A Particle-Partition of Unity Method Part VII: Adaptivity

Michael Griebel and Marc Alexander Schweitzer

Institut für Numerische Simulation, Universität Bonn, Wegelerstr. 6, D-53115 Bonn, Germany
{griebel, schweitzer}@ins.uni-bonn.de

Summary. This paper is concerned with the adaptive multilevel solution of elliptic partial differential equations using the partition of unity method. While much of the work on meshfree methods is concerned with convergence-studies, the issues of fast solution techniques for the discrete system of equations and the construction of optimal order algorithms are rarely addressed. However, the treatment of large scale real-world problems by meshfree techniques will become feasible only with the availability of fast adaptive solvers.

The adaptive multilevel solver proposed in this paper is a main step toward this goal. In particular, we present an h-adaptive multilevel solver for the partition of unity method which employs a subdomain-type error indicator to control the refinement and an efficient multilevel solver within a nested iteration approach. The results of our numerical experiments in two and three space dimensions clearly show the efficiency of the proposed scheme.

Key words: meshfree method, partition of unity method, adaptive refinement, multilevel method

1 Introduction

One main purpose of this paper is to investigate adaptive h-type refinement strategies for meshfree methods and their interplay with multilevel solution techniques; in particular we address these issues for the partition of unity method [1, 15]. To this end, we employ a classical a posteriori error estimation technique due to Babuška and Rheinboldt [2]. The resulting error indicator is used to steer the local refinement procedure in our tree-based cover construction procedure. To obtain an adaptive solver with optimal complexity we combine the multilevel techniques developed in [7, 9, 15] for the PUM with the nested iteration approach [12]. The results of our numerical experiments in two and three space dimensions demonstrate the effectiveness of the proposed approach and its overall efficiency.

In this paper we restrict ourselves to the study of a scalar elliptic partial differential equation, namely we consider the diffusion problem

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega \subset \mathbb{R}^d, \\ u &= g_D \quad \text{on } \Gamma_D \subset \partial\Omega, \\ \frac{\partial u}{\partial n} &= g_N \quad \text{on } \Gamma_N = \partial\Omega \setminus \Gamma_D. \end{aligned} \tag{1.1}$$

The remainder of this paper is organized as follows. In section 2 we give a short overview of the PUM and its convergence properties. Furthermore, we outline the implementation of essential boundary conditions using Nitsche's method and the Galerkin discretization of the arising variational problem. The main theme of this paper, the adaptive meshfree multilevel solution of an elliptic PDE, is presented in section 3. There, we introduce our refinement algorithm and show that it leads to point sets and covers that are consistent with the multilevel construction of [9]. Moreover, we present the construction of our error indicator and discuss how we obtain an adaptive multilevel solver with optimal complexity using the nested iteration approach. Then, we present the results of our numerical experiments in two and three space dimensions in section 4. These results clearly demonstrate the efficiency of the proposed scheme. Finally, we conclude with some remarks in section 5.

2 Partition of Unity Method

In the following, we shortly review the construction of a partition of unity space V^{PU} and the Galerkin discretization of an elliptic partial differential equation using V^{PU} as trial and test space, see [15] for details.

2.1 Construction of a Partition of Unity Space

In a PUM, 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 $\omega_i := \operatorname{supp}(u_i)$, 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

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

The starting point in the implementation of a PUM approximation space V^{PU} is the construction of an appropriate PU, see Definition 1 and Definition 2.

Definition 1 (Partition of Unity). *Let $\Omega \subset \mathbb{R}^d$ be an open set. Let $\{\varphi_i\}$ be a collection of Lipschitz functions with*

$$\begin{array}{ll} 0 \;\le\; \varphi_i(x) \le 1, & \sum_i \varphi_i \equiv 1 \;\text{ on } \overline{\Omega}, \\ \|\varphi_i\|_{L^\infty(\mathbb{R}^d)} \le C_\infty, & \|\nabla\varphi_i\|_{L^\infty(\mathbb{R}^d)} \le \frac{C_\nabla}{\mathrm{diam}(\omega_i)}, \end{array}$$

where $\omega_i := \mathrm{supp}(\varphi_i)$, C_∞ and C_∇ are two positive constants. The sets ω_i are called patches and their collection is referred to as a cover $C_\Omega := \{\omega_i\}$ of the domain Ω.

For PUM spaces (2.2) which employ such a PU $\{\varphi_i\}$ there hold the following error estimates due to [1].

Theorem 1. *Let $\Omega \subset \mathbb{R}^d$ be given. Let $\{\varphi_i\}$ be a partition of unity according to Definition 1. Let us further introduce the covering index $\lambda_{C_\Omega} : \Omega \to \mathbb{N}$ such that*

$$\lambda_{C_\Omega}(x) = \mathrm{card}(\{i \,|\, x \in \omega_i\}) \tag{2.3}$$

and let us assume that $\lambda_{C_\Omega}(x) \le M \in \mathbb{N}$ for all $x \in \Omega$. Let a collection of local approximation spaces $V_i^{p_i} = \mathrm{span}\langle\{\psi_i^n\}\rangle \subset H^1(\Omega \cap \omega_i)$ be given. Let $u \in H^1(\Omega)$ be the function to be approximated. Assume that the local approximation spaces $V_i^{p_i}$ have the following approximation properties: On each patch $\Omega \cap \omega_i$, the function u can be approximated by a function $u_i \in V_i^{p_i}$ such that

$$\|u - u_i\|_{L^2(\Omega\cap\omega_i)} \le \hat{\epsilon}_i, \quad \text{and} \quad \|\nabla(u - u_i)\|_{L^2(\Omega\cap\omega_i)} \le \tilde{\epsilon}_i \tag{2.4}$$

hold for all i. Then the function

$$u^{\mathrm{PU}} := \sum_{\omega_i \in C_\Omega} \varphi_i u_i \in V^{\mathrm{PU}} \subset H^1(\Omega)$$

satisfies the global estimates

$$\|u - u^{\mathrm{PU}}\|_{L^2(\Omega)} \le \sqrt{M} C_\infty \Big(\sum_{\omega_i \in C_\Omega} \hat{\epsilon}_i^2 \Big)^{\frac{1}{2}}, \tag{2.5}$$

$$\|\nabla(u - u^{\mathrm{PU}})\|_{L^2(\Omega)} \le \sqrt{2M} \Big(\sum_{\omega_i \in C_\Omega} \big(\frac{C_\nabla}{\mathrm{diam}(\omega_i)}\big)^2 \hat{\epsilon}_i^2 + C_\infty^2 \tilde{\epsilon}_i^2 \Big)^{\frac{1}{2}}. \tag{2.6}$$

The estimates (2.5) and (2.6) show that the global error is of the same order as the local errors provided that the covering index is bounded independent of the size of the cover, i.e. $M = O(1)$. Note that we need to assume a slightly stronger condition to obtain a *sparse* linear system by the Galerkin approach. To this end, we introduce the notion of a local neighborhood or local cover $C_{\Omega,i} \subset C_\Omega$ of a particular cover patch $\omega_i \in C_\Omega$ by

$$C_{\Omega,i} := \{\omega_j \in C_\Omega \,|\, \omega_j \cap \omega_i \neq \emptyset\} \tag{2.7}$$

and require $\max_{\omega_i \in C_\Omega} \operatorname{card}(C_{\Omega,i}) = O(1)$.

Note furthermore that the conditions imposed on the PU in Definition 1 do not ensure that the product functions $\varphi_i \psi_i^n$ of (2.2) are linearly independent. However, to obtain the linear independence of the product functions $\varphi_i \psi_i^n$ it is sufficient to require that the PU has the following property.

Definition 2 (Flat top property). *Let $\{\varphi_i\}$ be a partition of unity according to Definition 1. Let us define the sub-patches $\omega_{\mathrm{FT,i}} := \{x \,|\, \lambda_{C_\Omega}(x) = 1\}$ such that $\varphi_i|_{\omega_{\mathrm{FT,i}}} \equiv 1$. Then, the PU is said to have the flat top property, if there exists a constant C_{FT} such that for all patches ω_i*

$$\mu(\omega_i) \leq C_{\mathrm{FT}}\, \mu(\omega_{\mathrm{FT,i}}) \tag{2.8}$$

where $\mu(A)$ denotes the Lebesgue measure of $A \subset \mathbb{R}^d$. We have $C_\infty = 1$ for a PU with the flat top property.

Obviously the product functions $\varphi_i \psi_i^n$ are linearly independent if we assume that the PU has the flat top property and that each of the local bases $\{\psi_i^n\}$ is locally linearly independent on the sub-patches $\omega_{\mathrm{FT,i}} \subset \omega_i$.[1]

The PU concept is employed in many meshfree methods. However, in most cases very smooth PU functions $\varphi_i \in C^k(\Omega)$ with $k \geq 2$ are used and the functions φ_i have rather large supports ω_i which overlap substantially. Hence in most meshfree methods $\operatorname{card}(C_{\Omega,i})$ is large and the employed PU does *not* have the flat top property. This makes it easier to control $\|\nabla \varphi_i\|_{L^\infty}$, compare Definition 1 and (2.6), but it can lead to ill-conditioned and even singular stiffness matrices. For a flat top PU we obviously have $\nabla \varphi_i|_{\omega_{\mathrm{FT,i}}} \equiv 0$ so that it is sufficient to bound $\nabla \varphi_i$ on the complement $\omega_i \setminus \omega_{\mathrm{FT,i}}$ which requires some additional properties, compare (2.12) and (2.13). Hence, the cover construction for a flat top PU is somewhat more challenging.

A PU can for instance be constructed by simple averaging, often referred to as Shepard's method. Let us assume that we have a cover $C_\Omega = \{\omega_i\}$ of the domain Ω such that $1 \leq \lambda_{C_\Omega}(x) \leq M$ for all $x \in \Omega$. With the help of non-negative weight functions W_k defined on these cover patches ω_k, i.e. $W_k(x) > 0$ for all $x \in \omega_k \setminus \partial\omega_k$, we can easily generate a partition of unity by

$$\varphi_i(x) := \frac{W_i(x)}{S_i(x)} \quad \text{where } S_i(x) := \sum_{\omega_j \in C_{\Omega,i}} W_j(x). \tag{2.9}$$

[1] Note that the flat top property is a sufficient condition only. It is not a necessary requirement. In practice we already obtain a linearly independent set of shape functions if the flat top property is satisfied by most but not necessarily all patches ω_i of the cover C_Ω.

Obviously, the smoothness of the resulting PU functions φ_i is determined entirely by the smoothness of the employed weight functions. Hence, on a cover with tensor product patches ω_i we can easily construct partitions of unity of any regularity for instance by using tensor products of splines with the desired regularity as weight functions.[2] Hence, let us assume that the weight functions W_i are all given as linear transformations of a generating normalized spline weight function $\mathcal{W} : \mathbb{R}^d \to \mathbb{R}$ with $\operatorname{supp}(\mathcal{W}) = [0,1]^d$, i.e.,

$$W_i(x) = \mathcal{W} \circ T_i(x), \quad T_i : \omega_i \to [0,1]^d, \quad \|DT_i\|_\infty \leq \frac{C_T}{\operatorname{diam}(\omega_i)} \tag{2.10}$$

and $\|\mathcal{W}\|_\infty = 1$, $\|\nabla \mathcal{W}\|_\infty \leq C_{\mathcal{W},U}$. To show that the PU arising from (2.9) is valid according to Definition 1 it is sufficient to make the following additional assumptions:

- Comparability of neighboring patches: There exist constants C_L and C_U such that for all local neighborhoods $C_{\Omega,i}$ there holds the implication
$$\omega_j \in C_{\Omega,i} \quad \Rightarrow \quad C_L \operatorname{diam}(\omega_i) \leq \operatorname{diam}(\omega_j) \leq C_U \operatorname{diam}(\omega_i) \tag{2.11}$$
with absolute constants C_L and C_U.
- Sufficient overlap: There exists a constant $K > 0$ such that for any $x \in \Omega$ there is at least one cover patch ω_i with the property
$$x \in \omega_i, \quad \operatorname{dist}(x, \partial\omega_i) \geq K \operatorname{diam}(\omega_i). \tag{2.12}$$
- Weight function and cover are compatible: There exists a constant $C_{\mathcal{W},L}$ such that for all cover patches ω_i
$$|\nabla W_i(x)| > \frac{C_{\mathcal{W},L}}{\operatorname{diam}(\omega_i)} \quad \text{holds for all } x \in \Omega \text{ with } \lambda_{C_\Omega}(x) > 1, \tag{2.13}$$
compare Figure 2.1.

