The International Conference on Computational Methods
December 15–17, 2004, Singapore

PARTICLE–PARTITION OF UNITY METHODS IN ELASTICITY

Michael Griebel and Marc Alexander Schweitzer
Institut für Numerische Simulation, Universität Bonn,
Wegelerstrasse 6, D-53115 Bonn, Germany
griebel@iam.uni-bonn.de, m.a.schweitzer@ins.uni-bonn.de

Abstract

We consider discretizations of problems in elasticity using the particle–partition of unity method (PUM). We focus on discretization issues and fast solution techniques. Numerical results for applications in two and three dimensions also for obstacle problems are presented.

Key words: meshfree method, partition of unity, multilevel solvers, Nitsche's method

1. Introduction

The particle–partition of unity method (PUM) [1, 2, 3, 4, 5, 8] is a meshfree Galerkin method for the numerical treatment of partial differential equations (PDE). In essence, it is a generalized finite element method (GFEM) which employs piecewise rational shape functions rather than piecewise polynomial functions. The PUM shape functions, however, make up a basis of the discrete function space unlike other GFEM approaches which allows us to construct fast multilevel solvers in a similar fashion as in the finite element method (FEM).

The paper is organized as follows: In section 2 we shortly review the construction of PUM spaces, the Galerkin discretization of a linear elliptic PDE using our PUM as well as the fast multilevel solution of the arising linear system. Then we present some numerical results with respect to approximation as well as fast solution techniques in two and three space dimensions obtained with our PUM for the numerical solution of the Navier–Lamé equations in section 2.4. The discretization of constrained minimization problems like the obstacle problem is the subject of section 3. Then, some numerical results for the obstacle problem in two space dimensions are given in section 3.2. Finally, we conclude with some remarks.

2. Partition of Unity Method

In the following, we shortly review the construction partition of unity spaces and the meshfree Galerkin discretization of an elliptic PDE, see [1, 2] for details. Furthermore, we give a summary of the efficient multilevel solution of the arising linear block-system, see [3] for details.

2.1. Construction of Partition of Unity Spaces

In a partition of unity method, we define a global approximation u^{PU} simply as a weighted sum of local approximations u_i,

$$u^{\mathrm{PU}}(x) := \sum_{i=1}^{N} \varphi_i(x) u_i(x). \tag{2.1}$$

These local approximations u_i are completely independent of each other, i.e., the local supports $\omega_i := \mathrm{supp}(u_i)$, the local basis $\{\psi_i^n\}$ and the order of approximation p_i for every single $u_i := \sum u_i^n \psi_i^n \in V_i^{p_i}$ can be chosen independently of all other u_j. Here, the functions φ_i form a partition of unity (PU). They are used to splice the local approximations u_i together in such a way that the global approximation u^{PU} benefits from the local approximation orders p_i yet it still fulfills global regularity conditions. Hence, the global approximation space on Ω is defined as

$$V^{\mathrm{PU}} := \sum_i \varphi_i V_i^{p_i} = \sum_i \varphi_i \operatorname{span}\langle\{\psi_i^n\}\rangle = \operatorname{span}\langle\{\varphi_i \psi_i^n\}\rangle. \tag{2.2}$$

The starting point for any meshfree method is a collection of N independent points $P := \{x_i \in \mathbb{R}^d \,|\, x_i \in \overline{\Omega}, i = 1, \ldots, N\}$. In the PU approach we need to construct a partition of unity $\{\varphi_i\}$ on the domain of interest Ω to define an approximate solution (2.1) where the union of the supports $\operatorname{supp}(\varphi_i) = \overline{\omega_i}$ covers the domain $\overline{\Omega} \subset \bigcup_{i=1}^N \omega_i$ and $u_i \in V_i^{p_i}(\omega_i)$ is some locally defined approximation of order p_i to u on ω_i. Thus, the first (and most crucial) step in a PUM is the efficient construction of an appropriate cover $C_\Omega := \{\omega_i\}$. Throughout this paper we use a tree-based construction algorithm for d-rectangular covers C_Ω presented in [2, 8]. Here, the cover patches ω_i are products of intervals $(x_i^l - h_i^l, x_i^l + h_i^l)$ for $l = 1, \ldots, d$. With the help of weight functions W_k defined on these cover patches ω_k we can easily generate a partition of unity by *Shepard's method*, i.e., we define

