} \sin \theta \cos \theta = -\frac{4M}{\pi r^2} \sin 2\theta \cos^2 \theta \\ \sigma_y &= \sigma_r \sin^2 \theta + 2\tau_{r\theta} \sin \theta \cos \theta = \frac{M}{\pi r^2} \sin 4\theta \\ \tau_{xy} &= \sigma_r \sin \theta \cos \theta + \tau_{r\theta}(\cos^2 \theta - \sin^2 \theta) = \frac{M}{\pi r^2}(\cos 4\theta + \cos 2\theta)\end{aligned}\tag{25}$$

As we can observe from (25), the stress components are infinite at the point where the concentrated moment is applied ( $r = 0$ ). Furthermore, it is not difficult to show that the strain energy of the whole domain is also infinite. Nonetheless, Babuska and Nistor [3] have shown that the strain energy is finite in any subdomain not containing a small neighborhood of the point  $r = 0$ .

In this section, the  $C^k$  generalized finite element method is used to solve the problem described above. The finite domain BCDE shown in Figure 21 is used in our analysis. Neumann boundary conditions computed from the exact solution are applied on the boundary segments BC, CD, and DE (cf. Figure 21). The components of the traction vector,  $\mathbf{t} = (t_x, t_y)$ , along the boundary are given by

$$\begin{aligned}t_x &= \frac{M}{\pi r^2}(\cos 4\theta + \cos 2\theta), \quad t_y = \frac{M}{\pi r^2} \sin 4\theta \quad \text{at } y = d \\ t_x &= -\frac{M}{\pi r^2}(\cos 4\theta + \cos 2\theta), \quad t_y = -\frac{M}{\pi r^2} \sin 4\theta \quad \text{at } y = -d \\ t_x &= -\frac{4M}{\pi r^2} \sin 2\theta \cos^2 \theta, \quad t_y = \frac{M}{\pi r^2}(\cos 4\theta + \cos 2\theta) \quad \text{at } x = w\end{aligned}\tag{26}$$

In addition, a concentrated moment  $M$  is applied at the origin as discussed below. With these boundary conditions, the exact solution in the finite domain BCDE is the same as for the infinite domain case. Point Dirichlet boundary conditions are prescribed in order to prevent rigid body motions. The following parameters are adopted for the analysis: A state of plane stress; Poisson's ratio  $\nu = 0.3$ ; Young's modulus  $E = 1.0$ ; Domain edge dimensions  $w = d = 8.0$ ; Thickness  $t = 1.0$ ; Concentrated moment at origin  $M = 10.0$ .

The vector of nodal loads that are work-equivalent to the concentrated moment  $M$  can be computed as follows. For simplicity of notation, we consider the case of a concentrated moment applied at  $(x_0, y_0)$ , as illustrated in Figure 21.

Let a  $C^k$  generalized finite element test function or virtual displacement be given by

$$\mathbf{v}(x, y) = \begin{bmatrix} v_x(x, y) \\ v_y(x, y) \end{bmatrix} = \begin{bmatrix} \phi_1 & 0 & \phi_2 & 0 & \dots & \phi_{N_{uv}} & 0 \\ 0 & \phi_1 & 0 & \phi_2 & \dots & 0 & \phi_{N_{uv}} \end{bmatrix} \begin{bmatrix} v_{1x} \\ v_{1y} \\ v_{2x} \\ v_{2y} \\ \vdots \\ v_{N_{uv}x} \\ v_{N_{uv}y} \end{bmatrix}$$

where  $N_{uv}$  is the number of degrees of freedom for each component of the displacement vector and  $\phi_j$  is a  $C^k$  GFE shape function.

The concentrated moment  $M$  applied on the boundary segment BE of Figure 21 can be written as

$$\mathbf{m} = \begin{bmatrix} -M\delta'_y(y - y_0) \\ 0 \end{bmatrix}$$

where  $\delta'_y(y - y_0)$  is a distribution (also known as derivative Dirac delta generalized function) defined by

$$\int_{-\infty}^{\infty} \delta'_y(y - y_0)v(x_0, y)dy = -\frac{\partial v}{\partial y}(x_0, y_0)$$

where  $v(x_0, y)$  is a function such that  $\partial v/\partial y$  is continuous and bounded at  $(x_0, y_0)$ .

The virtual work of the concentrated moment  $\mathbf{m}$  is then given by

$$\begin{aligned} W_M &= \int_{y=-d}^{y=d} \mathbf{m} \cdot \mathbf{v} dy = \int_{y=-d}^{y=d} -M\delta'_y(y - y_0)v_x(x_0, y)dy = M\frac{\partial v_x}{\partial y}(x_0, y_0) \\ &= M \sum_{j=1}^{N_{uv}} v_{jx} \frac{\partial \phi_j}{\partial y}(x_0, y_0) \end{aligned}$$

The vector of equivalent nodal loads is then given by

$$\mathbf{f} = M \left[ \frac{\partial \phi_1(\mathbf{x}_0)}{\partial y} \quad 0 \quad \frac{\partial \phi_2(\mathbf{x}_0)}{\partial y} \quad 0 \quad \dots \quad \frac{\partial \phi_{N_{uv}}(\mathbf{x}_0)}{\partial y} \quad 0 \right]^T \quad (27)$$

where  $\mathbf{x}_0 = (x_0, y_0)$  is the position vector of the node where the concentrated moment is applied. From the above, we conclude that the nodal loads equivalent to a concentrated moment are expressed as the product of the concentrated moment and the derivative of shape functions in the direction tangent to the boundary of the body.

It is quite apparent that the problem of a concentrated moment in elasticity can not be solved with classical finite elements. Lagrangian finite elements provide only  $C^0$  shape functions and the derivatives of these functions do not exist at the nodes. However, as discussed in previous sections, the proposed  $C^k$  partition of unity provides arbitrary order of regularity for the GFEM shape functions.

The  $C^k$  GFEM shape functions are zero, together with their derivatives up to order  $k$ , at the boundary of their support. Therefore, at any node of an element the only non-zero functions and derivatives of functions up to order  $k$  are those associated with the node. Therefore, if a concentrated moment is applied at a given node  $\mathbf{x}_0$  of an element, the only non-zero components of the vector of equivalent nodal loads for this element are those corresponding to shape functions associated with node  $\mathbf{x}_0$ . This makes the implementation of this type of loading quite straightforward.

As in previous examples,  $h$ -extensions are performed in order to investigate the performance of the proposed  $C^k$  GFEM for this class of problem. Quadratic enrichment functions and uniform meshes are used. In the computations, only the lower half  $\Omega_L = \{(x, y) \in \mathbb{R}^2 : 0 < x < w, -d < y < 0\}$  of the domain shown in Figure 21 is discretized. Anti-symmetry boundary conditions  $u_x = 0$ ,  $t_y = 0$  are applied at  $y = 0$  and the concentrated moment is taken as  $M/2$ . Table 5 presents the computed strain energy for domain  $\Omega_L$  and for the subdomain  $A = \{(x, y) \mid 4 < x < 8, -8 < y < -4\}$  (cf. Figure 21). It can be observed that the computed strain energy for the domain  $\Omega_L$  grows very fast with mesh refinement, as expected. The strain energy in the subdomain  $A$  is finite and the computed values converge to the exact one, albeit not monotonically. This can be explained by the fact that this problem is *not* equivalent to the minimization of a potential as in most linear elasticity problems. The reference value for the strain energy in the subdomain  $A$  was computed using the exact solution (26). The rate of convergence in the energy norm ( $\beta_h$ ) with respect to the number of degrees of freedom ( $N$ ) is also presented in Table 5. It approaches the asymptotic value of 1.0 theoretically predicted by Babuska and Nistor [3] as the mesh is refined. The relative error in the energy norm for

region A is shown in Figure 22. Figure 23 illustrates the computed von-Mises stress distributions for several mesh sizes.

#### 6.4 Concentrated moment in a linear elasticity problem - Analysis using combined PoUs

The numerical integration of  $C^k$  GFE shape functions, while more straightforward than in case of meshfree methods, requires more integration points than in the case of the  $C^0$  GFEM (cf. Section 7). Therefore, these functions should be used only in parts of the computational domain where their high smoothness is required, as in the case of problems with distributional data, or is beneficial for some other reason. In situations like the example of Section 6.3, the combined  $C^0$ - $C^k$  GFEM presented in Section 5 can be used effectively. In this approach,  $C^k$  GFE shape functions are used at elements where concentrated moments are applied. Elsewhere,  $C^0$  GFE shape functions can be used and the procedure described in Section 5 is employed to combine the two types of shape functions. Discretizations with these features are illustrated in Figure 24. As a result, only a few elements in the computational domain uses  $C^k$  GFE shape functions leading to a reduction in the computational cost.

In this section, the concentrated moment problem of Section 6.3 is analyzed again using  $h$ -extensions on uniform  $C^0$ - $C^k$  GFE discretizations like those illustrated in Figure 24. Quadratic enrichment is used and all parameters are as in Section 6.3. Table 6 summarizes the results. The performance of the combined  $C^0$ - $C^k$  GFEM is essentially the same as the  $C^\infty$  GFEM used in the previous section. Figure 25 depicts the data from Table 6 and Figure 26 displays the computed von-Mises stress distributions for a number of different mesh sizes.

The a-priori error estimates for this class of problem presented in [3] require that a smooth approximation be used in the entire domain, like in the computations presented in Section 6.3. The numerical results presented here shows that this requirement is not always necessary.

## 7 NUMERICAL INTEGRATION

As discussed in Section 1, meshfree methods offer an attractive alternative for the solution of many classes of problems. The computational cost of meshless methods based on the Galerkin method, however, is one of the main drawbacks of these methods. This is mainly due to (i) larger bandwidth of the global matrices as compared with those in the FEM and (ii) difficulties in the

numerical integration of meshfree shape functions and their products. This issue is discussed in great detail in, for example, [15] and it has been an active area of research in the meshfree community. Dolbow and Belytschko [15] proposed the use of weighting functions with rectangular support and the use of integration cells at their intersections. Difficulties with the intersection of the supports and the boundary of the computational domain still remain. Griebel and Schweitzer [28, 30, 31] proposed a similar approach including several improvements in the computational implementation. De and Bathe [13] developed special integration rules for the integration over the intersection of the support of meshfree functions in two and three dimensions. Chen et al. [12] developed a very appealing approach based on nodal integration which has shown to be stable and suitable for linear approximations.

