A PARTICLE-PARTITION OF UNITY METHOD–PART III: A MULTILEVEL SOLVER

MICHAEL GRIEBEL[†] AND MARC ALEXANDER SCHWEITZER[†]

Abstract. In this sequel to [15, 16] we focus on the efficient solution of the linear block-systems arising from a Galerkin discretization of an elliptic partial differential equation of second order with the partition of unity method (PUM). We present a cheap multilevel solver for partition of unity discretizations of any order. The shape functions of a PUM are products of piecewise rational partition of unity (PU) functions φ_i with $\text{supp}(\varphi_i) = \omega_i$ and higher order local approximation functions ψ_i^n (usually a local polynomial of degree $\leq p_i$). Furthermore, they are non-interpolatory. In a multilevel approach we not only have to cope with non-interpolatory basis functions but also with a sequence of nonnested spaces due to the meshfree construction. Hence, injection or interpolatory interlevel transfer operators are not available for our multilevel PUM. Therefore, the remaining natural choice for the prolongation operators are L^2-projections. Here, we exploit the partition of unity construction of the function spaces and a hierarchical construction of the PU itself to localize the corresponding projection problem. This significantly reduces the computational costs associated with the setup and the application of the interlevel transfer operators. The second main ingredient for our multilevel solver is the use of a block-smoother to treat the local approximation functions ψ_i^n for all n simultaneously. The results of our numerical experiments in two and three dimensions show that the convergence rate of the proposed multilevel solver is independent of the number of patches $\text{card}(\{\omega_i\})$. The convergence rate is slightly dependent on the local approximation orders p_i.

Key words. meshfree method, gridless discretization, partition of unity method, Galerkin method, multigrid method, multilevel method, nonnested spaces

AMS subject classifications. 65N55, 65N22, 65F10, 65N30

1. Introduction.

Meshfree methods are promising approaches to overcome the problem of mesh generation which still is the most time-consuming part of any finite element simulation. Meshfree methods are based only on a (finite) collection of independent points within the domain of interest, i.e. there are no fixed connections between any two points like in a conventional mesh. These points can now be used as collocation nodes [1, 12, 13, 14, 20], for the construction of approximate densities [21, 22, 23] or even for the construction of trial and test spaces for a Galerkin method [2, 3, 4, 11, 15].

Since meshfree methods are independent of a mesh, they are especially well-suited for problems with complex geometries or problems which require highly adaptive discretizations. Furthermore, meshfree methods are interesting for the treatment of time-dependent problems from a Lagrangian point of view [15].

The shape functions of a meshfree Galerkin method are in general more complex than finite element shape functions. In a meshfree method the shape functions are usually piecewise rational functions, whereas in a finite element method (FEM) they are piecewise polynomials. This makes the meshfree Galerkin discretization of a partial differential equation more challenging than its discretization with a FEM. See [15, 16] for details on the discretization process with the partition of unity method (PUM).

For the efficient solution of linear systems derived from grid-based discretizations multigrid [19] and multilevel methods [29] have been developed in the last 25 years. They exhibit an optimal complexity, i.e. the number of operations necessary to obtain

[†] Sonderforschungsbereich 256 *Nonlinear Partial Differential Equations*, Project D *Meshfree numerical methods for the simulation of 3D flows with free boundaries*, Institut für Angewandte Mathematik, Universität Bonn, Wegelerstr. 6, D-53115 Bonn, {griebel, schweitz}@iam.uni-bonn.de

the solution up to a prescribed accuracy is proportional to the number of unknowns of the linear system. Furthermore, the constant of proportionality is quite small. In a multigrid method we usually deal with nested grids $\Omega_0 \subset \Omega_1 \subset \ldots \subset \Omega_J$. Here J denotes the finest level of discretization. In a finite element setting we have the associated *nested* function spaces V_k

$$V_0 \subset V_1 \subset V_2 \subset \cdots \subset V_{J-1} \subset V_J,$$

with *interpolatory* basis functions $\phi_{i,k} \in V_k$. These two properties contribute significantly to the optimal convergence of multigrid methods and they are also the standard prerequisites in the respective convergence proofs.

The shape functions $\varphi_i \psi_i^n$ of a PUM space V^{PU} are products of a piecewise rational partition of unity function φ_i with $\mathrm{supp}(\varphi_i) = \omega_i$ and a higher order local approximation function ψ_i^n. These product functions are *non-interpolatory* due to the meshfree construction. Furthermore, this construction leads to a *nonnested* sequence

$$V_0^{\mathrm{PU}} \not\subset V_1^{\mathrm{PU}} \not\subset V_2^{\mathrm{PU}} \not\subset \cdots \not\subset V_{J-1}^{\mathrm{PU}} \not\subset V_J^{\mathrm{PU}},$$

of function spaces $V_k^{\mathrm{PU}} := \sum_i \varphi_{i,k} V_{i,k}^{p_{i,k}} = \sum_i \varphi_{i,k} \,\mathrm{span}\langle \psi_{i,k}^n \rangle$ in a multilevel setting. Hence, the construction of prolongation operators $I_{k-1}^k : V_{k-1}^{\mathrm{PU}} \to V_k^{\mathrm{PU}}$ and restriction operators $I_k^{k-1} : V_k^{\mathrm{PU}} \to V_{k-1}^{\mathrm{PU}}$ which connect the PUM spaces is not an easy task. Furthermore, the construction of these interlevel transfer operators has to respect the approximation orders $p_{i,k}$ of the local approximation spaces $V_{i,k}^{p_{i,k}} = \mathrm{span}\langle \psi_{i,k}^n \rangle$.

In this paper we present a multilevel solver for the large sparse linear block-systems arising from a (higher order) partition of unity discretization [2, 3, 15, 16] of an elliptic partial differential equation (PDE) of second order. The main ingredients of our multilevel solver are the use of a hierarchical construction algorithm for the sequence of partitions of unity $\{\varphi_{i,k}\}$ and an L^2-projection approach to the construction of prolongation operators I_{k-1}^k. Here, we not only exploit the structure of the PUM function space to localize this projection problem but also our hierarchical construction of $\{\varphi_{i,k}\}$ can be utilized to further reduce the computational costs of the setup of the interlevel transfer operators. Furthermore, we employ a block-smoother in our multilevel iteration to treat all local approximation functions $\psi_{i,k}^n$ simultaneously. The resulting multilevel iteration scheme converges with a rate ρ which is independent of the number of discretization points, yet ρ is slightly dependent on the local approximation orders.

The remainder of the paper is organized as follows: in §2 we give a short recap of the construction of a PUM function space and its fundamental properties. In §3 we review the Galerkin discretization of an elliptic PDE using a PUM function space. Then we give the basic ingredients for an abstract multilevel algorithm in §4. We present the multilevel construction for our partition of unity method in §5. Here, we utilize the hierarchical construction proposed in [16] for the definition of a sequence of partitions of unity $\{\varphi_{i,k}\}$ which leads (in general) to a sequence of nonnested trial and test spaces V_k^{PU}. Since a direct interpolation between two successive PUM spaces is not available we use an L^2-projection approach to construct prolongation operators. We exploit the structure of the PUM function spaces as well as the hierarchical construction of the sequence of partitions of unity $\{\varphi_{i,k}\}$ to localize the projections. This localization in turn significantly reduces the operation count and storage requirements associated with the interlevel transfer. The results of our numerical examples in two and three dimensions are given in §6. They show that the presented multilevel iteration scheme converges with a rate ρ which is independent of the number of discretization points and their distribution. Furthermore, we see from these results that the localization of the L^2-projection has no significant effect on the convergence behavior; i.e. the rates ρ of the multilevel iteration with the global—and expensive—L^2-projection are (almost) identical to those of the iteration which employs our very cheap localized L^2-projection. The convergence rates ρ are slightly dependent on the local approximation orders $p_{i,k}$, since the employed block-smoother only eliminates local couplings within $\omega_{i,k}$ but does not eliminate couplings between neighboring patches $\omega_{i,k} \cap \omega_{j,k} \neq \emptyset$. Finally, we conclude with some remarks in §7.

2. Partition of Unity Spaces. In the following, we give a short recap of how to construct partition of unity spaces for a meshfree Galerkin method, see [15, 16] for details. In a partition of unity method, we define a global approximation u^{PU} simply as a weighted sum of local approximations u_i,

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

These local approximations u_i are completely independent of each other, i.e. the local supports $\omega_i := \mathrm{supp}(u_i)$, the local basis $\{\psi_i^n\}$ and the order of approximation p_i for every single $u_i := \sum u_i^n \psi_i^n$ 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, see [15].

The starting point for any meshfree method is a collection of N independent points $P := \{x_i \in \mathbb{R}^d \,|\, x_i \in \overline{\Omega}, i = 1, \ldots, N\}$. In the PU approach we need to construct a partition of unity $\{\varphi_i\}$ on the domain of interest Ω to define an approximate solution (2.1) where the union of the supports $\mathrm{supp}(\varphi_i) = \overline{\omega_i}$ covers the domain $\overline{\Omega} \subset \bigcup_{i=1}^{N} \omega_i$ and $u_i \in V_i^{p_i}(\omega_i)$ is some locally defined approximation of order p_i to u on ω_i. Given a cover $C_\Omega = \{\omega_i \,|\, i = 1, \ldots, N\}$ we then can define such a partition of unity and local approximations u_i by using *Shepard functions* as φ_i and local approximation spaces $V_i^{p_i} = \mathrm{span}\langle \psi_i^n \rangle$ on the patches ω_i. The efficient construction of an appropriate cover C_Ω for general point sets P is not an easy task [27]. Throughout this paper we use a tree-based construction algorithm for rectangular covers presented in [16].

With the help of weight functions W_k defined on the patches ω_k of the cover C_Ω we can easily generate a partition of unity by Shepard's method, i.e. we define

$$\varphi_i(x) = \frac{W_i(x)}{\sum_{\omega_k \in C_\Omega^i} W_k(x)}, \tag{2.2}$$

where $C_i := \{\omega_j \in C_\Omega \,|\, \omega_i \cap \omega_j \neq \emptyset\}$ is the set of all geometric neighbors of a cover patch ω_i. We restrict ourselves to the use of cover patches ω_i which are d-rectangular, i.e. they are products of intervals $[x_i^l - h_i^l, x_i^l + h_i^l]$. Therefore, the most natural choice for a weight function W_i is a product of one-dimensional functions, i.e. $W_i\,(x) = \prod_{l=1}^{d} W_i^l\,(x^l) = \prod_{l=1}^{d} \mathcal{W}\,(\frac{x - x_i^l + h_i^l}{2h_i^l})$ with $\mathrm{supp}(\mathcal{W}) = [0, 1]$ such that $\mathrm{supp}(W_i) = \overline{\omega_i}$. It is sufficient for this construction to choose a one-dimensional weight function $\mathcal{W}$ which is non-negative. The partition of unity functions φ_i inherit the regularity of the generating weight function $\mathcal{W}$. We always use a normed B-spline [27] as the generating weight function $\mathcal{W}$.

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

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

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

In summary we can view the construction given above as follows

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

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

3. Galerkin Discretization. We want to solve elliptic boundary value problems of the type

$$\begin{array}{rcll} Lu & = & f & \text{in } \Omega \subset \mathbb{R}^d\,, \\ Bu & = & g & \text{on } \partial\Omega\,, \end{array}$$

where L is a symmetric partial differential operator of second order and B expresses suitable boundary conditions. For reasons of simplicity we consider in the following the model problem

$$\begin{array}{rcll} -\Delta u + u & = & f & \text{in } \Omega \subset \mathbb{R}^d\,, \\ \nabla u & = & g & \text{on } \partial\Omega\,, \end{array} \tag{3.1}$$

of Helmholtz type with natural boundary conditions. The Galerkin discretization of (3.1) leads to a definite linear system[1].

In the following let $a\,(\cdot,\cdot)$ be the continuous and elliptic bilinear form induced by L on $V := H^1(\Omega)$. We discretize the partial differential equation using Galerkin's method. Then, we have to compute the stiffness matrix

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

and the right hand side vector

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

[1] The implementation of Neumann boundary conditions with our partition of unity method is straightforward and similar to their treatment within the FEM. The realization of essential boundary conditions with meshfree methods is more involved than with a finite element method due to the non-interpolatory character of the meshfree shape functions. There are several different approaches to the implementation of essential boundary conditions with meshfree approximations, see [15, 18, 27]. The resulting linear systems may be indefinite, e.g. when we use Lagrangian multipliers to enforce the essential boundary conditions.

The integrands of the weak form of (3.1) may have quite a number of jumps of significant size since we use piecewise polynomial weights W_i whose supports ω_i overlap in the Shepard construction (2.2). Therefore, the integrals of the weak form have to be computed using an appropriate numerical quadrature scheme, see [15, 16].

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

1. The stiffness matrix A can be arranged in *spatial blocks*. A spatial block A_{nm} corresponds to a discretization of the PDE on the complete domain Ω using the trial functions $\varphi_j \psi_j^m$ and the test function $\varphi_i \psi_i^n$ with fixed n and m. Here, all blocks A_{nm} are sparse matrices and have the same row and column dimensions which corresponds to the number of partition of unity functions φ_i.
2. The stiffness matrix A may also be arranged in *polynomial blocks*. Here, a single block A_{ij} corresponds to a local discretization of the PDE on the domain $\omega_i \cap \omega_j \cap \Omega$. The polynomial blocks A_{ij} are dense matrices and may have different dimensions corresponding to the dimensions of the local approximation spaces $V_j^{p_j}$ and $V_i^{p_i}$.

The separation of degrees of freedom into local approximation functions ψ_i^n and partition of unity functions φ_i may also be utilized in the design of multilevel solvers, see §5.