$$\varphi_i(x) = \frac{W_i(x)}{\sum_{\omega_k \in C_\Omega^i} W_k(x)}, \tag{2.3}$$

where $C_i := \{\omega_j \in C_\Omega \,|\, \omega_i \cap \omega_j \neq \emptyset\}$ is the set of all geometric neighbors of a cover patch ω_i. Due to the use of d-rectangular patches ω_i, the most natural choice for a weight function W_i is a product of one-dimensional functions, i.e., $W_i\,(x) = \prod_{l=1}^d W_i^l\,(x^l) = \prod_{l=1}^d \mathcal{W}\,(\frac{x - x_i^l + h_i^l}{2h_i^l})$ with $\operatorname{supp}(\mathcal{W}) = [0, 1]$ such that $\operatorname{supp}(W_i) = \overline{\omega_i}$. It is sufficient for this construction to choose a one-dimensional weight function $\mathcal{W}$ with the desired regularity which is non-negative. The partition of unity functions φ_i inherit the regularity of the generating weight function $\mathcal{W}$. Throughout this paper we use a linear B-spline as the generating weight function $\mathcal{W}$.

In general, a partition of unity $\{\varphi_i\}$ can only recover the constant function on the domain Ω. Hence, we need to improve the approximation quality to use the method for the discretization of a PDE. To this end, we multiply the partition of unity functions φ_i locally with polynomials ψ_i^n. Since we use d-rectangular patches ω_i only, a local tensor product space is the most natural choice. Here, we use products of univariate Legendre polynomials as local approximation spaces $V_i^{p_i}$, i.e., we choose

$$V_i^{p_i} = \operatorname{span}\langle\{\psi_i^n \,|\, \psi_i^n = \prod_{l=1}^d \mathcal{L}_i^{\hat{n}_l}, \|\hat{n}\|_1 = \sum_{l=1}^d \hat{n}_l \leq p_i\}\rangle,$$

where $\hat{n}$ is the multi-index of the polynomial degrees $\hat{n}_l$ of the univariate Legendre polynomials $\mathcal{L}_i^{\hat{n}_l} : [x_i^l - h_i^l, x_i^l + h_i^l] \to \mathbb{R}$, and n is the index associated with the product function $\psi_i^n = \prod_{l=1}^d \mathcal{L}_i^{\hat{n}_l}$.

For the approximation of vector-fields we employ vector-valued shape functions $\varphi_i \psi_i^n$; i.e., we simply change the definition of our local approximation spaces $V_i^{p_i} = \operatorname{span}\langle\vec{\psi}_i^n\rangle$ but keep the partition of unity functions φ. To this end, we choose the local vector-valued basis functions

$$\vec{\psi}_i^n := \vec{\psi}_i^{\tilde{n},l} := \psi_i^{\tilde{n}} \vec{e}_l$$

where we simply multiply the scalar functions $\psi_i^{\tilde{n}}$ with an appropriate unit vector $\vec{e}_l$. In the following we will drop the explicit vector notation and use the symbol ψ_i^n also for vector-valued functions.

In summary, we can view the construction given above as follows

$$\begin{pmatrix} \{x_i\} \\ \mathcal{W} \\ \{p_i\} \end{pmatrix} \rightarrow \begin{pmatrix} \{\omega_i\} \\ \{W_i\} \\ \{V_i^{p_i} = \operatorname{span}\langle \psi_i^n \rangle\} \end{pmatrix} \rightarrow \begin{pmatrix} \{\varphi_i\} \\ \{V_i^{p_i}\} \end{pmatrix} \rightarrow V^{\mathrm{PU}} = \sum \varphi_i V_i^{p_i},$$

where the set of points $P = \{x_i\}$, the generating weight function $\mathcal{W}$ and the local approximation orders p_i are assumed to be given.

