Schweitzer and Ziegenhagel *Adv. Model. and Simul. in Eng. Sci.(2018)5:13*
https://doi.org/10.1186/s40323-018-0107-6

Advanced Modeling and Simulation in Engineering Sciences

RESEARCH ARTICLE Open Access

Multilevel preconditioners for embedded enriched partition of unity approximations

Marc Alexander Schweitzer[1,2] and Albert Ziegenhagel[1*]

*Correspondence:
albert.ziegenhagel@scai.fraunhofer.de
[1] Fraunhofer-Institut für Algorithmen und Wissenschaftliches Rechnen SCAI, Schloss Birlinghoven, 53757 Sankt Augustin, Germany
Full list of author information is available at the end of the article

Abstract

In this paper we are concerned with the non-invasive embedding of enriched partition of unity approximations in classical finite element simulations and the efficient solution of the resulting linear systems. The employed embedding is based on the partition of unity approach introduced in Schweitzer and Ziegenhagel (Embedding enriched partition of unity approximations in finite element simulations. In: Griebel M, Schweitzer MA, editors. Meshfree methods for partial differential equations VIII. Lecture notes in science and engineering, Cham, Springer International Publishing; 195–204, 2017) which is applicable to any finite element implementation and thus allows for a stable enrichment of e.g. commercial finite element software to improve the quality of its approximation properties in a non-invasive fashion. The major remaining challenge is the efficient solution of the arising linear systems. To this end, we apply classical subspace correction techniques to design non-invasive efficient multilevel solvers by blending a non-invasive algebraic multigrid method (applied to the finite element components) with a (geometric) multilevel solver (Griebel and Schweitzer in SIAM J Sci Comput 24:377–409, 2002; Schweitzer in Numer Math 118:307–28, 2011) (applied to the enriched embedded components). We present first numerical results in two and three space dimensions which clearly show the (close to) optimal performance of the proposed solver.

Keywords: Partition of unity method, Generalized finite element method, Multilevel solver, Subspace correction, Domain decompositon

Mathematics Subject Classification: Primary 65N55, 65N30

Introduction

The direct generalization and extension of the classical finite element method (FEM) to allow for the use of arbitrary non-polynomial basis functions as in partition of unity (PU) based approaches like XFEM/GFEM [1–5] usually requires a fair amount of implementational work within the original finite element (FE) code. Thus, the timely evaluation of novel generalizations of the FEM in large-scale industrial applications, which in general rely on commercial software packages, is usually not feasible. This issue can, however, be overcome with the help of the embedding approach presented in [6]. It allows for the non-invasive stable embedding of an arbitrary approximation space V_{ENR} into a classical FE space V_{HOST} and thereby enables the easy evaluation of novel generalizations of the FEM empolying arbitrary approximation functions in an industrial context. In [6] it was demonstrated that the approach is free from artifacts and yields substantial improvements in terms of accuracy. Note that this approach is very different from classical global–local techniques where you consider an independent auxilliary local problem. We, however, blend two function spaces V_{HOST} and V_{ENR} to discretize the global problem directly with a single larger function space V_{BND} which comprises V_{HOST} and V_{ENR}, compare also [7–9].

In this paper we are concerned with the construction of highly efficient solvers and preconditioners for the linear system arising from the discretization of the global problem with this blended function space V_{BND}. To this end, we are concerned with the non-invasive construction of multilevel preconditioners based on subspace correction methods [10] which are also referred to as Schwarz methods e.g. in the domain decomposition context [11]. We construct the coarsening process for V_{BND} with the help of an available (geometric) multilevel structure of V_{ENR} and a multilevel decomposition of V_{HOST} obtained by an algebraic multigrid (AMG) method in a non-invasive fashion. The remainder of this paper is structured as follows: we first quickly introduce the mathematical foundation of our embedding approach, the partition of unity method (PUM), in "A partition of unity method for the embedding of arbitrary approximation spaces in finite element spaces" and summarize the actual embedding procedure. In "Subspace correction methods" we introduce efficient subspace corrections preconditioners for the linear system arising from the discretization of the global problem by our blended function space V_{BND}. The results of our numerical experiments with these preconditioners are presented in "Numerical results" before we conclude with some remarks in "Concluding remarks".

A partition of unity method for the embedding of arbitrary approximation spaces in finite element spaces

The PUM was introduced in [1,12] as a generalization of the FEM and is based on [13]. The abstract ingredients which make up a PUM space

$$V^{\mathrm{PU}} := \sum_{i=1}^{N} \varphi_i V_i = \mathrm{span}\langle \varphi_i \vartheta_i^m \rangle; \tag{1}$$

are a partition of unity (PU) $\{\varphi_i : i = 1, \ldots, N\}$ and a collection of local approximation spaces $V_i := V_i(\omega_i) := \mathrm{span}\langle \vartheta_i^m \rangle_{m=1}^{d_{V_i}}$ defined on the patches $\omega_i := \mathrm{supp}(\varphi_i)$ for $i = 1, \ldots, N$. Thus, the shape functions of a PUM space are simply defined as the products of the PU functions φ_i and the local approximation functions ϑ_i^m. The PU functions provide the locality and global regularity of the product functions $\varphi_i \vartheta_i^m$ whereas the functions ϑ_i^m equip V^{PU} with its approximation power. Note that there are no constraints imposed on the choice of the local spaces V_i, i.e. they are completely independent of each other. Thus, this very local interpretation of the PUM approach allows to utilize local a priori information about the sought solution by using so-called enrichment functions or physics-based basis functions in general [5]. Here, we usually employ local approximation spaces V_i of the form $V_i = \mathcal{P}_i + \mathcal{E}_i$ where $\mathcal{P}_i$ denotes a space of local polynomials and $\mathcal{E}_i$ accounts for non-smooth local features such as kinks, discontinuities and singularities of the solution on the patch ω_i. In our setting, however, we will not follow this local approach but we take a more global point of view which is in spirit closer to a domain decomposition line of thought, see e.g. [11], and can be viewed as a generalization of [7–9], see [6] for details.

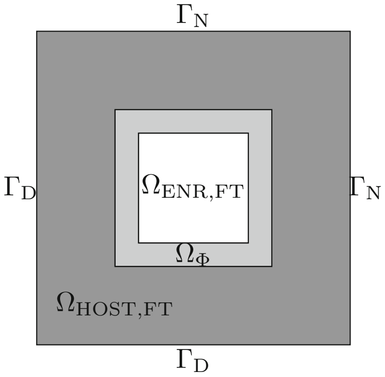

Fig. 1 Sketch of smooth model problem in two dimensions depicting the employed partitioning of the domain Ω into $\Omega_\mathrm{HOST,FT}$ (drak gray), $\Omega_\Phi = \Omega_\mathrm{HOST} \cap \Omega_\mathrm{ENR}$ (light gray), and $\Omega_\mathrm{ENR,FT}$ (white). The boundary data for this model problem is given by $g = 0$ on Γ_D and $h = (1,1)^T \cdot n$ on Γ_N where n denotes the outer normal