The number of nonzeros of the stiffness matrix A is given by the number of neighbors $\operatorname{card}(N_i)$ of each cover patch ω_i and the local approximation order p_i, i.e. we have

$$\text{nonzeros}(A) \sim \sum_{\omega_i \in C_\Omega} \operatorname{card}(N_i) p_i^{2d},$$

whereas the number of degrees of freedom, i.e. the number of rows of A, is given by

$$\mathsf{dof} = \text{rows}(A) \sim \sum_{\omega_i \in C_\Omega} p_i^d.$$

4. Variational Multilevel Algorithm.

Now we consider the efficient solution of the large sparse linear (block-)system $A\tilde{u} = \hat{f}$. Of course, the computational work associated with it should be comparable to the computational work associated with the discretization process. We have to find a solver for the linear system which scales linearly with the number of unknown coefficients (i.e. actually the solver should scale with the number of nonzeros of A). It is well-known that the convergence rates ρ of classical iteration schemes like the Jacobi- or Gauß–Seidel-method grow with the number of unknowns. Also, the corresponding single-level preconditioners B result in condition numbers $\kappa(BA)$ that are dependent on the number of unknown coefficients. Hence, the computational costs during the solution of the linear system does *not* scale linearly with the number of unknowns. To overcome this problem multilevel techniques can be used. In the following we state the basic assumptions for a multilevel algorithm for a sequence of (nonnested) discretization spaces V_k.

1. Let $V_0, \dots, V_J$ be a sequence of (nonnested) finite dimensional vector spaces where V_J is the finest discretization space.
2. Assume that we have a linear prolongation operator $I_{k-1}^k : V_{k-1} \to V_k$ for $k = 1, \dots, J$.

3. Assume that we have a linear restriction operator $I_k^{k-1} : V_k \to V_{k-1}$ for $k = 1, \dots, J$.
4. Assume that we have a symmetric positive definite bilinear form $a(\cdot, \cdot)$ on the function space V and its respective representation A_k on the discretization spaces V_k for $k = 0, \dots, J$.
5. Assume that we have linear smoothing operators $S_k^{\text{pre}} : V_k \times V_k \to V_k$ and $S_k^{\text{post}} : V_k \times V_k \to V_k$ on the spaces V_k for $k = 1, \dots, J$.

We can then define an abstract multilevel algorithm:

Algorithm 1 (Multilevel Algorithm $M_\gamma^{\nu_1,\nu_2}(k, x_k, b_k)$).

```
1. if $k > 0$:
    (a) For $l = 1, \dots, \nu_1$: Set $x_k = S_k^{\text{pre}}(x_k, b_k)$.
    (b) Set $d_{k-1} := I_k^{k-1}(b_k - A_k x_k)$.
    (c) Set $e_{k-1} := 0$.
    (d) For $i = 1, \dots, \gamma$: $e_{k-1} = M_\gamma^{\nu_1,\nu_2}(k-1, e_{k-1}, d_{k-1})$.
    (e) Set $x_k = C_k(x_k, e_{k-1}) := x_k + I_{k-1}^k e_{k-1}$.
    (f) For $l = 1, \dots, \nu_2$: Set $x_k = S_k^{\text{post}}(x_k, b_k)$.
2. else:
    (a) Set $x_k = A_k^{-1} b_k$.
```

In the variational setting the prolongation operators I_{k-1}^k, i.e. their matrix representations, are used to transport coefficient vectors $\tilde{u}$, whereas the restriction operators I_k^{k-1} are used to transport moment vectors $\hat{f}$.

The iteration $M_\gamma^{\nu_1,\nu_2}$ is an optimal solver for discretizations of continuous problems with full elliptic regularity on nested grids if we have the approximation property for I_{k-1}^k, I_k^{k-1} and the smoothing property for S^{pre}, S^{post} [5]. A different convergence theory based on iterative subspace splittings and Schwarz theory was introduced in [8, 28]. This theory was also extended to the case of nonnested spaces V_k [7] where the variational assumption

$$a(I_{k-1}^k \phi_{j,k-1}, I_{k-1}^k \phi_{i,k-1}) = a(\phi_{j,k-1}, \phi_{i,k-1}) \tag{4.1}$$

and subsequently the Galerkin identity

$$A_{k-1} = I_k^{k-1} A_k I_{k-1}^k \tag{4.2}$$

are not valid. The general convergence theory developed in [7] is based on the weaker assumption

$$a(I_{k-1}^k \phi_{j,k-1}, I_{k-1}^k \phi_{i,k-1}) \leq C a(\phi_{j,k-1}, \phi_{i,k-1}) \tag{4.3}$$

besides further conditions on the regularity of the underlying problem and the approximation properties of the transfer operators. In [24] a convergence theory for additive multilevel iterations is presented which exploits an estimate of the growth of the iterated prolongations in the energy norm instead of a two-level estimate like (4.3). Nonconforming multigrid methods, where special prolongations and restrictions for a certain element are constructed (see [9] and the references cited therein), are special cases of the general nonnested situation.

We now apply the general PUM construction given in §2 to every point set of a nonnested sequence of point sets $P_k = \{x_{i,k}\}$ to define a sequence of function spaces V_k^{PU}. Hence, we not only have to deal with the general situation of nonnested spaces but also with non-interpolatory shape functions on every level. Our multilevel construction can be visualized with the diagram

$$\begin{tikzcd}
\begin{pmatrix}\{x_{i,0}\}\\ \mathcal{W}\\ \{p_{i,0}\}\end{pmatrix} \arrow[r] \arrow[d, shift right] \arrow[d, shift left, <-] & \begin{pmatrix}\{\omega_{i,0}, W_{i,0}\}\\ \{V_{i,0}^{p_{i,0}}\}\end{pmatrix} \arrow[r] \arrow[d, shift right] \arrow[d, shift left, <-] & \begin{pmatrix}\{\varphi_{i,0}\}\\ \{V_{i,0}^{p_{i,0}}\}\end{pmatrix} \arrow[r] \arrow[d, shift right] \arrow[d, shift left, <-] & V_0^{\mathrm{PU}} = \sum \varphi_{i,0} V_{i,0}^{p_{i,0}} \arrow[d, shift right] \arrow[d, shift left, <-] \\
\vdots \arrow[d, shift right] \arrow[d, shift left, <-] & \vdots \arrow[d, shift right] \arrow[d, shift left, <-] & \vdots \arrow[d, shift right] \arrow[d, shift left, <-] & \vdots \arrow[d, shift right] \arrow[d, shift left, <-] \\
\begin{pmatrix}\{x_{i,k}\}\\ \mathcal{W}\\ \{p_{i,k}\}\end{pmatrix} \arrow[r] \arrow[d, shift right] \arrow[d, shift left, <-] & \begin{pmatrix}\{\omega_{i,k}, W_{i,k}\}\\ \{V_{i,k}^{p_{i,k}}\}\end{pmatrix} \arrow[r] \arrow[d, shift right] \arrow[d, shift left, <-] & \begin{pmatrix}\{\varphi_{i,k}\}\\ \{V_{i,k}^{p_{i,k}}\}\end{pmatrix} \arrow[r] \arrow[d, shift right] \arrow[d, shift left, <-] & V_k^{\mathrm{PU}} = \sum \varphi_{i,k} V_{i,k}^{p_{i,k}} \arrow[d, shift right] \arrow[d, shift left, <-] \\
\vdots \arrow[d, shift right] \arrow[d, shift left, <-] & \vdots \arrow[d, shift right] \arrow[d, shift left, <-] & \vdots \arrow[d, shift right] \arrow[d, shift left, <-] & \vdots \arrow[d, shift right] \arrow[d, shift left, <-] \\
\begin{pmatrix}\{x_{i,J}\}\\ \mathcal{W}\\ \{p_{i,J}\}\end{pmatrix} \arrow[r] & \begin{pmatrix}\{\omega_{i,J}, W_{i,J}\}\\ \{V_{i,J}^{p_{i,J}}\}\end{pmatrix} \arrow[r] & \begin{pmatrix}\{\varphi_{i,J}\}\\ \{V_{i,J}^{p_{i,J}}\}\end{pmatrix} \arrow[r] & V_J^{\mathrm{PU}} = \sum \varphi_{i,J} V_{i,J}^{p_{i,J}}
\end{tikzcd}$$

where all interlevel transfer operators have to cope with the nonnestedness of the point sets and of the function spaces. The construction of a sequence of point sets $P_k = \{x_{i,k}\}$ from a given initial point set $\tilde{P} = \{x_i \,|\, i = 1, \ldots, \tilde{N}\}$ is presented in §5. There we also develop appropriate prolongation operators I_{k-1}^k for the resulting PUM spaces V_k^{PU} of any order.

Prior to our multilevel PUM construction we give a short analysis of the computational costs associated with the abstract multilevel iteration scheme $M_\gamma^{\nu_1,\nu_2}$.

4.1. Computational Costs. The number of operations $\mathcal{C}_{M_\gamma^{\nu_1,\nu_2}}$ associated with the abstract multilevel iteration $M_\gamma^{\nu_1,\nu_2}$ can be estimated with the help of the average number of nonzeros per degree of freedom for the operator matrix A_k, the prolongation matrix I_{k-1}^k and the restriction matrix I_k^{k-1}, i.e. with the constants

$$\begin{aligned}\mathcal{C}(A_k) \quad := \quad \frac{\text{nonzeros}(A_k)}{\mathsf{dof_k}}, \quad \mathcal{C}(I_{k-1}^k) \quad &:= \quad \frac{\text{nonzeros}(I_{k-1}^k)}{\mathsf{dof_k}} \\ \text{and} \quad \mathcal{C}(I_k^{k-1}) := \frac{\text{nonzeros}(I_k^{k-1})}{\mathsf{dof_k}}&.\end{aligned}$$

These averages $\mathcal{C}(A_k)$, $\mathcal{C}(I_{k-1}^k)$ and $\mathcal{C}(I_k^{k-1})$ are also the relevant measures for the storage requirement of the method.

With these estimates for the matrices we can estimate the operation counts per unknown coefficient on level k. For the defect computation (step 1b) and the correction step 1e we get the estimates

$$\mathcal{C}_{D,k} = 2\mathcal{C}(I_k^{k-1})\big(2\mathcal{C}(A_k) + 1\big) \quad \text{and} \quad \mathcal{C}_{C,k} = 1 + 2\mathcal{C}(I_{k-1}^k).$$

The operation count $\mathcal{C}_{S,k}$ for the smoothing steps 1d and 1f is given by

$$\mathcal{C}_{S,k} = 2\big(\nu_1 + \nu_2\big)\big(\mathcal{C}(A_k) - 1\big)$$

if we apply Gauß–Seidel smoothing.

With these stepwise operation counts we can bound the number of operations per iteration of Algorithm 1 on level k by

$$\mathsf{dof}_k \mathcal{C}^k_{M_\gamma^{\nu_1,\nu_2}} \leq \mathsf{dof}_k(\mathcal{C}_{D,k} + \mathcal{C}_{C,k} + \mathcal{C}_{S,k}) + \gamma \mathsf{dof}_{k-1} \mathcal{C}^{k-1}_{M_\gamma^{\nu_1,\nu_2}}, \tag{4.4}$$

where $\mathcal{C}^k_{M_\gamma^{\nu_1,\nu_2}}$ denotes the number of operations per unknown on level k. If we assume that the averages $\mathcal{C}(A_k)$, $\mathcal{C}(I_k^{k-1})$ and $\mathcal{C}(I_{k-1}^k)$ can be bounded independent of the level, i.e. $\mathcal{C}(A_k) \leq \mathcal{C}_A$, $\mathcal{C}(I_k^{k-1}) \leq \mathcal{C}_R$ and $\mathcal{C}(I_{k-1}^k) \leq \mathcal{C}_P$ for all levels k, then the estimates for the number of operations given above are also independent of the level k, i.e. $\mathcal{C}_{D,k} \leq \mathcal{C}_D$, $\mathcal{C}_{C,k} \leq \mathcal{C}_C$ and $\mathcal{C}_{S,k} \leq \mathcal{C}_S$. Hence, the summation of (4.4) over all levels gives the estimate

$$\mathcal{C}^J_{M_\gamma^{\nu_1,\nu_2}} \leq (\mathcal{C}_D + \mathcal{C}_C + \mathcal{C}_S)\left(1 + \sum_{k=1}^{J-1} \gamma^k \frac{\mathsf{dof}_{J-k}}{\mathsf{dof}_J}\right) + \gamma^J \frac{\mathsf{dof}_0}{\mathsf{dof}_J}\mathcal{C}_0. \tag{4.5}$$

Therefore, one iteration of the multilevel algorithm $M_\gamma^{\nu_1,\nu2}$ is of linear complexity with respect to dof_J if the series

$$\sum_{k=0}^{J} \gamma^k \frac{\mathsf{dof}_{J-k}}{\mathsf{dof}_J} < \infty \quad \text{for} \quad J \to \infty$$

converges.

5. Multilevel Partition of Unity Method. In this section we present the multilevel construction of PUM spaces V_k^{PU} and appropriate prolongation operators $I_{k-1}^k : V_{k-1}^{\mathrm{PU}} \to V_k^{\mathrm{PU}}$. Recall from §2 that a general PUM space V^{PU} is defined as

$$V^{\mathrm{PU}} := \sum_{i=1}^{N} \varphi_i V_i^{p_i},$$

where $\{\varphi_i\}$ is a partition of unity based only on a set of points $P = \{x_i \in \overline{\Omega}\}$ with $\mathrm{card}(P) = N$ and the $V_i^{p_i}$ are local approximation spaces of degree p_i defined on the supports $\omega_i = \mathrm{supp}(\varphi_i)$. Due to this separation of the degrees of freedom in *h-type components* ($\{\varphi_i\}$) and *p-type components* ($V_i^{p_i}$) we can define two abstract multilevel constructions.