Lemma 1. *The PU defined by* (2.9) *with weights* (2.10) *is valid according to Definition 1 under the assumptions* (2.11), (2.12), *and* (2.13).

Proof. For $x \in \Omega$ with $\lambda_{C_\Omega}(x) = 1$ we have $\nabla\varphi_i(x) = 0$. Note that we have

$$|S_i(x)| \geq |W_l(x)| = |W_l(x) - W_l(y)|$$

where ω_l denotes the cover patch with property (2.12) for $x \in \Omega$ and $y \in \partial\omega_l$ is arbitrary. For any $x \in \Omega$ with $\lambda_{C_\Omega} > 1$ we therefore obtain with the mean value theorem, (2.12) and (2.13)

[2] Other shapes of the cover patches $\omega_i \in C_\Omega$ are of course possible, e.g. balls or ellipsoids, but the resulting partition of unity functions φ_i are more challenging to integrate numerically. For instance a subdivision scheme based on the piecewise constant covering index λ_{C_Ω} leads to integration cells with very complicated geometry.

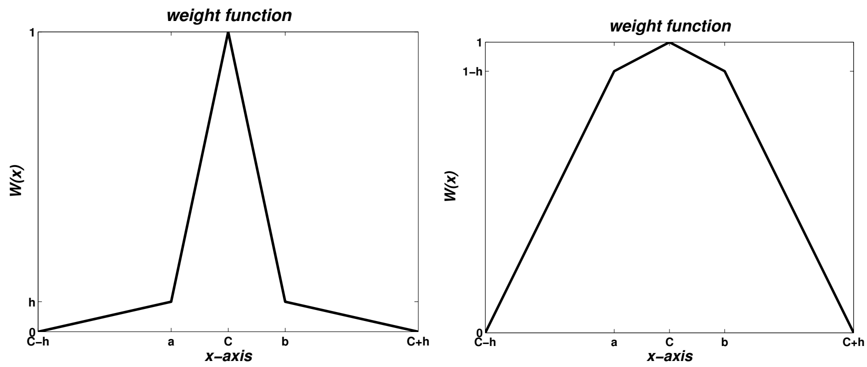

Figure 2.1. A one-dimensional weight function W_i on a patch $\omega_i = (C-h, C+h)$ with $\omega_{\mathrm{FT},i} = (a,b)$ that does not satisfy (left) the compatibility condition (2.13), and one that does (right).

$$|S_i(x)| \geq C_{\mathcal{W},L} K.$$

Together with (2.10) and (2.11) this yields the point-wise estimate

$$\begin{aligned}
|\nabla \varphi_i(x)| &= \Big| \frac{W_i(x)\nabla S_i(x) - \nabla W_i(x) S_i(x)}{S_i^2(x)} \Big| \\
&\leq \frac{\Big(|\nabla \mathcal{W} \circ T_i(x) DT_i(x) S_i(x)| + |W_i(x) \sum_k \nabla \mathcal{W} \circ T_k(x) DT_k(x)|\Big)}{|S_i^2(x)|} \\
&\leq (C_{\mathcal{W},L} K)^{-2} \frac{2MC_T C_{\mathcal{W},U}}{\mathrm{diam}(\omega_i)},
\end{aligned}$$

which gives the asserted bound $\|\nabla \varphi_i\|_{L^\infty(\mathbb{R}^d)} \leq \frac{C_\nabla}{\mathrm{diam}(\omega_i)}$ with $C_\nabla \geq \frac{2MC_T C_{\mathcal{W},U}}{(C_{\mathcal{W},L}K)^{-2}}$. $\square$

In general any local space which provides some approximation property such as (2.4) can be used in a PUM. Furthermore, the local approximation spaces are independent of each other. Hence, if there is a priory knowledge about the (local) behavior of the solution u available, it can be utilized to choose operator-dependent approximation spaces. For instance, the a priori information can be used to enrich a space of polynomials by certain singularities or it maybe used to choose systems of eigenfunction of (parts) of the considered differential operator as local approximation spaces. Such specialized local function spaces may be given analytically or numerically.

If no a priori knowledge about the solution is available classical multipurpose expansion systems like polynomials are used. In this paper we employ products of univariate Legendre polynomials throughout, i.e., we use

$$\begin{gathered}
V_i^{p_i}(\omega_i) := \mathcal{P}^{p_i} \circ \widetilde{T}_i, \quad \widetilde{T}_i : \omega_i \to (-1,1)^d \\
\mathcal{P}^{p_i}((-1,1)^d) = \mathrm{span}\langle \{\psi^n \mid \psi^n = \prod_{l=1}^d \mathcal{L}^{\hat{n}_l}, \|\hat{n}\|_1 = \sum_{l=1}^d \hat{n}_l \leq p_i\} \rangle,
\end{gathered}$$

where $\mathcal{L}^k$ denotes the Legendre polynomial of degree k.

2.2 Essential Boundary Conditions and Galerkin Discretization

The treatment of essential boundary conditions in meshfree methods is not straightforward and a number of different approaches have been suggested. In [10] we have presented how Nitsche's method [13] can be applied successfully in the meshfree context.

In order to formulate the resulting weak formulation of (1.1) arising from Nitsche's approach, we introduce some additional notation. Let $\partial_n u := \frac{\partial u}{\partial n}$ denote the normal derivative, $\Gamma_{D,i} := \omega_i \cap \Gamma_D$, and

$$C_{\Gamma_D} := \{\omega_i \in C_\Omega \mid \Gamma_{D,i} \neq \emptyset\}$$

denote the cover of the Dirichlet boundary. Furthermore, we define the cover-dependent norm

$$\|\partial_n u\|^2_{-\frac{1}{2},C_\Omega} := \sum_{\omega_i \in C_{\Gamma_D}} \operatorname{diam}(\Gamma_{D,i}) \|\partial_n u\|^2_{L^2(\Gamma_{D,i})}.$$

With these conventions we obtain the weak formulation

$$a_\beta(u,v) = l_\beta(v) \quad \text{for all } v \in V^{\mathrm{PU}} \tag{2.14}$$

with the cover-dependent bilinear form

$$a_\beta(u,v) := \int_\Omega \nabla u \nabla v - \int_{\Gamma_D} (\partial_n u v + u \partial_n v) + \beta \sum_{\omega_i \in C_{\Gamma_D}} \operatorname{diam}(\Gamma_{D,i})^{-1} \int_{\Gamma_{D,i}} uv \tag{2.15}$$

and the corresponding linear form

$$l_\beta(v) := \int_\Omega f v - \int_{\Gamma_D} g_D \partial_n v + \int_{\Gamma_N} g_N v + \beta \sum_{\omega_i \in C_{\Gamma_D}} \operatorname{diam}(\Gamma_{D,i})^{-1} \int_{\Gamma_{D,i}} g_D v \tag{2.16}$$

from the minimization of the functional

$$J_\beta(w) := \int_\Omega |\nabla w|^2 - 2\int_{\Gamma_D} \partial_n w w + \beta \sum_{\omega_i \in C_{\Gamma_D}} \operatorname{diam}(\Gamma_{D,i})^{-1} \int_{\Gamma_{D,i}} |w|^2. \tag{2.17}$$

Note that this minimization is completed for the error in V^{PU}, i.e., we are looking for $\min_{u^{\mathrm{PU}} \in V^{\mathrm{PU}}} J_{N,\beta}(u - u^{\mathrm{PU}})$. There is a unique solution u^{PU} if the regularization parameter β is chosen large enough; i.e., the regularization parameter is dependent on the discretization space V^{PU}. The solution u^{PU} of (2.14) satisfies an optimal error estimate if the space V^{PU} admits the following inverse estimate

$$\|\partial_n u\|_{-\frac{1}{2},C_\Omega} \leq C_{\mathrm{inv}} \|\nabla u\|_{L^2(\Omega)} \quad \text{for all } v \in V^{\mathrm{PU}} \tag{2.18}$$

with an absolute constant C_{inv} depending on the cover C_Ω, the generating weight function $\mathcal{W}$ and the employed local bases $\{\psi_i^n\}$. If C_{inv} is known, the regularization parameter β can be chosen as $\beta > 2C_{\text{inv}}^2$. Hence, the main task is the automatic computation of the constant C_{inv}. Fortunately, C_{inv}^2 can be approximated very efficiently, see [10]. To this end, we consider the inverse assumption (2.18) as a generalized eigenvalue problem locally on each patch $\omega_i \in C_{\Gamma_D}$ which intersects the Dirichlet boundary and solve for the largest eigenvalue to obtain an approximation to C_{inv}^2.

Note that this (overlapping) variant of Nitsche's approach is slightly different from the one employed in [10, 15], e.g., there the inverse assumption (2.18) was formulated using a cover-independent norm. To attain a convergent scheme from (2.17) it is essential that the covering index $\lambda_{C_{\Gamma_D}}(x) < M$ is bounded. The implementation of (2.15) and (2.16) is somewhat more involved since the numerical integration scheme must be capable of handling the overlap region correctly. The main advantage of this overlapping variant is that the regularization parameter β is dependent on the employed local approximation spaces, i.e., on the employed polynomial degrees, and on the maximal level difference L close to the boundary only. It is not dependent on $\text{diam}(\omega_i)$. Hence, it is sufficient to pre-compute β for the maximal allowable value of L and the maximal polynomial degree. This value can then be used for all patches on all levels.

For the Galerkin discretization of (2.14), which yields the linear system

$$A\tilde{u} = \hat{f}, \quad \text{with } A_{(i,k),(j,n)} = a_\beta(\varphi_j \psi_j^n, \varphi_i \psi_i^k), \text{ and } \hat{f}_{(i,k)} = l_\beta(\varphi_i \psi_i^k),$$

we need to employ numerical integration since the PU functions are in general piecewise rational functions. Note that the flat top property is also beneficial to the numerical integration since all PU functions are constant on each $\omega_{\text{FT,i}}$ so that the integration on a large part of the domain $\bigcup_i \omega_{\text{FT,i}} \subset \Omega$ involves only the local basis functions ψ_i^n. Therefore, a subdivision scheme based on the covering index λ_{C_Ω} which employs sparse grid numerical integration rules of higher order on the cover-dependent integration cells seems to be the best approach, see [8] for details. Note that the use of an automatic construction procedure for the numerical integration scheme is a must for adaptive computations since an a priori prescribed background integration scheme can hardly account for the (possibly) huge variation in the support sizes $\text{diam}(\omega_i)$ and may lead to stability problems.