In the proposed  $C^k$  GFEM, the support of the shape functions are the same as the corresponding FE shape functions defined on the same mesh. This property considerably facilitates the numerical integration of these functions and also leads to the same bandwidth of matrices as in the case of  $C^0$  GFEM. The proposed  $C^k$  GFEM shape functions are, however, non-polynomial functions and an analysis of the numerical integration of these functions using Gauss-Legendre rules is needed. In addition, the  $C^\infty$  GFEM shape functions proposed by Edwards [24, 25] (cf. Section 3) was used with a collocation procedure, not in a Galerkin method as in this paper. The next section presents some numerical experiments aimed at shedding some light on the issue of numerical integration of  $C^\infty$  and  $C^k$  GFE shape functions.

### *7.1 Numerical integration of $C^k$ generalized finite element approximations*

The problem described in Section 6.1 is used in the numerical experiments presented in this section. Gauss-Legendre integration rules are used to integrate the element stiffness matrices and right hand sides. The effect of numerical integration errors is measured against the computed strain energy in the entire domain. The mesh of Figure 12(b) which has only convex clouds and the mesh of Figure 12(d) which has non-convex clouds, are used. Reference values for the strain energy are computed using  $50 \times 50$  integration points for the computation of the stiffness matrix in each element. These values are listed in Tables 7 and 8. The computations were done for both linear and quadratic approximations. The problem was then solved using the same meshes and the integration rules indicated in the tables. The computed relative error of the strain energy with respect to the reference values are listed in Table 7 for the mesh with only convex clouds and in Table 8 for the mesh with non-convex clouds. From the tables, we can observe that the error on the computed strain energy decreases reasonably fast as the number of integration points is increased, albeit not monotonically. As expected, the integration error for the

case of quadratic enrichment is higher than the linear case when the same number of integration points is used. Also, the required number of integration points is larger than in the case of generalized finite element methods based on  $C^0$  Lagrangian partitions of unity.

The number of integration points to be used in a given problem depends on the level of the discretization error ones wishes to attain. For discretization errors of, for example, 0.1 % in energy a  $7 \times 7$  rule suffices for this particular problem. Adaptive integration procedures could, of course, also be used at the price of increasing the complexity of the implementation [67, 69]. Alternatively, the  $C^0$ - $C^k$  GFEM described in Section 5 can be used as illustrated in Section 6.4. This is a good approach for solving problems like the one discussed in Section 6.3 in which  $C^k$  GFEM shape functions are needed only in a few elements and  $C^0$  GFEM shape functions are used elsewhere.

## 8 SUMMARY AND CONCLUDING REMARKS

In this paper, we have proposed a procedure to build generalized finite element shape functions with the following features:

- (i) Arbitrarily smooth;
- (ii) Non-structured meshes composed of triangular or quadrilateral elements can be used and extensions to higher dimensions are possible;
- (iii) The functions have the same support as corresponding global finite element Lagrangian shape functions defined on the same mesh;
- (iv) The implementation can be done using existing FE data structures.

The authors are not aware of other finite element shape functions satisfying all these requirements. While several meshfree functions have property (i), they are difficult to numerically integrate, lead to matrices with large bandwidths, as discussed in Section 7, and their combination with finite element approximations is not trivial. The high smoothness of the proposed  $C^k$  GFEM can be beneficial in many applications as discussed in Section 1. In this work, this is demonstrated through the solution of problems with distributional data. A linear elasticity problem with a concentrated moment is considered. Higher order distributional boundary conditions can also be handled. A detailed numerical study of the convergence properties of the method is presented for this class of problems as well as for problems in energy space. In all problems solved, meshes with convex and non-convex clouds are considered. The studies demonstrate that the convergence rates of the method are not sensitive to the convexity of the support of the shape functions.

The partition of unity property of the proposed  $C^k$  finite element-based PoU is used to build GFE shape functions that can exactly reproduce linear and quadratic polynomials. Higher order polynomials and non-polynomial enrichment functions can, of course, be used. The later case is very effective to represent local features like cracks, holes and voids, as demonstrated in, e.g., [4, 16, 18, 19, 53, 55, 56, 68, 70, 71].

The proposed  $C^k$  GFE shape functions removes the requirement of convex supports of the  $C^\infty$  GFE shape functions proposed by Edwards [24, 25]. In fact, the proposed functions are also  $C^\infty$ , except at re-entrant corners (edges) of two-dimensional (three-dimensional) supports where the functions are  $C^k$ , with  $k$  arbitrarily large.

The integrability of the functions using standard Gauss-Legendre rules is investigated in Section 7. The study shows that while property (iii) listed above facilitates the numerical integration of the proposed  $C^k$  GFE shape functions, the required number of integration points is larger than in the case of generalized finite element methods based on  $C^0$  Lagrangian partitions of unity.

In Section 5, a technique to combine a  $C^0$  linear finite element partition of unity with a  $C^k$  finite element-based partition of unity is presented. The technique allows the use of  $C^k$  GFE shape functions in parts of the computational domain where their high smoothness is required, as in the case of problems with distributional data, and the less computationally demanding  $C^0$  GFE shape functions elsewhere in the domain. The approach, while simple, is quite flexible and generic. It can, in fact, be used to couple finite element approximations with any (meshfree or not) approximation that possess the partition of unity property as discussed in [20]. The effectiveness of this approach is demonstrated in Section 6.4 to solve a problem with distributional boundary conditions. The performance of the combined  $C^0$ - $C^k$  GFEM is shown to be essentially the same as in the case of  $C^\infty$  GFEM while requiring less computational effort for the numerical integration of the matrices.

## ACKNOWLEDGMENTS

The partial support of this work by The University of Illinois at Urbana-Champaign and by the National Science Foundation under grant DMS-0341982 to the first author and the Korea Research Foundation to the second author (Project Number : M06-2003-000-10312-0) is gratefully acknowledged. The authors also wish to thank Prof. Ivo Babuška from the University of Texas at Austin for fruitful discussions during the course of this research.

## References

- [1] S. N. Atluri and T. Zhu. A new meshless local Petrov-Galerkin (mlpg) approach in computational mechanics. *Computational Mechanics*, 22: 117–127, 1998.
- [2] I. Babuška, U. Banerjee, and J. E. Osborn. Survey of meshless and generalized finite element methods: A unified approach. *Acta Numerica*, 12: 1–125, May 2003.
- [3] I. Babuška and V. Nistor. Interior numerical approximation of boundary value problems with a distributional data. Technical Report ICES 05-11, The University of Texas at Austin, Austin, Texas, February 2005. Available at <http://www.ices.utexas.edu/>.
- [4] T. Belytschko and T. Black. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering*, 45:601–620, 1999.
- [5] 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.
- [6] T. Belytschko, Y. Y. Lu, and L. Gu. Crack propagation by element free Galerkin methods. In *Advanced Computational Methods for Material Modeling*, pages 191–205, 1993. AMD-Vol. 180/PVP-Vol. 268, ASME 1993.
- [7] T. Belytschko, Y. Y. Lu, and L. Gu. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 37:229–256, 1994.
- [8] T. Belytschko, D. Organ, and Y. Krongauz. A coupled finite element–element-free Galerkin method. *Computational Mechanics*, 17:186–195, 1995.
- [9] B. Calvo, M. A. Martinez, and M. Doblare. On solving large strain hyperelastic problems with the natural element method. *International Journal for Numerical Methods in Engineering*, 62:159–185, 2005.
- [10] J. S. Chen, C. Pan, and C. T. Wu. Large deformation analysis of rubber based on a reproducing kernel particle method. *Computational Mechanics*, 19:211–227, 1997.
- [11] J. S. Chen, C. Pan, C. T. Wu, and W. K. Liu. Reproducing kernel particle methods for large deformation analysis of nonlinear structures. *Computer Methods in Applied Mechanics and Engineering*, 139:195–227, 1996.
- [12] J.-S. Chen, C.-T. Wu, S. Yoon, and Y. You. A stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, 50:435–466, 2001.
- [13] S. De and K. J. Bathe. The method of finite spheres. *Computational Mechanics*, 25:329–345, 2000.
- [14] Carl deBoor. *A Practical Guide to Splines*. Springer-Verlag, New York, 1978.
- [15] J. Dolbow and T. Belytschko. Numerical integration of the Galerkin weak

- form in meshfree methods. *Computational Mechanics*, 23:219–230, 1999.
- [16] J. Dolbow, N. Moes, and T. Belytschko. Discontinuous enrichment in finite elements with a partition of unity method. *Finite Elements in Analysis and Design*, 36:235–260, 2000.
- [17] C. A. Duarte, I. Babuška, and J. T. Oden. Generalized finite element methods for three dimensional structural mechanics problems. In S. N. Atluri and P. E. O’Donoghue, editors, *Modeling and Simulation Based Engineering*, volume I, pages 53–58. Tech Science Press, October 1998. Proceedings of the International Conference on Computational Engineering Science, Atlanta, GA, October 5-9, 1998.
- [18] C. A. Duarte, I. Babuška, and J. T. Oden. Generalized finite element methods for three dimensional structural mechanics problems. *Computers and Structures*, 77:215–232, 2000.
- [19] C. A. Duarte, O. N. Hamzeh, T. J. Liszka, and W. W. Tworzydło. A generalized finite element method for the simulation of three-dimensional dynamic crack propagation. *Computer Methods in Applied Mechanics and Engineering*, 190:2227–2262, 2001.
- [20] C. A. Duarte, D. Q. Migliano, and E. B. Becker. A technique to combine meshfree- and finite element-based partition of unity approximations. Structural Research Series 638, Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, May 2005. ISSN-0197-9191.
- [21] 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.
- [22] C. A. M. Duarte and J. T. Oden. *Hp* clouds—an *hp* meshless method. *Numerical Methods for Partial Differential Equations*, 12:673–705, 1996.
- [23] C. Armando Duarte. *The hp Cloud Method*. PhD dissertation, The University of Texas at Austin, December 1996. Austin, TX, USA.
- [24] H. C. Edwards.  $C^\infty$  finite element basis functions. Technical report, TICAM Report 96-45, The University of Texas at Austin, 1996.
- [25] H. C. Edwards. *A Parallel Infrastructure for Scalable Adaptive Finite Element Methods and Its Application to Least Squares  $C^\infty$  Collocation*. PhD dissertation, The University of Texas at Austin, May 1997. Austin, TX, USA.
- [26] S. Fernández-Méndez and A. Huerta. Imposing essential boundary conditions in mesh-free methods. *Computer Methods in Applied Mechanics and Engineering*, 193:1257–1275, 2004.
- [27] R. A. Gingold and J. J. Monaghan. Kernel estimates as a basis for general particle methods in hydrodynamics. *Journal of Computational Physics*, 46:429–453, 1982.
- [28] M. Griebel and M. A. Schweitzer. A particle-partition of unity method for the solution of elliptic, parabolic and hyperbolic pdes. *SIAM Journal Scientific Computing*, 22(3):853–890, 2000.
- [29] M. Griebel and M. A. Schweitzer, editors. *Meshfree Methods for Par-*