2.2. Galerkin Discretization

Consider the elliptic boundary value problem

$$Lu = f \quad \text{in } \Omega \subset \mathbb{R}^d , \qquad Bu = g \quad \text{on } \partial\Omega , \tag{2.4}$$

where L is a symmetric partial differential operator of second order and B expresses suitable boundary conditions. The imposition of essential boundary conditions within meshfree methods is more involved than in the FEM for a number of reasons and many different approaches have been proposed. We use Nitsche's method [7] to enforce Dirichlet boundary conditions. The main advantages of this approach are that it does not require a second function (or multiplier) space and that it leads to a positive definite linear system, see [5, 8] for a more detailed discussion of Nitsche's method in the PUM context. Here, we just state resulting weak formulation $a(u, v) = l(v)$ of the simple Poisson problem

$$\begin{aligned} -\Delta u &= f && \text{in } \Omega \subset \mathbb{R}^d, \\ u &= g_D && \text{on } \Gamma_D \subset \partial\Omega, \\ u_n &= g_N && \text{on } \Gamma_N = \partial\Omega \setminus \Gamma_D, \end{aligned}$$

with mixed boundary conditions which reads as

$$\int_\Omega \nabla u \nabla v + \int_{\Gamma_D} u(\beta v - v_n) - u_n v = \int_\Omega f v + \int_{\Gamma_D} g_D(\beta v - v_n) + \int_{\Gamma_N} g_N v, \tag{2.5}$$

where the subscript n denotes the normal derivative and β is the Nitsche regularization parameter which depends on the employed PUM space but can be pre-computed without much additional cost. Finally, for the Galerkin discretization of (2.5) we have to compute the stiffness matrix

$$A = \left(A_{(i,n),(j,m)}\right), \text{ with } A_{(i,n),(j,m)} = a\left(\varphi_j \psi_j^m, \varphi_i \psi_i^n\right) \in \mathbb{R} ,$$

and the right-hand side vector

$$\hat{f} = \left(f_{(i,n)}\right), \text{ with } f_{(i,n)} = \langle f, \varphi_i \psi_i^n \rangle_{L^2} = \int_\Omega f \varphi_i \psi_i^n \in \mathbb{R} .$$

The stable approximation of these integrals is somewhat more involved than in the finite element method (FEM). Due to the meshfree construction given above the shape functions $\varphi_i \psi_i^n$ are piecewise rational functions only so that the respective integrands have a number of jumps within the integration domain which need to the resolved. For the stable numerical integration of the weak form we use a tree-based decomposition scheme together with efficient sparse grid integration rules, see [2, 8].

2.3. Multilevel Solution of Resulting Linear System

The product structure of the shape functions $\varphi_i \psi_i^n$ implies two natural block-partitions of the resulting linear system $A\tilde{u} = \hat{f}$, where $\tilde{u}$ denotes a coefficient vector and $\hat{f}$ denotes a moment vector.

1. The stiffness matrix A can be arranged in *spatial blocks*. A spatial block A_{nm} corresponds to a discretization of the PDE on the complete domain Ω using the trial functions $\varphi_j \psi_j^m$ and the test function $\varphi_i \psi_i^n$ with fixed n and m. Here, all blocks A_{nm} are sparse matrices and have the same row and column dimensions which corresponds to the number of partition of unity functions φ_i.

2. The stiffness matrix A may also be arranged in *polynomial blocks*. Here, a single block A_{ij} corresponds to a local discretization of the PDE on the domain $\omega_i \cap \omega_j \cap \Omega$. The polynomial blocks A_{ij} are dense matrices and may have different dimensions corresponding to the dimensions of the local approximation spaces $V_j^{p_j}$ and $V_i^{p_i}$.