Let us consider a very simple cover of the domain Ω into just two overlapping patches (or subdomains) Ω_0 and Ω_1 with respective PU functions Φ_0 and Φ_1, i.e. $\Phi_0 + \Phi_1 \equiv 1$ on $\Omega \subseteq \Omega_0 \cup \Omega_1$. Then, let us choose the approximation space V_0 on the patch Ω_0 to be a classical FE space defined on a respective mesh $\Omega_{0,h}$ which discretizes Ω_0. According to the general PUM approximation theory this choice of V_0 imposes absolutely no constraints on our choice of the approximation space V_1 on the other subdomain Ω_1. For instance, we could choose another FE space on a non-matching mesh $\Omega_{1,H}$ and blend these non-matching spaces smoothly together via the PUM [7,8]. Another admissible choice for V_1 is the use of a locally enriched PUM space [6]. Throughout this paper we focus on the latter case where we blend an enriched approximation space with a classical FE space. Thereby, we can equip any classical FE simulation with stable enrichment capabilities via the PUM in a non-invasive fashion. To this end, let us introduce some more descriptive notation to identify the various components employed in our overall scheme. We refer to the two patches or subdomains by Ω_HOST and Ω_ENR, compare Fig. 1. Moreover, we denote the respective function spaces defined on these subdomains by V_HOST and V_ENR where V_HOST is a classical FE space and V_ENR is an enriched approximation space. Note that in this simplistic two subdomain case we can rewrite the PU functions as $\Phi_\mathrm{HOST} := \Phi$ and $\Phi_\mathrm{ENR} := (1 - \Phi)$ with some non-negative weight function Φ with $\mathrm{supp}(\Phi) = \bar{\Omega}_\mathrm{HOST}$.

With the help of this notation and (1) we define our blended global approximation space V_BND on the complete domain Ω by

$$V_\mathrm{BND} := \Phi_\mathrm{HOST} V_\mathrm{HOST} + \Phi_\mathrm{ENR} V_\mathrm{ENR} = \Phi V_\mathrm{HOST} + (1 - \Phi) V_\mathrm{ENR} \tag{2}$$

such that any element $v \in V_\mathrm{BND}$ can be written as

$$v_\mathrm{BND} = \Phi_\mathrm{HOST} v_\mathrm{HOST} + \Phi_\mathrm{ENR} v_\mathrm{ENR}$$

with $v_\mathrm{HOST} \in V_\mathrm{HOST}$ and $v_\mathrm{ENR} \in V_\mathrm{ENR}$. Note that the general convergence theory of the PUM [1,12] allows to obtain some straightforward error bounds for this blending approach. To this end, let us consider the following estimate from [14].

Theorem 1 *Let* $\Omega \subset \mathbb{R}^D$ *be a Lipschitz domain. Let* $\{\varphi_i : i = 1, \ldots, N\}$ *be an admissible non-negative partition of unity defined on patches* $\omega_i := \mathrm{supp}(\varphi_i)$.

Let us further introduce the covering index $\lambda_{C_\Omega} : \Omega \to \mathbb{N}$ *such that*

$$\lambda_{C_\Omega}(x) := \operatorname{card}(\{\omega_i \in C_\Omega : x \in \omega_i\}) \tag{3}$$

and let us assume that

$$\lambda_{C_\Omega}(x) \leq \Lambda \in \mathbb{N} \quad \textit{for all } x \in \Omega \tag{4}$$

with Λ *independent of the number of patches* N*. Let a collection of local approximation spaces* $V_i = \operatorname{span}\langle \vartheta_i^m \rangle \subset H^1(\omega_i)$ *be given. Let* $f \in H^1(\Omega)$ *be the function to be approximated. Assume that the local approximation spaces* V_i *have the following approximation properties: On each patch* $\Omega \cap \omega_i$*, the function* f *can be approximated by a function* $v_i \in V_i$ *such that*

$$\|f - v_i\|_{L^2(\Omega\cap\omega_i)} \leq \hat{\epsilon}_i, \quad \textit{and} \quad \|\nabla(f - v_i)\|_{L^2(\Omega\cap\omega_i)} \leq \tilde{\epsilon}_i \tag{5}$$

hold for all $i = 1, \ldots, N$*. Then the function*

$$v := \sum_{i=1}^{N} \varphi_i v_i \in V^{\mathrm{PU}} \subset H^1(\Omega)$$

satisfies the global estimates

$$\|f - v\|_{L^2(\Omega)} \leq \left(\sum_{i=1}^{N} \|\varphi_i\|_{L^\infty(\mathbb{R}^d)} \hat{\epsilon}_i^2 \right)^{1/2}, \tag{6}$$

$$\|\nabla(f - v)\|_{L^2(\Omega)} \leq \sqrt{2} \left(\sum_{i=1}^{N} \Lambda \left(\|\nabla\varphi_i\|_{L^\infty(\mathbb{R}^d)} \hat{\epsilon}_i \right)^2 + \|\varphi_i\|_{L^\infty(\mathbb{R}^d)} \tilde{\epsilon}_i^2 \right)^{1/2}. \tag{7}$$

In our setting we only have two approximation spaces V_{HOST} and V_{ENR} and let us, for the ease of notation, assume that both spaces ($i = \mathrm{HOST}, \mathrm{ENR}$) admit error bounds (5) of the form

$$\|f - v_i\|_{L^2(\Omega\cap\omega_i)} \leq Ch_{\mathrm{HOST}}^{p+1}, \quad \text{and} \quad \|\nabla(f - v_i)\|_{L^2(\Omega\cap\omega_i)} \leq Ch_{\mathrm{HOST}}^{p}. \tag{8}$$

Then it is sufficient to ensure that the estimates

$$\|\nabla\Phi\|_{L^\infty(\mathbb{R}^d)} \leq \frac{C_\nabla}{h_{\mathrm{HOST}}}, \quad \|\Phi\|_{L^\infty(\mathbb{R}^d)} \leq C_\infty$$

are satisfied by our PU function Φ to attain optimal convergence with the blended function space. Thus, our blending by the PUM provides optimal convergence even for very small overlap regions $\Omega_{\mathrm{HOST}} \cap \Omega_{\mathrm{ENR}}$.

The Galerkin discretization of a partial differential equation (PDE) using this blended function space V_{BND} yields the linear system

$$K_{\mathrm{BND}} \tilde{u}_{\mathrm{BND}} = \hat{f}_{\mathrm{BND}}, \tag{9}$$

where $\hat{f}_{\mathrm{BND}}$ denotes the load vector and $\tilde{u}_{\mathrm{BND}}$ the respective coefficient vector of the solution. If we assume that the PU function Φ satisfies the so-called flat-top property

$$\Phi|_{\Omega_{\mathrm{HOST,FT}}} = \Phi_{\mathrm{HOST}}|_{\Omega_{\mathrm{HOST,FT}}} \equiv 1$$

for some subset $\Omega_{\text{HOST,FT}} \subset \Omega_{\text{HOST}} = \Omega_{\text{HOST,FT}} \cup \Omega_\Phi$ with $\Omega_\Phi := \Omega_{\text{HOST}} \setminus \Omega_{\text{HOST,FT}}$. Obviously, the other PU function $(1-\Phi) = \Phi_{\text{ENR}}$ then satisfies

$$(1-\Phi)|_{\Omega_{\text{ENR,FT}}} = \Phi_{\text{ENR}}|_{\Omega_{\text{ENR,FT}}} \equiv 1$$