- tial Differential Equations*, volume 26 of *Lecture Notes in Computational Science and Engineering*, 2002. Springer.
- [30] M. Griebel and M. A. Schweitzer. A particle-partition of unity method—Part II: Efficient cover construction and reliable integration. *SIAM Journal Scientific Computing*, 23(5):1655–1682, 2002.
- [31] M. Griebel and M. A. Schweitzer. A particle-partition of unity method—Part III: A multilevel solver. *SIAM Journal Scientific Computing*, 24(2):377–409, 2002.
- [32] M. Griebel and M. A. Schweitzer, editors. *Meshfree Methods for Partial Differential Equations II*, volume 43 of *Lecture Notes in Computational Science and Engineering*, 2005. Springer.
- [33] W. Han, H. Lu, S. Li, J. Cao, and W.K. Liu. Reproducing kernel element method. Part I: Theoretical formulation. *Computer Methods in Applied Mechanics and Engineering*, 193:933–951, 2004.
- [34] D. Hegen. Element-free Galerkin methods in combination with finite element approaches. *Computer Methods in Applied Mechanics and Engineering*, 135(1–2):143–166, 1996.
- [35] A. Huerta and S. Fernandez-Mendez. Enrichment and coupling of the finite element and meshless methods. *International Journal for Numerical Methods in Engineering*, 48:1615–1636, 2000.
- [36] Y. Krongauz and T. Belytschko. Enforcement of essential boundary conditions in meshless approximations using finite elements. *Computer Methods in Applied Mechanics and Engineering*, 131:133–145, 1996.
- [37] P. Krysl and T. Belytschko. Analysis of thin plates by the element-free Galerkin method. *Computational Mechanics*, 17:26–35, 1996.
- [38] P. Krysl and T. Belytschko. Analysis of thin shells by the element-free galerkin method. *Int. J. Solids and Structures*, 33:3057–3080, 1996.
- [39] P. Lancaster and K. Salkauskas. *Curve and Surface Fitting, an Introduction*. Academic Press, San Diego, 1986.
- [40] F. Lekien and J. Marsden. Tricubic interpolation in three dimensions. *International Journal for Numerical Methods in Engineering*, 63:455–471, 2005.
- [41] Francois Lekien, Chad Coulliette, and Jerry Marsden. Lagrangian structures in very high-frequency radar data and optimal pollution timing. In *American Institute of Physics Conference Proceedings*, volume 676, pages 162–168, 2003.
- [42] S. Li and W. K. Liu. *Meshfree Particle Methods*. Springer, November 2004. ISBN: 3540222561.
- [43] Tadeusz Liszka and J. Orkisz. Finite difference method for arbitrary irregular meshes in nonlinear problems of applied mechanics. In *IV SMiRt*, San Francisco, 1977.
- [44] Tadeusz Liszka and J. Orkisz. The finite difference method at arbitrary irregular grids and its application in applied mechanics. *Computers and Structures*, 11:83–95, 1980.
- [45] G. R. Liu. *Mesh Free Methods: Moving Beyond the Finite Element*

- Method*. CRC Press, July 2002.
- [46] W. K. Liu and Y. Chen. Wavelet and multiple scale reproducing kernel methods. *International Journal for Numerical Methods in Fluids*, 21: 901–931, 1995.
  - [47] W. K. Liu, Y. Chen, S. Jun, J. S. Chen, T. Belytschko, C. Pan, R. A. Uras, and C. T. Chang. *Overview and Applications of the Reproducing Kernel Particle Methods*, volume 3 of *Archives in Computational Methods in Engineering: State of Art Reviews*. CIMNE, Barcelona, Spain, 1996.
  - [48] W. K. Liu, S. Jun, and Y. F. Zhang. Reproducing kernel particle methods. *International Journal for Numerical Methods in Engineering*, 20:1081–1106, 1995.
  - [49] W. K. Liu, R. A. Uras, and Y. Chen. Enrichment of the finite element method with reproducing kernel particle method. *Journal of Applied Mechanics*, 64:861–870, 1997.
  - [50] H. Lu, W. Han, and D.C. Liu, W.K. Simkins. Reproducing kernel element method Part II: Globally conforming  $i^m/c^n$  hierarchies. *Computer Methods in Applied Mechanics and Engineering*, 193:953–987, 2004.
  - [51] H. Lu, S. Li, D.C. Simkins Jr., W.K. Liu, and J. Cao. Reproducing kernel element method Part III: Generalized enrichment and applications. *Computer Methods in Applied Mechanics and Engineering*, 193:989–1011, 2004.
  - [52] 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.
  - [53] N. Moes, J. Dolbow, and T. Belytschko. A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering*, 46:131–150, 1999.
  - [54] B. Nayroles, G. Touzot, and P. Villon. Generalizing the finite element method: Diffuse approximation and diffuse elements. *Computational Mechanics*, 10:307–318, 1992.
  - [55] J. T. Oden and C. A. Duarte. Chapter: Clouds, Cracks and FEM’s. In B. D. Reddy, editor, *Recent Developments in Computational and Applied Mechanics*, pages 302–321, Barcelona, Spain, 1997. International Center for Numerical Methods in Engineering, CIMNE.
  - [56] J. T. Oden and C. A. M. Duarte. Chapter: Solution of singular problems using  $hp$  clouds. In J. R. Whiteman, editor, *The Mathematics of Finite Elements and Applications– Highlights 1996*, pages 35–54, New York, NY, 1997. John Wiley & Sons.
  - [57] E. Onate and S. Idelsohn. A mesh-free finite point method for advective-diffusive transport and fluid flow problems. *Computational Mechanics*, 21:283–292, 1998.
  - [58] E. Onate, S. Idelsohn, and O. C. Zienkiewicz. Finite point methods in computational mechanics. Technical report, International Center for Numerical Methods in Engineering, Barcelona, Spain, July 1995.
  - [59] E. Onate, S. Idelsohn, O. C. Zienkiewicz, R. L. Taylor, and C. Sacco. A

- stabilized finite point method for analysis of fluid mechanics problems. *Computer Methods in Applied Mechanics and Engineering*, pages 315–346, 1996.
- [60] D. Organ, M. Fleming, T. Terry, and T. Belytschko. Continuous meshless approximations for nonconvex bodies by diffraction and transparency. *Computational Mechanics*, 18:225–235, 1996.
- [61] D. M. Quaresma. Element-based *hp*-cloud method for large deformation analysis of rubber. Ph.D. Proposal, The University of Texas at Austin, January 2003.
- [62] E. Ramm and W. A. Wall. Preface of computational methods for shells. *Computer Methods in Applied Mechanics and Engineering*, 194:iii–v, 2005.
- [63] V. L. Rvachev and T. I. Sheiko. R-functions in boundary value problems in mechanics. *Applied Mechanics Review*, 48(4):151–188, 1995.
- [64] V. Shapiro. Theory of R-functions and applications: a primer. Technical Report TR91-1219, Computer Science Department, Cornell University, Ithaca, NY, 1991.
- [65] D. Shepard. A two-dimensional function for irregularly spaced data. In *ACM National Conference*, pages 517–524, 1968.
- [66] D.C. Simkins Jr., S. Li, H. Lu, and W.K. Liu. Reproducing kernel element method. Part IV: Globally compatible  $c^n (n \geq 1)$  triangular hierarchy. *Computer Methods in Applied Mechanics and Engineering*, 193:1013–1034, 2004.
- [67] T. Strouboulis, I. Babuška, and K. Copps. The design and analysis of the generalized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 81(1–3):43–69, 2000.
- [68] T. Strouboulis, K. Copps, and I. Babuška. The generalized finite element method: An example of its implementation and illustration of its performance. *International Journal for Numerical Methods in Engineering*, 47(8):1401–1417, 2000.
- [69] T. Strouboulis, K. Copps, and I. Babuška. The generalized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 190:4081–4193, 2001.
- [70] T. Strouboulis, L. Zhang, and I. Babuška. Generalized finite element method using mesh-based handbooks: application to problems in domains with many voids. *Computer Methods in Applied Mechanics and Engineering*, 192:3109–3161, 2003.
- [71] N. Sukumar, D. Chopp, N. Moes, and T. Belytschko. Modeling holes and inclusions by level sets in the extended finite element method. *Computer Methods in Applied Mechanics and Engineering*, 190:6183–6200, 2001.
- [72] K. S. Surana, S. R. Petti, A. R. Ahmadi, and J. N. Reddy. On p-version hierarchical interpolation functions for higher-order continuity finite element models. *International Journal of Computational Engineering Science*, 2(4):653–673, 2001.
- [73] J. W. Swegle, S. W. Attaway, M. W. Heinstein, F. J. Mello, and D. L.

- Hicks. An analysis of the smoothed particle hydrodynamics. Technical Report SAND93-2513 UC-705, Sandia, 1994.
- [74] J. W. Swegle, D. L. Hicks, and S. W. Attaway. Smoothed particle hydrodynamics stability analysis. *Journal of Computational Physics*, 116: 123–134, 1995.
- [75] S. P. Timoshenko and J. N. Goodier. *Theory of Elasticity*. McGraw-Hill Book Company, London, 1982.
- [76] W. Tworzydło. Analysis of large deformations of membrane shells by the generalized finite difference method. *Computers and Structures*, 27(1): 39–59, 1987.
- [77] W. Tworzydło. The fdm in arbitrary curvilinear co-ordinates - formulation, numerical approach and applications. *International Journal for Numerical Methods in Engineering*, 28:261–277, 1989.
- [78] A. C. Ugural and S. K. Fenster. *Advanced Strength and Applied Elasticity*. Prentice-Hall, Upper Saddle River, New Jersey, 1995.
- [79] G. J. Wagner and W. K. Liu. Hierarchical enrichment for bridging scales and meshfree boundary conditions. *International Journal for Numerical Methods in Engineering*, 50(3):507–524, 2000.