This separation of the degrees of freedom into local approximation functions ψ_i^n and partition of unity functions φ_i can be used to define two different multilevel concepts [3]. Throughout this paper we assume that the stiffness matrix is given in polynomial block-form and we use the corresponding spatial multilevel solver developed in [3] for the fast and efficient solution of the resulting large sparse linear block-system $A\tilde{u} = \hat{f}$, where $\tilde{u}$ denotes a coefficient block-vector and $\hat{f}$ a moment block-vector.
In a multilevel method we need a sequence of discretization spaces V_k with $k = 0, \ldots, J$ where J denotes the finest level. To this end we construct a sequence of PUM spaces V_k^{PU} with the help of a tree-based algorithm developed in [2, 3]. As a first step we generate a sequence of point sets P_k and covers C_Ω^k from a given initial point set $\tilde{P}$ with this algorithm, see Figure 2.1. Following the construction given in §2.1 we can then define an associated sequence of PUM spaces V_k^{PU}. Note that these spaces are nonnested, i.e., $V_{k-1}^{\mathrm{PU}} \not\subset V_k^{\mathrm{PU}}$, and that the shape functions $\varphi_{i,k}\psi_{i,k}^n$ are non-interpolatory. Thus, we need to construct appropriate transfer operators $I_{k-1}^k : V_{k-1}^{\mathrm{PU}} \to V_k^{\mathrm{PU}}$ and $I_k^{k-1} : V_k^{\mathrm{PU}} \to V_{k-1}^{\mathrm{PU}}$. With such transfer operators I_{k-1}^k, I_k^{k-1} and the stiffness matrices A_k coming from the Galerkin discretization on each level k we can then set up a standard multiplicative multilevel iteration to solve the linear system $A_J \tilde{u}_J = \hat{f}_J$. Our multilevel solver utilizes special localized L^2-projections for the interlevel transfers and a block-smoother to treat all local degrees of freedom ψ_i^n within a patch ω_i simultaneously. Namely, we use the so-called local-to-local L^2-projections as prolongation operators I_k^{k-1} for scalar as well as vector-valued problems. For further details see [3, 8].

2.4. Numerical Results for Linear Elasticity

In the following we consider the numerical solution of the Navier–Lamé equations

$$-\mu \boldsymbol{\Delta} u - (\lambda + \mu)\nabla(\nabla \cdot u) = f \quad \text{in} \quad \Omega \subset \mathbb{R}^d, \quad d = 2, 3$$

together with suitable boundary conditions $u_D = g_D$ on $\Gamma_D \subset \partial\Omega$ and $\boldsymbol{\sigma}(u) \cdot n = g_N$ on $\Gamma_N = \partial\Omega \setminus \Gamma_D$ where $\boldsymbol{\sigma}(u) := \lambda \nabla \cdot u I + 2\mu\, \boldsymbol{\epsilon}(u)$ denotes the symmetric stress tensor and $\boldsymbol{\epsilon}(u) := \frac{1}{2}(\partial_i u_j + \partial_j u_i)$ the strain tensor associated with the displacement field $u = (u_i)$, $i = 1, \ldots, d$. The parameters λ and μ are the so-called Lamé parameters. They are related to the Poisson ratio ν and the Young modulus E of the material via $\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$ and