With respect to the assembly of the system matrix A for a refined PUM space it is important to note that we do not need to compute all its entries $A_{(i,k),(j,n)}$. We can re-use the entries $A_{(i,k),(j,n)}$ which stem from a patch ω_i with the property that none of its neighbors $\omega_j \in C_{\Omega,i}$ have been refined. Hence, there are a number of complete block-rows $A_{(i,\cdot),(\cdot,\cdot)}$ that do not need to be computed for the refined space and we need to compute only a minimal number of matrix entries $A_{(i,k),(j,n)}$ from level to level.

3 Adaptive Multilevel Solution

In [8, 9] we have developed a tree-based cover construction scheme that gives a sequence of covers $\{C_\Omega^k\}$ based on a given point set $P = \{x_i\}$. The fundamental construction principle employed in [8] is a d-binary tree. Based on the given point data P, we sub-divide a bounding-box $\mathcal{C}_\Omega \supset \Omega$ until each of the tree cells

$$\mathcal{C}_i = \prod_{l=1}^{d} (c_i^l - h_i^l, c_i^l + h_i^l)$$

contains at most a single point $x_i \in P$, see Figure 3.2 (left). We obtain a valid cover from this tree by choosing

$$\omega_i = \prod_{l=1}^{d} (c_i^l - \alpha h_i^l, c_i^l + \alpha h_i^l), \quad \text{with } \alpha > 1. \tag{3.1}$$

Note that we define a cover patch ω_i and a corresponding PU function φ_i for cells that contain a point $x_i \in P$ as well as for *empty* cells that do not contain any point from P.[3] This procedure increases the dimension of the resulting PUM space, yet (under some assumptions) only by a constant factor [5, 8]. The main benefit of using a larger number of cover patches is that the resulting neighborhoods $C_{\Omega,i}$ are smaller and that we therefore obtain a smaller number of entries in the stiffness matrix. The coarser covers C_Ω^k are defined considering coarser versions of the constructed tree, i.e., by removing the complete set of leaves of the tree. For details of this construction see [8].

The constructed covers C_Ω^k all satisfy the conditions of the previous section.

Lemma 1. *A cover $C_\Omega = \{\omega_i\}$ arising from the stretching of a d-binary tree cell decomposition $\{\mathcal{C}_i\}$ according to* (3.1) *with $\alpha > 1$ satisfies conditions* (2.10), (2.11) *and* (2.12).

Proof. For the ease of notation let us assume $h_i = h_i^k$ for $k = 1, \dots, d$. Then, we have $h_i \asymp 2^{-l_i} \operatorname{diam}(\Omega)$ where l_i refers to the tree-level of the cell $\mathcal{C}_i$. Obviously, we have $C_T = 1$,

$$C_U = \max_{\omega_i \in C_\Omega} \max_{\omega_j \in C_{\Omega,i}} 2^{|l_i - l_j|}, \quad \text{and} \quad C_L = C_U^{-1}.$$

Due to the stretching of the tree cells we can find for any $x \in \Omega$ at least one cover patch ω_i such that $x \in \omega_i$ and that the inequality

$$\operatorname{dist}(x, \partial\omega_i) \geq \frac{\alpha - 1}{2} \min_{\omega_j \in C_{\Omega,i}} 2^{-l_j}$$

[3] This approach can be interpreted as a saturation technique for our d-binary tree. To this end, we can define an additional point set $\tilde{P} = \{\xi_i\}$ such that each cell of the tree now contains *exactly* one point of the union $\tilde{P} \cup P$, compare [8, 9].

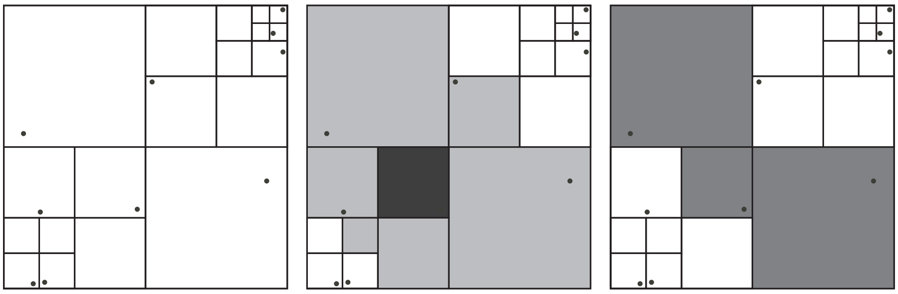

Figure 3.2. Subdivision corresponding to an initial cover (left). Subdivision with cells of neighborhood $C_{\Omega,i}$ (light gray) and cell corresponding to patch ω_i (dark gray) (center). Subdivision with cells of subset $R_{\Omega,i}$ (gray) (right).

holds. Hence, with the maximal difference in tree levels of two overlapping cover patches $L := \max_{\omega_i \in C_\Omega} \max_{\omega_j \in C_{\Omega,i}} |l_i - l_j|$ we obtain $C_U = 2^L$ and $K = (\alpha - 1)2^{-L-1}$. $\square$

Therefore, the resulting PU defined by (2.9) using a tensor product B-spline as generating weight function satisfies the assumptions of Definition 1 and the error estimates (2.5) and (2.6) hold for our multilevel PUM on each level k. Furthermore, we can easily enforce that each PU of the resulting sequence $\{\varphi_{i,k}\}_k$ has the flat top property.

Corollary 1. *The PU resulting from* (2.9) *based on a cover* $C_\Omega = \{\omega_i\}$ *arising from the stretching of a d-binary tree cell decomposition* $\{\mathcal{C}_i\}$ *with* $\alpha > 1$ *according to* (3.1) *has the flat top property if* $\alpha \in (1, 1 + 2^{-L})$.

3.1 Particle Refinement

In the following we consider the refinement of the given point set P and the respective sequence of covers C_Ω^k obtained from the tree construction reviewed above. One of the properties this refinement procedure should have is that we are able to bound the maximal level difference L of the resulting tree. Only then will we obtain a sequence of covers $\{C_\Omega^k\}$ and a sequence of PUs $\{\varphi_{i,k}\}$ which satisfy the conditions given above with *uniform* constants.

In general any local refinement procedure employs a Boolean refinement indicator function $r : C_\Omega \to \{\texttt{true}, \texttt{false}\}$ which identifies or marks regions which should be refined. Often the refinement indicator is based on thresholding of local error estimates $\eta_i \approx \|u - u^{\mathrm{PU}}\|_{H^1(\Omega\cap\omega_i)}$, e.g.

$$r(\omega_i) = \begin{cases} \texttt{true} & \text{if} \quad 2\eta_i \geq \eta_{\max} \\ \texttt{false} & \text{else} \end{cases}, \quad \text{with } \eta_{\max} := \max_{\omega_i \in C_\Omega} \eta_i. \tag{3.2}$$

Based on this refinement indicator we then employ certain refinement rules to improve the resolution on a patch ω_i. Since we are interested in the refinement of a particle set, these refinement rules must essentially create new particles in the regions $\{\omega_i \in C_\Omega \,|\, r(\omega_i) = \texttt{true}\}$. Furthermore, this refinement process should be consistent with our tree construction.

Before we consider our refinement rules for particles, let us first consider if a simple refinement indicator function like (3.2) is suitable for the PUM. To this end let us assume that we have $\eta_i \approx \|u - u^{\mathrm{PU}}\|_{H^1(\Omega\cap\omega_i)}$ an estimate of the error $u - u^{\mathrm{PU}}$ locally on each patch ω_i, see section 3.3 for the construction of such an error estimator. Note that the η_i are (overlapping) subdomain estimators which are fundamentally different from the more common (disjoint) element estimators in the FEM.

Recall that the local error is given by

$$\|u - u^{\mathrm{PU}}\|_{H^1(\Omega\cap\omega_i)} = \|\sum_{\omega_j \in C_{\Omega,i}} \varphi_j(u - u_j)\|_{H^1(\Omega\cap\omega_i)} \tag{3.3}$$

since $\sum_{\omega_j \in C_{\Omega,i}} \varphi_j(x) \equiv 1$ for all $x \in \omega_i$. Thus, using a simple indicator like (3.2) which would just mark the patch ω_i for refinement may not be sufficient to reduce the error on ω_i. It seems necessary that at least some of the neighbors are also marked for refinement to achieve a sufficient error reduction. Another reason why the highly local indicator (3.2) is not suitable for our PUM is the fact that we need to bound the maximal level difference L of neighboring patches. This can hardly be achieved using (3.2). A simple solution to this issue could be to refine all patches $\omega_j \in C_{\Omega,i}$ since they all contribute to the error on ω_i. This will certainly ensure that the error on ω_i is reduced, however, it may lead to a substantial yet unnecessary increase in computational work and storage. Taking a closer look at (3.3) we find that the contribution of a patch ω_j to the error on ω_i lives on the intersection $\omega_j \cap \omega_i$ only. Hence, it is a promising approach to select an appropriate subset of neighbors $\omega_j \in C_{\Omega,i}$ via the (relative) size of these intersections. This approach is further in perfect agreement with our constraint of bounding the maximal level difference since the intersection $\omega_j \cap \omega_i$ will be (relatively) large if $l_j < l_i$, i.e., ω_j is a coarser patch than ω_i. Hence, we introduce the sets

$$R_{\Omega,i} := \{\omega_i\} \cup \{\omega_j \in C_{\Omega,i} \,|\, l_j < l_i\}$$

of patches with a large contribution to the local error on ω_i. With the help of these sets we define our refinement indicator function as

$$r(\omega_j) = \begin{cases} \texttt{true} & \text{if} \quad \omega_j \in R_{\Omega,i} \text{ and } 2\eta_i \geq \eta_{\max} \\ \texttt{false} & \text{else} \end{cases}, \tag{3.4}$$

see Figure 3.2. Note however that this does not guarantee that the maximal level difference L stays constant. But it ensures that L increases somewhat slower and only in regions where the error is already small. Hence, the adverse effect of larger constants in the error bound (2.6) is almost negligible. Also note that the presented procedure can be interpreted as a refinement with implicit smoothing of the cover due to the selection of a subset of neighbors. This strategy gave very favorable results in all our numerical studies and it is employed in the numerical experiments presented in this paper.

Now that we have identified the refinement region, we need to consider the refinement rules for our particle set P based on the patch-wise refinement indicator function (3.4). Our goal is to define a set of refinement rules which create new points x_n and a respective cover $\tilde{C}_\Omega$, so that our original cover construction algorithm with the input $P \cup \{x_n\}$ will give the refined cover $\tilde{C}_\Omega$. Hence, the locations of the created points are constrained to certain cells of our tree. In this paper we employ a very simple and numerically cheap positioning scheme[4] for the new points based on a local center of gravity, which is defined as

$$g_i = \frac{1}{\mathrm{card}(G_{\Omega,i})} \sum_{x_k \in G_{\Omega,i}} x_k, \quad G_{\Omega,i} = \{x_k \in P \,|\, x_k \in \omega_k \in C_{\Omega,i}\}.$$

Note that g_i is well-defined for all patches. Due to our tree-based cover construction we can always find at least one given point x_i in the local neighborhoods $C_{\Omega,e}$ even for empty patches ω_e, i.e. $\omega_e \cap P = \emptyset$. Besides the local centers of gravity g_i we furthermore use the geometric centers c_i of the tree-cell $\mathcal{C}_i$ associated with the considered patch ω_i and the centers $c_{i,q}$ of the refined tree-cells $\mathcal{C}_{i,q}$ with $q = 1, \ldots, 2^d$ for our positioning scheme. The overall refinement scheme for a patch ω_i reads as follows.

Algorithm 1 (Particle Refinement).