# FIGURES

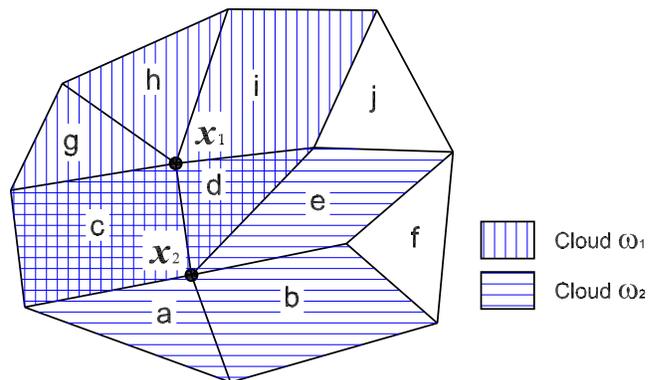


Fig. 1. Examples of convex and non-convex finite element clouds.

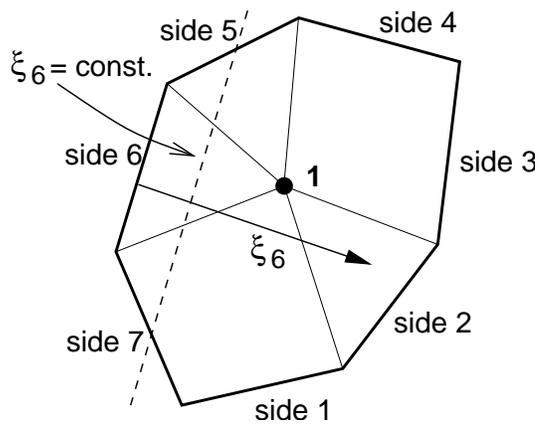


Fig. 2. Setup for the construction of cloud boundary functions.

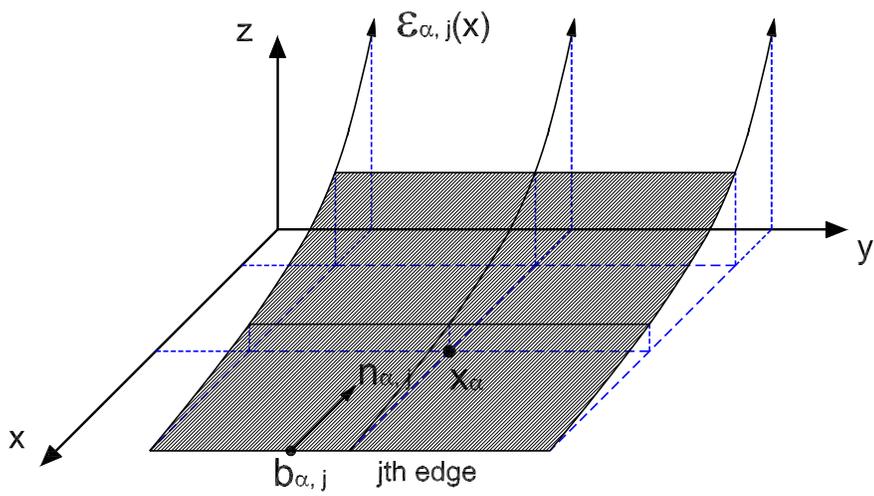


Fig. 3. Example of a two-dimensional cloud boundary function.

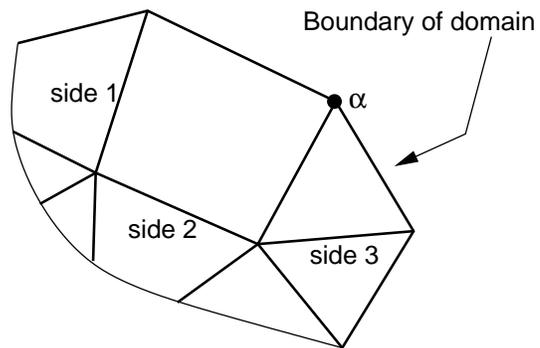
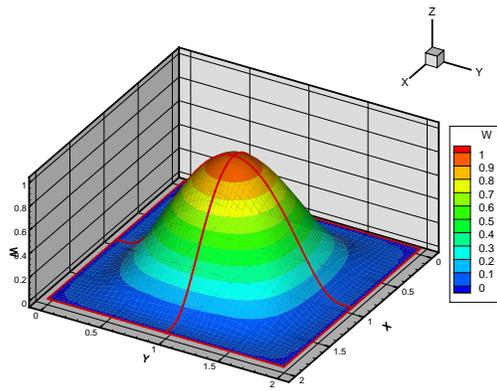
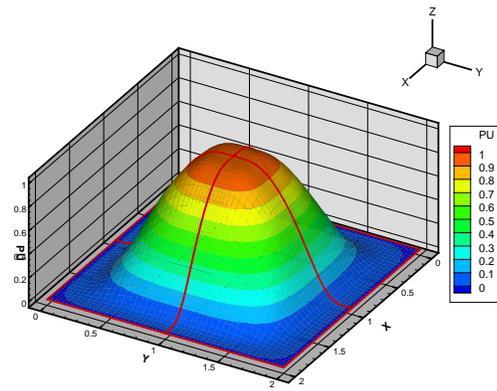


Fig. 4. Example of a two-dimensional cloud associated with a node at the boundary of the domain.



(a) Weighting function for convex cloud.



(b) Partition of unity function for convex cloud

Fig. 5. Weighting function and partition of unity function for a convex cloud.

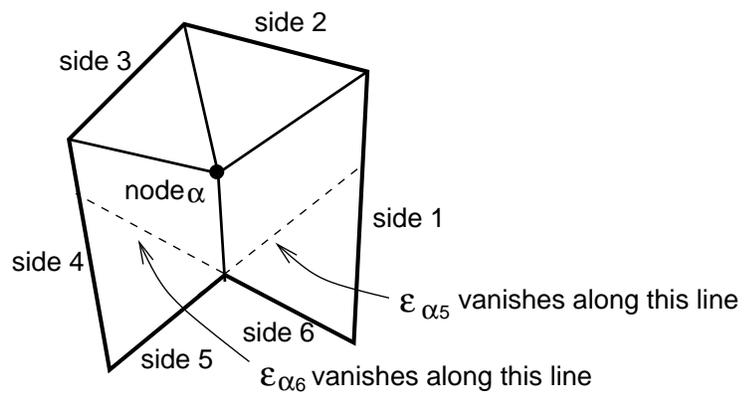
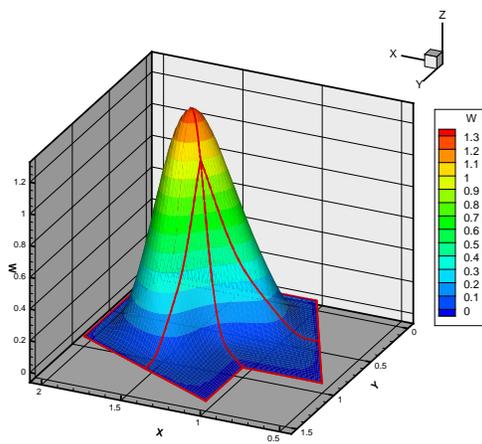
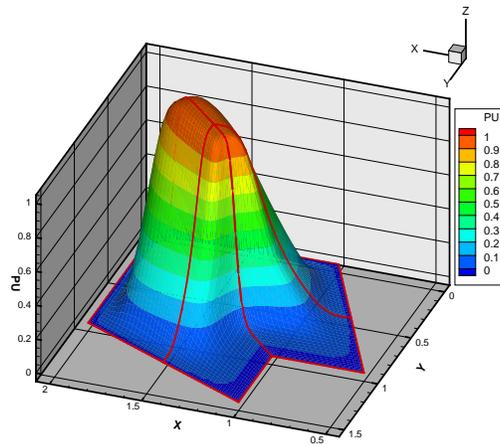


Fig. 6. Non-convex finite element cloud.



(a) Weighting function for a non-convex cloud.



(b) Partition of unity function for a non-convex cloud.

Fig. 7.  $C^{k=2}$  finite element-based weighting function and partition of unity function for a non-convex cloud.

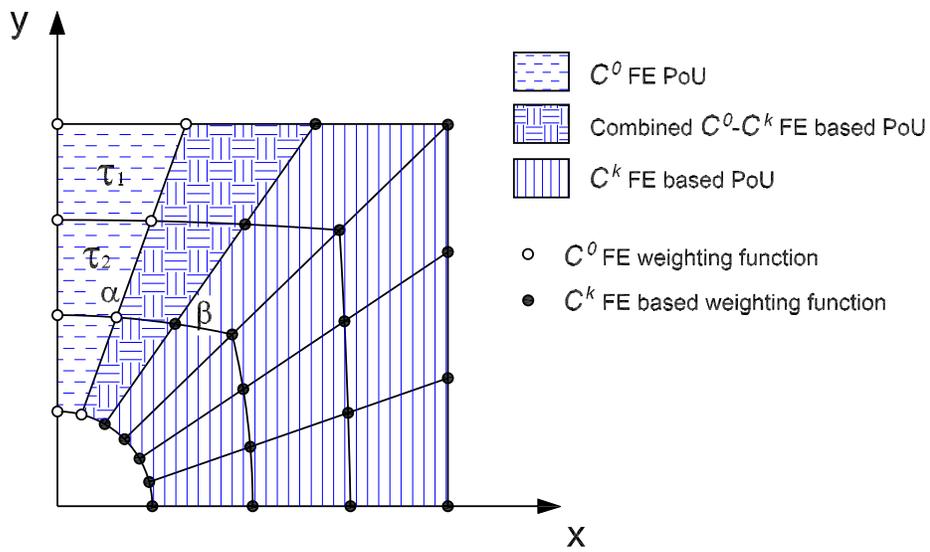
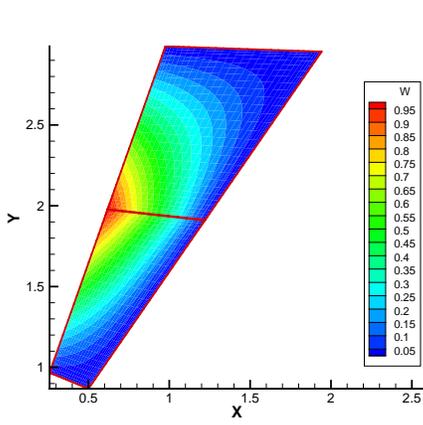
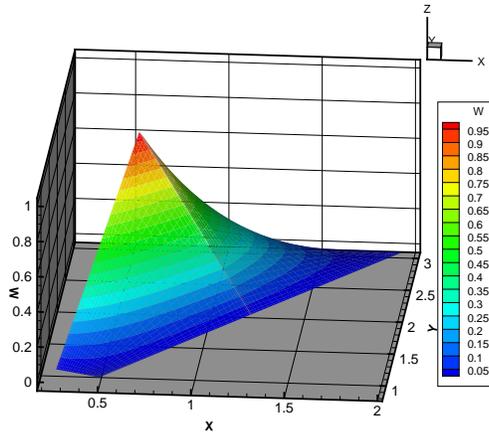


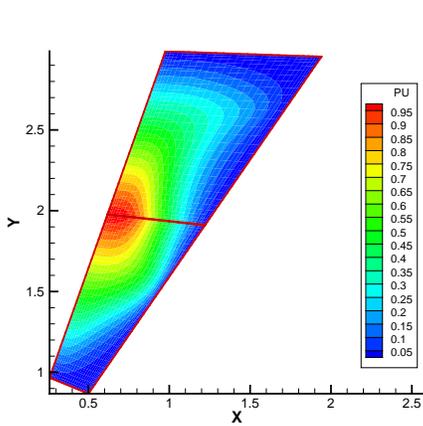
Fig. 8. Example of a discretization with a combined  $C^0$ - $C^k$  FE-based partition of unity.