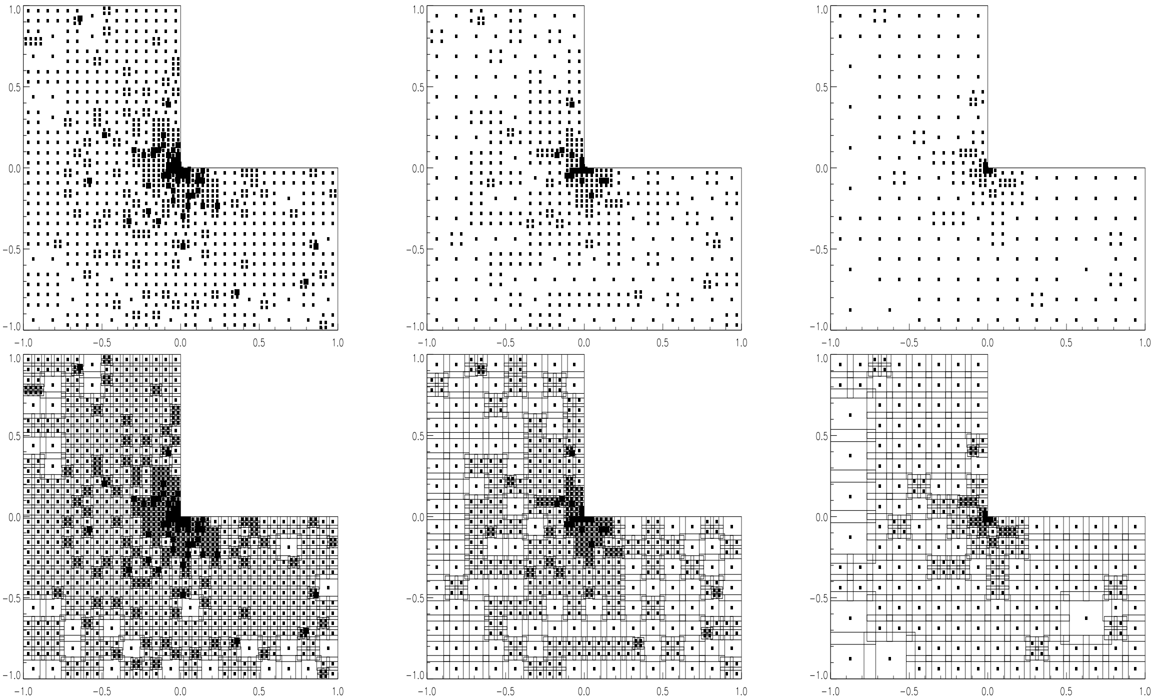

FIG. 2.1. *Point sets P_k and covers C_Ω^k for $k = 10, \ldots, 8$ generated for an initial graded Halton$(2,3)$ point set $\tilde{P}$ with $\tilde{N} = 678$ points. The number N of generated points on the finest level $J = 10$ is $N = 1293$.*

$\mu = \frac{E}{2(1+\nu)}$. The associated bilinear form arising from Nitsche's approach is given by

$$a(u,v) = \int_\Omega \boldsymbol{\sigma}(u) : \boldsymbol{\epsilon}(v) + \int_{\Gamma_D} 2\mu\beta_\epsilon u \cdot v + \lambda\beta_{\mathrm{div}}(u \cdot n)(v \cdot n) - \big((\boldsymbol{\sigma}(u) \cdot n) \cdot v + u \cdot (\boldsymbol{\sigma}(v) \cdot n)\big)$$

and the linear form on the right-hand side by

$$l(v) = \int_\Omega f \cdot v + \int_{\Gamma_N} g_N \cdot v + \int_{\Gamma_D} 2\mu\beta_\epsilon g_D \cdot v + \lambda\beta_{\mathrm{div}}(g_D \cdot n)(v \cdot n) - g_D \cdot (\boldsymbol{\sigma}(v) \cdot n).$$

We measure the convergence of our multilevel solver via a simple test problem, where we use a sequence of uniform point sets in the domain $\Omega = [-1,1]^d$ for $d = 2,3$. In this example we use the parameters $E = 1$ and $\nu = \frac{1}{3}$ for the material and the boundary conditions $u = 0$ on $\Gamma_D := \{x \in \partial\Omega : x_0 = -1\}$, $\boldsymbol{\sigma}(u) \cdot n = 0$ on $\Gamma_{N_1} := \{x \in \partial\Omega : x_1 = -1 \text{ or } x_1 = 1\}$, and $\boldsymbol{\sigma}(u) \cdot n = (0,-1)^T$. In two dimensions the finest discretization is based on $N = 1048576$ points and employs dof $= 6291456$ degrees of freedom, in three dimensions the finest discretization uses $N = 262144$ and dof $= 1048576$.

In Table 2.1 we give the measured convergence rates ρ for our multilevel solver using the V- and W-cycle (ρ^1 and ρ^2 respectively) with 2, 3, and 4 smoothing steps, the number of points N on the finest level J, the polynomial degree p and the dimension D_p of the local approximation spaces V_i^p. These rates $\rho := \|u_r\|_{L^2}^{1/r}$ are determined using a vanishing right-hand side and the stopping criteria $\|u_r\|_{L^2} < 10^{-12}$ or $r = 50$ where u_r denotes the current iterate. From these numbers we can observe an optimal level-independent convergence of our solver. The slight fluctuations in the measured rates are due to the parallelization of the block-Gauss–Seidel smoother, see [4, 8] for details on the parallelization.