Polynomial Multilevel Approach. Here, we keep the PU fixed and define local hierarchies for the local spaces $V_i^{p_i}$. Since these local spaces are polynomial spaces we can easily define a nested sequence of local spaces by

$$V_i^0 \subset V_i^1 \subset V_i^2 \subset \cdots \subset V_i^{p_i}$$

embedding lower order spaces V_i^l into $V_i^{p_i}$. With this choice we can even define a direct splitting of the local spaces $V_i^{p_i}$ and the partitioning of the stiffness matrix into spatial blocks given above can be interpreted as an implementation of this direct splitting.

This p-multilevel approach though suffers from several drawbacks concerning the optimal complexity of the resulting solver. The reduction in the number of degrees of freedom is (in general) level-dependent and it is close to one. In the context of Schwarz methods we can cure this problem by coarsening directly to the $p_i = 0$ case [25] and by limiting ourselves to a polynomial two-level approach. In the multilevel

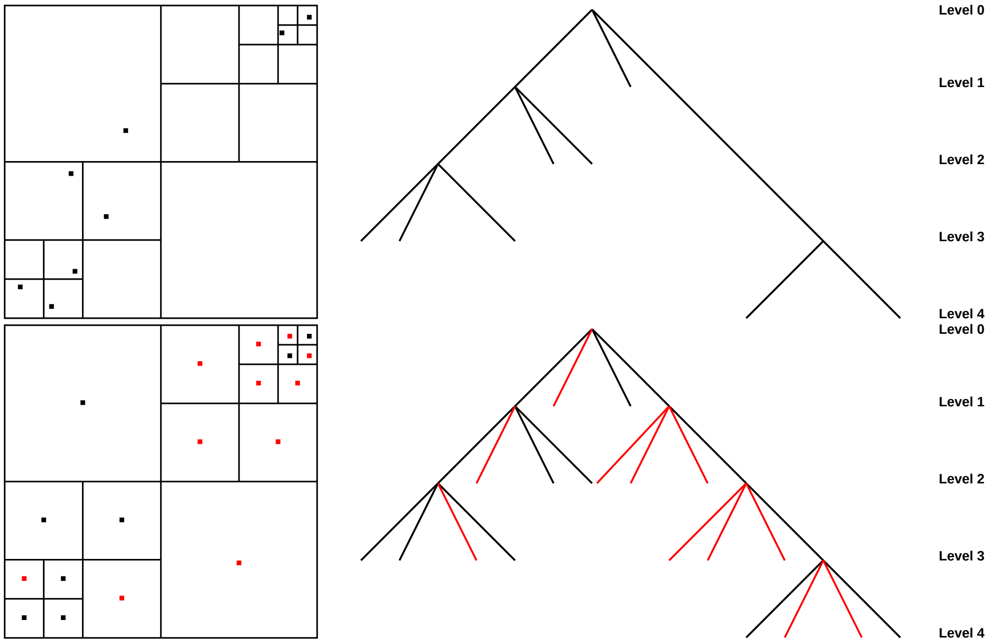

FIGURE 5.1. *Hierarchical cover construction with Algorithm 2 in two dimensions. The initial cell decomposition induced by $\tilde{P}$ (upper left) and its corresponding tree representation (upper right) after step 2 of Algorithm 2. The final cell decomposition with the cell center point set P (lower left) and its tree representation (lower right) after the completion of Algorithm 2.*

as well as the two-level approach the problem remains that the coarsest linear system is of dimension $N = \operatorname{card}(P)$, i.e. it is of non-constant size and its condition number is also not constant. Hence, we still need an optimal solver for the spatial degrees of freedom.

Spatial Multilevel Approach. Here, we "fix" the local approximation spaces $V_i^{p_i}$ and work only with the h-components of our PUM space, i.e. the partition of unity $\{\varphi_i\}$. Due to the overlap of the support patches ω_i, the definition of a nested sequence of function spaces is usually not possible. Furthermore, the meshfree construction—where we have no fixed connections between any two points—does not lead to a natural ordering or hierarchy of the partition of unity functions φ_i. In [16] a hierarchical cover construction algorithm was proposed which simplifies this problem substantially. Due to the construction principles given in §2 we only need to specify a sequence of point sets $P_k = \{x_{i,k}\}$ (besides the generating weight function $\mathcal{W}$ and the polynomial degrees $p_{i,k}$) for our multilevel PUM setup. The sequences of covers $C_\Omega^k := \{(\omega_{i,k}, W_{i,k}) \,|\, W_{i,k} : \mathbb{R}^d \to \mathbb{R}, \operatorname{supp}(W_{i,k}) = \omega_{i,k}\}$, partitions of unity $\{\varphi_{i,k}\}$ and PUM spaces V_k^{PU} are then constructed according to our single level PUM construction

$$\begin{pmatrix} P_k = \{x_{i,k}\} \\ \mathcal{W} \\ \{p_{i,k}\} \end{pmatrix} \to \begin{pmatrix} \{\omega_{i,k}, W_{i,k}\} \\ \{V_{i,k}^{p_{i,k}}\} \end{pmatrix} \to \begin{pmatrix} \{\varphi_{i,k}\} \\ \{V_{i,k}^{p_{i,k}}\} \end{pmatrix} \to V_k^{\mathrm{PU}} = \sum \varphi_{i,k} V_{i,k}^{p_{i,k}}$$

for all $k = 0, \ldots, J$.

The cover construction is the most crucial step in a PUM. The cover has a significant impact on the computational costs associated with the assembly of the stiffness

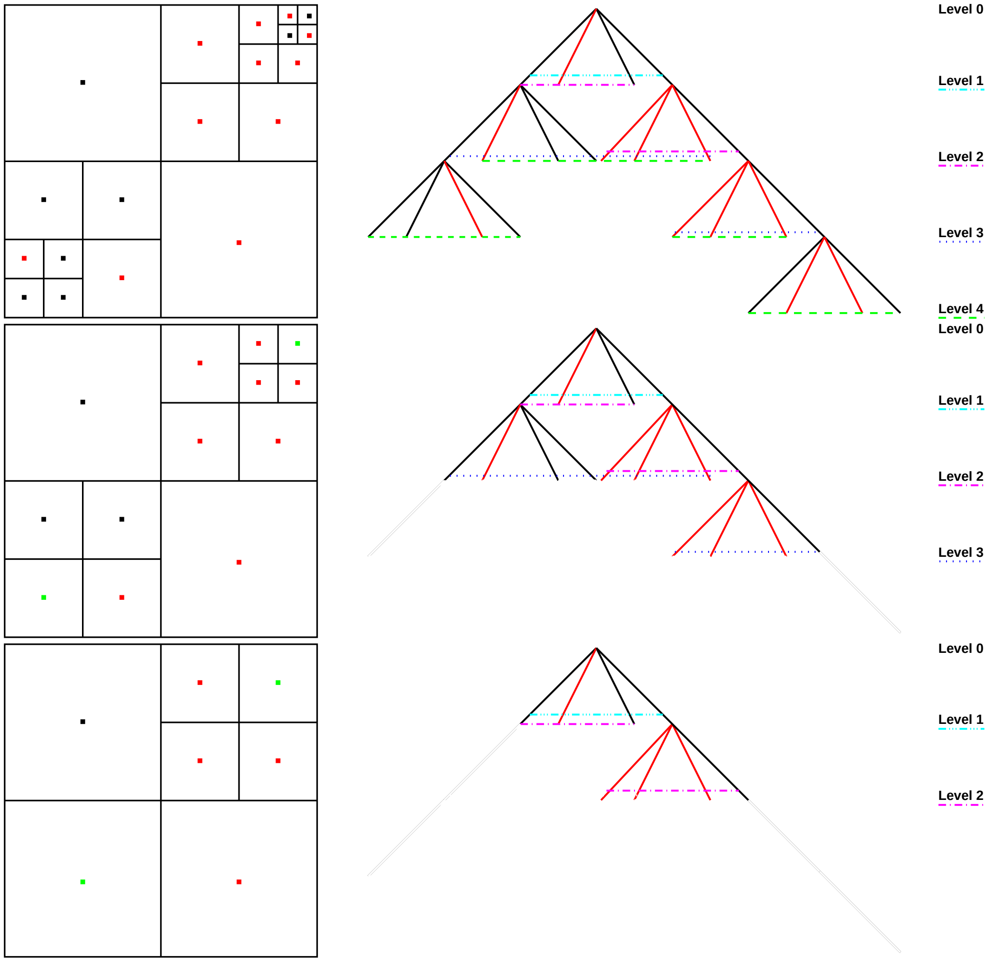

FIGURE 5.2. *The cell decomposition (left) and its respective tree representation (right) for the fine point set P_4 (upper) given in Figure 5.1 and two subsequent coarser point sets (P_3 and P_2) due to our cell agglomeration principle. The horizontal lines indicate the active levels for a tree node, i.e. its respective cell.*

matrix A, since the cover already defines the sparsity pattern of the stiffness matrix, i.e. the number of integrals to be evaluated. Furthermore, the cover influences the algebraic structure of the partition of unity functions φ_i, which has to be resolved for the proper integration of a stiffness matrix entry. In [16] the following hierarchical cover construction algorithm was proposed to reduce the computational costs of the assembly of the stiffness matrix. Due to this hierarchical construction we can now use this algorithm to define a sequence of point sets P_k, i.e. we can directly define a sequence of covers C_Ω^k, which can then be used to define a sequence of function spaces V_k^{PU}.

Algorithm 2 (Hierarchical Regular Cover Construction).

1. Given: the domain $\Omega \subset \mathbb{R}^d$, a bounding box $R_\Omega = \bigotimes_{i=1}^d [l_\Omega^i, u_\Omega^i] \supset \overline{\Omega}$, the initial point set $\tilde{P} = \{x_j \in \mathbb{R}^d \,|\, x_j \in \overline{\Omega}, j = 1, \dots \tilde{N}\}$ and a parameter $q \in \mathbb{N}$.
2. Set $P := \tilde{P}$ and build a d-binary tree (quadtree, octree) over R_Ω, such that per

leaf L at most one $x_i \in P$ lies within the associated cell $C_L := \bigotimes_{i=1}^{d} [l_L^i, u_L^i]$, and the difference of the levels (with respect to the tree) of two adjacent cells is at most q, see Figure 5.1.

3. For all cells $C_L = \bigotimes_{i=1}^{d} [l_L^i, u_L^i]$ with $C_L \cap \Omega \neq \emptyset$:
   (a) Set $x_L^i = l_L^i + \frac{1}{2}(u_L^i - l_L^i)$.
   (b) If there is *no* $x_j \in P$ with $x_j \in C_L$, set $P = P \cup \{x_L\}$.
   (c) Set $\omega_L = R_L = \bigotimes_{i=1}^{d} [x_L^i - h_L^i, x_L^i + h_L^i] \supset C_L$, where $h_L^i = \frac{\alpha_l}{2}(u_L^i - l_L^i)$.
   (d) Set the associated weight function $W_L\,(x) := \Pi_{i=1}^{d} \mathcal{W}\,(\frac{x - x_L^i + h_L^i}{2h_L^i})$.

Here, the parameter α_l is only dependent on the order l of the spline $\mathcal{W}$ used in the construction of the partition of unity, see [16]. Throughout this paper we use a linear spline $\mathcal{W}$ to generate the partition of unity with $\alpha_l = 1.3$ and $q = \infty$.

The d-binary tree of step 2 can be used to build a hierarchy of covers C_Ω^k and subsequently to define a sequence of function spaces V_k^{PU}. To this end we define a coarser cover C_Ω^{k-1} to a cover C_Ω^k by collapsing those leaves of the tree into its parent tree node whose siblings are also leaves, i.e. do not have children, see Figures 5.1 and 5.2. The construction of C_Ω^{k-1} then proceeds from this reduced tree with step 3 of Algorithm 2. This cover is then used to define the corresponding coarse partition of unity $\{\varphi_{i,k-1}\}$ according to (2.2). Finally, we use the maximal polynomial degree $p_{j,k-1} := \max_{\omega_{i,k}} p_{i,k}$ of all collapsed children $\omega_{i,k}$ to define a local approximation space $V_{j,k-1}^{p_{j,k-1}}$ on the coarse cover patch $\omega_{j,k-1}$. With these we then define the respective coarser function space $V_{k-1}^{\mathrm{PU}} := \sum \varphi_{i,k-1} V_{i,k-1}^{p_{i,k-1}}$.

The described patch agglomeration principle though does not translate (in general) to a nested sequence of function spaces due to the Shepard construction (2.2) for the partition of unity. Note further that although a geometric patch ω may be resident on multiple levels, its corresponding shape functions may *not* be the same on different levels. Since the geometric neighboring relations and the weight functions of the respective neighbors on different levels can change, the corresponding partition of unity function φ can change. Hence, the shape functions $\varphi_{i,k}\psi_{i,k}^n$ associated with $\omega_{i,k} = \omega$ on level k are different from those $\varphi_{j,k-1}\psi_{j,k-1}^n$ on level $k-1$, even if the local approximation space $V_{i,k}^{p_{i,k}} = V_{j,k-1}^{p_{j,k-1}}$ on the cover patch $\omega_{i,k} = \omega_{j,k-1} = \omega$ is not changed between levels k and $k-1$.