for some $\Omega_{\text{ENR,FT}} \subset \Omega_{\text{ENR}} = \Omega_{\text{ENR,FT}} \cup \Omega_\Phi$ with $\Omega_{\text{HOST,FT}} \cap \Omega_{\text{ENR,FT}} = \emptyset$ and $\Omega_{\text{HOST}} \cap \Omega_{\text{ENR}} = \Omega_\Phi$. With the help of the flat-top property of the PU we can thus introduce the splitting

$$V_{\text{BND}} := V_{\text{HOST,FT}} + \Phi_{\text{HOST}} V^{\perp}_{\text{HOST,FT}} + \Phi_{\text{ENR}} V^{\perp}_{\text{ENR,FT}} + V_{\text{ENR,FT}} \tag{10}$$

of our global blended function space V_{BND} into four components where any $v \in V_{\text{HOST,FT}}$ satisfies $\text{supp}(v) \subset \bar{\Omega}_{\text{HOST,FT}}$ and $V^{\perp}_{\text{HOST,FT}}$ denotes the complement of $V_{\text{HOST,FT}}$ in V_{HOST}. The respective block-partitioning of the global stiffness matrix then reads as

$$K_{\text{BND}} = \begin{pmatrix} K_{\text{HF,HF}} & K_{\text{HF,HF}^\perp} & 0 & 0 \\ K_{\text{HF}^\perp\text{,HF}} & K_{\text{HF}^\perp\text{,HF}^\perp} & K_{\text{HF}^\perp\text{,EF}^\perp} & 0 \\ 0 & K_{\text{EF}^\perp\text{,HF}^\perp} & K_{\text{EF}^\perp\text{,EF}^\perp} & K_{\text{EF}^\perp\text{,EF}} \\ 0 & 0 & K_{\text{EF,EF}^\perp} & K_{\text{EF,EF}} \end{pmatrix}. \tag{11}$$

The load vector $\hat{f}_{\text{BND}}$ and coefficient vector $\tilde{u}_{\text{BND}}$ in this block-form are given by

$$\hat{f}_{\text{BND}} = \begin{pmatrix} \hat{f}_{\text{HF}} \\ \hat{f}_{\text{HF}^\perp} \\ \hat{f}_{\text{EF}^\perp} \\ \hat{f}_{\text{EF}} \end{pmatrix} \quad \text{and} \quad \tilde{u}_{\text{BND}} = \begin{pmatrix} \tilde{u}_{\text{HF}} \\ \tilde{u}_{\text{HF}^\perp} \\ \tilde{u}_{\text{EF}^\perp} \\ \tilde{u}_{\text{EF}} \end{pmatrix}.$$

Note that the sub-matrix $K_{\text{HF,HF}}$ is the classical FE stiffness matrix on the sub-domain $\Omega_{\text{HOST,FT}}$ and thus can be provided by any FE package whereas all other sub-matrices in (11) need to be computed by the embedding code. Hence, our approach is completely non-invasive to a (commercial) FE package with respect to the disjoint partitioning of the domain Ω into $\Omega_{\text{HOST,FT}} \subset \Omega$, which is discretized by the (commerical) FE package, and $\Omega \setminus \Omega_{\text{HOST,FT}}$, compare Fig. 1. However, we actually employ a FE discretization on $\Omega_{\text{HOST}} = \Omega_{\text{HOST,FT}} \cup \Omega_\Phi$ which is obtained by merging the FE discretization on $\Omega_{\text{HOST,FT}}$ provided by the HOST code and the discretization on Ω_Φ provided by our embedding code, see [6] for details. Obviously, the overall computational effort associated with the assembly of the linear system (9) thus scales with the size of the overlap region and thus a small overlap region is preferable from this point of view.

Considering the disjoint partitioning of the domain Ω into $\Omega_{\text{HOST,FT}}$, Ω_Φ and $\Omega_{\text{ENR,FT}}$ we can introduce the block-partitioning

$$K_{\text{BND}} = \begin{pmatrix} K_{\text{HF,HF}} & K_{\text{HF},\Phi} & 0 \\ K_{\Phi\text{,HF}} & K_{\Phi,\Phi} & K_{\Phi\text{,EF}} \\ 0 & K_{\text{EF},\Phi} & K_{\text{EF,EF}} \end{pmatrix}, \tag{12}$$

where

$$K_{\Phi,\Phi} := \begin{pmatrix} K_{\text{HF}^\perp\text{,HF}^\perp} & K_{\text{HF}^\perp\text{,EF}^\perp} \\ K_{\text{EF}^\perp\text{,HF}^\perp} & K_{\text{EF}^\perp\text{,EF}^\perp} \end{pmatrix},$$

which may serve as a natural starting point for the development of classical (single level) domain decomposition solvers and preconditioners. We, however, are interested in the construction of multigrid-like solvers for (9) via the blending of respective multilevel sequences of V_{HOST} and V_{ENR} and thus employ the even more compact block-partitioning

$$K_{\mathrm{BND}} = \begin{pmatrix} K_{\mathrm{H,H}} & K_{\mathrm{H,E}} \\ K_{\mathrm{E,H}} & K_{\mathrm{E,E}} \end{pmatrix}, \quad \hat{f}_{\mathrm{BND}} = \begin{pmatrix} \hat{f}_{\mathrm{H}} \\ \hat{f}_{\mathrm{E}} \end{pmatrix}, \quad \tilde{u}_{\mathrm{BND}} = \begin{pmatrix} \tilde{u}_{\mathrm{H}} \\ \tilde{u}_{\mathrm{E}} \end{pmatrix}, \tag{13}$$

where

$$K_{\mathrm{H,H}} := \begin{pmatrix} K_{\mathrm{HF,HF}} & K_{\mathrm{HF,HF}^{\perp}} \\ K_{\mathrm{HF}^{\perp},\mathrm{HF}} & K_{\mathrm{HF}^{\perp},\mathrm{HF}^{\perp}} \end{pmatrix}.$$

Note, however, that this disjoint partitioning of the matrix corresponds to an overlapping partitioning of the domain Ω into $\Omega_{\mathrm{HOST}} = \Omega_{\mathrm{HOST,FT}} \cup \Omega_{\Phi}$ and $\Omega_{\mathrm{ENR}} = \Omega_{\mathrm{ENR,FT}} \cup \Omega_{\Phi}$. To introduce our proposed multilevel solver for (9) based on the partitioning (13) let us shortly review the general components of such solvers in the context of subspace correction methods.

Subspace correction methods

The computational effort associated with the solution of linear systems like (9) account for a very large part (often even the largest) of the overall computational cost in any implicit or stationary simulation. Thus, the development of efficient linear solvers is of great practical relevance and is still an active research field today. Even though classical general purpose numerical linear algebra techniques such as (sparse) matrix factorizations, see e.g. [15], are widely used in practice, it is well-known that their computational complexity is not optimal and that specialized iterative linear solvers are needed to tackle large scale problems with millions of unknowns efficiently.

A very sophisticated class of iterative methods which not only show an optimal scaling in the storage demand but also in the operation count are so-called multilevel iterative solvers or (geometric) multigrid methods which are particular instances of subspace correction methods [10]. Note, however, that these multilevel and multigrid solvers are not general algebraic methods but involve a substantial amount of information about the discretization and possibly the PDE. Thus, the introduction of such a (geometric) multilevel solver in a commercial software package is very much invasive and typically infeasible. However, there exist extensions of (geometric) multigrid methods, so-called algebraic multigrid methods (AMG) [16–19], which can be used as a non-invasive plugin solver also in commercial software [20,21]. Such AMG solvers are successfully utilized in many different application fields yet they are essentially designed for classical mesh-based piecewise linear discretizations and thus are in general not directly applicable to discretizations with arbitrary approximation functions, i.e. V_{BND} and V_{ENR}. Therefore, no optimal linear solver for (9) is readily available and we need to take the specific construction of our blended approximation space V_{BND} into account when designing a respective iterative linear solver. To this end, we employ classical subspace correction techniques which can utilize splittings such as (2) and (10).