For the approximation of more complicated geometries we reuse the tree-based cover construction to define so-called domain integration cells, see Figures 2.2 and 2.3. Note, however, that the resolution of the domain is not directly coupled to the cover construction.

TABLE 2.1

Convergence rates $\rho_1^{\nu,\nu}$ for the $V^{\nu,\nu}$-cycle and convergence rates $\rho_2^{\nu,\nu}$ for the $W^{\nu,\nu}$-cycle with $\nu = 2, 3, 4$ in two and three dimensions.

N	J	p	D_p	$\rho_1^{2,2}$	$\rho_1^{3,3}$	$\rho_1^{4,4}$	$\rho_2^{2,2}$	$\rho_2^{3,3}$	$\rho_2^{4,4}$
16384	7	1	3	0.502	0.438	0.388	0.240	0.182	0.150
65536	8	1	3	0.503	0.437	0.389	0.229	0.170	0.141
262144	9	1	3	0.504	0.437	0.389	0.221	0.162	0.136
1048576	10	1	3	0.505	0.437	0.389	0.217	0.158	0.134
512	3	1	4	0.769	0.712	0.678	0.549	0.415	0.398
4096	4	1	4	0.741	0.714	0.677	0.367	0.268	0.222
32768	5	1	4	0.735	0.706	0.675	0.285	0.221	0.195
262144	6	1	4	0.726	0.694	0.671	0.261	0.202	0.175

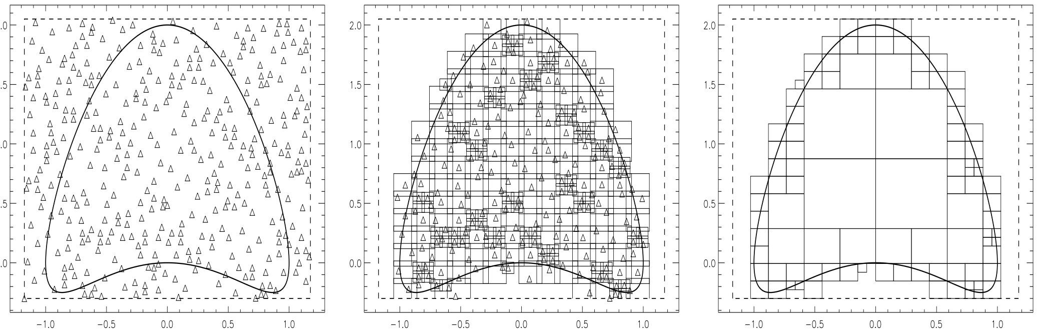

FIG. 2.2. *Domain approximation using the cover tree. Left: Domain and all sampling points. Center: Domain and constructed cover. Right: Domain and respective integration cells.*

Currently, the approximation of the boundary of the domain is given by the boundary of the respective tree-cells. However, a higher order reconstruction of the boundary is straightforward, see e.g. [6].

In summary, these results show that the PUM can be used effectively for the numerical solution of linear elliptic PDE, i.e. for unconstrained minimization problems. Let us now focus on the approximation of constrained minimization problems.

3. Constrained Minimization Problems

A classical example for such a minimization problem with constraints is the Poisson-Obstacle problem

$$\begin{array}{rcll} -\Delta u & \leq & f & \text{on } \Omega, \\ u & = & 0 & \text{on } \partial\Omega, \\ u & \leq & o & \text{on } \Omega, \\ (-\Delta u - f)(u - o) & = & 0 & \text{on } \Omega, \end{array} \tag{3.1}$$

or the more involved Poisson–Signorini problem

$$\begin{array}{rcll} -\Delta u & = & f & \text{on } \Omega, \\ u & = & g_D & \text{on } \Gamma_D \subset \partial\Omega, \\ \dfrac{\partial u}{\partial n} & = & g_N & \text{on } \Gamma_N \subset \partial\Omega, \\ \dfrac{\partial u}{\partial n} \geq 0, \quad u\dfrac{\partial u}{\partial n} \geq 0, \quad u & \geq & 0 & \text{on } \Gamma_C \subset \partial\Omega. \end{array} \tag{3.2}$$

In the obstacle problem (3.1) the constraints are enforced throughout the entire domain Ω, whereas in the more involved Signorini problem the constraints are enforced on a certain part of the boundary Γ_C, the contact boundary, only.