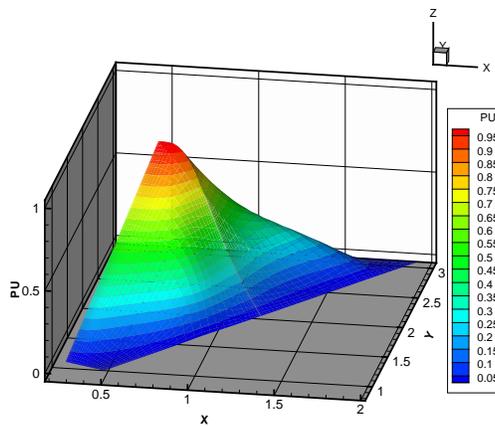
(a) Iso contour of the  $C^0$  (linear) finite element weighting function at node  $\alpha$  of the discretization shown in Figure 8.



(b) Three-dimensional representation of the finite element weighting function shown in Figure 9(a).

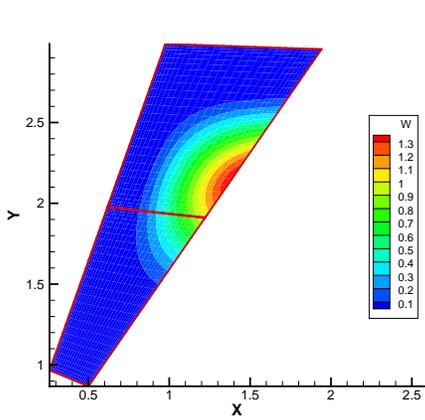


(c) Iso contour of the  $C^0$  PoU function at node  $\alpha$  of the discretization shown in Figure 8.

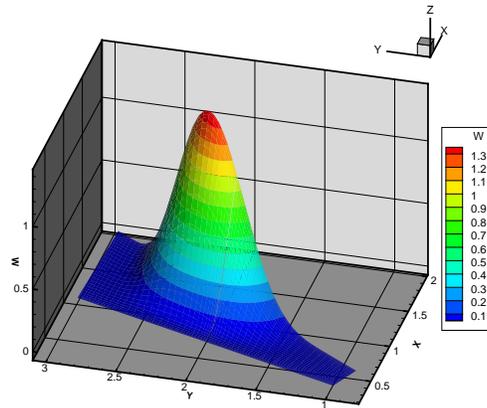


(d) Three-dimensional representation of the PoU function shown in Figure 9(c).

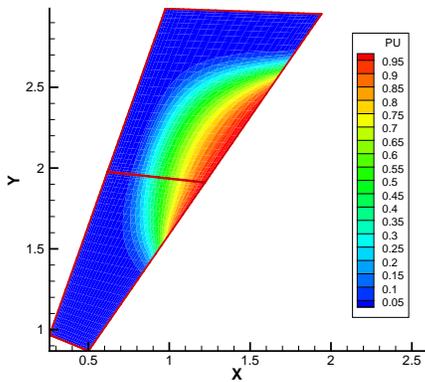
Fig. 9.  $C^0$  FE-based weighting function and PoU function at node  $\alpha$  of the discretization shown in Figure 8.



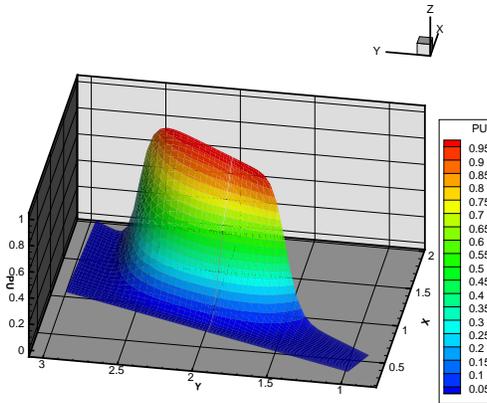
(a) Iso contour of the  $C^k$  FE-based weighting function at node  $\beta$  of the discretization shown in Figure 8.



(b) Three-dimensional representation of the  $C^k$  FE-based weighting function shown in Figure 10(a).



(c) Iso contour of the  $C^0$ - $C^k$  PoU function at node  $\beta$  of the discretization shown in Figure 8.



(d) Three-dimensional representation of the PoU function shown in Figure 10(c).

Fig. 10.  $C^k$  FE-based weighting function and  $C^0$ - $C^k$  PoU function at node  $\beta$  of the discretization shown in Figure 8.

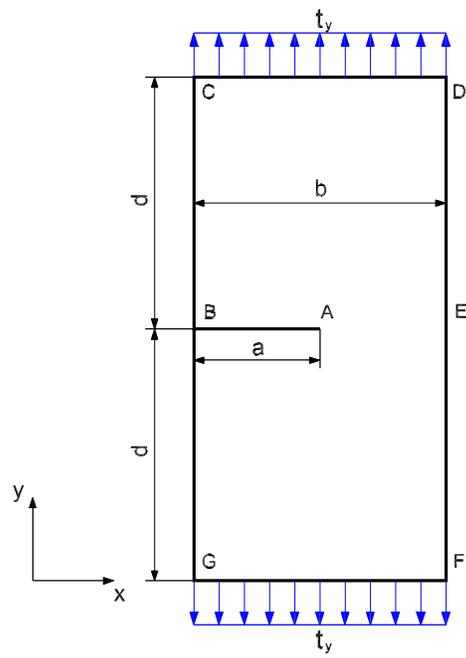


Fig. 11. Single edge-notch test specimen.

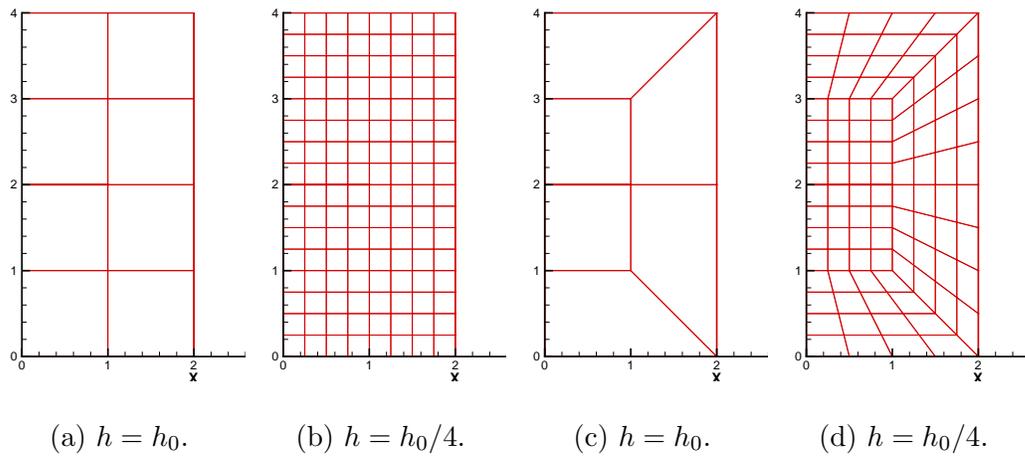


Fig. 12. (a),(b) Meshes with convex clouds only. (c), (d) Meshes with convex and non-convex clouds.

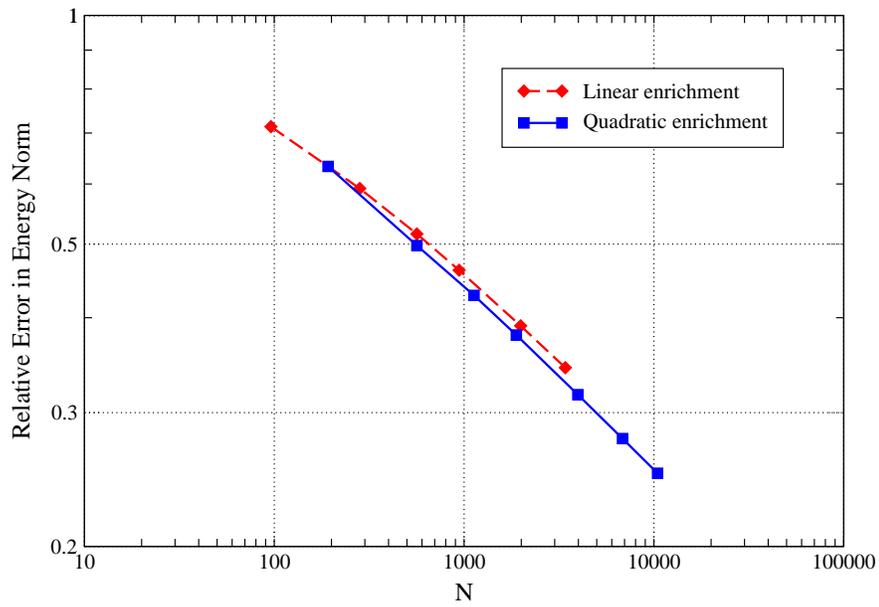


Fig. 13.  $H$ -convergence for the single edge-notch problem on a sequence of meshes with convex clouds. The resulting GFEM shape functions are  $C^\infty$  in this case.