Due to the tree-based construction we have “transfer maps” between the point sets $P_k = \{x_{i,k}\}$ on different levels and the cover patches $C_\Omega^k = \{\omega_{i,k}\}$ on different levels via a descent or ascent operation in the tree. But such tree-operations cannot be used to transport information between V_{k-1}^{PU} and V_k^{PU}. To this end, the prolongations $I_{k-1}^k : V_{k-1}^{\mathrm{PU}} \to V_k^{\mathrm{PU}}$ have to be constructed explicitly. Yet we can exploit our tree construction to simplify the construction of I_{k-1}^k, see §5.1.

Note that our coarsening strategy is different from the usual level oriented coarsening in multigrid methods. The approximation property (with respect to the solution) of the resulting coarser PUM spaces should benefit from this coarsening strategy since we have a global coarsening which keeps local differences in the resolution constant (we assume that such differences are induced by the solution). With our strategy we also have a very fast coarsening, i.e. the number of points of a coarse level card(P_{k-1}) generated by our coarsening strategy is (in general) smaller than the number of points would be with a level oriented coarsening. A fast coarsening is useful for the optimal complexity of the multilevel iteration scheme. But still the coarsening rate may only be of algebraic type for highly irregular point sets $\tilde{P}$. The optimal operation count of the multilevel iteration $M_\gamma^{\nu_1,\nu_2}$ in such cases has to be ensured employing similar strategies as we have in multigrid for adaptive grids, see [30] and the references cited therein for details.

5.1. Interlevel Transfer Operators. Now we turn to the question of interlevel transfer operators, i.e. the construction of the prolongation operators

$$I_{k-1}^{k} : V_{k-1}^{\mathrm{PU}} \to V_{k}^{\mathrm{PU}}$$

and the restriction operators

$$I_{k}^{k-1} : V_{k}^{\mathrm{PU}} \to V_{k-1}^{\mathrm{PU}}.$$

Coarser shape functions $\varphi_{i,k-1}\psi_{i,k-1}$ cannot be represented exactly on finer levels, i.e.

$$\varphi_{i,k-1}\psi_{i,k-1}^{l} \neq \sum \beta_{j,k}^{m}\varphi_{j,k}\psi_{j,k}^{m} \tag{5.1}$$

due to the nonnestedness of the spaces V_{k}^{PU}. Furthermore, the functions $\varphi_{i,k}\psi_{i,k}^{n}$ are non-interpolatory and therefore a simple interpolation I_{k-1}^{k} between the spaces V_{k-1}^{PU} and V_{k}^{PU} is not available.

L^2-projections for Interlevel Transfer. One approach toward the construction of the prolongation operators I_{k-1}^{k} for our nonnested spaces is the use of L^2-projections Π_{k-1}^{k}

$$I_{k-1}^{k} := \Pi_{k-1}^{k} : V_{k-1}^{\mathrm{PU}} \to V_{k}^{\mathrm{PU}}$$

from V_{k-1}^{PU} onto V_{k}^{PU}. In general an L^2-projection $\Pi_{W}^{\tilde{W}} : W \to \tilde{W}$ from $W \subset L^2$ onto $\tilde{W} \subset L^2$ can be defined with the help of two moment matrices

$$(M_{\tilde{W}}^{\tilde{W}})_{ij} := \langle \phi_{j}^{\tilde{W}}, \phi_{i}^{\tilde{W}} \rangle_{L^2} \quad (M_{W}^{\tilde{W}})_{ij} := \langle \phi_{j}^{W}, \phi_{i}^{\tilde{W}} \rangle_{L^2}$$

where $\{\phi_{j}^{W}\}$ denotes a basis for W and $\{\phi_{j}^{\tilde{W}}\}$ a basis for $\tilde{W}$. The L^2-projection $\Pi_{W}^{\tilde{W}} : W \to \tilde{W}$ can then be defined as

$$\Pi_{W}^{\tilde{W}} := (M_{\tilde{W}}^{\tilde{W}})^{-1}(M_{W}^{\tilde{W}}). \tag{5.2}$$

The projection $\Pi_{W}^{\tilde{W}}$ maps coefficients $\tilde{u}_{W}$ to coefficients $\tilde{u}_{\tilde{W}}$. Its transpose $(\Pi_{W}^{\tilde{W}})^{T}$ obviously transports moment vectors $\hat{f}_{W}$ to moment vectors $\hat{f}_{\tilde{W}}$. Hence, if we use projections Π as the prolongations it is legitimate to use the transposed projections Π^{T} as restrictions.

Note that in the context of nonconforming discretizations L^2-projections are also used for the interlevel transfer, e.g. for the Crouzeix–Raviart element [6].

5.1.1. Global L^2-projection. The selection of $W = V_{k-1}^{\mathrm{PU}}$ with the PUM basis $\{\phi_{j}^{W}\} = \{\varphi_{i,k-1}\psi_{i,k-1}^{n}\}$ and $\tilde{W} = V_{k}^{\mathrm{PU}}$ with $\{\phi_{j}^{\tilde{W}}\} = \{\varphi_{i,k}\psi_{i,k}^{n}\}$ leads to the mass matrix M_{k}^{k} on V_{k}^{PU} and the interlevel mass matrix M_{k-1}^{k} from V_{k-1}^{PU} to V_{k}^{PU}. The global L^2-projection Π_{k-1}^{k} is then given by

$$\Pi_{k-1}^{k} = (M_{k}^{k})^{-1}(M_{k-1}^{k}). \tag{5.3}$$

This global projection Π_{k-1}^{k} though suffers from three major drawbacks:

1. The mass matrix M^k_k has to be inverted. Although the global basis $\{\varphi_{i,k}\psi^n_{i,k}\}$ is stable with respect to the number of cover patches $\text{card}(P_k)$, see [16], the condition number κ_k of M^k_k is dependent on the local approximation orders $p_{i,k}$.
2. The sparsity pattern of the mass matrix M^k_k is identical to that of the operator matrix A_k and therefore the storage requirement per level k is doubled.
3. The sparsity pattern of the interlevel mass matrix M^k_{k-1} is given by the geometric neighbor relations $\omega_{j,k-1} \cap \omega_{i,k} \neq \emptyset$. Due to the overlap of the cover patches the number of interlevel neighbors is rather large which further increases the storage requirement per level.

5.1.2. Global-to-Local L^2-projection. From the basic PUM error estimate [2, 3]

$$\|v - v^{\text{PU}}\|^2_{L^2(\Omega)} \leq C \sum_i \|v - v_i\|^2_{L^2(\omega_i \cap \Omega)}, \tag{5.4}$$

where $v^{\text{PU}} := \sum_i \varphi_i \sum_n u^n_i \psi^n_i$ and $v_i := \sum_n u^n_i \psi^n_i$, we know that it is sufficient to control the local errors $\|v - v_i\|_{L^2(\omega_i \cap \Omega)}$ on each cover patch ω_i. In the context of our projection problem we have $v = u^{\text{PU}}_{k-1}$ and $v^{\text{PU}} = I^k_{k-1} u^{\text{PU}}_{k-1}$ and (5.4) leads to the estimate

$$\|u^{\text{PU}}_{k-1} - I^k_{k-1} u^{\text{PU}}_{k-1}\|^2_{L^2(\Omega)} \leq C \sum_i \|u^{\text{PU}}_{k-1} - u_{i,k}\|^2_{L^2(\omega_{i,k} \cap \Omega)} \tag{5.5}$$

for the interlevel transfer problem. Hence, we can localize the approximation of the coarse function u^{PU}_{k-1} to the fine patches $\omega_{i,k}$ but still construct a valid global approximation $u^{\text{PU}}_k = I^k_{k-1} u^{\text{PU}}_{k-1}$. The corresponding moment matrices associated with this localized projection are

$$\begin{array}{rcl} (\tilde{M}^k_k)_{(i,n),(i,m)} & := & \langle \psi^m_{i,k}, \psi^n_{i,k} \rangle_{L^2(\omega_{i,k} \cap \Omega)} \quad \text{and} \\ (\hat{M}^k_{k-1})_{(i,n),(j,m)} & := & \langle \varphi_{j,k-1} \psi^m_{j,k-1}, \psi^n_{i,k} \rangle_{L^2(\omega_{i,k} \cap \Omega)}. \end{array} \tag{5.6}$$

The respective projection is given by

$$\hat{\Pi}^k_{k-1} = (\tilde{M}^k_k)^{-1} (\hat{M}^k_{k-1}). \tag{5.7}$$

Here, the localized moment matrix $\tilde{M}^k_k$ is block-diagonal. Therefore, the above-mentioned disadvantages 1 and 2 associated with the mass matrix M^k_k are eliminated. But since the sparsity pattern of the localized interlevel moment matrix $\hat{M}^k_{k-1}$ is identical to that of the global interlevel mass matrix M^k_{k-1} we still have to cope with the corresponding storage overhead.

5.1.3. Local-to-Local L^2-projection. The sparsity pattern of the moment matrices $\hat{M}^k_{k-1}$ and M^k_{k-1} is given by the geometric neighbor relations $\omega_{j,k-1} \cap \omega_{i,k} \neq \emptyset$. These have to be taken into account since we approximate the *global function* u^{PU}_{k-1} in V^{PU}_k. But if we also exploit the partition of unity construction of the coarse space V^{PU}_{k-1} and our hierarchical multilevel cover construction we can further localize the approximation problem associated with the interlevel transfer.

Due to our tree-based cover construction we have exactly one coarse cover patch $\omega_{j,k-1}$ with $\omega_{j,k-1} \supseteq \omega_{i,k}$ for every fine cover patch $\omega_{i,k}$, i.e. the coarse index $j, k-1$ is unique for every fine index i, k.

Every cover patch ω_L corresponds to a tree-cell C_L and vice versa. Either a fine cover patch $\omega_{i,k}$ is also element of the coarse cover C_Ω^{k-1}, then we have $\omega_{i,k} = \omega_{j,k-1}$, or the cover patch $\omega_{j,k-1}$ which corresponds to the parent tree-cell of $\omega_{i,k}$ is element of C_Ω^{k-1} and is the only coarse patch $\omega_{l,k-1}$ that fulfills $\omega_{l,k-1} \supseteq \omega_{i,k}$; i.e. in this case $\omega_{j,k-1} \supset \omega_{i,k}$ holds. We plug the associated coarse local approximation $u_{j,k-1} \in V_{j,k-1}^{p_{j,k-1}}$ into the right hand side of (5.5) and get the estimate

$$\begin{array}{rcl} \|u_{k-1}^{\mathrm{PU}} - I_{k-1}^k u_{k-1}^{\mathrm{PU}}\|_{L^2(\Omega)}^2 & \leq & C\sum_i \big(\|u_{k-1}^{\mathrm{PU}} - u_{j,k-1}\|_{L^2(\omega_{i,k}\cap\Omega)} + \\ & & \qquad \|u_{j,k-1} - u_{i,k}\|_{L^2(\omega_{i,k}\cap\Omega)}\big)^2. \end{array} \tag{5.8}$$

Now, let u denote the continuous function which is approximated by $u_{k-1}^{\mathrm{PU}} \in V_{k-1}^{\mathrm{PU}}$ and introduce u into the first term of the right hand side of (5.8). This leads to the estimate

$$\begin{array}{rcl} \|u_{k-1}^{\mathrm{PU}} - I_{k-1}^k u_{k-1}^{\mathrm{PU}}\|_{L^2(\Omega)}^2 & \leq & C\sum_i \big(\|u_{k-1}^{\mathrm{PU}} - u\|_{L^2(\omega_{i,k}\cap\Omega)} + \\ & & \qquad \|u - u_{j,k-1}\|_{L^2(\omega_{i,k}\cap\Omega)} + \\ & & \qquad \|u_{j,k-1} - u_{i,k}\|_{L^2(\omega_{i,k}\cap\Omega)}\big)^2. \end{array} \tag{5.9}$$

Therefore, it is sufficient to control the error $\|u_{j,k-1} - u_{i,k}\|_{L^2(\omega_{i,k}\cap\Omega)}$ of the local approximation $u_{i,k}$ on the fine cover patch $\omega_{i,k}$ to the coarse local approximation $u_{j,k-1}$ with $\omega_{i,k} \subseteq \omega_{j,k-1}$. The respective moment matrices are then defined by

$$\begin{array}{rcl} (\tilde{M}_k^k)_{(i,n),(i,m)} & := & \langle \psi_{i,k}^m, \psi_{i,k}^n \rangle_{L^2(\omega_{i,k}\cap\Omega)} \quad \text{and} \\ (\tilde{M}_{k-1}^k)_{(i,n),(j,m)} & := & \langle \psi_{j,k-1}^m, \psi_{i,k}^n \rangle_{L^2(\omega_{i,k}\cap\Omega)}, \end{array} \tag{5.10}$$

where the sparsity pattern of $\tilde{M}_{k-1}^k$ is now given by the hierarchical condition $\omega_{i,k} \subseteq \omega_{j,k-1}$ instead of the neighbor relation $\omega_{i,k} \cap \omega_{j,k-1} \neq \emptyset$. Hence, the storage requirement for the projection

$$\tilde{\Pi}_{k-1}^k = (\tilde{M}_k^k)^{-1}(\tilde{M}_{k-1}^k) \tag{5.11}$$

is reduced to one block-entry $(\tilde{\Pi}_{k-1}^k)_{ij}$ per fine level patch $\omega_{i,k}$ without (a significant) loss of accuracy of the approximation. Note that all projections Π_{k-1}^k, $\hat{\Pi}_{k-1}^k$ and $\tilde{\Pi}_{k-1}^k$ are exact for polynomials of degree $\min_i p_{i,k}$.

5.2. Smoothing Operators. The remaining ingredients for our multilevel algorithm are the smoothers S^{pre} and S^{post}. Since the coarsening procedure involves only the covers C_Ω^k and does not coarsen the local degrees of freedom $V_{i,k}^{p_{i,k}}$ we use a block-smoother which treats all local approximation functions $\psi_{i,k}^n$, i.e. the respective coefficients, simultaneously.