There are two main variants of subspace correction approaches: the parallel subspace correction (PSC) scheme (or additive Schwarz method) and the successive subspace correction (SSC) scheme (or multiplicative Schwarz method). Assuming a splitting

$$V = \sum_{i=0}^{J} V_i \tag{14}$$

of a global function space V, the PSC iteration reads

$$\tilde{u} \leftarrow \tilde{u} + \sum_{i=0}^{J} B_i(\hat{f} - K\tilde{u}) \tag{15}$$

with

$$B_i := P_i K_i^{-1} P_i^T, \quad \text{the prolongation} \quad P_i : V_i \to V, \tag{16}$$

and K_i denoting the stiffness matrix with respect to subspace V_i. The SSC scheme is defined by

$$\text{For } i = 0, \ldots, J: \quad \tilde{u} \leftarrow \tilde{u} + B_i(\hat{f} - K\tilde{u}) \tag{17}$$

and thus involves a successive update of the residual $\hat{f} - K\tilde{u}$ after each subspace correction. Note that the use of the exact inverse K_i^{-1} in (16) is not necessary. In fact, the use of approximate subspace solvers is usually advisable and much more common, i.e. we define

$$B_i := P_i W_i P_i^T \quad \text{with} \quad W_i \approx K_i^{-1}. \tag{18}$$

The main ingredients which control the performance of the iterations (15) and (17) are the specific choices of the subspace splitting (14), the prolongations (16) and the approximate subspace solvers W_i in (18) where it is important to note that we do not assume that (14) allows for a unique decomposition $v = \sum_{i=0}^{J} v_i$ of a function $v \in V$. In fact, the redundancy in the splitting (14) has a substantial impact on the convergence properties of (15) and (17).

In classical multigrid terminology the approximate subspace solvers W_i in (18) are referred to as smoothers and the subspaces V_i correspond to the employed approximation spaces defined on different refinement levels of the underlying mesh, i.e. $V_J = V$ denotes the finest discretization space and V_i with $i < J$ are referred to as coarse spaces. The role of the smoothers W_i is to reduce high frequency error components whereas the corrections $B_i(\hat{f} - K\tilde{u})$ obtained on the coarser levels should reduce low frequency errors so that all error frequencies are efficiently reduced in each iteration.

In the following we first focus on the construction of the coarse spaces V_i with $i < J$ for our finest discretization space $V_J = V = V_{\text{BND}}$ and the definition of the respective prolongations $P_i : V_i \to V = V_J = V_{\text{BND}}$. To this end, let us review an essential property that the prolongations P_i and coarse spaces V_i should satisfy. Since the role of the coarse spaces V_i is the resolution of low frequency errors they should at least contain the constant functions, i.e. $1 \in V_i$, and the prolongations should be exact for the constant functions. Thus, we need to specify a coarsening process for the blended space $V_{\text{BND}} = V_J$ such that the resulting coarse space V_{J-1} contains constants and a respective prolongation $P_{J-1} : V_{J-1} \to V_J = V_{\text{BND}}$ which preserves constants. To this end, let us consider the representation of the constant functions in $V_J = V_{\text{BND}}$

$$1_{\text{BND}} = \Phi_{\text{HOST}} \cdot 1_{\text{HOST}} + \Phi_{\text{ENR}} \cdot 1_{\text{ENR}},$$

where $1_{\text{HOST}} \in V_{\text{HOST}}$ and $1_{\text{ENR}} \in V_{\text{ENR}}$ denote the constant function on the overlapping sub-domains Ω_{HOST} and Ω_{ENR} respectively. Therefore, we can localize the coarsening process for the space V_{BND} to the two spaces V_{HOST} and V_{ENR} and then define the prolongation as

$$P_{J-1} := \begin{pmatrix} P_{\text{HOST},J-1} & 0 \\ 0 & P_{\text{ENR},J-1} \end{pmatrix}$$

provided that the two prolongations $P_{\text{HOST},J-1} : V_{\text{HOST},J-1} \to V_{\text{HOST}}$ and $P_{\text{ENR},J-1} : V_{\text{ENR},J-1} \to V_{\text{ENR}}$ preserve constants in V_{HOST} and V_{ENR} respectively. Hence, let us now focus on the definition of a non-invasive coarsening process for V_{HOST} which comprises the FE space $V_{\text{HOST,FT}}$ (handled by the HOST code) and the FE space $V_{\text{HOST,FT}}^{\perp}$ via AMG techniques.

In general, AMG constructs a suitable coarse space $V_{J-1} \subset V_J$ and a respective prolongation $P_{J-1} : V_{J-1} \to V_J$ that preserves constants from the stiffness matrix K_J obtained by the Galerkin discretization of the PDE using the space V_J. In our setting, however, the block $K_{H,H}$ of the stiffness matrix (13) does not correspond to the Galerkin discretization by V_{HOST} but by the space $\Phi_{\text{HOST}} V_{\text{HOST}}$ (which does not contain the constant function). Thus, only the block $K_{\text{HF,HF}}$ of the matrix

$$K_{\text{H,H}} := \begin{pmatrix} K_{\text{HF,HF}} & K_{\text{HF,HF}^{\perp}} \\ K_{\text{HF}^{\perp},\text{HF}} & K_{\text{HF}^{\perp},\text{HF}^{\perp}} \end{pmatrix}$$

is given by a classical FE discretization and all other blocks are computed using basis functions that are products $\Phi_{\text{HOST}}\phi_k$ of our PU function Φ_{HOST} and classical FE basis functions ϕ_k. Applying AMG directly to $K_{\text{H,H}}$ therefore does not provide a suitable prolongation that preserves constants in V_{HOST}, compare Fig. 2. To overcome this issue in a way that is non-invasive also to AMG we simply set up an auxiliary matrix

$$\bar{K}_{\text{H,H}} := \begin{pmatrix} K_{\text{HF,HF}} & \bar{K}_{\text{HF,HF}^{\perp}} \\ \bar{K}_{\text{HF}^{\perp},\text{HF}} & \bar{K}_{\text{HF}^{\perp},\text{HF}^{\perp}} \end{pmatrix}$$

where we exchange the matrix blocks that involve Φ_{HOST} in $K_{\text{H,H}}$ by their unweighted counterparts, i.e. which are computed using the classical FE basis functions ϕ_k instead of the products $\Phi_{\text{HOST}}\phi_k$. The matrix block $K_{\text{HF,HF}}$ is unchanged so that the non-invasive character of our approach to the HOST code is fully maintained and no additional assembly on the sub-domain $\Omega_{\text{HOST,FT}}$ that is handled by the HOST code is necessary. Since $\bar{K}_{\text{H,H}}$ now satisfies all assumptions of AMG, we obtain a suitable coarse space $V_{\text{HOST},J-1}$ and associated prolongation $\bar{P}_{\text{HOST},J-1}$ from the application of AMG to $\bar{K}_{\text{H,H}}$, compare Fig. 2. In fact, AMG automatically computes a sequence of coarse spaces $V_{\text{HOST},i}$ and associated prolongations $\bar{P}^{i}_{\text{HOST},i-1} : V_{\text{HOST},i-1} \to V_{\text{HOST},i}$.