1. Set counter $w = 0$.
2. If there is $x_i \in P$ with $x_i \in \mathcal{C}_i \subset \omega_i$, then determine sub-cell $\mathcal{C}_{i,\tilde{q}} \subset \omega_{i,\tilde{q}}$ with $x_i \in \mathcal{C}_{i,\tilde{q}}$.
3. If $g_i \in \mathcal{C}_i \subset \omega_i$, then determine sub-cell $\mathcal{C}_{i,\hat{q}} \subset \omega_{i,\hat{q}}$ with $g_i \in \mathcal{C}_{i,\hat{q}}$. Set $P = P \cup \{g_i\}$ and $w = w + 1$. If $w \geq 2^{d-1}$, then stop.
4. If $\hat{q} \neq \tilde{q}$, then
   For $q = 1, \ldots, 2^d$ compute projection $p_{i,q}$ of sub-cell center $c_{i,q}$ on line $\overline{x_i g_i}$ and the projection $\tilde{p}_{i,q}$ of $c_{i,q}$ on line $\overline{x_i c_i}$.
   If $p_{i,q} \in \mathcal{C}_{i,q}$, then set $P = P \cup \{p_{i,q}\}$ and $w = w + 1$. If $w \geq 2^{d-1}$, then stop.
   If $p_{i,q} \notin \mathcal{C}_{i,q}$ and $\tilde{p}_{i,q} \in \mathcal{C}_{i,q}$, then set $P = P \cup \{\tilde{p}_{i,q}\}$ and $w = w + 1$. If $w \geq 2^{d-1}$, then stop.
5. If $\hat{q} = \tilde{q}$ and $g_i = c_i$, then assume that data is gridded and set $P = P \cup \{c_{i,q}\}$ with $q = 1, \ldots, 2^d$.

[4] Note that many other approaches to the construction of new points are possible. For instance we can minimize the local fill distance or the separation radius under the constraint of positioning the new points within the sub-cells of the tree construction. Such approaches, however, involve the solution of a linear system and hence are computationally more expensive.

Now that we have our refined point set P, let us consider the question of how to define the respective cover patches ω_i. The refinement of a cover is straightforward since a d-binary tree is an adaptive data-structure. Here, it is sufficient to use a single subdivision step to split the tree-cell $\mathcal{C}_i$ into 2^d sub-cells $\mathcal{C}_{i,q}$ if the refinement indicator function $r(\omega_i) = \texttt{true}$ for the associated cover patch ω_i. Then, we insert the created particles in the respective cells and set the patches $\omega_{i,q}$ on all refined (sub-)cells $\mathcal{C}_{i,q}$ with $q = 1, \ldots, 2^d$ using (3.1) with the same overlap parameter α.

Due to our careful selection of the positions of the new points $\{x_n\}$ in our refinement scheme we ensure that the refined cover $\tilde{C}_\Omega$ is identical to the cover obtained by our original cover construction algorithm using the refined point set $P \cup \{x_n\}$ as input. Hence, a refined cover is guaranteed to have the same properties as the covers obtained from the original algorithm.

Recall that our tree-based cover construction algorithm provides a complete sequence of covers C_Ω^k. Hence, we must deal with the question how to introduce a refined cover into an existing sequence of covers C_Ω^k such that the resulting refined sequence is consistent with our multilevel construction [9, 15].

3.2 Iterative Solution

In [9] we have constructed a sequence of covers $\{C_\Omega^k = \{\omega_{i,k}\}\}$ with $k = 0, \ldots, J$ where J denotes the maximal subdivision level of the tree, that is all covers C_Ω^k have the property

$$k = \max_{\omega_i \in C_\Omega^k} l_i. \tag{3.5}$$

The respective PUM spaces V_k^{PU} are defined as

$$V_k^{\mathrm{PU}} := \sum_{\omega_{i,k} \in \Omega_k} \varphi_{i,k} V_{i,k}^{p_{i,k}}$$

with the PU functions (2.9) based on the cover $C_{\Omega,k}$ and local approximation spaces $V_{i,k}^{p_{i,k}}$ of degree $p_{i,k}$.

Note that property (3.5) ensures that we have a minimal number of levels $J+1$ and thus minimal work and storage in the iterative multilevel solver. Hence, the covers of a refined sequence must also satisfy (3.5).

Let us assume that we have a sequence of covers $\{C_\Omega^k\}$ with $k = 0, \ldots, J$ satisfying the level property (3.5) and that we refine the cover C_Ω^J. To obtain a sequence $\{C_\Omega^k\}$ with this property by the refinement scheme presented above we need to distinguish two cases.

First we consider the simple case where we refine the cover C_Ω^J in such a way that at least one patch ω_i with $l_i = J$ is marked for refinement. Then, the resulting cover $R(C_\Omega^J)$ has at least one element ω_j with $l_j = J+1$ and we can extend our sequence of covers $\{C_\Omega^k\}$ with $k = 0, \ldots, J+1$ where

$C_\Omega^{J+1} = R(C_\Omega^J)$. In the case where we refine only patches ω_i with $l_i < J$, we obtain a refined cover $R(C_\Omega^J)$ for which $\max_{\omega_i \in R(C_\Omega^J)} l_i = J$ holds and we cannot extend the existing sequence of covers by $R(C_\Omega^J)$. We rather need to replace the cover C_Ω^J by its refined version $R(C_\Omega^J)$ to obtain a consistent sequence of covers. Thus, we end up with the modified sequence $\{C_\Omega^k\}$ with $k = 0, \dots, J$ where we assign $C_\Omega^J = R(C_\Omega^J)$.

With these conventions it is clear that our refinement scheme leads to a sequence of covers $\{C_\Omega^k\}$ that satisfies all assumptions of our multilevel construction and our iterative multilevel solver is applicable also in adaptive computations.

To reduce the computational work even further, we couple our multilevel solver with the nested iteration technique [12], see Algorithm 2.

Algorithm 2 (Nested Iteration).

1. If $l > 0$, then set initial guess
   $\tilde{u}_l^0 = P_{l-1}^l \tilde{u}_{l-1}^{k_{l-1}}$.
   Else, set initial guess
   $\tilde{u}_l^0 = 0$.
2. Set $\tilde{u}_l^{k_l} \leftarrow \mathcal{IS}_l^{k_l}(\tilde{u}_l^0, \hat{f}_l, A_l)$.

The ingredients of a nested iteration are the basic iterative solution procedure $\mathcal{IS}_l$ (in our case $\mathcal{IS}_l$ will be a multilevel iteration $\mathcal{MG}_l(0, \hat{f}_J)$) defined on each level l and prolongation operators P_{l-1}^l. One key observation which lead to the development of Algorithm 2 is that the approximate solution $\tilde{u}_{l-1}^{k_{l-1}}$ obtained on level $l-1$ is a good initial guess $\tilde{u}_l^0$ for the iterative solution on level l. To this end, we need the prolongation operator P_{l-1}^l to transfer a coarse solution on level $l-1$ to the next finer level l, see step 1 of Algorithm 2. Another property that is exploited in our nested iteration solver is that there is nothing to gain from solving the linear system of equations (almost) exactly since its solution describes an approximate solution of the considered PDE only. The iterative solution process on level l can be stopped once the error of the iteration is of the same order as the discretization error on level l. Thus, if the employed iterative solver $\mathcal{IS}_l$ has a constant error reduction rate, as it is the case for an optimal multilevel iteration $\mathcal{MG}_l$, then a (very small) constant number of iterations k_l that is independent of l in step 2 is sufficient to obtain an approximate solution $\tilde{u}_l$ on each level l within discretization accuracy. The overall iterative process is also referred to as full multigrid [6, 11].

In all our numerical studies no more than 2 applications of a $V(1,1)$-cycle with block-Gauss-Seidel smoothing (compare [9]) were necessary to obtain an approximate solution within discretization accuracy.

3.3 Error Estimation

The final ingredient of our adaptive multilevel PUM is the local error estimator $\eta_i \approx \|u - u^{\mathrm{PU}}\|_{H^1(\Omega\cap\omega_i)}$ which steers our particle refinement process and can be used to assess the quality of the computed global approximation [2, 3, 16, 17]. In this section we now construct an error estimator η_i for the PUM based on the subdomain approach due to [2].

We employ an a posteriori error estimation technique based on the solution of local Dirichlet problems defined on (overlapping) subdomains introduced in [2] which is very natural to the PUM. To this end let us consider the additional local problems

$$\begin{aligned} -\Delta w_i &= f && \text{in } \Omega \cap \omega_i, \\ w_i &= u^{\mathrm{PU}} && \text{on } \partial(\Omega \cap \omega_i) \setminus \Gamma_N, \\ \frac{\partial w_i}{\partial n} &= g_N && \text{on } \Gamma_N \cap \partial(\Omega \cap \omega_i) \end{aligned} \tag{3.6}$$

to approximate the error $u - u^{\mathrm{PU}}$ on $\Omega \cap \omega_i$ by $w_i - u^{\mathrm{PU}} \in H^1(\Omega \cap \omega_i)$, see [2]. This leads to the local error estimator $\eta_i := \|w_i - u^{\mathrm{PU}}\|_{H^1(\Omega\cap\omega_i)}$.

Note that the local problems (3.6) employ inhomogeneous Dirichlet boundary values. As discussed in section 2.2, the implementation of essential boundary conditions is somewhat more involved in the PUM. There we have presented a non-conforming approach due to Nitsche to realize the global Dirichlet conditions of our model problem (1.1). Of course this technique can also be pursued here, however, since we consider (3.6) on very special subdomains, i.e., on the support of a PU function φ_i, there is a much simpler and conforming approach.

Enforcing homogeneous boundary conditions on the boundary $\partial\omega_i$ is trivial since $\varphi_i|_{\partial\omega_i} \equiv 0$. For patches close to the boundary we can easily enforce homogeneous boundary values on $\partial(\Omega\cap\omega_i)\setminus\Gamma_N\setminus\Gamma_D$. Hence, if we reformulate (3.6) in such a way that we have to deal with vanishing boundary data on $\partial(\Omega\cap\omega_i)\setminus\Gamma_N\setminus\Gamma_D$ only, we can realize the (artificial) boundary conditions in a conforming way. Only for the global boundary data on Γ_D we need to employ the non-conforming Nitsche technique. Therefore, we employ a discrete version of the following equivalent formulation of (3.6)

$$\begin{aligned} -\Delta \tilde{w}_i &= f - f^{\mathrm{PU}} && \text{in } \Omega \cap \omega_i, \\ \tilde{w}_i &= 0 && \text{on } \partial(\Omega \cap \omega_i) \setminus \Gamma_N \setminus \Gamma_D, \\ \tilde{w}_i &= g_D - u^{\mathrm{PU}} && \text{on } \partial(\Omega \cap \omega_i) \cap \Gamma_D, \\ \frac{\partial \tilde{w}_i}{\partial n} &= g_N - \frac{\partial u^{\mathrm{PU}}}{\partial n} && \text{on } \Gamma_N \cap \partial(\Omega \cap \omega_i), \end{aligned} \tag{3.7}$$

where f^{PU} denotes the best approximation of f in V^{PU}, with mostly homogeneous boundary conditions within our implementation.