It is well-known that the smoothing property of the block-Gauß–Seidel smoother is dependent on the ordering of the unknown coefficients. Due to the meshfree construction of the function space there is no natural ordering scheme for the degrees of freedom. However, we can define ordering schemes for the cover patches $\omega_{i,k}$ with the help of our tree construction. We can identify the cover patches $\omega_{i,J}$ on the finest level J with the leaves of the tree and use e.g. a depth-first ordering to index the leaves; a similar approach can be applied for covers C_Ω^k on coarser levels $k < J$. But note that the resulting ordering of the cover patches would have no data locality property. It is similar to the result we can obtain from a space filling curve (SFC) ordering scheme [26] with the Lebesgue curve. However, the data locality property where the indices of geometrically neighboring cover patches are clustered together is desirable not only

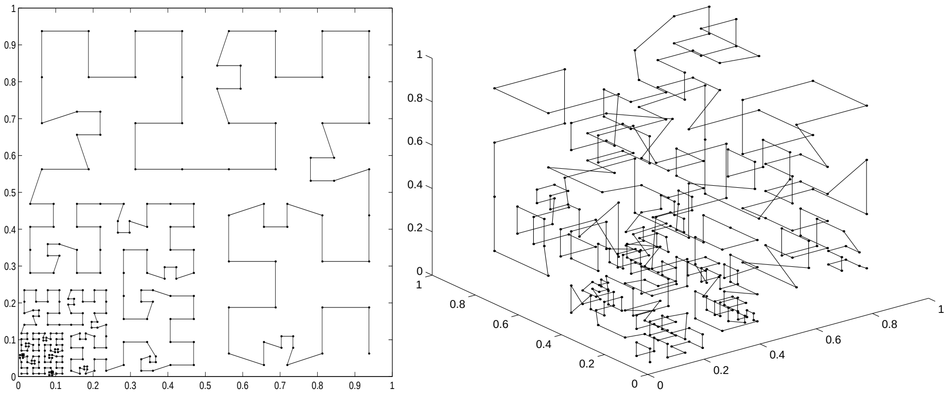

FIGURE 5.3. *An adaptive Hilbert curve which is used to index the cover patches in two (left) and three (right) dimensions.*

from a parallelization point of view [10, 17, 30] but also for the smoothing property of the block-Gauß–Seidel smoother. Such an ordering with the data locality property can be obtained by using an SFC ordering based on the Hilbert curve, see Figure 5.3. Hence, throughout this paper we always use such an Hilbert ordering scheme for the cover patches and subsequently for the block-rows of the stiffness matrix in polynomial block form.

5.3. Computational Costs. In this section we analyze the computational costs according to §4.1 associated with the multilevel iteration $M_\gamma^{\nu_1,\nu_2}$ based on the projections Π_{k-1}^k, $\hat{\Pi}_{k-1}^k$ and $\tilde{\Pi}_{k-1}^k$. Again, we exploit the separation of the degrees of freedom into p-type and h-type components to simplify the presentation.

In our partition of unity method the number of degrees of freedom on level k

$$\mathsf{dof_k} = \mathrm{rows}(A_k) = \sum_{i=1}^{\mathrm{card}(P_k)} \dim(V_{i,k}^{p_{i,k}})$$

is given by the number of cover patches $\mathrm{card}(P_k)$ and the dimensions of the local approximation spaces $V_{i,k}^{p_{i,k}}$. Since our multilevel construction does not "change" the local approximation spaces $V_{i,k}^{p_{i,k}}$ between levels we can assume that $\dim(V_{i,k}^{p_{i,k}})$ can be estimated by a constant C_{loc} which depends on $\max_{i,k} p_{i,k}$. Therefore, we can restrict ourselves to the use of $\mathrm{card}(P_k)$ instead of dof_k for the discussion, i.e. we restrict ourselves to block-storage and block-operation estimates. Here, we have to keep in mind that the storage associated with a block-entry is of order $p_{i,k}^{2d}$ whereas a block-inversion has an operation count of order $p_{i,k}^{3d}$.

With this notation we get the optimal complexity of the multilevel iteration if the series

$$\sum_{k=0}^{J} \gamma^k \frac{\mathrm{card}(P_{J-k})}{\mathrm{card}(P_J)} < \infty \quad \text{for} \quad J \to \infty \tag{5.12}$$

converges and the block-operation counts $\mathcal{C}_{D,k}$, $\mathcal{C}_{C,k}$ and $\mathcal{C}_{S,k}$, i.e. the block-storage estimates $\mathcal{C}(A_k)$, $\mathcal{C}(I_{k-1}^k)$ and $\mathcal{C}(I_k^{k-1})$, are level-independent.

For reasons of simplicity we restrict ourselves to uniform point sets P_k to estimate the average storage costs $\mathcal{C}_A$ for the discrete operator, $\mathcal{C}_P$ for the prolongations and $\mathcal{C}_R$ for the restrictions. The average number of nonzero blocks per row of the discrete operator A_k is then given by the average number of geometric neighbors

$$\mathcal{C}_A = \mathcal{C}(A_k) = \operatorname{avg}_i N_{i,k} = \operatorname{avg}_i \operatorname{card}\{\omega_{j,k} \,|\, \omega_{j,k} \cap \omega_{i,k} \neq \emptyset\} = 3^d$$

which is level independent.

The global L^2-projection Π^k_{k-1} involves the inverse of the mass matrix M^k_k on level k which is a dense matrix. But the inverse can be computed more efficiently by an iterative method since the condition number κ_k of M^k_k is independent of the number of patches $\operatorname{card}(P_k)$; yet it may strongly depend on the local approximation orders $p_{i,k}$. Then, the operation counts $\mathcal{C}_D$ and $\mathcal{C}_C$ for a multilevel iteration with the global L^2-projection can be estimated by

$$\mathcal{C}_{D,k} = 2(\mathcal{C}(M^{k-1}_k) + C_{M^k_k})\big(2\mathcal{C}(A_k) + 1\big), \text{ and } \quad \mathcal{C}_{C,k} = 1 + 2(\mathcal{C}(M^k_{k-1}) + C_{M^k_k})$$

only, where $C_{M^k_k}$ estimates the costs of the iterative inversion of M^k_k and may be strongly dependent on $p_{i,k}$. The necessary storage for Π^k_{k-1} and $(\Pi^k_{k-1})^T$ is given by the storage for M^k_k and $M^k_{k-1} = (M^{k-1}_k)^T$

$$\mathcal{C}(M^k_k) + \mathcal{C}(M^k_{k-1}) = 3^d + 2^{-1}3^d.$$

For the Global-to-Local L^2-projection $\hat{\Pi}^k_{k-1}$ we need the inverse of $\tilde{M}^k_k$ which is readily available since $\tilde{M}^k_k$ is block-diagonal. We only need to invert a single block-entry per block-row of $\hat{\Pi}^k_{k-1}$. In general the inverse can be computed in $\bar{p}^{3d}_k$ operations and it can be applied in $\bar{p}^{2d}_k$ operations, where $\bar{p}_k = \max_i p_{i,k}$. If the local basis functions $\psi^m_{i,k}$ are L^2-orthogonal (like the Legendre polynomials) the inverse can be computed in $\bar{p}^d_k$ operations (disregarding boundary effects). The average storage costs for $\hat{\Pi}^k_{k-1}$ is given by the storage costs for $\hat{M}^k_{k-1}$ which has an identical sparsity pattern as M^k_{k-1}. Hence, we obtain the block-storage estimate

$$\mathcal{C}_P = \mathcal{C}(\hat{\Pi}^k_{k-1}) = \mathcal{C}(\hat{M}^k_{k-1}) = \mathcal{C}(M^k_{k-1}) = 2^{-1}3^d$$

for the Global-to-Local L^2-projection $\hat{\Pi}^k_{k-1}$ on uniform point sets.

Finally, we obtain a minimal block-storage estimate for the Local-to-Local L^2-projection $\tilde{\Pi}^k_{k-1}$ for any point set $\tilde{P}$ since no geometric neighbor relations are involved in its construction, i.e.

$$\mathcal{C}_P = \mathcal{C}(\tilde{\Pi}^k_{k-1}) = \mathcal{C}(\tilde{M}^k_{k-1}) = 1$$

holds independent of the distribution of $\tilde{P}$. Hence, the conditions for an optimal complexity multilevel iteration based on the Local-to-Local projection $\tilde{\Pi}^k_{k-1}$ are the convergence of the series (5.12) and the validity of the estimate $\mathcal{C}(A_k) = \operatorname{avg}_i N_{i,k} \leq \mathcal{C}_A$. The overall complexity of the iteration with respect to $\bar{p}_k$ is at least $\bar{p}^{2d}_k$ since the block-entries of the operator A_k are dense as well as the blocks of $\tilde{\Pi}^k_{k-1}$. The block entries of $\tilde{\Pi}^k_{k-1}$ are dense since the blocks of $\tilde{M}^k_{k-1}$ are dense. We employ block-smoothers in the iteration, therefore we have $\bar{p}^{3d}_k$ as the $\bar{p}_k$-complexity of the iteration.

The assembly of the global L^2-projection Π_{k-1}^k (and the Global-to-Local L^2-projection $\hat{\Pi}_{k-1}^k$) requires the integration of the global shape functions $\varphi_{i,k}\psi_{i,k}^m$ which is more challenging than the integration of a finite element shape function [15, 16]. In the assembly of the Local-to-Local L^2-projection though the partition of unity functions $\varphi_{i,k}$ are completely eliminated. Here, we only have to compute integrals of local approximation functions $\psi_{i,k}^m$, i.e. usually polynomials, which can be computed very efficiently and without any additional error due to numerical integration.

In summary, the choice of the projection essentially influences only the constants in the operation count, but not the overall complexity of the multilevel iteration scheme. The constants though may vary quite dramatically. The use of the global L^2-projection not only involves the inverse of the mass matrix on level k but also the interlevel mass matrix M_{k-1}^k from level $k-1$ to level k which has a rather large number of nonzero blocks depending on the distribution of the point set $\tilde{P}$. For the Local-to-Local projection $\tilde{\Pi}_{k-1}^k$ on the other hand, there is no need for a global inverse and we have only one block-entry per block-unknown $u_{i,k}$ on the fine level k independent of the geometric point distribution. Here, the constants $\mathcal{C}_P$ and $\mathcal{C}_R$ are always minimal. This will also have a significant impact on parallel applications [17]. With only one block-entry in the prolongation and restriction operators, communication can be eliminated (almost) completely in the interlevel transfers.

For highly irregular point sets $\tilde{P}$ the series (5.12) may well not converge which would lead (at least) to a logarithmic complexity of the global multilevel iteration $M_\gamma^{\nu_1,\nu_2}$. But even for such point sets $\tilde{P}$ we can achieve an optimal complexity implementation. To this end we can employ similar strategies as we have in multigrid methods on adaptive grids where an equivalent problem arises, see [30] and the references cited therein for details.[2]

6. Numerical Examples.

The model problem we apply our multilevel PUM to is the PDE

$$-\Delta u + u = 0 \text{ in } \Omega = (0,1)^d \text{ where } d = 2, 3 \tag{6.1}$$

of Helmholtz type with vanishing Neumann boundary conditions $\nabla u = 0$ on $\partial\Omega$. In all our experiments we use a linear normed B-spline as the generating weight function $\mathcal{W}$ for the partition of unity construction and $\alpha_l = 1.3$ in step 3c of Algorithm 2. The initial value $\tilde{u}_0$ for the multilevel iteration is random valued with $\|\tilde{u}_0\| = 1$. The stopping criterion for the iteration is $\|\tilde{u}_r\| < 10^{-10}$. The convergence rate of our iteration with respect to the error is then given by

$$\rho := \|\tilde{u}_r\|^{\frac{1}{r}}.$$

We use the l^2-norm of the coefficient vectors $\tilde{u}$ for the stopping criterion and the computation of the convergence rate ρ. Note that even for uniform point sets $\tilde{P}$ we have no uniform correspondence between the L^2-norm of the function u and the l^2-norm of its associated coefficient vector $\tilde{u}$ due to the use of local polynomials, just like in the p-version of the finite element method. Furthermore when we are dealing with irregular point sets the relation between the L^2- and l^2-norms now also depends on

[2] The basic idea is to restrict the iteration on each level to a subset of patches in such a way that the respective series converges yet there is no deterioration in the quality of the iteration. To this end we define a set of *inactive* patches on each level and its complement, the *active* patches, on which the iteration $M_\gamma^{\nu_1,\nu_2}$ is carried out. Note that the number of active patches per level is (in general) larger than the number of active nodes due to our meshfree construction.

the varying size of the patches. One way of dealing with this problem is to introduce the mass matrix into the norm measurement

$$\|u\|_{L^2}^2 \simeq \tilde{u}^T M \tilde{u}.$$

We have also computed all convergence rates and norms using this approximation to the L^2-norm of u. Here, we found that the absolute value of the l^2-norm and the (approximate) L^2-norm may vary significantly (depending on the local approximation orders $p_{i,J}$, the distribution of $\tilde{P}$ and the number of the points in $\tilde{P}$) but the respective convergence rates are (almost) identical. Here, the l^2-norm of the coefficient vector $\tilde{u}$ was always larger than the L^2-norm of u. Hence, our stopping criterion overestimates the error norm. In that sense we therefore complete too many iterations, but the convergence rates ρ given are accurate. A similar observation holds also for the l^2-norm of the residual.