For the embedded space V_{ENR}, which in our case is itself a PUM space, we construct a sequence of suitable prolongations $\bar{P}^{i}_{\text{ENR},i-1} : V_{\text{ENR},i-1} \to V_{\text{ENR},i}$ directly by so-called global-to-local L^2-projections [22,23] based on a geometric coarsening process. Thus, we can now define a sequence of prolongations

$$P^{i}_{i-1} := \begin{pmatrix} \bar{P}^{i}_{\text{HOST},i-1} & 0 \\ 0 & \bar{P}^{i}_{\text{ENR},i-1} \end{pmatrix}$$

and respective coarser versions $K_{\text{BND},i}$ of our overall stiffness matrix $K_{\text{BND}} = K_{\text{BND},J}$ by the so-called Galerkin operators

$$K_{\text{BND},i-1} := (P^{i}_{i-1})^{T} K_{\text{BND},i} P^{i}_{i-1}.$$

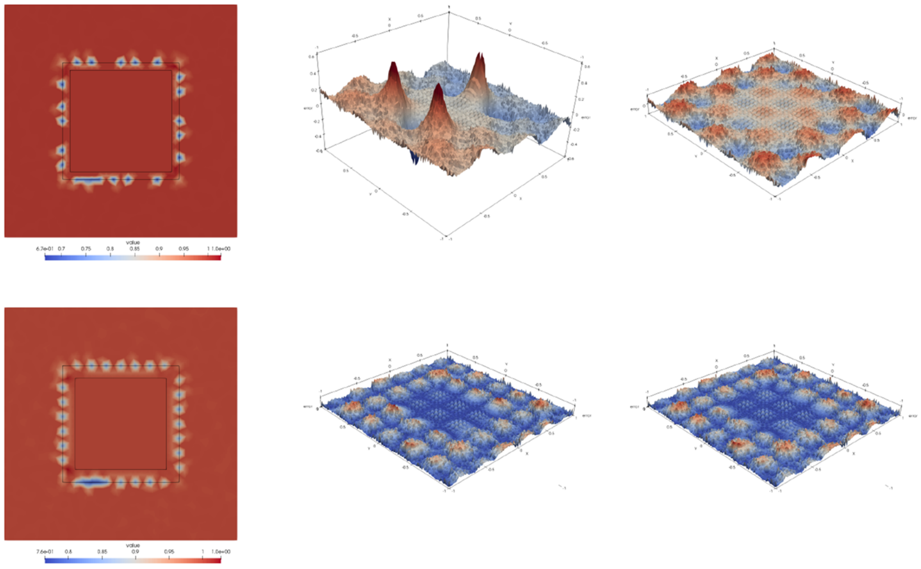

Fig. 2 Contour plots of the prolongation errors (left) attained for the constant function when using the weighted space $\Phi_{\mathrm{HOST}} V_{\mathrm{HOST}}$ in the AMG construction of the prolongation $P^{J}_{\mathrm{HOST},J-1}$ for a small (top) and a large overlap region (bottom). Surface plots of the prolongation error of the smooth function $\sin(3\pi x)\cos(3\pi y)$ using $P^{J}_{\mathrm{HOST},J-1}$ (center) and $\tilde{P}^{J}_{\mathrm{HOST},J-1}$ (right)

Note, however, that this overall coarsening process in fact coarsens the two componentes V_{HOST} and V_{ENR} independently and thus may not be optimal on the overlap region $\Omega_\Phi = \Omega_{\mathrm{HOST}} \cap \Omega_{\mathrm{ENR}}$ where the two spaces interact substantially. Therefore, the overall performance of the proposed scheme will be affected somewhat by the absolute size of the overlap region Ω_Φ. Unlike in classical domain decomposition methods where larger overlap yields a convergence improvement, we, however, anticipate that an increasing overlap will rather worsen the convergence behavior since our independent coarsening process then ignores stronger interactions between the spaces.

As a final component we need to specify the approximate subspace solvers or smoothers on the resulting coarse spaces $V_{\mathrm{BND},i}$ to instantiate the iterations (15) and (17). In the following we focus on iterations of the form (17), in particular we employ the classical multigrid iteration $M_\gamma^{\nu_1,\nu_2}$ given in Algorithm 3.1 and consider different numbers of smoothing steps $\nu = \nu_1 = \nu_2$ as well as the V-cycle ($\gamma = 1$) and the W-cycle ($\gamma = 2$).

```
Algorithm 3.1 (Multilevel Iteration $M_\gamma^{\nu_1,\nu_2}(l, \tilde{u}_l, \hat{f}_l)$).
1 if $l > 0$:
    (a) For $i = 1, \dots, \nu_1$: Set $\tilde{u}_l = S_l^{\mathrm{pre}}(\tilde{u}_l, \hat{f}_l)$.
    (b) Set $\hat{d}_{l-1} := (P_l^{l-1})^T (\hat{f}_l - K_l \tilde{u}_l)$.
    (c) Set $\tilde{e}_{l-1} := 0$.
    (d) For $i = 1, \dots, \gamma$: $\tilde{e}_{l-1} = M_\gamma^{\nu_1,\nu_2}(l-1, \tilde{e}_{l-1}, \hat{d}_{l-1})$.
    (e) Set $\tilde{u}_l = C_l(\tilde{u}_l, \tilde{e}_{l-1}) := \tilde{u}_l + P_{l-1}^{l} \tilde{e}_{l-1}$.
    (f) For $i = 1, \dots, \nu_2$: Set $\tilde{u}_l = S_l^{\mathrm{post}}(\tilde{u}_l, \hat{f}_l)$.
2 else:
    (a) Set $\tilde{u}_l = K_l^{-1} \hat{f}_l$.
```

As smoothers S_l^{pre} and S_l^{post} in Algorithm 3.1 we simply use Gauß–Seidel iterations, which provide acceptable error reduction but their smoothing effect seems to be somewhat less pronounced in Ω_Φ for larger overlap regions Ω_Φ, see Fig. 3.