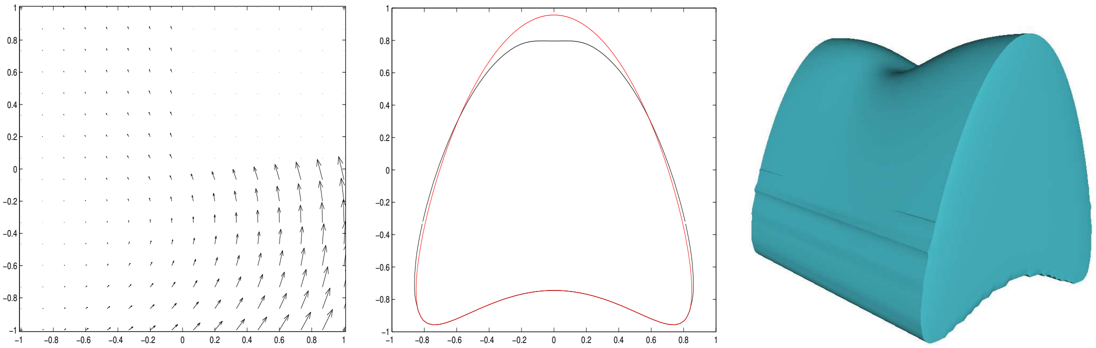

FIG. 2.3. *Displacement field on a L-shaped domain (left), comparison of deformed geometry (red) and original geometry (black) in two dimensions (center), and deformed geometry in three dimensions.*

Let us consider the respective weak formulation of such a constrained problem. To this end, we define the classical energy function

$$\mathcal{J}(u) := \frac{1}{2}a(u,u) - \langle f, u \rangle_{L^2}$$

associated with the underlying PDE problem, i.e. for (3.1) $a(u,v) = \|\nabla u\|_{L^2}^2$ and restrict its minimization to the closed cone

$$K := \{v \in H^1 \,|\, v(x) \leq o(x) \text{ a.e. in } \Omega\}.$$

That is we are now looking for the minimum in a *convex* subset $K \subset H^1$ only, i.e., we try to find $u \in K$ such that $\mathcal{J}(u) \leq \mathcal{J}(v)$ for all $v \in K$ holds. This is unlike in the unconstrained minimization case where we are looking for the minimum in the linear space H^1. The discretization of this cone K of valid functions is the main issue in the numerical treatment of problems like (3.1) and (3.1). For instance, within the FEM pointwise conditions like

$$v(x) \leq o(x) \text{ for almost all } x \in \Omega$$

on the functions $v \in H^1$ are approximated in the vertices x_i of the mesh. Since the linear FEM shape functions are interpolatory, this approximation directly translates into a simple comparison of the coefficient vectors $\tilde{v}_h = (v_i) \in \mathbb{R}^n$ and $\tilde{o}_h = (o_i) \in \mathbb{R}^n$ associated with the discrete function $v_h \in V_h$ and the obstacle $o_h \in V_h$. Hence, in the FEM the cone K is usually discretized as

$$K_h := \{v_h \in V_h \,|\, v_h(x_i) \leq o_h(x_i)\} = \{\tilde{v}_h \in \mathbb{R}^n \,|\, v_i \leq o_i\}.$$

3.1. Partition of Unity Discretization

In the PUM, however, this approach is not valid since the shape functions of the PUM are non-interpolatory. Hence, we cannot directly compare the discrete coefficients of the current solution and the obstacle to determine whether the solution is valid. However, if we employ just linear local spaces $V_i^{p_i=1}$ then we can easily compute the *minimum* and *maximal* function value of the difference $u_i - o_i$ locally on the patch ω_i. Hence, we discretize the closed cone K_{PUM} of valid functions within our PUM via

$$K_{\text{PUM}} := \{v \in V^{\text{PUM}} \,|\, \max_{\omega_i}(v_i - o_i) \leq 0 \quad \text{for all } i\}.$$

Here, we exploit the PU property of the functions φ_i to localize the pointwise conditions $v = \sum \varphi_i v_i \leq \sum_i \varphi_i o_i = o$ on the global shape functions $\varphi_i \psi_i^n$ to the local shape functions ψ_i^n patches ω_i.