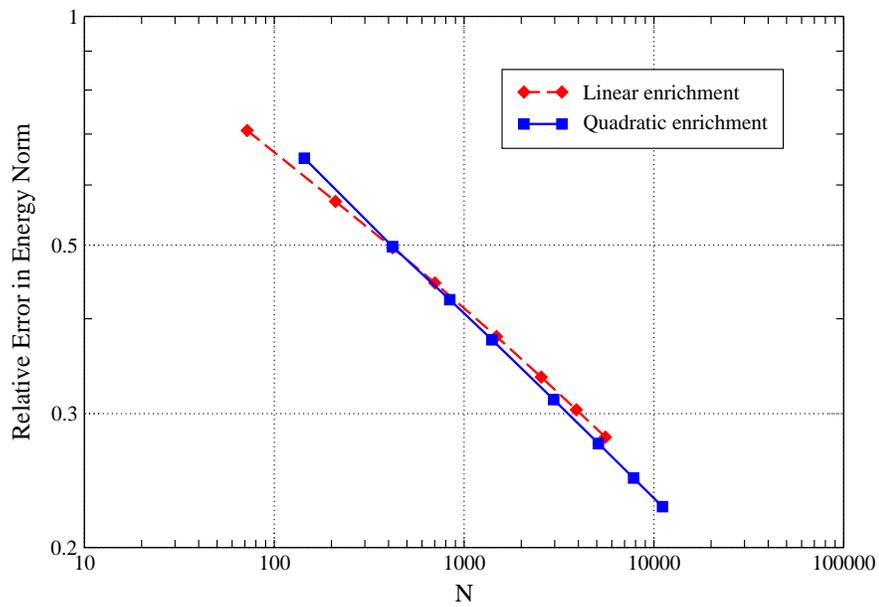


Fig. 14.  $H$ -convergence for the single edge-notch problem on a sequence of meshes with non-convex clouds. The proposed approach based on the concept of R-functions is used to handle the non-convex supports.

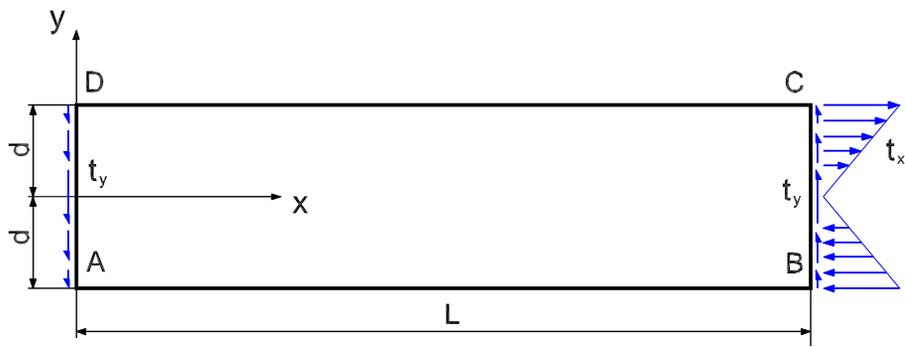


Fig. 15. Beam-like problem.

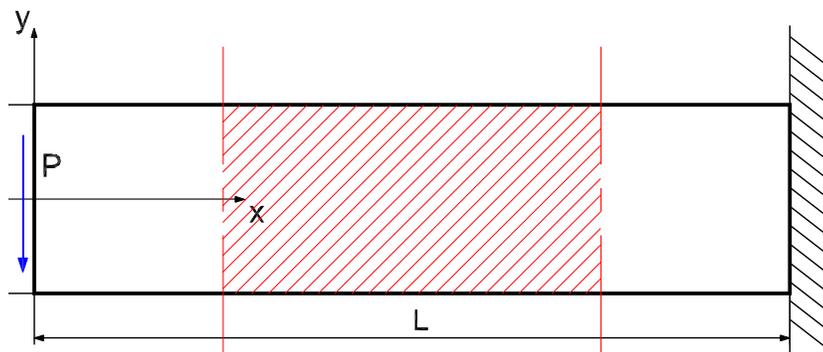
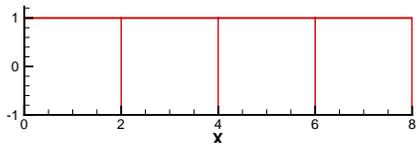
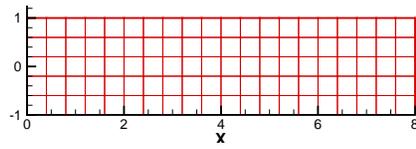


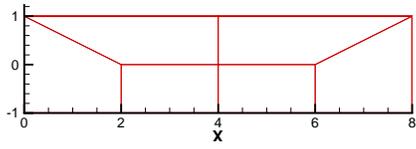
Fig. 16. Cantilever beam problem analogous to the problem of Figure 15 (in the shaded area, far from the support and tip of the beam).



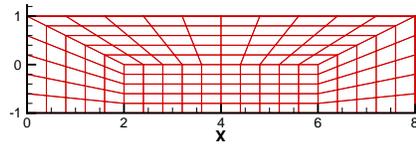
(a)  $h = h_0$ .



(b)  $h = h_0/5$ .



(c)  $h = h_0$ .



(d)  $h = h_0/5$ .

Fig. 17. (a),(b) Meshes with convex clouds only. (c), (d) Meshes with non-convex clouds.

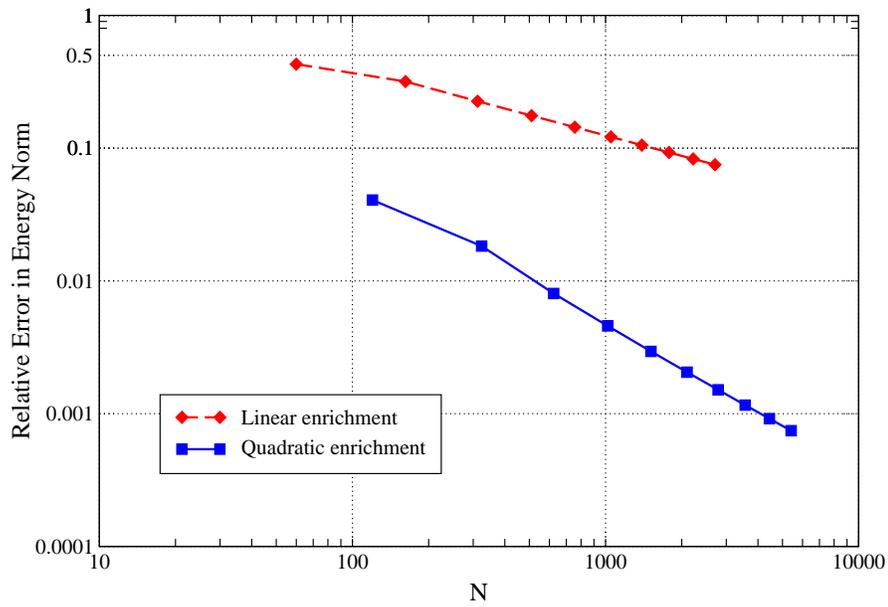


Fig. 18.  $H$ -convergence for the beam-like problem on a sequence of meshes with convex clouds. The resulting GFEM shape functions are  $C^\infty$  in this case.

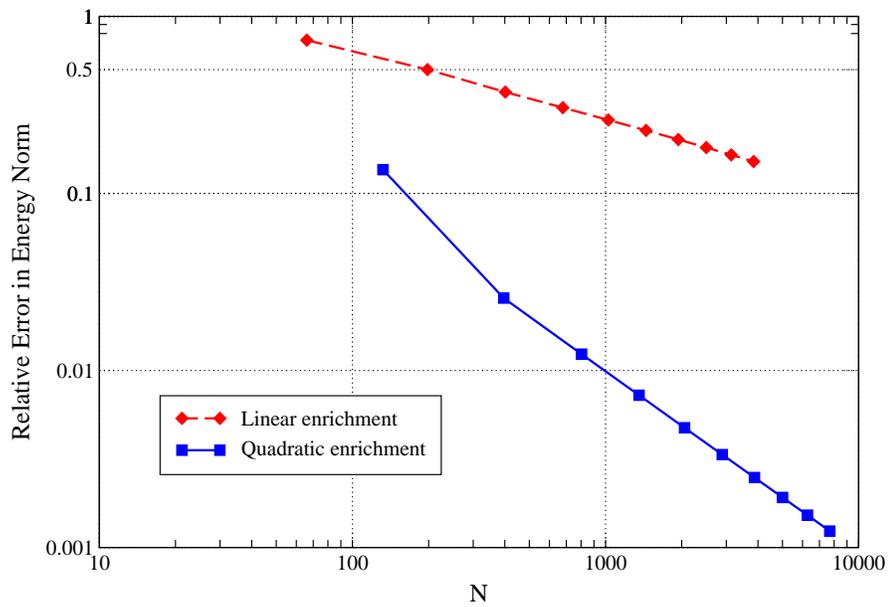


Fig. 19.  $H$ -convergence for the beam-like problem on a sequence of meshes with non-convex clouds. The proposed approach based on the concept of R-functions is used to handle the non-convex supports.

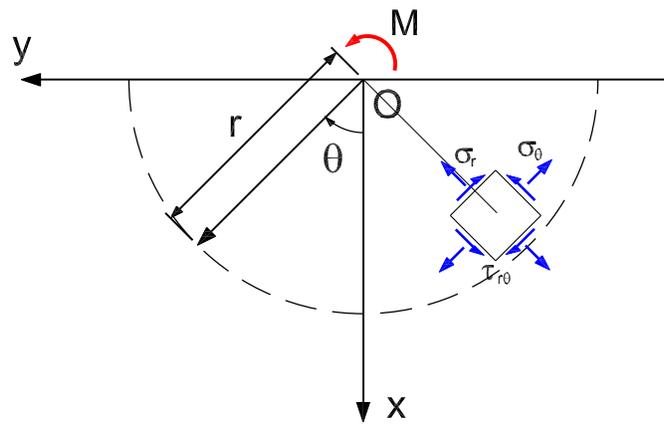


Fig. 20. Concentrated bending moment at a point of a straight boundary.

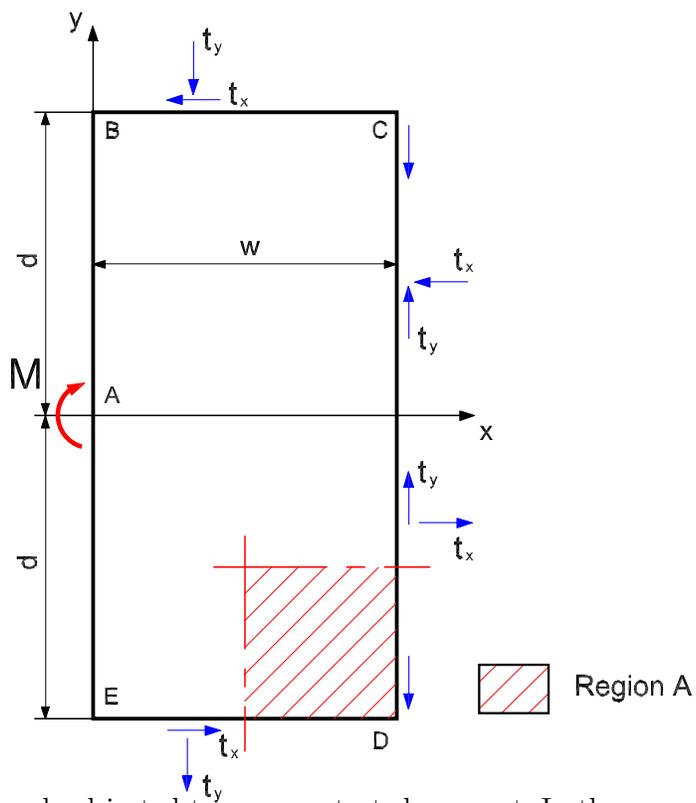


Fig. 21. Panel subjected to a concentrated moment. In the computations, only the lower half  $\Omega_L = \{(x, y) \in \mathbb{R}^2 : 0 < x < w, -d < y < 0\}$  of the domain is discretized. Anti-symmetry boundary conditions  $u_x = 0, t_y = 0$  are applied at  $y = 0$ .