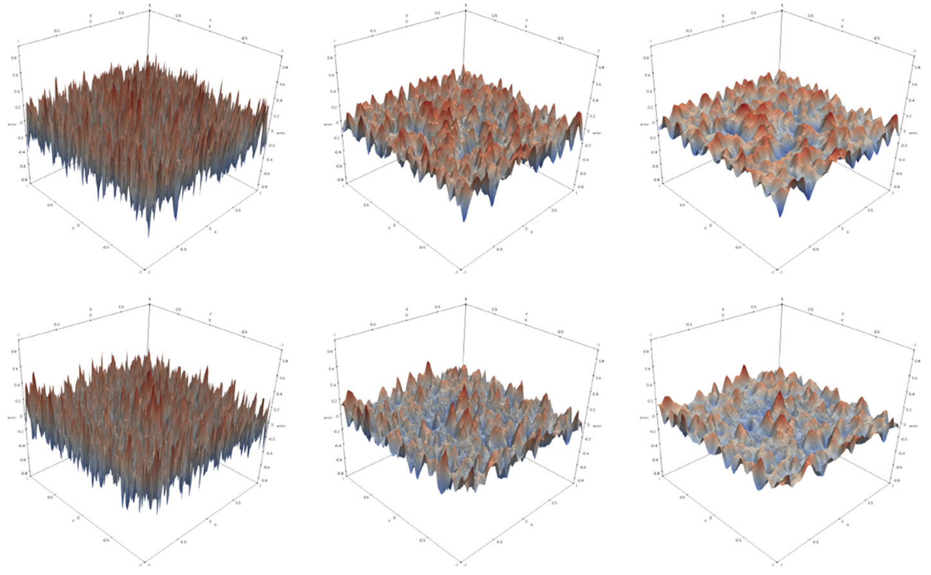

Fig. 3 Surface plots of iterates of a Gauß–Seidel smoother for iteration 1, 3, 5 (left to right) using a random initial guess on a blended discretization using a small (top) and a large overlap region (bottom)

Numerical results

In this section we present some results of our numerical experiments using the embedded enriched PUM within a classical FE simulation as discussed above. To this end, we consider a set of isotropic model problems focussing on the optimality of the proposed solver only. A detailed study of the solvers robustness with respect to varying material coefficients is the subject of future research. In particular, we are concerned with the approximation of the Poisson problem

$$\begin{aligned} -\Delta u &= f && \text{in } \Omega \subset \mathbb{R}^d, \\ u &= g && \text{on } \Gamma_D \subset \partial\Omega, \\ \frac{\partial u}{\partial n} &= h && \text{on } \Gamma_N = \partial\Omega \backslash \Gamma_D, \end{aligned} \tag{19}$$

in two space dimensions on a square domain, see Fig. 1, where we embed a smooth enrichment space V_{ENR} to identify the best performance of our proposed solver. Then, we consider a non-convex domain, see Fig. 4, where we embed an enrichment space V_{ENR} that contains singular functions to efficiently resolve the corner singularity. Finally, we consider a linearly elastic bar in three space dimensions, see Fig. 5, i.e.

$$\begin{aligned} -\mathbf{div}\boldsymbol{\sigma}(\vec{u}) &= 0 && \text{in } \Omega \subset \mathbb{R}^d, \\ \vec{u} &= \vec{g} && \text{on } \Gamma_D \subset \partial\Omega, \\ \boldsymbol{\sigma}(\vec{u}) \cdot \vec{n} &= 0 && \text{on } \Gamma_C, \\ \boldsymbol{\sigma}(\vec{u}) \cdot \vec{n} &= \vec{h} && \text{on } \Gamma_N = \partial\Omega \backslash \Gamma_D, \end{aligned} \tag{20}$$

with the stress tensor $\boldsymbol{\sigma}(\vec{u}) := \mathcal{C}\boldsymbol{\varepsilon}(\vec{u}) = 2\mu\boldsymbol{\varepsilon}(\vec{u}) + \lambda\, \mathrm{trace}(\boldsymbol{\varepsilon}(\vec{u}))\mathbb{I}$ and the infinitesimal strain tensor $\boldsymbol{\varepsilon}(u) := \frac{1}{2}(\nabla\vec{u} + (\nabla\vec{u})^T)$, to study the performance of our proposed solver for systems of equations in higher dimensions. Moreover, the selected model problems

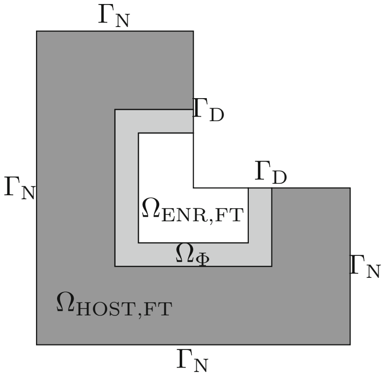

Fig. 4 Sketch of a non-convex domain Ω in two dimensions depicting the employed partitioning into $\Omega_{\rm HOST,FT}$ (drak gray), $\Omega_\Phi = \Omega_{\rm HOST} \cap \Omega_{\rm ENR}$ (light gray), and $\Omega_{\rm ENR,FT}$ (white). The boundary data for this model problem is given by $g = 0$ on Γ_D and $h = \nabla\left((x^2 + xy + y^2 + 1)(r^{\frac{2}{3}} \sin(\frac{2\theta-\pi}{3}))\right) \cdot n$ on Γ_N where n denotes the outer normal

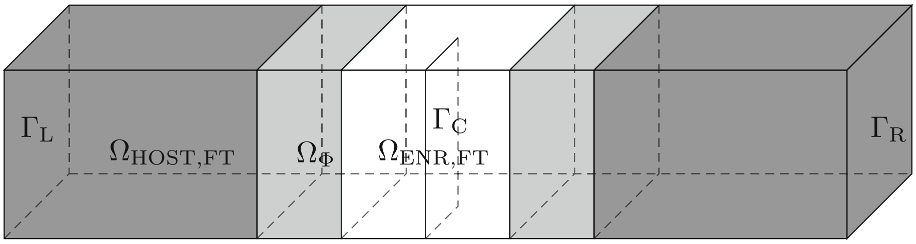

Fig. 5 Sketch of a pre-cracked bar in three dimensions depicting the partitioning of the domain Ω into $\Omega_{\rm HOST,FT}$ (drak gray), $\Omega_\Phi = \Omega_{\rm HOST} \cap \Omega_{\rm ENR}$ (light gray), and $\Omega_{\rm ENR,FT}$ (white). The boundary data for this model problem is given by $\Gamma_D = \Gamma_L \cup \Gamma_R$, $g = (0, 0, 0)^T$ on Γ_L, $g = (0.02, 0, 0)^T$ on Γ_R and $h = (0, 0, 0)^T$ on $\Gamma_N = \partial\Omega \setminus \Gamma_D$

employ different embedding configurations with respect to the intersection of $\Omega_{\rm ENR}$ with the boundaries Γ_D and Γ_N of the global simulation domain Ω, compare Figs. 1, 4 and 5.

In all our experiments we consider discretizations which satisfy our assumption (8) by choosing the support sizes of basis functions in $V_{\rm HOST}$ and $V_{\rm ENR}$ of comparable size, see Fig. 6, and use (bi/tri-)linear finite elements in $V_{\rm HOST}$ and linear polynomials with additional enrichments in $V_{\rm ENR}$. We employ the proposed multilevel iteration as a stand-alone solver as well as a preconditioner for the conjugate gradient method (CG) and measure the number of iterations required to reduce the initial residual by ten orders of magnitude. From an optimality point of view, we are mostly interested in obtaining iteration numbers n that are independent of the number of employed levels k. From the measured iteration numbers given in Table 1, we see that the V-cycle stand-alone solver with only a single pre- and post-smoothing step already provides acceptable iteration numbers $n_{\rm V(1,1)} < 45$ which, however, are not completely independent of the number of employed levels k. Yet, increasing the number of smoothing steps to $\nu = 3$ or changing to the more expensive W-cycle yields constant iteration numbers $n_{\rm V(3,3)}$ and $n_{\rm W(1,1)}$ independent of k, see also Fig. 7. In fact, further experiments with the proposed multilevel iteration showed that it is already sufficient to increase the number of smoothing steps on coarser levels only. Thus,