We approximate (3.7) using the trial and test spaces $V_{i,*}(\Omega \cap \omega_i) := \varphi_i V_i^{p_i+q_i}$ with $q_i > 0$. Obviously, the functions $w_i \in V_{i,*}(\Omega \cap \omega_i)$ satisfy the homogeneous boundary conditions on $\partial(\Omega \cap \omega_i) \setminus \Gamma_N \setminus \Gamma_D$ due to the multiplication with the partition of unity function φ_i. Note that these local problems fit very well with the *global* Nitsche formulation (2.17) since the solution of (3.7) coincides with the minimizer of

$$J_{\gamma_i}(u - u^{\mathrm{PU}} - \tilde{w}_i) \to \min\{\tilde{w}_i \in V_{i,*}(\Omega \cap \omega_i)\}$$

where the parameter γ_i now depends on the local discretization space $V_{i,*}(\Omega\cap\omega_i)\subset H^1(\Omega\cap\omega_i)$ and not on $V^{\mathrm{PU}}\subset H^1(\Omega)$.[5] Note that the utilization of the global Nitsche functional is possible due to the use of a conforming approach for the additional boundary $\partial(\Omega\cap\omega_i)\setminus\Gamma_N\setminus\Gamma_D$ only.

We obtain our local approximate error estimator

$$\eta_i := \|\tilde{w}_i\|_{H^1(\Omega\cap\omega_i)} \tag{3.8}$$

from the approximate solution $\tilde{w}_i\in V_{i,*}(\Omega\cap\omega_i)$ of the local problem (3.7). The global error is then estimated by

$$\eta := \Big(\sum_{\omega_i\in C_\Omega}(\eta_i)^2\Big)^{\frac{1}{2}} = \Big(\sum_{\omega_i\in C_\Omega}\|\tilde{w}_i\|^2_{H^1(\Omega\cap\omega_i)}\Big)^{\frac{1}{2}}. \tag{3.9}$$

Note that we solve (3.7) in the complete space $V_{i,*}(\Omega\cap\omega_i)=\varphi_i V_i^{p_i+q_i}$ and not just the space $\varphi_i V_i^{p_i+q_i\setminus p_i}$ where $V_i^{p_i+q_i\setminus p_i}$ denotes the hierarchical complement of $V_i^{p_i}$ in $V_i^{p_i+q_i}$.

This subdomain error estimation approach was already analyzed in the PUM context in [1]. There it was shown that the subdomain estimator is efficient and reliable, i.e., there holds the equivalence

$$C^{-1}\sum_{\omega_i\in C_\Omega}\|w_i\|^2_{H^1(\Omega\cap\omega_i)} \le \|u-u^{\mathrm{PU}}\|^2_{H^1(\Omega)} \le C\sum_{\omega_i\in C_\Omega}\|w_i\|^2_{H^1(\Omega\cap\omega_i)}. \tag{3.10}$$

Yet, it was assumed that the variational problem is globally positive definite and that a globally conforming implementation of essential boundary conditions is employed. However, both these assumptions are not satisfied in our PUM due to the Nitsche approach. The analysis of the presented estimator is an open issue.

Also note that there are other a posteriori error estimation techniques based on the strong residual in Mortar finite elements based Nitsche's approach e.g. [4] which can be used in the PUM context. Finally, let us point out an interesting property of the PUM which might be beneficial in the construction of error estimators based on the strong residual. Recall from Theorem 1 that the global error is essentially given as an overlapping sum of the local errors with respect to the local approximation spaces. The properties of the PU required by Definition 1 enter in the constants of the estimates (2.5) and (2.6) only. They do not affect the attained approximation order. Hence, the global approximation error in a PUM is essentially invariant of the employed PU — if the PU is based on the same cover C_Ω.

Corollary 2. *Let $\Omega\subset\mathbb{R}^d$ be given. Let $\{\varphi_i^1\}$ and $\{\varphi_i^2\}$ be partitions of unity according to Definition 1 employing the same cover $C_\Omega=\{\omega_i\}$, i.e. for all i assume that $\omega_i^1=\omega_i^2=\omega_i$. Let us assume that $\lambda_{C_\Omega}(x)\le M\in\mathbb{N}$ for all $x\in\Omega$.*

[5] We may also pre-compute the Nitsche regularization parameter $\beta_{\max}$ for maximal *total* degree $\max p_i+q_i$ and employ $\beta_{\max}$ on all levels and for all local problems.

Let a collection of local approximation spaces $V_i^{p_i} = \operatorname{span}\langle\{\psi_i^n\}\rangle \subset H^1(\Omega\cap\omega_i)$ *be given as in Theorem 1. Let* $u \in H^1(\Omega)$ *be the function to be approximated. Then there hold the global equivalencies*

$$C_{1,2}^{-1}\|u^{\mathrm{PU},1} - u\|_{L^2(\Omega)} \leq \|u^{\mathrm{PU},2} - u\|_{L^2(\Omega)} \leq C_{2,1}\|u^{\mathrm{PU},1} - u\|_{L^2(\Omega)} \quad (3.11)$$

and

$$C_{1,2,\nabla}^{-1}\|\nabla(u^{\mathrm{PU},1}-u)\|_{L^2(\Omega)} \leq \|\nabla(u^{\mathrm{PU},2}-u)\|_{L^2(\Omega)} \leq C_{2,1,\nabla}\|\nabla(u^{\mathrm{PU},1}-u)\|_{L^2(\Omega)} \quad (3.12)$$

for the functions

$$u^{\mathrm{PU},1} := \sum_{\omega_i\in C_\Omega} \varphi_i^1 u_i \in V^{\mathrm{PU},1} \quad \text{and} \quad u^{\mathrm{PU},2} := \sum_{\omega_i\in C_\Omega} \varphi_i^2 u_i \in V^{\mathrm{PU},2}$$

with constants $C_{1,2}$*,* $C_{2,1}$*,* $C_{1,2,\nabla}$*,* $C_{2,1,\nabla}$ *depending on the partitions of unity only.*

Due to this equivalence it is easily possible to obtain an approximation $\tilde{u}^{\mathrm{PU}}$ with higher regularity $k > 0$ from a $\mathcal{C}^0$ approximation u^{PU} in our PUM simply be changing the employed generating weight function $\mathcal{W}$. The smoother approximation $\tilde{u}^{\mathrm{PU}}$ can for instance be used to evaluate/approximate higher order derivatives without the need to consider jumps or other discontinuities explicitly.

3.4 Overall Algorithm

Let us shortly summarize our overall adaptive multilevel algorithm which employs three user-defined parameters: $\epsilon > 0$ a global error tolerance, $q > 0$ the increment in the polynomial degree for the estimation of the error and $k > 0$ the number of multilevel iterations employed in the nested iteration.

Algorithm 3 (Adaptive Multilevel PUM).

1. Let us assume that we are given an initial point set P and that we have a sequence of PUM spaces $V_k^{\mathrm{PU}} = \sum_{\omega_{i,k}\in C_{\Omega,k}} \varphi_{i,k} V_{i,k}^{p_{i,k}}$ with $k = 0, \ldots, J$ based on a respective sequence of covers $C_{\Omega,k} = \{\omega_{i,k}\}$ arising from a d-binary tree construction using the point set P, see [8, 15] for details. Let $P_{l-1}^l : V_{l-1}^{\mathrm{PU}} \to V_l^{\mathrm{PU}}$ and $R_l^{l-1} : V_l^{\mathrm{PU}} \to V_{l-1}^{\mathrm{PU}}$ denote transfer operators and $\mathcal{S}_l : V_l^{\mathrm{PU}} \times V_l^{\mathrm{PU}} \to V_l^{\mathrm{PU}}$ appropriate smoothing schemes so that we can define a multilevel iteration $\mathcal{MG}_J : V_J^{\mathrm{PU}} \times V_J^{\mathrm{PU}} \to V_J^{\mathrm{PU}}$, see [7, 9, 15] for details. Set $\tilde{u}_J = \mathcal{MG}_J^{k_{\mathrm{init}}}(0, \hat{f}_J)$ where the number of iterations k_{init} is assumed to be large enough.
2. Compute the local error estimates η_i from (3.8) using the local spaces $\varphi_{J,i} V_{J,i}^{p_{J,k}+q}$. Estimate the global error by (3.9).
3. If the global estimate satisfies $\eta < \epsilon$: STOP.

4. Define the refinement indicator function (3.4) on the cover $C_{\Omega,J}$ based on the local estimates η_i.
5. Using the refinement rules of section 3.1 define a refined point set P, a refined cover $R(C_{\Omega,J})$ and its associated PUM space $R(V_J^{\mathrm{PU}})$.
6. If $R(C_{\Omega,J})$ satisfies the level property (3.5) with J:
   a) Delete the transfer operators P_{J-1}^J and R_J^{J-1}.
   b) Compute an intermediate transfer operator $\mathcal{P}_J : V_J^{\mathrm{PU}} \to R(V_J^{\mathrm{PU}})$.
   c) Set $\tilde{v}_J = \mathcal{P}_J \tilde{u}_J$.
   d) Delete the intermediate transfer operator $\mathcal{P}_J$.
   e) Remove the cover $C_{\Omega,J}$ and its associated PUM space V_J^{PU} from the respective sequences.
   f) Set $C_{\Omega,J} := R(C_{\Omega,J})$ and $V_J^{\mathrm{PU}} := R(V_J^{\mathrm{PU}})$.
7. If $R(C_{\Omega,J})$ satisfies the level property (3.5) with $J+1$:
   a) Extend the sequence of covers by $C_{\Omega,J+1} := R(C_{\Omega,J})$ and the sequence of PUM spaces by $V_{J+1}^{\mathrm{PU}} := R(V_J^{\mathrm{PU}})$.
   b) Set $\tilde{v}_J = 0$.
   c) Set $J = J+1$.
8. Set up the stiffness matrix A_J and right-hand side $\hat{f}_J$ using an appropriate numerical integration scheme.
9. Compute transfer operators $P_{J-1}^J : V_{J-1}^{\mathrm{PU}} \to V_J^{\mathrm{PU}}$ and $R_J^{J-1} : V_J^{\mathrm{PU}} \to V_{J-1}^{\mathrm{PU}}$ and define appropriate smoother $\mathcal{S}_J : V_J^{\mathrm{PU}} \times V_J^{\mathrm{PU}} \to V_J^{\mathrm{PU}}$ on level J.
10. If $\tilde{v}_J = 0$, set $\tilde{v}_J = P_{J-1}^J \tilde{u}_{J-1}$.
11. Apply $k > 0$ iterations and set $\tilde{u}_J = \mathcal{MG}_J^k(\tilde{v}_J^{\mathrm{PU}}, \hat{f}_J)$.
12. GOTO 2.

4 Numerical Results

In this section we present some results of our numerical experiments using the adaptive PUM discussed above. To this end, we introduce some shorthand notation for various error norms, i.e., we define

$$e_{L^\infty} := \frac{\|u - u^{\mathrm{PU}}\|_{L^\infty}}{\|u\|_{L^\infty}}, \quad e_{L^2} := \frac{\|u - u^{\mathrm{PU}}\|_{L^2}}{\|u\|_{L^2}}, \quad \text{and} \quad e_{H^1} := \frac{\|(u - u^{\mathrm{PU}})\|_{H^1}}{\|u\|_{H^1}}. \tag{4.1}$$

Analogously, we introduce the notion

$$e^*_{H^1} := \frac{\eta}{\|u\|_{H^1}} = \frac{\left(\sum_{\omega_i \in C_\Omega} \eta_i^2\right)^{\frac{1}{2}}}{\|u\|_{H^1}}$$

for the estimated (relative) error using (3.8) and (3.9). These norms are approximated using a numerical integration scheme with very fine resolution, see [15]. For each of these error norms we can compute the respective convergence rate ρ by considering the error norms of two consecutive levels $l-1$ and l

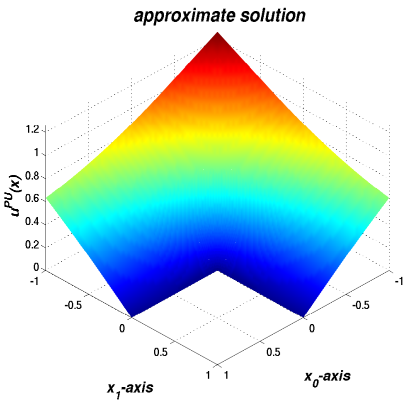

Figure 4.3. Surface plot of approximate solution u^{PU} on level $J = 11$.