Besides the convergence rates ρ we also give the number of initial points $\tilde{N}$, the number of generated patches $\operatorname{card}(P) = \operatorname{card}(P_J)$ on the finest level J, the polynomial degree $p = \max_i p_{i,J}$ and the dimension D_p of the associated local approximation space $V_{i,J}^{p_{i,J}}$ in all tables. Furthermore, we give the V-cycle ($M_1^{\nu,\nu}$) complexity C_1 and the W-cycle ($M_2^{\nu,\nu}$) complexity C_2, which are defined as

$$C_1 := \sum_{k=0}^{J} \frac{\operatorname{card}(P_{J-k})}{\operatorname{card}(P_J)} \quad \text{and} \quad C_2 := \sum_{k=0}^{J} \frac{2^k \operatorname{card}(P_{J-k})}{\operatorname{card}(P_J)}.$$

Note that the number of degrees of freedom on the finest level J is given by $\operatorname{card}(P)D_p$.

Example 1 (Halton Point Sets). In our first example we use a Halton[3] sequence as the initial point set $\tilde{P}$ for our cover construction, see Figure 6.1 for several levels of the constructed cover hierarchy for a $\text{Halton}_0^{4095}(2,3)$ set in two dimensions. The local approximation spaces $V_{i,k}^{p_{i,k}}$ we use in this experiment are based on isotropic Legendre polynomials, i.e. we choose $p_{i,k} = p$ with $p = 1, 2, 3, 4, 5$ for all i and k. Before we study higher order approximations $p > 1$, we begin with the linear case $p = 1$. Here, we have $D_p = 3$ in two dimensions and $D_p = 4$ in three dimensions.

The measured convergence rates $\rho_1^{\nu,\nu}$ for a multilevel V-cycle ($M_1^{\nu,\nu}$) and $\rho_2^{\nu,\nu}$ for a W-cycle ($M_2^{\nu,\nu}$) with up to 6 iterations of block-Gauß–Seidel smoothing ($\nu = 1, 2, 3$) are given in Tables 6.1, and 6.2 for the two-dimensional and three-dimensional model problem respectively. Since the Halton sequence is uniformly distributed we have (almost) standard coarsening rates between levels, i.e. the number of patches decreases by a factor of 2^{-d} from level to level. Hence, C_1 as well as C_2 are bounded in two and three dimensions. The numbers given in the corresponding tables indicate this behavior. Therefore the V-cycle and W-cycle iteration are of optimal complexity. This behavior can be observed from Figure 6.3 where we have plotted the iteration times against the number of degrees of freedom. From these results we clearly observe that the multilevel V-cycle for the global L^2-projection—which provides the best approximation property of the three presented interlevel transfer operators—converges with a rate ρ_1 which is independent of the number of patches $\operatorname{card}(P)$. Here, we have $\rho_1^{1,1} \simeq 0.23$, $\rho_1^{2,2} \simeq 0.13$ and $\rho_1^{3,3} \simeq 0.09$. The reduced approximation quality of the

[3] Halton–sequences are pseudo Monte Carlo sequences, which are used in sampling and numerical integration. Consider $n \in \mathbb{N}_0$ given as $\sum_j n_j p^j = n$ for some prime p. We can define the transformation H_p from $\mathbb{N}_0$ to $[0,1]$ with $n \mapsto H_p(n) = \sum_j n_j p^{-j-1}$. Then, the (p,q) Halton–sequence with $\tilde{N}$ points is defined as $\text{Halton}_0^N(q,p) := \{(H_p(n), H_q(n)) \,|\, n = 0, \dots, N\}$.

Global-to-Global L^2-projection

$\tilde{N}$	card(P)	C_1	C_2	J	p	D_p	$\rho_1^{1,1}$	$\rho_1^{2,2}$	$\rho_1^{3,3}$	$\rho_2^{1,1}$	$\rho_2^{2,2}$	$\rho_2^{3,3}$
16	28	1.64	2.79	3	1	3	0.108	0.063	0.047	0.079	0.046	0.031
64	106	1.87	4.53	5	1	3	0.196	0.091	0.058	0.183	0.080	0.046
256	406	1.65	3.06	5	1	3	0.200	0.115	0.077	0.181	0.102	0.067
1024	1729	1.84	4.64	7	1	3	0.224	0.123	0.086	0.200	0.110	0.076
4096	6364	1.73	3.99	8	1	3	0.228	0.130	0.087	0.199	0.111	0.072
16384	27673	1.84	5.20	10	1	3	0.233	0.140	0.094	0.214	0.128	0.084
65536	101314	1.71	3.87	10	1	3	0.227	0.131	0.088	0.189	0.105	0.068

Global-to-Local L^2-projection

$\tilde{N}$	card(P)	C_1	C_2	J	p	D_p	$\rho_1^{1,1}$	$\rho_1^{2,2}$	$\rho_1^{3,3}$	$\rho_2^{1,1}$	$\rho_2^{2,2}$	$\rho_2^{3,3}$
16	28	1.64	2.79	3	1	3	0.117	0.065	0.047	0.085	0.047	0.032
64	106	1.87	4.53	5	1	3	0.203	0.095	0.066	0.190	0.081	0.050
256	406	1.65	3.06	5	1	3	0.212	0.120	0.082	0.196	0.107	0.070
1024	1729	1.84	4.64	7	1	3	0.248	0.138	0.094	0.211	0.116	0.083
4096	6364	1.73	3.99	8	1	3	0.254	0.143	0.095	0.219	0.119	0.076
16384	27673	1.84	5.20	10	1	3	0.272	0.159	0.109	0.245	0.137	0.090
65536	101314	1.71	3.87	10	1	3	0.255	0.142	0.095	0.207	0.112	0.072

Local-to-Local L^2-projection

$\tilde{N}$	card(P)	C_1	C_2	J	p	D_p	$\rho_1^{1,1}$	$\rho_1^{2,2}$	$\rho_1^{3,3}$	$\rho_2^{1,1}$	$\rho_2^{2,2}$	$\rho_2^{3,3}$
16	28	1.64	2.79	3	1	3	0.112	0.068	0.050	0.083	0.049	0.032
64	106	1.87	4.53	5	1	3	0.197	0.097	0.059	0.180	0.083	0.047
256	406	1.65	3.06	5	1	3	0.210	0.121	0.084	0.179	0.104	0.070
1024	1729	1.84	4.64	7	1	3	0.220	0.127	0.088	0.185	0.107	0.074
4096	6364	1.73	3.99	8	1	3	0.234	0.133	0.090	0.185	0.108	0.071
16384	27673	1.84	5.20	10	1	3	0.230	0.138	0.094	0.197	0.118	0.077
65536	101314	1.71	3.87	10	1	3	0.234	0.137	0.094	0.184	0.104	0.069

Table 6.1

Convergence rates $\rho_1^{\nu,\nu}$ *for the* $V^{\nu,\nu}$*-cycle and convergence rates* $\rho_2^{\nu,\nu}$ *for the* $W^{\nu,\nu}$*-cycle with* $\nu = 1, 2, 3$. *Covers are based on Halton*$(2, 3)$ *point sets in two dimensions.*

Global-to-Local and the Local-to-Local projections though only slightly effects the convergence rates, i.e. the quality of the localized projections is comparable to that of the global L^2-projection. We have $\rho_1^{1,1} \simeq 0.26$ for the Global-to-Local projection. For the Local-to-Local projection we have essentially the same convergence rates as we have for the global L^2-projection with $\rho_1^{1,1} \simeq 0.23$. A similar behavior can be observed for V-cycles with multiple smoothing steps.

The measured convergence rates ρ_2 for the W-cycles are essentially the same as the V-cycle rates ρ_1, i.e. the increased work on coarser levels does not improve the convergence rate and does not pay off. This is due to the approximation quality of the L^2-projection approach, the loss of information between levels is "minimal".

The computational effort due to the different interlevel transfers is significant, see §5.3. From the plots of the execution times against the number of degrees of freedom depicted in Figure 6.3 we clearly see the increased operation count due to the iterative solution of the mass matrix problem in every interlevel transfer for the global L^2-projection. The execution times for the iterations based on the Global-to-Local and Local-to-Local projections are significantly smaller. We can also observe that the overall execution times for cycles with multiple smoothing steps are smaller than the execution time of the respective cycle with $(1, 1)$ smoothing for the global L^2-projection. This is due to the fact that multiple smoothing steps improve the convergence rates substantially which decreases the number of mass matrix inversions. This in turn reduces the computational work of the iteration since a single smoothing step is cheaper than the iterative solution of the mass matrix problem. The performance improvement of the iteration due to the use of the Local-to-Local projection

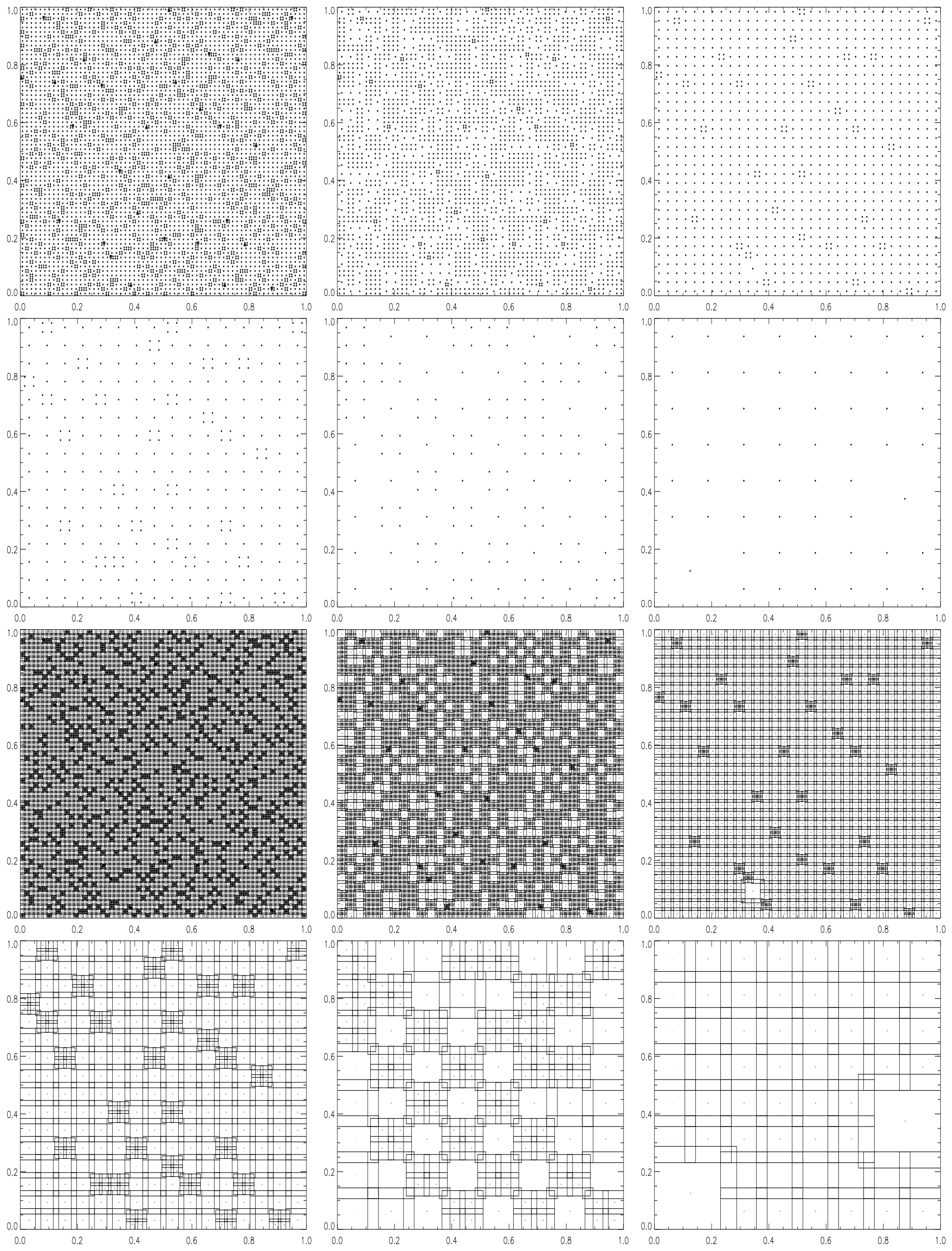

FIGURE 6.1. *Point set and cover hierarchy constructed by Algorithm 2 for a* $\mathrm{Halton}_0^{4095}(2,3)$ *point set in two dimensions.*

instead of the Global-to-Local projection is not as significant. This is due to the fact that the computational effort involved in the interlevel transfer with these projections is already small compared with the work due to smoothing. Note that the curves are not perfectly linear due to the slight variations in the complexities C_1 and C_2 which come from the irregularity of the Halton point set.