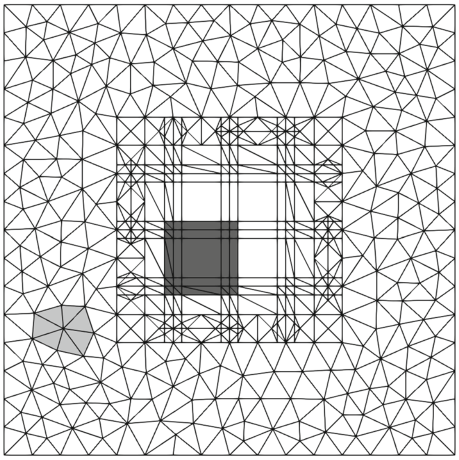

Fig. 6 Sketch of the blended function discretization space depicting the supports of a basis function $\phi_{\mathrm{HOST}} \in V_{\mathrm{HOST}}$ (light gray) and $\phi_{\mathrm{ENR}} \in V_{\mathrm{ENR}}$ (dark gray). The support sizes are of comparable size throughout the paper; i.e. diam(supp(ϕ_{HOST}) $\sim$ diam(supp(ϕ_{ENR})

Table 1 Measured iteration numbers *n* for the stand-alone solver and n_{CG} for the preconditioned CG method attained for the model problem (19) on the configuration depicted in Fig. 1 using a single element overlap

k	Dof	$n_{\mathrm{V}(1,1)}$	$n_{\mathrm{V}(3,3)}$	$n_{\mathrm{W}(1,1)}$	$n_{\mathrm{CGV}(1,1)}$	$n_{\mathrm{CGV}(3,3)}$	$n_{\mathrm{CGW}(1,1)}$
3	1479	23	9	23	12	7	12
4	5386	22	11	22	12	8	12
5	20,249	19	12	19	12	9	11
6	79,681	22	15	18	12	10	11
7	315,046	26	16	17	14	11	11
8	1,245,303	38	22	18	17	12	11
9	4,958,241	43	22	18	18	12	11

indicating that the quality of the corrections from coarser levels obtained in a $V(1,1)$-cycle is somewhat deminished for larger k. Nevertheless, the use of the $V(1,1)$-cycle as a preconditioner in CG yields fairly stable iteration numbers $n_{\mathrm{CGV}(1,1)} < 20$ and provides the fastest time-to-solution for the considered numbers of levels k. Note that the results summarized in Table 1 were obtained with a small overlap region Ω_Φ of a single element on the finest level; i.e. the overlap region is in fact shrinking as we refine the discretization. As mentioned above, we anticipate that for a larger overlap region Ω_Φ with fixed volume for all levels k, i.e. an increasing number of elements in the overlap as we refine, the convergence behavior of the proposed solver will deteriorate somewhat. In fact, the results given in Table 2 show that the number of iterations increases not only but also grows with larger number of levels k even when we use the rather expensive $W(1,1)$-cycle as a preconditioner in CG. Thus, the results confirm our expectation that it is advisable to choose an overlap region Ω_Φ whose diameter is proportational to the meshwidth on the finest level employed (unlike in classical domain decomposition approaches) since such a choice yields the least amount of work in the assembly of the blended linear system K_{BND} and it gives the best solver performance.

Now that we have in principle identified the best attainable convergence behavior of the proposed method for a smooth problem employing a smooth enrichment space V_{ENR},

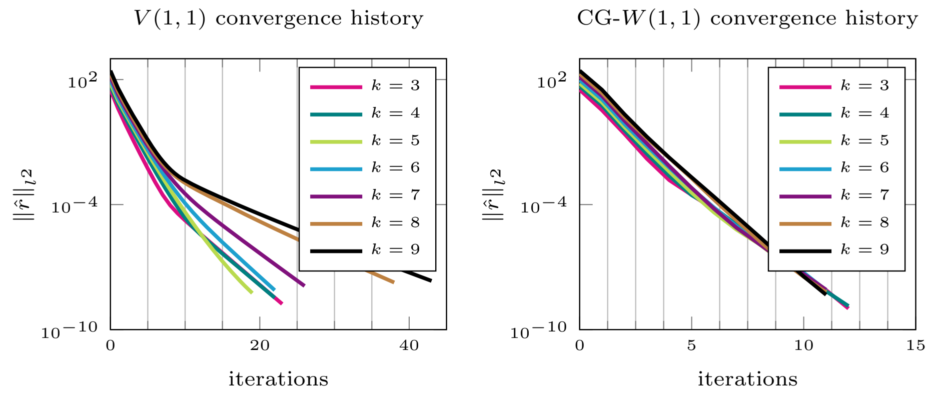

Fig. 7 Convergence histories for the $V(1, 1)$-cyle as a stand-alone solver (left) and the $W(1, 1)$-cycle as a preconditioner in CG (right)

Table 2 Measured iteration numbers n_{CG} for the preconditioned CG method attained for the model problem (19) on the configuration depicted in Fig. 1 using a fixed volume overlap

k	Dof	$n_{\text{CGV(1,1)}}$	$n_{\text{CGV(3,3)}}$	$n_{\text{CGW(1,1)}}$
3	1475	30	20	30
4	5710	37	25	32
5	21,965	53	36	39
6	88,022	56	60	54
7	350,511	133	96	75
8	1,392,765	233	168	116

let us consider a more relevant case where V_{ENR} contains problem-dependent singular functions. To this end, we consider (19) on a non-convex domain, see Fig. 4. Here, the discretization space V_{ENR} on the region Ω_{ENR} is defined as

$$V_{\text{ENR}} = \sum_{i=1}^{N} \varphi_i \Big(\mathcal{P}_1 + \text{span}\langle r^{\frac{2}{3}} \sin\Big(\frac{2\theta - \pi}{3}\Big)\rangle \Big) \tag{21}$$

where (r, θ) denote polar coordinates with respect to the re-entrant corner of Ω. Thus, we employ enrichment by a singular function everywhere in Ω_{ENR}, see [24] for details on the construction of a stable basis for V_{ENR}. Moreover, we use so-called block-Gauß–Seidel relaxation in V_{ENR} where we collect all degrees of freedom defined on the same patch into a single block, see [22,23] for details. The attained number of iterations are given in Table 3. The overall number of iterations is somewhat larger than in the previous case, however, the number of iterations are essentially independent of the number of levels. It is also noteworthy to point out, that in this model configuration the $V(3, 3)$-cycle substantially outperforms the $W(1, 1)$-cycle which shows the improved smoothing property of the patch-based block-Gauß–Seidel relaxation in V_{ENR}. Nevertheless, the fastest time-to-solution for the considered discretizations with more than 13 million degrees of freedom is still obtained by CG preconditioned by $V(1, 1)$-cycle which of course also benefits from the improved smoothing property.