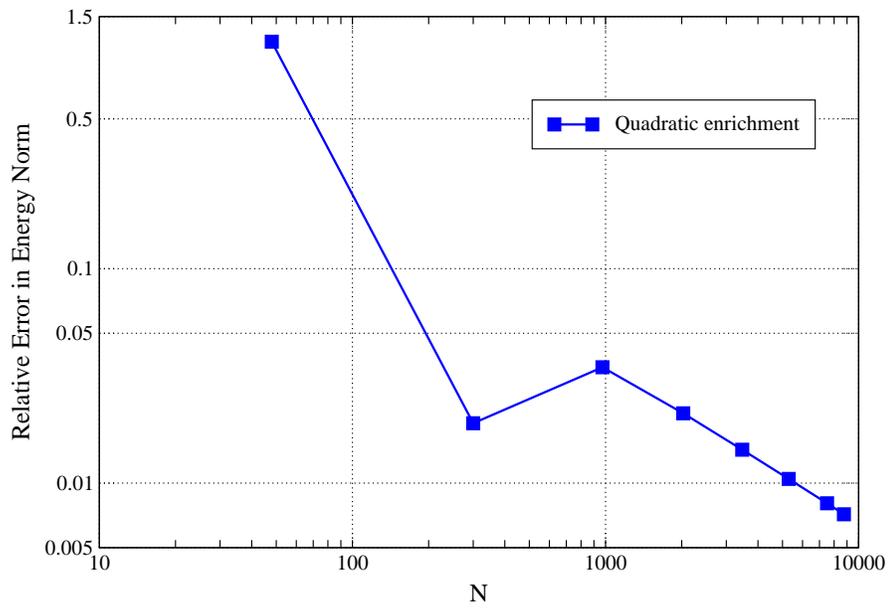
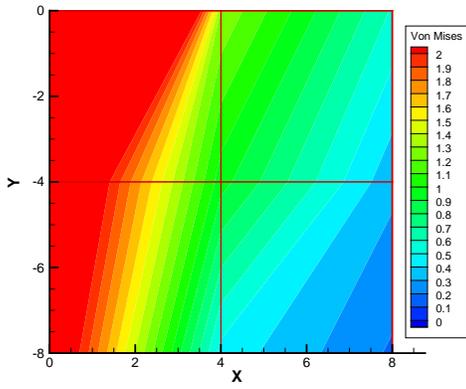
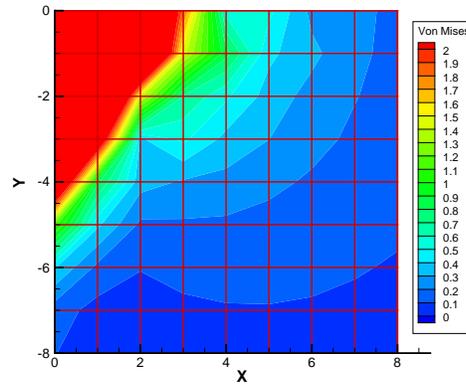


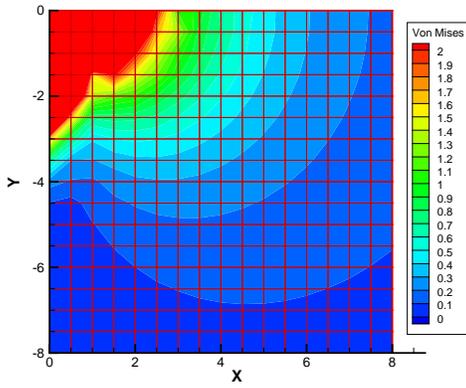
Fig. 22. The performance of  $h$ -extension on the concentrated moment problem ( $C^\infty$  GFEM).



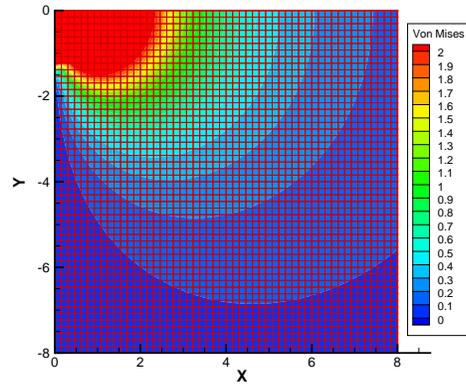
(a) .



(b) .



(c) .



(d) .

Fig. 23. Von-Mises stress contour on domain  $\Omega_L$  ( $C^\infty$  GFEM).

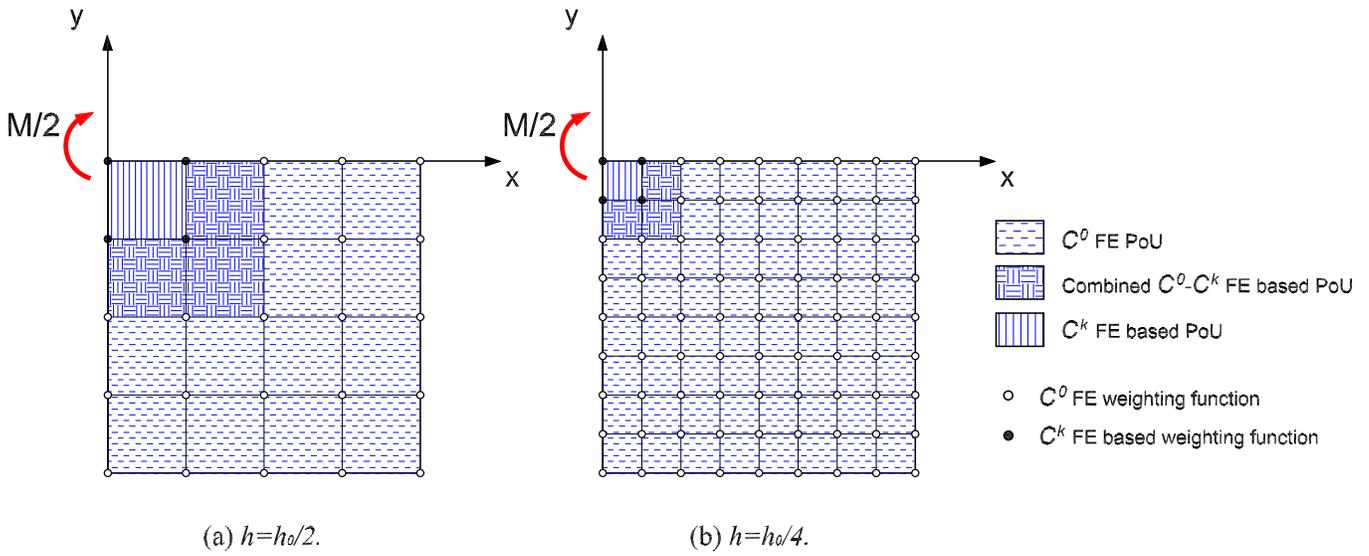


Fig. 24. Discretizations combining  $C^0$  and  $C^\infty$  FE-based PoUs for the concentrated moment problem.

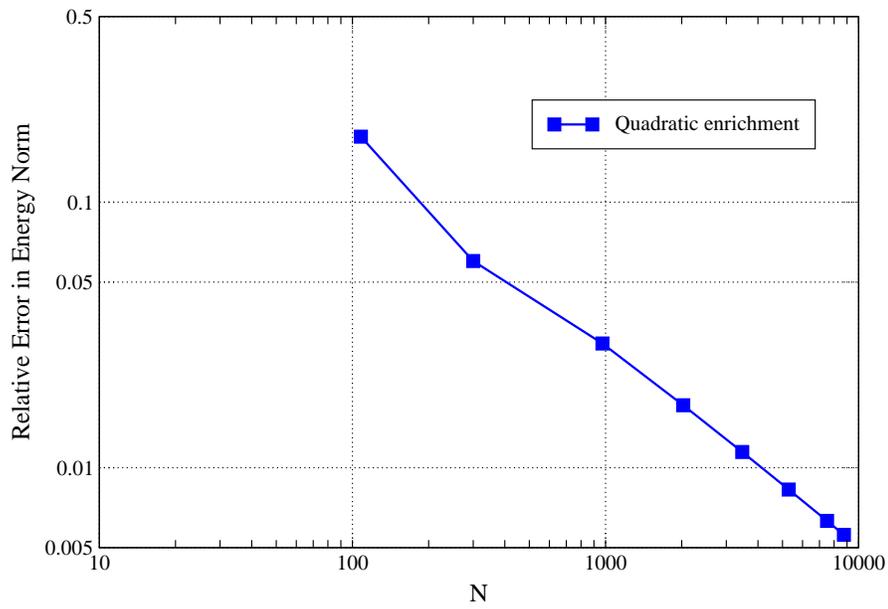
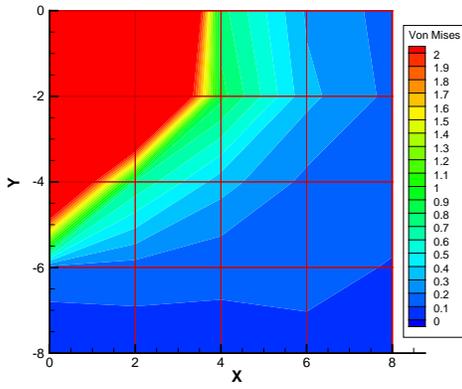
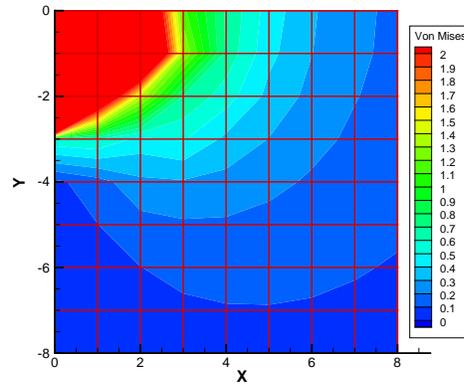


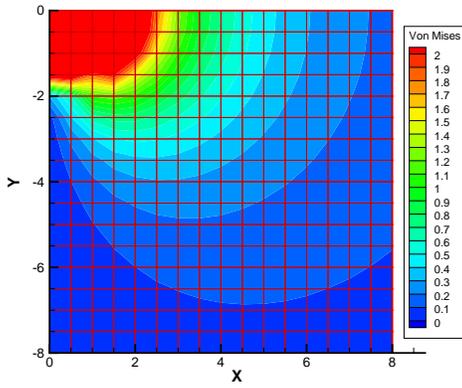
Fig. 25. The performance of  $h$ -extension on the concentrated moment problem (Combined PoUs).