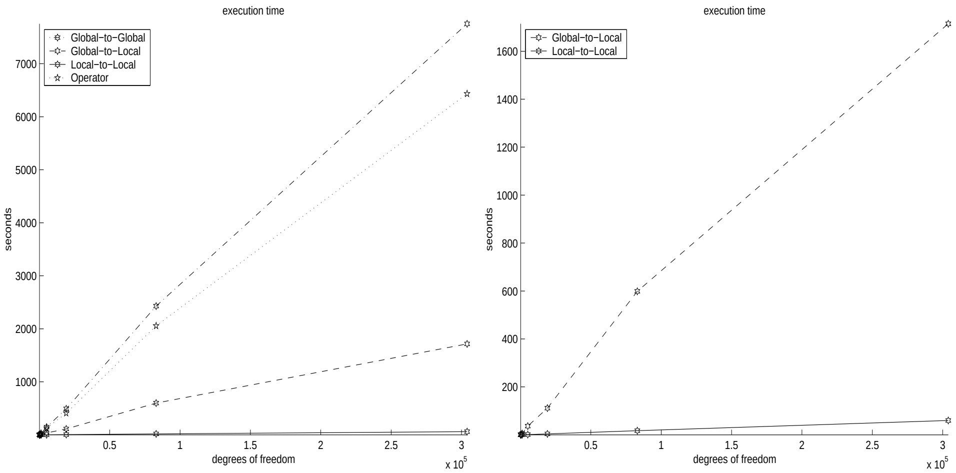

FIGURE 6.2. *Setup times for the transfer operators based on the global L^2-projection (Global-to-Global), the Global-to-Local and Local-to-Local projections as defined in §5.1 for Example 1 in two dimensions. The setup times for the operator assembly are also given.*

Global-to-Global L^2-projection												
$\tilde{N}$	card(P)	C_1	C_2	J	p	D_p	$\rho_1^{1,1}$	$\rho_1^{2,2}$	$\rho_1^{3,3}$	$\rho_2^{1,1}$	$\rho_2^{2,2}$	$\rho_2^{3,3}$
16	50	1.62	2.68	3	1	4	0.064	0.015	0.007	0.063	0.014	0.007
128	414	1.30	1.88	4	1	4	0.145	0.084	0.056	0.126	0.064	0.042
1024	3543	1.38	2.24	6	1	4	0.176	0.088	0.055	0.128	0.064	0.042
8192	26699	1.37	2.10	6	1	4	0.186	0.098	0.068	0.136	0.061	0.055
Global-to-Local L^2-projection												
$\tilde{N}$	card(P)	C_1	C_2	J	p	D_p	$\rho_1^{1,1}$	$\rho_1^{2,2}$	$\rho_1^{3,3}$	$\rho_2^{1,1}$	$\rho_2^{2,2}$	$\rho_2^{3,3}$
16	50	1.62	2.68	3	1	4	0.068	0.024	0.013	0.068	0.021	0.011
128	414	1.30	1.88	4	1	4	0.178	0.097	0.064	0.155	0.075	0.048
1024	3543	1.38	2.24	6	1	4	0.203	0.103	0.065	0.156	0.074	0.047
8192	26699	1.37	2.10	6	1	4	0.227	0.117	0.077	0.158	0.072	0.055
Local-to-Local L^2-projection												
$\tilde{N}$	card(P)	C_1	C_2	J	p	D_p	$\rho_1^{1,1}$	$\rho_1^{2,2}$	$\rho_1^{3,3}$	$\rho_2^{1,1}$	$\rho_2^{2,2}$	$\rho_2^{3,3}$
16	50	1.62	2.68	3	1	4	0.075	0.017	0.008	0.074	0.016	0.007
128	414	1.30	1.88	4	1	4	0.154	0.088	0.060	0.131	0.065	0.044
1024	3543	1.38	2.24	6	1	4	0.182	0.094	0.062	0.136	0.070	0.048
8192	26699	1.37	2.10	6	1	4	0.192	0.110	0.076	0.142	0.066	0.055

TABLE 6.2

Convergence rates $\rho_1^{\nu,\nu}$ for the $V^{\nu,\nu}$-cycle and convergence rates $\rho_2^{\nu,\nu}$ for the $W^{\nu,\nu}$-cycle with $\nu = 1, 2, 3$. Covers are based on Halton$(2, 3, 5)$ point sets in three dimensions.

When we take the costs of the assembly of the transfer operators into account the performance improvement due to the hierarchical localization though is substantial. In Figure 6.2 we give plots of the execution times for the setup associated with the different projection approaches and the operator assembly. Here, both the prolongation and its transpose are computed and stored explicitly. From these plots we clearly see that the assembly of the global L^2-projection, i.e. the assembly of M_{k-1}^k and M_k^{k-1}, is even more expensive than the operator assembly (which also includes the setup for M_k^k). The impact of the first localization which eliminates the partition of unity functions on the test side is obvious. The setup times for the Global-to-Local projections $\hat{\Pi}_{k-1}^k$ are significantly smaller than the times spent in the operator assembly. But the

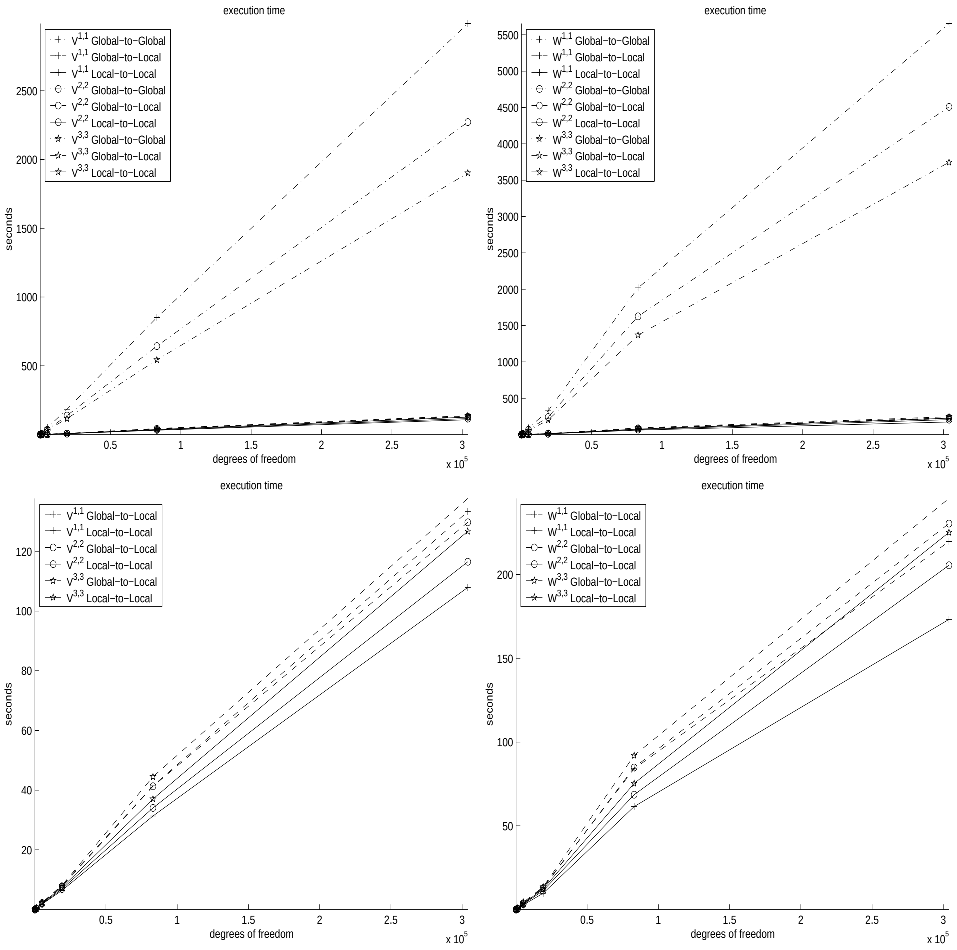

FIGURE 6.3. *Iteration times (V-cycle (left), W-cycle (right)) for the multilevel iteration scheme based on the global L^2-projection (Global-to-Global), the Global-to-Local and Local-to-Local projections as defined in §5.1 for Example 1 in two dimensions.*

number of integrals which have to be computed for $\hat{M}^k_{k-1}$ is the same as we have for M^k_{k-1} in the global L^2-projection Π^k_{k-1}. The second localization of the projections now reduces the number of block-integrals to be computed for $\tilde{M}^k_{k-1}$ to one per fine cover patch $\omega_{i,k}$. Furthermore, this block-integral only involves local polynomials and can be evaluated very efficiently. Hence, the reduction in the computational work is again substantial. Now the setup time for the computation of the interlevel transfer operators is negligible compared with the time spent in the operator assembly if we use the Local-to-Local projections $\tilde{\Pi}^k_{k-1}$.

The convergence behavior for the model problem in three dimensions (see Table 6.2) is similar to the two-dimensional case. We find e.g. $\rho_1^{1,1} \simeq 0.20$, $\rho_1^{2,2} \simeq 0.11$ and $\rho_1^{2,2} \simeq 0.08$.

In summary we obtain that the projection approach leads to multilevel iterations in which the W-cycle or even more expensive cycle types do not seem to pay off. The loss in the approximation quality due to the localization of the L^2-projection is very

Global-to-Global L^2-projection

p	D_p	$\tilde{\rho}_1^{1,1}$	$\tilde{\rho}_1^{2,2}$	$\tilde{\rho}_1^{3,3}$	$\tilde{\rho}_2^{1,1}$	$\tilde{\rho}_2^{2,2}$	$\tilde{\rho}_2^{3,3}$	p	D_p	$\tilde{\rho}_1^{1,1}$	$\tilde{\rho}_1^{2,2}$	$\tilde{\rho}_1^{3,3}$	$\tilde{\rho}_2^{1,1}$	$\tilde{\rho}_2^{2,2}$	$\tilde{\rho}_2^{3,3}$
1	3	0.23	0.13	0.09	0.20	0.11	0.08	1	4	0.17	0.09	0.06	0.13	0.06	0.05
2	6	0.10	0.04	0.02	0.10	0.04	0.02	2	10	0.08	0.02	0.01	0.08	0.02	0.01
3	10	0.18	0.04	0.01	0.18	0.04	0.01	3	20	0.22	0.05	0.01	0.20	0.04	0.01
4	15	0.39	0.17	0.08	0.39	0.17	0.08	4	35	0.41	0.17	0.08	0.41	0.17	0.08
5	21	0.60	0.41	0.27	0.60	0.41	0.27	5	56	0.64	0.44	0.30			

Global-to-Local L^2-projection

p	D_p	$\tilde{\rho}_1^{1,1}$	$\tilde{\rho}_1^{2,2}$	$\tilde{\rho}_1^{3,3}$	$\tilde{\rho}_2^{1,1}$	$\tilde{\rho}_2^{2,2}$	$\tilde{\rho}_2^{3,3}$	p	D_p	$\tilde{\rho}_1^{1,1}$	$\tilde{\rho}_1^{2,2}$	$\tilde{\rho}_1^{3,3}$	$\tilde{\rho}_2^{1,1}$	$\tilde{\rho}_2^{2,2}$	$\tilde{\rho}_2^{3,3}$
1	3	0.26	0.15	0.10	0.22	0.12	0.08	1	4	0.20	0.11	0.07	0.16	0.08	0.05
2	6	0.10	0.04	0.02	0.10	0.04	0.02	2	10	0.08	0.02	0.01	0.08	0.02	0.01
3	10	0.21	0.05	0.01	0.21	0.05	0.01	3	20	0.20	0.04	0.01	0.20	0.04	0.01
4	15	0.44	0.19	0.08	0.44	0.19	0.08	4	35	0.42	0.18	0.08	0.42	0.18	0.08
5	21	0.63	0.44	0.27	0.63	0.44	0.27	5	56	0.64	0.45	0.31			

Local-to-Local L^2-projection

p	D_p	$\tilde{\rho}_1^{1,1}$	$\tilde{\rho}_1^{2,2}$	$\tilde{\rho}_1^{3,3}$	$\tilde{\rho}_2^{1,1}$	$\tilde{\rho}_2^{2,2}$	$\tilde{\rho}_2^{3,3}$	p	D_p	$\tilde{\rho}_1^{1,1}$	$\tilde{\rho}_1^{2,2}$	$\tilde{\rho}_1^{3,3}$	$\tilde{\rho}_2^{1,1}$	$\tilde{\rho}_2^{2,2}$	$\tilde{\rho}_2^{3,3}$
1	3	0.23	0.14	0.09	0.19	0.11	0.07	1	4	0.18	0.10	0.07	0.14	0.07	0.05
2	6	0.12	0.04	0.02	0.12	0.04	0.02	2	10	0.10	0.03	0.01	0.10	0.03	0.01
3	10	0.19	0.04	0.01	0.19	0.04	0.01	3	20	0.20	0.04	0.01	0.20	0.04	0.01
4	15	0.41	0.18	0.08	0.41	0.18	0.08	4	35	0.42	0.18	0.08	0.42	0.18	0.08
5	21	0.61	0.41	0.27	0.61	0.41	0.27	5	56	0.64	0.45	0.31			

Table 6.3

Average convergence rates (two dimensions (left), three dimensions (right)) $\tilde{\rho}_1^{\nu,\nu}$ *for the* $V^{\nu,\nu}$*-cycle and convergence rates* $\tilde{\rho}_2^{\nu,\nu}$ *for the* $W^{\nu,\nu}$*-cycle with* $\nu = 1, 2, 3$*. Covers based on Halton*$(2, 3)$ *point sets in two dimensions, and on Halton*$(2, 3, 5)$ *point sets in three dimensions.*

small and does not seem to effect the convergence rate significantly. A larger number of smoothing steps improves the convergence rates quite a bit but the overall execution times may even increase for the Local-to-Local projection multilevel iteration due to the larger computational work in the smoothing steps. The overall performance of this approach—which exploits not only the general PUM localization but also our hierarchical cover construction—is impressive. The convergence rate is essentially the same as we have for the global L^2-projection, yet the solution times as well as the setup times and associated storage costs are significantly smaller.

Let us now consider higher order discretizations. Here, we use $p_{i,k} = p = 1, 2, 3, 4, 5$ for the local spaces $V_{i,k}^{p_{i,k}}$. Recall that all three projection approaches are exact for polynomials of degree p.