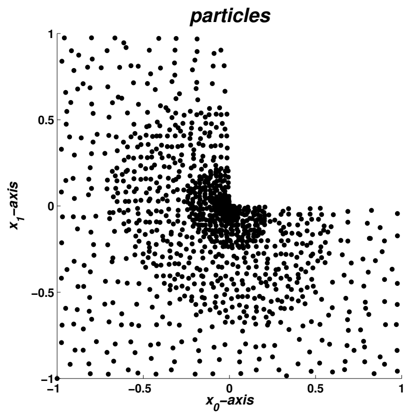

Figure 4.4. Refined point set P on level $J = 11$ for Example 1 using quadratic polynomials for error estimation.

$$\rho := -\frac{\log\left(\frac{\|u-u_l^{\mathrm{PU}}\|}{\|u-u_{l-1}^{\mathrm{PU}}\|}\right)}{\log(\frac{\mathsf{dof}_l}{\mathsf{dof}_{l-1}})}, \quad \mathsf{dof}_k := \sum_{\omega_{i,k}\in C_\Omega^k} \dim(V_{i,k}^{p_{i,k}}). \tag{4.2}$$

To assess the quality of our error estimator we give its effectivity index with respect to the H^1-norm

$$\epsilon_{H^1}^* := \frac{e_{H^1}^*}{e_{H^1}} = \frac{\eta}{\|(u - u^{\mathrm{PU}})\|_{H^1}}$$

in the tables. We also give the maximal subdivision level J of our tree for the cover construction, and the total number of degrees of freedom dof of the constructed PUM space V^{PU} on level J.

Example 1. In our first example we consider the standard test case of an L-shaped domain in two space dimensions with homogeneous boundary conditions at the re-entrant corner. That is we discretize the problem

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega = (-1,1)^2 \setminus [0,1)^2, \\ u &= g_D \quad \text{on } \partial\Omega \end{aligned}$$

with our adaptive PUM where we choose f and g_D such that the solution $u \in H^{\frac{3}{2}}(\Omega)$ in polar coordinates is given by $u(r,\theta) = r^{\frac{2}{3}} \sin(\frac{2\theta-\pi}{3})$, see Figure 4.3. We employ linear Legendre polynomials as local approximation spaces V_i^1 and estimate the local errors once with quartic (Table 4.1 and Figure 4.5) and once with quadratic (Table 4.2 and Figure 4.6) Legendre polynomials, i.e. $V_{i,*} = \varphi_i V_i^4$ and $V_{i,*} = \varphi_i V_i^2$ respectively.

It is well-known that in this two-dimensional example uniform refinement will yield a convergence rate of $\rho_{H^1} = \frac{1}{3}$ only instead of the optimal $\rho_{H^1} =$

Table 4.1. Relative errors e (4.1) and convergence rates ρ (4.2) for Example 1 using quartic Legendre polynomials for error estimation.

J	dof	e_{L^∞}	ρ_{L^∞}	e_{L^2}	ρ_{L^2}	e_{H^1}	ρ_{H^1}	$e^*_{H^1}$	$\rho^*_{H^1}$	$\epsilon^*_{H^1}$
0	3	3.303_{-1}	1.01	2.469_{-1}	1.27	5.272_{-1}	0.58	4.150_{-1}	0.80	0.79
1	9	1.162_{-1}	0.95	5.301_{-2}	1.40	2.153_{-1}	0.82	2.327_{-1}	0.53	1.08
2	36	7.431_{-2}	0.32	2.084_{-2}	0.67	1.572_{-1}	0.23	1.344_{-1}	0.40	0.85
3	144	4.628_{-2}	0.34	8.947_{-3}	0.61	1.034_{-1}	0.30	8.702_{-2}	0.31	0.84
4	252	2.908_{-2}	0.83	4.207_{-3}	1.35	7.291_{-2}	0.62	6.030_{-2}	0.66	0.83
5	360	1.830_{-2}	1.30	2.774_{-3}	1.17	5.632_{-2}	0.72	4.566_{-2}	0.78	0.81
6	468	1.152_{-2}	1.76	2.389_{-3}	0.57	4.821_{-2}	0.59	3.836_{-2}	0.66	0.80
7	576	7.252_{-3}	2.23	2.277_{-3}	0.23	4.457_{-2}	0.38	3.505_{-2}	0.43	0.79
8	1008	4.564_{-3}	0.83	1.164_{-3}	1.20	3.244_{-2}	0.57	2.554_{-2}	0.57	0.79
9	1566	2.874_{-3}	1.05	6.580_{-4}	1.29	2.524_{-2}	0.57	1.981_{-2}	0.58	0.78
10	2124	1.811_{-3}	1.52	5.282_{-4}	0.72	2.174_{-2}	0.49	1.701_{-2}	0.50	0.78
11	3636	1.140_{-3}	0.86	2.669_{-4}	1.27	1.635_{-2}	0.53	1.280_{-2}	0.53	0.78
12	5418	7.183_{-4}	1.16	1.834_{-4}	0.94	1.306_{-2}	0.56	1.023_{-2}	0.56	0.78
13	8226	4.525_{-4}	1.11	1.230_{-4}	0.96	1.063_{-2}	0.49	8.324_{-3}	0.49	0.78
14	13491	2.850_{-4}	0.93	6.996_{-5}	1.14	8.298_{-3}	0.50	6.493_{-3}	0.50	0.78
15	20412	1.796_{-4}	1.12	4.731_{-5}	0.94	6.618_{-3}	0.55	5.191_{-3}	0.54	0.78
16	30438	1.131_{-4}	1.16	3.305_{-5}	0.90	5.455_{-3}	0.48	4.277_{-3}	0.48	0.78
17	49842	7.125_{-5}	0.94	1.926_{-5}	1.09	4.288_{-3}	0.49	3.359_{-3}	0.49	0.78
18	77256	4.489_{-5}	1.05	1.225_{-5}	1.03	3.385_{-3}	0.54	2.657_{-3}	0.53	0.79
19	115326	2.828_{-5}	1.15	8.611_{-6}	0.88	2.786_{-3}	0.49	2.187_{-3}	0.49	0.78
20	189585	1.781_{-5}	0.93	5.119_{-6}	1.05	2.193_{-3}	0.48	1.720_{-3}	0.48	0.78
21	298440	1.122_{-5}	1.02	3.129_{-6}	1.09	1.719_{-3}	0.54	1.350_{-3}	0.53	0.79
22	446850	7.069_{-6}	1.14	2.201_{-6}	0.87	1.411_{-3}	0.49	1.109_{-3}	0.49	0.79
23	737478	4.453_{-6}	0.92	1.321_{-6}	1.02	1.110_{-3}	0.48	8.715_{-4}	0.48	0.78
24	1171548	2.805_{-6}	1.00	7.915_{-7}	1.11	8.672_{-4}	0.53	6.814_{-4}	0.53	0.79
25	1756818	1.767_{-6}	1.14	5.570_{-7}	0.87	7.109_{-4}	0.49	5.585_{-4}	0.49	0.79

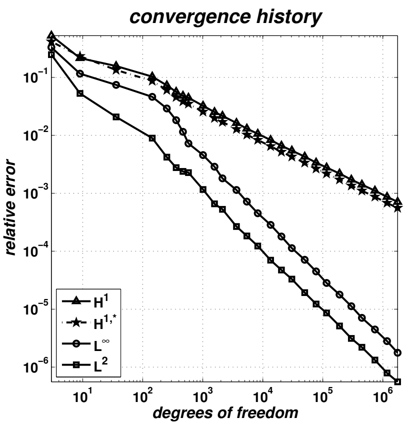

Figure 4.5. Convergence history for Example 1 using quartic polynomials for error estimation.

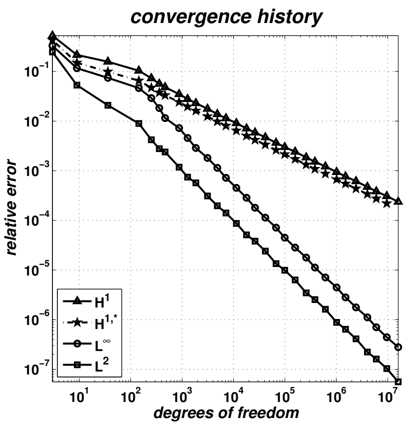

Figure 4.6. Convergence history for Example 1 using quadratic polynomials for error estimation.

Table 4.2. Relative errors e (4.1) and convergence rates ρ (4.2) for Example 1 using quadratic Legendre polynomials for error estimation.

J	dof	e_{L^∞}	ρ_{L^∞}	e_{L^2}	ρ_{L^2}	e_{H^1}	ρ_{H^1}	$e^*_{H^1}$	$\rho^*_{H^1}$	$\epsilon^*_{H^1}$
0	3	3.303_{-1}	1.01	2.469_{-1}	1.27	5.272_{-1}	0.58	4.287_{-1}	0.77	0.81
1	9	1.162_{-1}	0.95	5.301_{-2}	1.40	2.153_{-1}	0.82	1.496_{-1}	0.96	0.69
2	36	7.431_{-2}	0.32	2.084_{-2}	0.67	1.572_{-1}	0.23	9.882_{-2}	0.30	0.63
3	144	4.628_{-2}	0.34	8.947_{-3}	0.61	1.034_{-1}	0.30	6.502_{-2}	0.30	0.63
4	252	2.908_{-2}	0.83	4.207_{-3}	1.35	7.291_{-2}	0.62	4.696_{-2}	0.58	0.64
5	360	1.830_{-2}	1.30	2.774_{-3}	1.17	5.632_{-2}	0.72	3.753_{-2}	0.63	0.67
6	468	1.152_{-2}	1.76	2.389_{-3}	0.57	4.821_{-2}	0.59	3.306_{-2}	0.48	0.69
7	873	7.250_{-3}	0.74	1.181_{-3}	1.13	3.484_{-2}	0.52	2.418_{-2}	0.50	0.69
8	1278	4.566_{-3}	1.21	7.443_{-4}	1.21	2.778_{-2}	0.59	1.950_{-2}	0.56	0.70
9	1854	2.875_{-3}	1.24	5.693_{-4}	0.72	2.298_{-2}	0.51	1.615_{-2}	0.51	0.70
10	3123	1.811_{-3}	0.89	3.101_{-4}	1.16	1.765_{-2}	0.51	1.243_{-2}	0.50	0.70
11	4842	1.140_{-3}	1.05	1.963_{-4}	1.04	1.380_{-2}	0.56	9.784_{-3}	0.55	0.71
12	7146	7.184_{-4}	1.19	1.408_{-4}	0.85	1.136_{-2}	0.50	8.054_{-3}	0.50	0.71
13	11385	4.525_{-4}	0.99	8.684_{-5}	1.04	9.076_{-3}	0.48	6.422_{-3}	0.49	0.71
14	17937	2.851_{-4}	1.02	5.109_{-5}	1.17	7.102_{-3}	0.54	5.047_{-3}	0.53	0.71
15	26235	1.796_{-4}	1.22	3.806_{-5}	0.77	5.863_{-3}	0.50	4.173_{-3}	0.50	0.71
16	41598	1.131_{-4}	1.00	2.404_{-5}	1.00	4.710_{-3}	0.47	3.347_{-3}	0.48	0.71
17	67266	7.126_{-5}	0.96	1.344_{-5}	1.21	3.654_{-3}	0.53	2.602_{-3}	0.52	0.71
18	99162	4.489_{-5}	1.19	9.895_{-6}	0.79	2.999_{-3}	0.51	2.138_{-3}	0.51	0.71
19	157779	2.828_{-5}	0.99	6.324_{-6}	0.96	2.410_{-3}	0.47	1.714_{-3}	0.48	0.71
20	259047	1.781_{-5}	0.93	3.465_{-6}	1.21	1.861_{-3}	0.52	1.325_{-3}	0.52	0.71
21	383805	1.122_{-5}	1.18	2.532_{-6}	0.80	1.521_{-3}	0.51	1.085_{-3}	0.51	0.71
22	612792	7.070_{-6}	0.99	1.621_{-6}	0.95	1.220_{-3}	0.47	8.686_{-4}	0.47	0.71
23	1014804	4.454_{-6}	0.92	8.828_{-7}	1.20	9.396_{-4}	0.52	6.695_{-4}	0.52	0.71
24	1509102	2.806_{-6}	1.16	6.403_{-7}	0.81	7.659_{-4}	0.51	5.465_{-4}	0.51	0.71
25	2412603	1.767_{-6}	0.98	4.109_{-7}	0.95	6.143_{-4}	0.47	4.375_{-4}	0.47	0.71
26	4014459	1.113_{-6}	0.91	2.230_{-7}	1.20	4.723_{-4}	0.52	3.366_{-4}	0.51	0.71
27	5983155	7.014_{-7}	1.16	1.610_{-7}	0.82	3.844_{-4}	0.52	2.743_{-4}	0.51	0.71
28	9575469	4.419_{-7}	0.98	1.034_{-7}	0.94	3.082_{-4}	0.47	2.195_{-4}	0.47	0.71
29	15969915	2.784_{-7}	0.90	5.600_{-8}	1.20	2.368_{-4}	0.52			