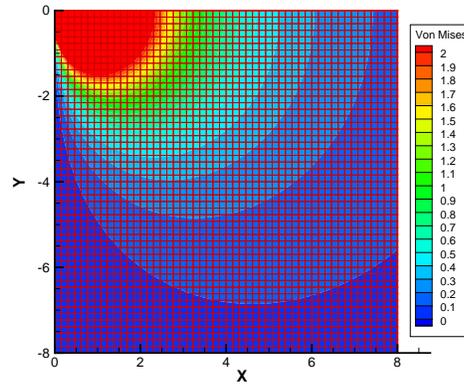
(a)  $h = h_0/2$ .



(b)  $h = h_0/4$ .



(c)  $h = h_0/8$ .



(d)  $h = h_0/26$ .

Fig. 26. Von-Mises stress contour on domain  $\Omega_L$  (combined PoUs).

## TABLES

Table 1

$H$ -convergence for the single edge-notch problem on a sequence of meshes with convex clouds. Here,  $N$  is the number of degrees of freedom,  $U_h$  is the computed strain energy and  $\beta_h$  the rate of convergence with respect to  $N$ .

No. of nodes	Linear enrichment			Quadratic enrichment		
	$N$	$U_h$	$\beta_h$	$N$	$U_h$	$\beta_h$
16	96	4.5126648		192	5.5156362	
47	282	5.9779874	0.1740040	564	6.9178478	0.2223645
94	564	6.7532853	0.1986614	1128	7.5132489	0.2182363
157	942	7.2357070	0.2142114	1884	7.8731232	0.2341456
331	1986	7.7974465	0.2260002	3972	8.2751758	0.2423046
569	3414	8.1116503	0.2343572	6828	8.4908804	0.2455494
871	5226	8.3115249	0.2386456	10452	8.6249768	0.2467264
$\infty$	$\infty$	9.198545583	0.25	$\infty$	9.198545583	0.25

Table 2

$H$ -convergence for the single edge-notch problem on a sequence of meshes with non-convex clouds. Here,  $N$  is the number of degrees of freedom,  $U_h$  is the computed strain energy and  $\beta_h$  the rate of convergence with respect to  $N$ .

No. of nodes	Linear enrichment			Quadratic enrichment		
	$N$	$U_h$	$\beta_h$	$N$	$U_h$	$\beta_h$
12	72	4.5916574		144	5.3041828	
35	210	6.2022707	0.2009371	420	6.9192913	0.2502154
70	420	6.9354983	0.2024516	840	7.5464177	0.2321183
117	702	7.3695487	0.2072741	1404	7.9033429	0.2369155
247	1482	7.8765075	0.2172021	2964	8.2967956	0.2422900
425	2550	8.1649671	0.2267801	5100	8.5072965	0.2449205
651	3906	8.3508900	0.2325263	7812	8.6381018	0.2459677
925	5550	8.4804900	0.2361727	11100	8.7272551	0.2466014
$\infty$	$\infty$	9.198545583	0.25	$\infty$	9.198545583	0.25

Table 3

$H$ -convergence for the beam-like problem on a sequence of meshes with convex clouds. Here,  $N$  is the number of degrees of freedom,  $U_h$  is the computed strain energy and  $\beta_h$  the rate of convergence with respect to  $N$ .

No. of nodes	Linear enrichment			Quadratic enrichment		
	$N$	$U_h$	$\beta_h$	$N$	$U_h$	$\beta_h$
10	60	109.5041701		120	134.0183082	
27	162	120.7009461	0.3033845	324	134.1950813	0.8036406
52	312	127.4188425	0.5229951	624	134.2313316	1.2550778
85	510	130.0964492	0.5071921	1020	134.2371822	1.1433572
126	756	131.4671410	0.5102171	1512	134.2388346	1.1214568
175	1050	132.2568467	0.5101774	2100	134.2394338	1.0987603
232	1392	132.7524721	0.5099683	2784	134.2396929	1.0847807
297	1782	133.0835100	0.5099359	3564	134.2398193	1.0869617
370	2220	133.3154015	0.5095943	4440	134.2398869	1.0755088
451	2706	133.4840625	0.5091633	5412	134.2399256	1.0665487
$\infty$	$\infty$	134.24	0.5	$\infty$	134.24	1.0

Table 4

$H$ -convergence for the beam-like problem on a sequence of meshes with non-convex clouds. Here,  $N$  is the number of degrees of freedom,  $U_h$  is the computed strain energy and  $\beta_h$  the rate of convergence with respect to  $N$ .

No. of nodes	Linear enrichment			Quadratic enrichment		
	$N$	$U_h$	$\beta_h$	$N$	$U_h$	$\beta_h$
11	66	61.8589432		32	131.7506602	
33	198	100.5671200	0.3482811	396	134.1518807	1.5205918
67	402	115.4896466	0.4133666	804	134.2194901	1.0292376
113	678	121.7554098	0.3890580	1356	134.2329543	1.0221019
171	1026	125.1845749	0.3875813	2052	134.2369752	1.0205396
241	1446	127.3294940	0.3939010	2892	134.2384967	1.0187778
323	1938	128.7808280	0.4024951	3876	134.2391719	1.0180741
417	2502	129.8151611	0.4016782	5004	134.2395074	1.0180139
523	3138	130.5804174	0.4102391	6276	134.2396891	1.0170659
641	3846	131.1631527	0.4183038	7692	134.2397943	1.0160395
$\infty$	$\infty$	134.24	0.5	$\infty$	134.24	1.0

Table 5  
*H*-Convergence in region A for the concentrated moment problem ( $C^\infty$  GFEM).

Region A			Domain $\Omega_L$	
$N$	$U_h$	$\beta_h$	$N$	$U_h$
48	0.277022351		108	
300	0.119865956	2.235908378	972	986.842185868
972	0.119764824	-0.511894009	3468	3948.254099930
2028	0.119855560	0.672145925	7500	8883.941155530
3468	0.119884740	0.729206484	13068	15793.903014900
5292	0.119896211	0.743467288	20172	24678.139692900
7500	0.119901547	0.748913729	28812	35536.651122600
8748	0.119903185	0.754807257	33708	41706.259873800
$\infty$	0.11990932575		$\infty$	$\infty$

Table 6  
*H*-Convergence in region A for the concentrated moment problem (Combined PoUs).

Region A			Domain $\Omega_L$	
$N$	$U_h$	$\beta_h$	$N$	$U_h$
108	0.123646142		300	
300	0.119477003	1.055554091	972	1018.370650450
972	0.119805893	0.608319918	3468	4074.369185970
2028	0.119873878	0.728030581	7500	9167.699979350
3468	0.119893581	0.756297812	13068	16298.362337700
5292	0.119901108	0.769227068	20172	25466.353943800
7500	0.119904556	0.780100119	28812	36671.668575800
8748	0.119905582	0.787028137	33708	43038.318242500
$\infty$	0.11990932575		$\infty$	$\infty$

Table 7

Study of the effect of integration errors on the computed strain energy. The mesh of Figure 12(b) which has only convex clouds is used. The GFEM shape functions built on this mesh are  $C^\infty$ . The reference values for the strain energy were computed using  $50 \times 50$  integration points in each element.

No. of IPs	Linear enrichment		Quadratic enrichment	
	$U_h$	Relative error(%)	$U_h$	Relative error(%)
$3 \times 3$	7.6149462	5.241	8.0695748E+09	
$4 \times 4$	7.1341657	1.403	7.8688713	0.0540
$5 \times 5$	7.2359236	0.003	7.8649924	0.1033
$6 \times 6$	7.2590545	0.323	7.9581462	1.0799
$7 \times 7$	7.2314774	0.058	7.8679146	0.0662
$8 \times 8$	7.2328343	0.040	7.8619269	0.1422
$9 \times 9$	7.2364752	0.011	7.8737481	0.0079
$10 \times 10$	7.2360834	0.005	7.8751317	0.0255
$11 \times 11$	7.2356665	0.001	7.8735128	0.0049
$12 \times 12$	7.2356558	0.001	7.8729874	0.0017
$13 \times 13$	7.2356897	0.000	7.8730010	0.0016
$14 \times 14$			7.8730820	0.0005
$15 \times 15$			7.8731246	0.0000
Ref. Value	7.2357070		7.8731232	

Table 8

Study of the effect of integration errors on the computed strain energy. The mesh of Figure 12(d) which has non-convex clouds is used. The GFEM shape functions built on this mesh are  $C^{k=2}$  or  $C^\infty$ . The reference values for the strain energy were computed using  $50 \times 50$  integration points in each element.

No. of IPs	Linear enrichment		Quadratic enrichment	
	$U_h$	Relative error(%)	$U_h$	Relative error(%)
$3 \times 3$	9.3854834	27.355	2.4851409E+01	
$4 \times 4$	7.3767360	0.098	8.1171298	2.7050
$5 \times 5$	7.3670477	0.034	8.0248179	1.5370
$6 \times 6$	7.4027772	0.451	7.9652831	0.7837
$7 \times 7$	7.3750666	0.075	7.9005689	0.0351
$8 \times 8$	7.3661904	0.046	7.8968490	0.0822
$9 \times 9$	7.3682904	0.017	7.9027909	0.0070
$10 \times 10$	7.3709942	0.020	7.9054040	0.0261
$11 \times 11$	7.3698800	0.004	7.9036814	0.0043
$12 \times 12$	7.3690040	0.007	7.9032081	0.0017
$13 \times 13$	7.3696685	0.002	7.9033068	0.0005
$14 \times 14$	7.3695830	0.000	7.9032965	0.0006
$15 \times 15$			7.9033457	0.0000
Ref. Value	7.3695487		7.9033429	