Finally, we consider the system of partial differential equations (20) in three space dimensions. Here, we employ a so-called point-based AMG approach [21] for the coarsening in

Table 3 Measured iteration numbers n_{CG} for the preconditioned CG method attained for the model problem (19) on the configuration depicted in Fig. 4 using a single element overlap

k	Dof	$n_{\mathrm{CGV}(1,1)}$	$n_{\mathrm{CGV}(3,3)}$	$n_{\mathrm{CGW}(1,1)}$
3	3,843	18	10	18
4	14,063	18	10	18
5	54,408	19	11	18
6	214,527	20	11	19
7	847,974	20	12	19
8	3,372,646	21	14	19
9	13,380,007	22	16	13

Table 4 Measured iteration numbers n_{CG} for the preconditioned CG method attained for the model problem (20) on the configuration depicted in Fig. 5 using a single element overlap

k	Dof	$n_{\mathrm{CGV}(1,1)}$	$n_{\mathrm{CGV}(3,3)}$	$n_{\mathrm{CGW}(1,1)}$
3	15,666	17	11	17
4	106,482	19	14	18
5	774,738	22	17	17
6	5,883,282	24	20	16

Ω_{HOST} and utilize one more layer of bock-partitioning which collects all three displacement components in the Gauß–Seidel smoother, i.e. we now use a (3×3)-block relaxation in V_{HOST} combined with the block-relaxation in V_{ENR} described above. We discretize (20) with trilinear elements in V_{HOST} and use singular and discontinuous enrichments for the treatment of the crack in V_{ENR} (besides linear polynomials). The performance of our proposed solver is summarized in Table 4. Again, we find essentially constant iteration numbers for CG preconditioned by $W(1, 1)$-cycle and slightly increasing iteration numbers when using a V-cycle preconditioner. Yet, the fastest time-to-solution is still attained when using the $V(1, 1)$-cycle as preconditioner.

In summary, the presented results clearly show that the proposed solver yields close to optimal convergence in two and three dimensions when using a small overlap. Using a small overlap is moreover beneficial to the total computational cost in the assembly of the blended linear system and still yields optimal approximation errors.

Concluding remarks

In this paper we proposed a constructive non-invasive approach to the design of efficient multilevel solvers for embedded enriched approximations. The non-invasive embedding scheme is based on a partition of unity approach and can essentially blend arbitrary (overlapping) approximation spaces, yet we consider the special case of embedding an enriched partition of unity space into a classical finite element space. The proposed solver utilizes non-invasive algebraic multigrid technology [16–20] for the automatic construction of a sequence of coarser sub-spaces of the employed finite element space and a sequence of enriched partiton of unity spaces obtained via a geometric coarsening scheme [22]. The presented results clearly indicate that the proposed method can attain (close to) optimal convergence behavior when a small overlap or blending region is employed. A detailed study on the optimal selection of parameters and robustness properties of the proposed scheme is the subject of ongoing and future research.

Authors' contributions
MAS and AZ developed the method. AZ implemented the method and conducted the numerical experiments. All authors read and approved the final manuscript.

Author details
[1] Fraunhofer-Institut für Algorithmen und Wissenschaftliches Rechnen SCAI, Schloss Birlinghoven, 53757 Sankt Augustin, Germany, [2] Institut für Numerische Simulation, Rheinische Friedrich-Wilhelms-Universität Bonn, Wegelerstr. 6, 53115 Bonn, Germany.

Acknowledgements
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Availability of data and materials
Not applicable.

Consent for publication
Not applicable.

Ethics approval and consent to participate
Not applicable.

Funding
Not applicable.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 9 January 2018 Accepted: 30 April 2018
Published online: 18 May 2018

References

1. Babuška I, Melenk JM. The partition of unity method. Int J Numer Methods Eng. 1997;40:727–58.
2. Belytschko T, Black T. Elastic crack growth in finite elements with minimal remeshing. Int J Numer Methods Eng. 1999;45:601–20.
3. Duarte CA, Oden JT. An hp adaptive method using clouds. Comput Methods Appl Mech Eng. 1996;139:237–62.
4. Fries T-P, Belytschko T. The extended/generalized finite element method: an overview of the method and its applications. Int J Numer Methods Eng. 2010;84:253–304.
5. Schweitzer M. Variational mass lumping in the partition of unity method. SIAM J Sci Comput. 2013;35:A1073–97.
6. Schweitzer MA, Ziegenhagel A. Embedding enriched partition of unity approximations in finite element simulations. In: Griebel M, Schweitzer MA, editors. Meshfree methods for partial differential equations VIII., Lecture notes in science and engineeringCham: Springer International Publishing; 2017. p. 195–204.
7. Bacuta C, Xu J. Partition of unity for the Stokes problem on nonmatching grids. In: Proceedings of the 2003 copper mountain conference on multigrid. 2003.
8. Bakuta C, Chen J, Huang Y, Xu J, Zikatanov L. Partition of unity method on nonmatching grids for the Stokes problem. J Numer Math. 2005;13:157–69.
9. Gupta P, Pereira J, Kim D-J, Duarte C, Eason T. Analysis of three-dimensional fracture mechanics problems: a non-intrusive approach using a generalized finite element method. Eng Fract Mech. 2012;90:41–64.
10. Xu J. Iterative methods by space decomposition and subspace correction. SIAM Rev. 1992;34:581–613.
11. Smith BF, Bjørstad PE, Gropp WD. Domain decomposition: parallel multilevel methods for elliptic partial differential equations. Cambridge: Cambridge University Press; 1996.
12. Babuška I, Melenk JM. The partition of unity finite element method: basic theory and applications. Comput Methods Appl Mech Eng. 1996;139:289–314 (Special Issue on Meshless Methods).
13. Babuška I, Caloz G, Osborn JE. Special finite element methods for a class of second order elliptic problems with rough coefficients. SIAM J Numer Anal. 1994;31:945–81.
14. Schweitzer MA. Generalizations of the finite element method. Cent Eur J Math. 2012;10:3–24.
15. Amestoy PR, Duff IS, Koster J, L'Excellent JY. A fully asynchronous multifrontal solver using distributed dynamic scheduling. SIAM J Matrix Anal Appl. 2001;23:15–41.
16. Brandt A. Algebraic multigrid theory: the symmetric case. Appl Math Comput. 1986;19:23–56.
17. Brandt A, McCormick S, Ruge J. Algebraic multigrid (AMG) for sparse matrix equations. Sparsity and its applications (Loughborough, 1983). Cambridge: Cambridge Univ. Press; 1985. p. 257–84.

18. Ruge J, Stüben K. Efficient solution of finite difference and finite element equations. In: Multigrid methods for integral and differential equations (Bristol, vol. 3 of Inst. Math. Appl. Conf. Ser. New Ser.) New York: Oxford Univ. Press; 1983, 1985. p. 169–212.
19. Stüben K. A review of algebraic multigrid. J Comput Appl Math. 2001; 128:281–309. Numerical analysis, Partial differential equations: VII; 2000.
20. SAMG—efficiently solving large linear systems of equations. https://www.scai.fraunhofer.de/en/business-research-areas/fast-solvers/products/samg.html.
21. Stüben K, Ruge JW, Clees T, Gries S. Algebraic multigrid: from academia to industry. In: Griebel M, Schüller A, Schweitzer MA, editors. Scientific computing and algorithms in industrial simulation—projects and products of Fraunhofer SCAI. Cham: Springer International Publishing; 2017. p. 83–120.
22. Griebel M, Schweitzer MA. A particle-partition of unity method—part III: a multilevel solver. SIAM J Sci Comput. 2002;24:377–409.
23. Schweitzer MA. A parallel multilevel partition of unity method for elliptic partial differential equations., Lecture notes in computational science and engineeringBerlin: Springer; 2003.
24. Schweitzer MA. Stable enrichment and local preconditioning in the particle-partition of unity method. Numer Math. 2011;118:137–70.
25. Schweitzer MA. Multilevel particle-partition of unity method. Numer Math. 2011;118:307–28.