$\frac{1}{2}$. An efficient self-adaptive method however must achieve the optimal rate $\rho_{H^1} = \frac{1}{2}$ and show a very sharp local refinement near the re-entrant corner, compare Figure 4.4. From the numbers displayed in Table 4.1 and the graphs depicted in Figure 4.5 we can clearly observe that our adaptive PUM achieves this optimal value of $\rho_{H^1} \approx \frac{1}{2}$. The corresponding L^2-convergence rate ρ_{L^2} and L^∞-convergence rate ρ_{L^∞} are also optimal with a value close to 1. We can also observe the convergence of the effectivity index $\epsilon^*_{H^1}$ to 0.79 from Table 4.1. Hence, we see that our approximation to the local error using quartic Legendre polynomials is rather accurate. The convergence of $\epsilon^*_{H^1}$ is clear numerical evidence that the subdomain estimator described in section 3.3 satisfies an equivalence such as (3.10) also for the non-conforming Nitsche approach and solutions with less than full elliptic regularity.

Of course, an approximation of the error estimator using quartic polynomials is rather expensive. We have to solve a local problem of dimension 15 on each patch. If we are interested in steering the refinement only, then this amount of computational work might be too expensive. Hence, we carried out the same experiment using quadratic Legendre polynomials for the approximation of (3.7) only. Here, we need to solve local problems of dimension 6. The measured errors and convergence rates are displayed in Table 4.2 and in Figure 4.6. From these numbers we can clearly observe that we retain the optimal rates of $\rho_{H^1} \approx \frac{1}{2}$ and $\rho_{L^2} \approx 1$ also with this coarser approximation. However, we also see that the quality of our approximate estimate is slightly reduced, i.e., the effectivity index converges to the smaller value 0.71. Furthermore, we find that our refinement scheme based on the quadratic approximation selects more patches for refinement than in the quartic case; e.g., on level 9 we have $\mathsf{dof} = 1854$ for the quadratic polynomials and only $\mathsf{dof} = 1566$ for the quartic polynomials. However, we obtain covers with a maximal level difference of $L = 1$ for both approximation to the local errors. Obviously, the use of the quadratic approximation for the error estimation leads to an unnecessary increase in the total number of degrees of freedom, however, since we attain optimal convergence rates with both approximations this cheaper approximation for the error may well pay off with respect to the total compute time.

The solution of the arising linear systems using our nested iteration multilevel solver required an almost negligible amount of compute time. In this experiment it was sufficient to employ a single $V(1,1)$-cycle with block-Gauss-Seidel smoothing (compare [9]) within the nested iteration solver to obtain an approximate solution within discretization accuracy.

Finally, let us point out that the obtained numerical approximations u^{PU} to the considered singular solution are highly accurate with $e_{L^\infty} \approx 10^{-7}$. Such quality requires an accurate and stable numerical integration scheme which can account for the sharp localization in the adaptive refinement and the singular character of the solution u automatically. Our subdivision sparse grid integration scheme meets these requirements.

In summary we can say that the results of this numerical experiment indicate that our adaptive PUM can handle problems with singular solutions with optimal complexity. We obtain a stable approximation with optimal convergence rates already with a relatively cheap approximation to the local errors. The application of a nested iteration with our multilevel method as inner iteration yields approximate solution of very high quality with a minimal amount of computational work.

Example 2. In our second example we consider our model-problem (1.1) with Dirichlet boundary conditions on the cube $(0,1)^3$ in three space dimensions. We choose f and g_D such that the solution is given by $u(x) = |x|^{\frac{1}{3}}$. Again, we use linear Legendre polynomials for the approximation and estimate the local errors using quadratic Legendre polynomials. The optimal convergence

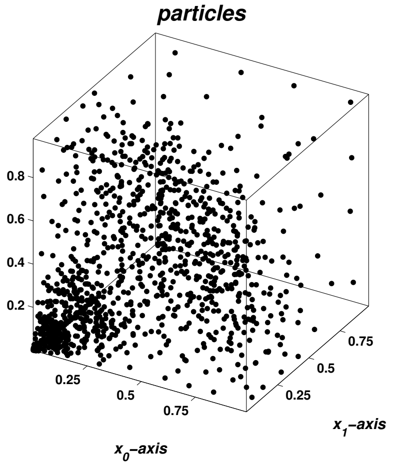

Figure 4.7. Refined point set P on level $J = 9$ for Example 2.

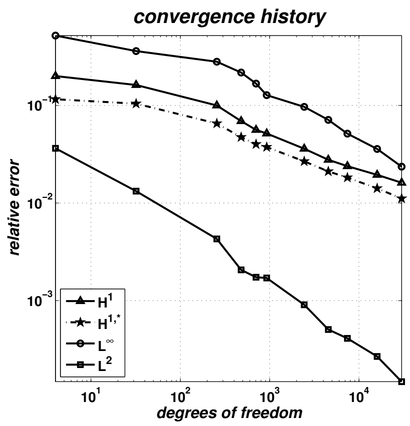

Figure 4.8. Convergence history for Example 2.

Table 4.3. Relative errors e (4.1) and convergence rates ρ (4.2) for Example 2.

J	dof	e_{L^∞}	ρ_{L^∞}	e_{L^2}	ρ_{L^2}	e_{H^1}	ρ_{H^1}	$e^*_{H^1}$	$\rho^*_{H^1}$	$\epsilon^*_{H^1}$
0	4	5.204_{-1}	0.47	3.635_{-2}	2.39	2.001_{-1}	1.16	1.157_{-1}	1.56	0.58
1	32	3.605_{-1}	0.18	1.323_{-2}	0.49	1.619_{-1}	0.10	1.043_{-1}	0.05	0.64
2	256	2.804_{-1}	0.12	4.285_{-3}	0.54	9.998_{-2}	0.23	6.530_{-2}	0.23	0.65
3	480	2.173_{-1}	0.41	2.061_{-3}	1.16	6.908_{-2}	0.59	4.728_{-2}	0.51	0.68
4	704	1.673_{-1}	0.68	1.733_{-3}	0.45	5.615_{-2}	0.54	4.003_{-2}	0.44	0.71
5	928	1.276_{-1}	0.98	1.703_{-3}	0.06	5.159_{-2}	0.31	3.745_{-2}	0.24	0.73
6	2440	9.678_{-2}	0.29	9.047_{-4}	0.65	3.600_{-2}	0.37	2.673_{-2}	0.35	0.74
7	4540	7.116_{-2}	0.50	5.046_{-4}	0.94	2.774_{-2}	0.42	2.108_{-2}	0.38	0.76
8	7424	5.140_{-2}	0.66	4.099_{-4}	0.42	2.386_{-2}	0.31	1.822_{-2}	0.30	0.76
9	15964	3.580_{-2}	0.47	2.673_{-4}	0.56	1.941_{-2}	0.27	1.412_{-2}	0.33	0.73
10	30076	2.355_{-2}	0.66	1.472_{-4}	0.94	1.613_{-2}	0.29	1.112_{-2}	0.38	0.69

rate with respect to the H^1-norm in three dimensions is $\rho_{H^1} = \frac{1}{3}$. From the numbers given in Table 4.3 we can clearly observe this optimal convergence behavior of our adaptive PUM. The rates ρ_{L^2} and ρ_{L^∞} obtained for the L^2-norm and L^∞-norm respectively are comparable to the optimal value of $\frac{2}{3}$. The effectivity index of our error estimator converges to a value of $\epsilon^*_{H^1} \approx 0.64$. Hence, the quality of the quadratic approximation to the error estimator in three dimensions is of comparable quality to that in two dimensions.

The maximal level difference in this example was $L = 1$ as in the previous example, see also Figure 4.7. Again, it was sufficient to use a single $V(1, 1)$-cycle within the nested iteration to obtain an approximate solution within discretization accuracy.

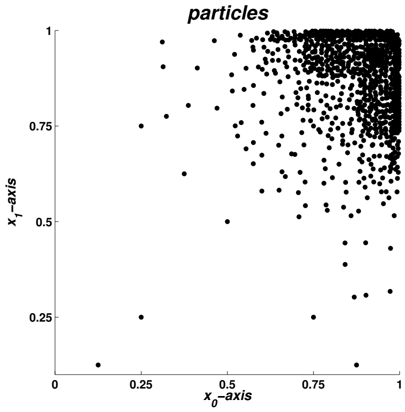

Figure 4.9. Refined point set P on level $J = 8$ for for Example 1 with $p = 1$.

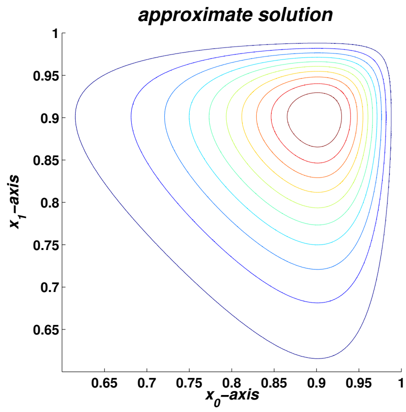

Figure 4.10. Zoom of isolines of the approximate solution computed on level $l = 10$ with $p = 1$.

Table 4.4. Relative errors e (4.1) and convergence rates ρ (4.2) for Example 3 using linear Legendre polynomials.