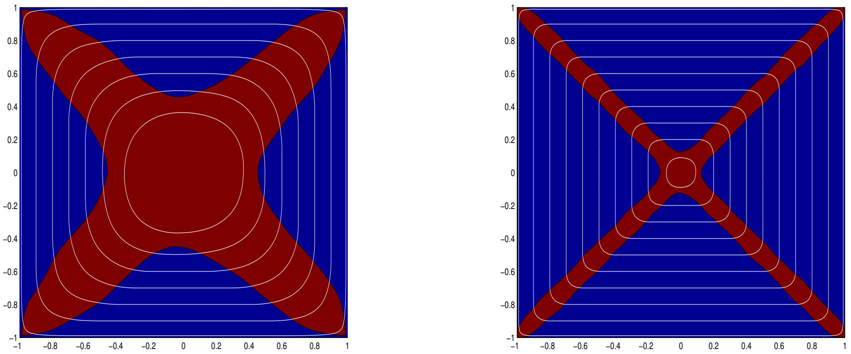

FIG. 3.1. *Active sets for right-hand side* $f = 2.5$ *(left) and* $f = 10$ *(right).*

3.2. Numerical Results for Obstacle Problem

Let us now present some numerical results obtained with our PUM for the obstacle problem (3.1) with $o = \mathrm{dist}_\Omega$ on the unit square $[-1, 1]^2$. From the isoline plots depicted in Figure 3.1 we can observe that we capture the active set, i.e. the part of the domain where the solution actually coincides with the obstacle, very well.

4. Concluding Remarks

In this paper we presented the PUM and its application to unconstrained as well as constrained minimization problems in elasticity. The presented numerical results clearly indicate the applicability of the PUM in this context. An open question at this time is the extension of our multilevel solver to the constrained minimization case, where certain monotony properties are essential. The implementation of these properties within our PUM, however, is not trivial due to the fact that the sequence of PUM function spaces are non-nested and the bilinear form involves level-dependent regularization parameters due to the Nitsche approach for essential boundary conditions.

REFERENCES

[1] M. GRIEBEL AND M. A. SCHWEITZER, *A Particle-Partition of Unity Method for the Solution of Elliptic, Parabolic and Hyperbolic PDE*, SIAM J. Sci. Comput., 22 (2000), pp. 853–890.

[2] ———, *A Particle-Partition of Unity Method—Part II: Efficient Cover Construction and Reliable Integration*, SIAM J. Sci. Comput., 23 (2002), pp. 1655–1682.

[3] ———, *A Particle-Partition of Unity Method—Part III: A Multilevel Solver*, SIAM J. Sci. Comput., 24 (2002), pp. 377–409.

[4] ———, *A Particle-Partition of Unity Method—Part IV: Parallelization*, in Meshfree Methods for Partial Differential Equations, M. Griebel and M. A. Schweitzer, eds., vol. 26 of Lecture Notes in Computational Science and Engineering, Springer, 2002, pp. 161–192.

[5] ———, *A Particle-Partition of Unity Method—Part V: Boundary Conditions*, in Geometric Analysis and Nonlinear Partial Differential Equations, S. Hildebrandt and H. Karcher, eds., Springer, 2002, pp. 517–540.

[6] O. KLAAS AND M. S. SHEPARD, *Automatic Generation of Octree-based Three-Dimensional Discretizations for Partition of Unity Methods*, Comput. Mech., 25 (2000), pp. 296–304.

[7] J. NITSCHE, *Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind*, Abh. Math. Sem. Univ. Hamburg, 36 (1970–1971), pp. 9–15.

[8] M. A. SCHWEITZER, *A Parallel Multilevel Partition of Unity Method for Elliptic Partial Differential Equations*, vol. 29 of Lecture Notes in Computational Science and Engineering, Springer, 2003.