J	dof	e_{L^∞}	ρ_{L^∞}	e_{L^2}	ρ_{L^2}	e_{H^1}	ρ_{H^1}	$e^*_{H^1}$	$\rho^*_{H^1}$	$\epsilon^*_{H^1}$
3	84	5.853_{-1}	0.92	4.360_{-1}	1.24	6.941_{-1}	0.52	5.591_{-1}	0.45	0.81
4	156	2.940_{-1}	1.11	1.371_{-1}	1.87	4.764_{-1}	0.61	3.117_{-1}	0.94	0.65
5	291	1.041_{-1}	1.67	5.489_{-2}	1.47	3.281_{-1}	0.60	2.332_{-1}	0.47	0.71
6	822	3.772_{-2}	0.98	1.944_{-2}	1.00	2.042_{-1}	0.46	1.566_{-1}	0.38	0.77
7	1524	2.102_{-2}	0.95	1.428_{-2}	0.50	1.508_{-1}	0.49	1.178_{-1}	0.46	0.78
8	5619	4.824_{-3}	1.13	4.716_{-3}	0.85	7.710_{-2}	0.51	5.963_{-2}	0.52	0.77
9	13332	2.648_{-3}	0.69	1.805_{-3}	1.11	4.947_{-2}	0.51	3.841_{-2}	0.51	0.78
10	74838	3.061_{-4}	1.25	2.875_{-4}	1.06	2.106_{-2}	0.50	1.649_{-2}	0.49	0.78
11	275997	1.168_{-4}	0.74	8.154_{-5}	0.97	1.090_{-2}	0.50	8.520_{-3}	0.51	0.78
12	899823	2.819_{-5}	1.20	2.872_{-5}	0.88	6.002_{-3}	0.51	4.667_{-3}	0.51	0.78
13	2885646	9.056_{-6}	0.97	1.025_{-5}	0.88	3.358_{-3}	0.50	2.612_{-3}	0.50	0.78
14	13579752	1.841_{-6}	1.03	2.062_{-6}	1.04	1.546_{-3}	0.50	1.201_{-3}	0.50	0.78

Example 3. In our last example we consider the Poisson problem (1.1) with Dirichlet boundary conditions where we choose f and g_D such that the solution [14] is given by

$$u(x) = \frac{1}{2000} \prod_{l=1}^{2} (x^l)^2 (1 - x^l)^2 (\exp(10(x^l)^2) - 1),$$

see also Figure 4.10.

Here, we now consider not only a linear approximation but also a higher order approach with quadratic polynomials since the solution is smooth enough. First, we approximate the solution using linear Legendre polynomials and estimate the error with quadratic Legendre polynomials as before. Then, we

Table 4.5. Relative errors e (4.1) and convergence rates ρ (4.2) for Example 3 using quadratic Legendre polynomials.

J	dof	e_{L^∞}	ρ_{L^∞}	e_{L^2}	ρ_{L^2}	e_{H^1}	ρ_{H^1}	$e^*_{H^1}$	$\rho^*_{H^1}$	$\epsilon^*_{H^1}$
3	168	2.073_{-1}	1.49	9.459_{-2}	2.48	4.184_{-1}	0.58	2.375_{-1}	1.36	0.57
4	240	4.879_{-2}	4.06	3.147_{-2}	3.09	2.504_{-1}	1.44	1.527_{-1}	1.24	0.61
5	600	1.608_{-2}	1.21	1.100_{-2}	1.15	9.248_{-2}	1.09	6.380_{-2}	0.95	0.69
6	1824	2.492_{-3}	1.68	1.742_{-3}	1.66	2.523_{-2}	1.17	1.628_{-2}	1.23	0.65
7	4974	4.917_{-4}	1.62	2.885_{-4}	1.79	7.416_{-3}	1.22	4.774_{-3}	1.22	0.64
8	18492	7.648_{-5}	1.42	6.033_{-5}	1.19	1.854_{-3}	1.06	1.158_{-3}	1.08	0.62
9	61134	9.078_{-6}	1.78	6.262_{-6}	1.89	5.518_{-4}	1.01	3.351_{-4}	1.04	0.61
10	222414	1.392_{-6}	1.45	1.216_{-6}	1.27	1.512_{-4}	1.00	9.025_{-5}	1.02	0.60
11	959100	1.359_{-7}	1.59	1.163_{-7}	1.61	3.444_{-5}	1.01	2.028_{-5}	1.02	0.59
12	3580440	1.952_{-8}	1.47	1.608_{-8}	1.50	9.302_{-6}	0.99	5.451_{-6}	1.00	0.59
13	13422120	2.766_{-9}	1.48	2.321_{-9}	1.46	2.507_{-6}	0.99	1.467_{-6}	0.99	0.59

consider the case when we approximate the solution with quadratic polynomials and use cubic Legendre polynomials to estimate the errors locally. With respect to the measured convergence rates we expect to find the optimal rates $\rho_{H^1} \approx \frac{1}{2}$ for the linear approximation, see Table 4.4 and Figure 4.11, and $\rho_{H^1} \approx 1$ for the quadratic approximation, see Table 4.5 and Figure 4.12.[6] Our adaptive PUM achieves this anticipated optimal convergence behavior with respect to the H^1-norm as well as in the L^2-norm for which we find the optimal rates $\rho_{L^2} \approx 1$ and $\rho_{L^2} \approx \frac{3}{2}$ respectively. The cover refinement carried out for a linear approximation (compare Figure 4.9) is in fact different from the one attained for the quadratic approximation. For instance we find a maximal level difference of $L = 3$ for the linear approximation and $L = 1$ for the quadratic approximation, i.e., the point sets and covers obtain for the higher order approximation is somewhat smoother.

The quality of the quadratic approximation of the error estimator is again similar to those obtain in the previous examples, i.e., we observe $\epsilon^*_{H^1} \approx 0.78$. Since the relative increase in the number of degrees of freedom going from quadratic to cubic polynomials is smaller than when we use quadratic polynomials to estimate the error of a linear approximation we can expect to find a smaller value of $\epsilon^*_{H^1}$ in Table 4.5. In fact the effectivity index converges to 0.59 only. Note that the results of further numerical experiments confirm that the quality of the estimator is essentially influenced by the relative increase of the polynomial degree. For instance we found a value of $\epsilon^*_{H^1} \approx 0.8$ again, when we use polynomials of order 6 to approximate the error of a quadratic approximation. Hence, this example demonstrates that using only a polynomial of degree $p + 1$ to estimate the error of an approximation of order p may

[6] Note that not all refinement steps are given in the tables and graphs for Example 3. Due to the smoothness of the solution our refinement scheme constructs several refined covers $R(C^J_\Omega)$ with $J = \max_{\omega_i \in R(C^J_\Omega)} l_i$. For better readability we only give the final results on the respective level J.

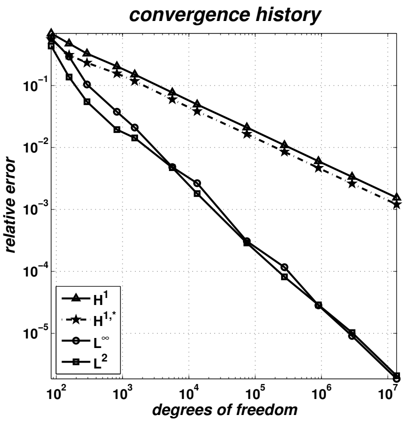

Figure 4.11. Convergence history for Example 3 using linear Legendre polynomials.

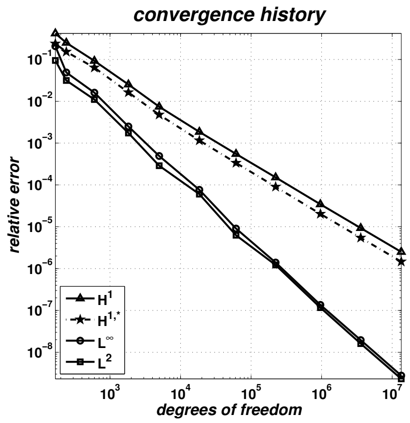

Figure 4.12. Convergence history for Example 3 using quadratic Legendre polynomials.

not yield very accurate estimates for large p. Nonetheless, our experiments also indicate that the actual refinement is only very slightly affected by this issue.

5 Concluding Remarks

In this paper we have considered the adaptive multilevel solution of a scalar elliptic PDE by the PUM. We have presented a particle refinement scheme for h-type adaptivity in the PUM which is steered by a local subdomain-type error estimator which is in turn approximated by local p-type enrichment. The results of our numerical experiments in two and three space dimensions are strong numerical evidence that the estimator is efficient and reliable.

Note that the adaptively constructed point sets may provide a novel way to approximate density distributions which can be related to solutions of PDEs. Also note that a local hp-type refinement is (in principle) straightforward in the PUM due to the independence of the local approximation spaces. The extension of the presented scheme to hp-type adaptivity however is subject of current research.

The nested multilevel iteration developed in this paper provides a highly efficient solver with optimal computational complexity. In summary, the presented meshfree scheme is a main step toward the availability of efficient adaptive meshfree methods which will enable us to tackle large scale complicated problems.

Acknowledgement. This work was supported in part by the Sonderforschungsbereich 611 *Singular phenomena and scaling in mathematical models* funded by the *Deutsche Forschungsgemeinschaft*.

References

1. I. Babuška and J. M. Melenk, *The Partition of Unity Method*, Int. J. Numer. Meth. Engrg., 40 (1997), pp. 727–758.
2. I. Babuška and W. C. Rheinboldt, *Error Estimates for Adaptive Finite Element Computations*, SIAM J. Numer. Anal., 15 (1978), pp. 736–754.
3. R. E. Bank and A. Weiser, *Some A Posteriori Error Estimators for Elliptic Partial Differential Equations*, Math. Comp., 44 (1985).
4. R. Becker, P. Hansbo, and R. Stenberg, *A Finite Element Method for Domain Decomposition with Non-Matching Grids*, Math. Modell. Numer. Anal., 37 (2003), pp. 209–225.
5. M. Bern, D. Eppstein, and J. Gilbert, *Provably Good Mesh Generation*, J. Comput. Sys. Sci., 48 (1994), pp. 384–409.
6. A. Brandt, *Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics*, tech. rep., GMD, 1984.
7. M. Griebel, P. Oswald, and M. A. Schweitzer, *A Particle-Partition of Unity Method—Part VI: A p-robust Multilevel Preconditioner*, in Meshfree Methods for Partial Differential Equations II, M. Griebel and M. A. Schweitzer, eds., vol. 43 of Lecture Notes in Computational Science and Engineering, Springer, 2005, pp. 71–92.
8. M. Griebel and M. A. Schweitzer, *A Particle-Partition of Unity Method—Part II: Efficient Cover Construction and Reliable Integration*, SIAM J. Sci. Comput., 23 (2002), pp. 1655–1682.
9. ———, *A Particle-Partition of Unity Method—Part III: A Multilevel Solver*, SIAM J. Sci. Comput., 24 (2002), pp. 377–409.
10. ———, *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.
11. W. Hackbusch, *Multi-Grid Methods and Applications*, vol. 4 of Springer Series in Computational Mathematics, Springer, 1985.
12. L. Kronsjø and G. Dahlquist, *On the Design of Nested Iterations for Elliptic Difference Equations*, BIT, 12 (1972), pp. 63–71.
13. 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.
14. N. Parés, P. Díez, and A. Huerta, *Subdomain-based flux-free a posteriori error estimators*, Comput. Meth. Appl. Mech. Engrg., 195 (2006), pp. 297–323.
15. 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.
16. R. Verfürth, *A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*, Wiley Teubner, 1996.
17. O. C. Zienkiewicz and J. Z. Zhu, *A Simple Error Estimator and Adapative Procedure for Practical Engineering Analysis*, Int. J. Numer. Meth. Engrg., (1987).