Here, we give the average

$$\tilde{\rho} := \frac{\rho(\tilde{N}_F) + \rho(2^{-d}\tilde{N}_F) + \rho(4^{-d}\tilde{N}_F)}{3}$$

of the convergence rates ρ for the three finest initial point sets $\tilde{P}$ in Table 6.3 for the two-dimensional (left) as well as for the three-dimensional model problem (right). We expect that the rates increase with p since the block-smoother only eliminates local couplings between the polynomials defined on the same cover patch but it does not eliminate couplings between polynomials of neighboring patches. Hence, the p-dependence of the smoother will lead to a p-dependence of the overall convergence rates. From the numbers given in Table 6.3 we clearly see this p-dependence in the convergence rates ρ_1 and ρ_2. It seems that for degrees $p = 2$ and $p = 3$ the local couplings within a patch are dominant over the neighbor couplings since the convergence rates ρ_1 and ρ_2 are at least as good as for $p = 1$. Only for $p > 3$ we see a deterioration of the rates. Again, the convergence rates ρ_2 for the W-cycle are essentially the same as the V-cycle rates, independent of the polynomial degree p.

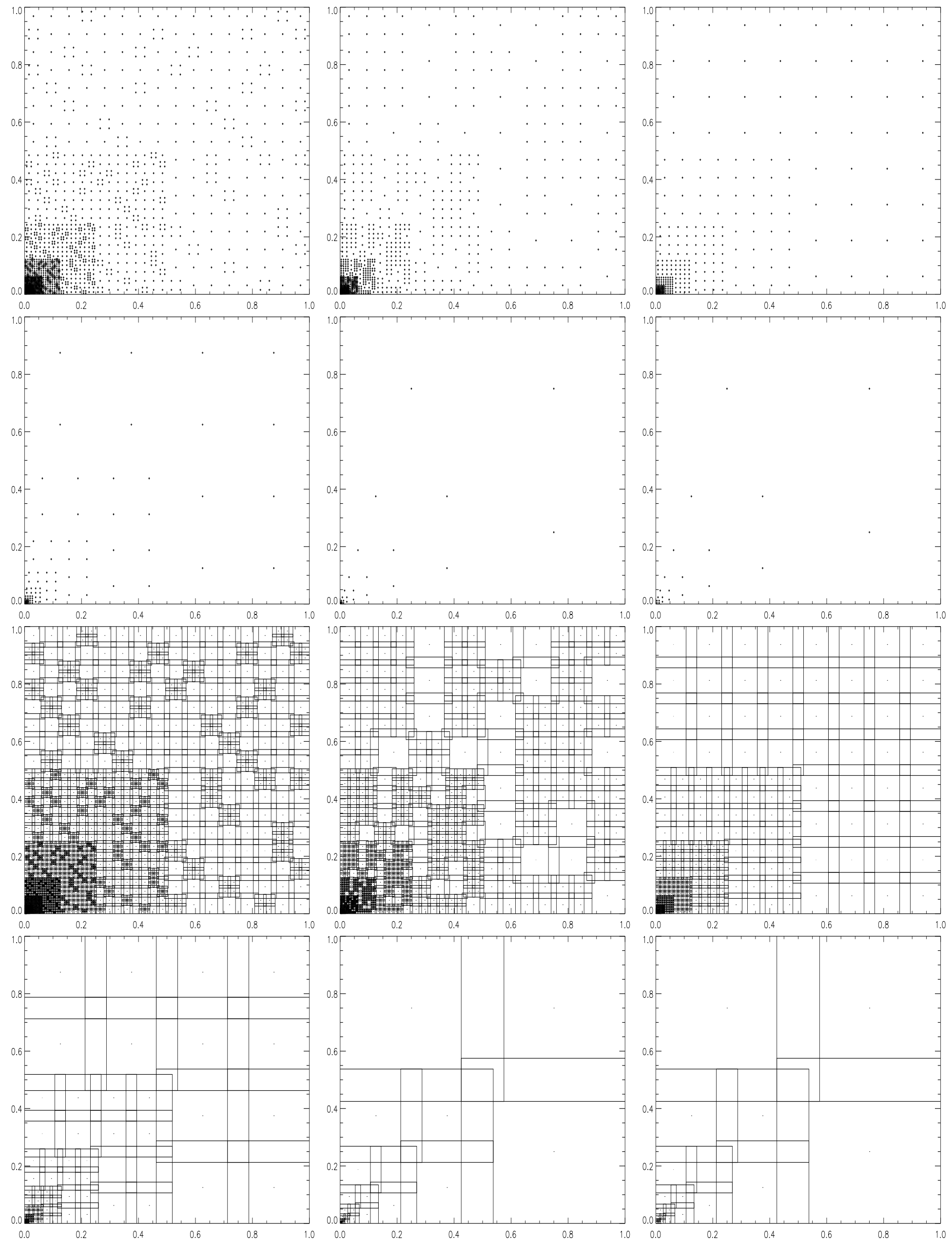

FIGURE 6.4. *Point set and cover hierarchy constructed by Algorithm 2 for a composite point set consisting of* 8 *layers of Halton*$_0^{255}(2,3)$ *point sets in two dimensions.*

Example 2 (Composite Halton Point Sets). In our second example we use the union of multiple Halton sets—each of which is distributed in a subset $\Omega_L \subseteq \Omega$ of the domain—as the initial point set $\tilde{P}$. These layers Ω_L are nested subsets of decreasing size, see Figure 6.4 for several levels of the constructed cover hierarchy for such a composite point set $\tilde{P}$ based on 8 layers of Halton$_0^{255}(2,3)$ point sets in two

Global-to-Global L^2-projection

C_L	card(P)	C_1	C_2	J	p	D_p	$\rho_1^{1,1}$	$\rho_1^{2,2}$	$\rho_1^{3,3}$	$\rho_2^{1,1}$	$\rho_2^{2,2}$	$\rho_2^{3,3}$
6	1921	1.69	6.91	10	1	3	0.242	0.144	0.099	0.210	0.119	0.083
7	2224	1.69	10.02	11	1	3	0.249	0.144	0.105	0.221	0.130	0.090
8	2527	1.70	15.63	12	1	3	0.243	0.144	0.098	0.213	0.119	0.082
9	2830	1.70	25.82	13	1	3	0.249	0.143	0.104	0.222	0.130	0.090
10	3133	1.71	44.50	14	1	3	0.245	0.144	0.097	0.212	0.119	0.083
11	3436	1.71	78.96	15	1	3	0.249	0.144	0.104	0.222	0.130	0.090
12	3739	1.72	142.89	16	1	3	0.245	0.143	0.097	0.213	0.121	0.085
13	4042	1.72	262.10	17	1	3	0.251	0.145	0.104	0.222	0.131	0.090

Global-to-Local L^2-projection

C_L	card(P)	C_1	C_2	J	p	D_p	$\rho_1^{1,1}$	$\rho_1^{2,2}$	$\rho_1^{3,3}$	$\rho_2^{1,1}$	$\rho_2^{2,2}$	$\rho_2^{3,3}$
6	1921	1.69	6.91	10	1	3	0.269	0.159	0.111	0.224	0.129	0.087
7	2224	1.69	10.02	11	1	3	0.277	0.160	0.112	0.237	0.138	0.095
8	2527	1.70	15.63	12	1	3	0.270	0.158	0.110	0.225	0.128	0.087
9	2830	1.70	25.82	13	1	3	0.282	0.159	0.111	0.236	0.138	0.095
10	3133	1.71	44.50	14	1	3	0.270	0.157	0.109	0.226	0.128	0.087
11	3436	1.71	78.96	15	1	3	0.282	0.159	0.111	0.236	0.138	0.094
12	3739	1.72	142.89	16	1	3	0.273	0.157	0.109	0.227	0.130	0.089
13	4042	1.72	262.10	17	1	3	0.283	0.160	0.111	0.236	0.138	0.095

Local-to-Local L^2-projection

C_L	card(P)	C_1	C_2	J	p	D_p	$\rho_1^{1,1}$	$\rho_1^{2,2}$	$\rho_1^{3,3}$	$\rho_2^{1,1}$	$\rho_2^{2,2}$	$\rho_2^{3,3}$
6	1921	1.69	6.91	10	1	3	0.240	0.145	0.100	0.205	0.117	0.082
7	2224	1.69	10.02	11	1	3	0.249	0.146	0.108	0.206	0.122	0.088
8	2527	1.70	15.63	12	1	3	0.236	0.145	0.099	0.206	0.117	0.081
9	2830	1.70	25.82	13	1	3	0.247	0.145	0.107	0.208	0.122	0.088
10	3133	1.71	44.50	14	1	3	0.240	0.145	0.099	0.206	0.116	0.082
11	3436	1.71	78.96	15	1	3	0.247	0.146	0.107	0.209	0.123	0.088
12	3739	1.72	142.89	16	1	3	0.236	0.145	0.098	0.206	0.118	0.083
13	4042	1.72	262.10	17	1	3	0.249	0.150	0.107	0.209	0.127	0.088

TABLE 6.4

Convergence rates $\rho_1^{\nu,\nu}$ for the $V^{\nu,\nu}$-cycle and convergence rates $\rho_2^{\nu,\nu}$ for the $W^{\nu,\nu}$-cycle with $\nu = 1, 2, 3$. Covers based on composite point sets (Halton$_0^{255}(2,3)$ point set per layer, C_L number of layers) in two dimensions.

dimensions. The local approximation spaces again are linear Legendre polynomials.

Due to the use of the composite point set $\tilde{P}$ the coarsening will start with the standard rate of 2^{-d} but the coarsening will eventually break down to a rate close to one. Hence, we expect that (at least) the W-cycle complexity C_2 will no longer be bounded in two and three dimensions. This can be observed from the numerical results presented in Tables 6.4 and 6.5 for the two- and three-dimensional model problem respectively. These numbers also indicate that the V-cycle is of optimal complexity without restricting the iteration to active subsets of patches (see §5.3 and [30]).[4]

From the rates ρ_1 displayed in Table 6.4 we see that the convergence behavior of the V-cycle based on composite point sets is very similar to the one we observed for the Halton point sets in Example 1. Here, we have $\rho_1^{1,1} \simeq 0.25$, $\rho_1^{2,2} \simeq 0.15$ and $\rho_1^{3,3} \simeq 0.10$ for the global L^2-projection independent of the number of discretization points card(P). Again, the difference in the convergence rates due to the change of the interlevel transfers is very small. We measure convergence rates of $\rho_1^{1,1} \simeq 0.28$, $\rho_1^{2,2} \simeq 0.16$ and $\rho_1^{3,3} \simeq 0.11$ for the Global-to-Local projection and $\rho_1^{1,1} \simeq 0.25$,

[4] This technique could be employed to reduce the overall complexity of the W-cycle but the extra work completed on coarser levels does not effect the convergence rate significantly and will only slightly improve the measured rate ρ_2.

Global-to-Global L^2-projection

C_L	card(P)	C_1	C_2	J	p	D_p	$\rho_1^{1,1}$	$\rho_1^{2,2}$	$\rho_1^{3,3}$	$\rho_2^{1,1}$	$\rho_2^{2,2}$	$\rho_2^{3,3}$
6	4243	1.47	5.77	10	1	4	0.222	0.132	0.099	0.208	0.115	0.081
7	4838	1.48	8.69	11	1	4	0.226	0.131	0.092	0.209	0.116	0.081
8	5433	1.48	13.98	12	1	4	0.230	0.135	0.094	0.211	0.118	0.082
9	6028	1.49	23.67	13	1	4	0.222	0.131	0.095	0.206	0.114	0.081
10	6623	1.49	41.50	14	1	4	0.229	0.133	0.092	0.210	0.118	0.083
11	7218	1.50	74.56	15	1	4	0.230	0.136	0.095	0.211	0.117	0.082
12	7813	1.51	136.14	16	1	4	0.223	0.132	0.095	0.205	0.115	0.082

Global-to-Local L^2-projection

C_L	card(P)	C_1	C_2	J	p	D_p	$\rho_1^{1,1}$	$\rho_1^{2,2}$	$\rho_1^{3,3}$	$\rho_2^{1,1}$	$\rho_2^{2,2}$	$\rho_2^{3,3}$
6	4243	1.47	5.77	10	1	4	0.268	0.154	0.114	0.231	0.131	0.089
7	4838	1.48	8.69	11	1	4	0.271	0.154	0.108	0.234	0.131	0.089
8	5433	1.48	13.98	12	1	4	0.272	0.156	0.111	0.235	0.134	0.090
9	6028	1.49	23.67	13	1	4	0.267	0.152	0.110	0.229	0.130	0.090
10	6623	1.49	41.50	14	1	4	0.274	0.156	0.109	0.236	0.133	0.090
11	7218	1.50	74.56	15	1	4	0.272	0.156	0.111	0.234	0.132	0.090
12	7813	1.51	136.14	16	1	4	0.269	0.153	0.110	0.229	0.130	0.090

Local-to-Local L^2-projection

C_L	card(P)	C_1	C_2	J	p	D_p	$\rho_1^{1,1}$	$\rho_1^{2,2}$	$\rho_1^{3,3}$	$\rho_2^{1,1}$	$\rho_2^{2,2}$	$\rho_2^{3,3}$
6	4243	1.47	5.77	10	1	4	0.208	0.128	0.099	0.197	0.110	0.076
7	4838	1.48	8.69	11	1	4	0.213	0.128	0.091	0.200	0.112	0.077
8	5433	1.48	13.98	12	1	4	0.215	0.130	0.095	0.200	0.112	0.081
9	6028	1.49	23.67	13	1	4	0.207	0.127	0.096	0.193	0.111	0.077
10	6623	1.49	41.50	14	1	4	0.213	0.130	0.092	0.199	0.114	0.081
11	7218	1.50	74.56	15	1	4	0.218	0.131	0.096	0.202	0.112	0.077
12	7813	1.51	136.14	16	1	4	0.208	0.128	0.097	0.192	0.112	0.077

TABLE 6.5

Convergence rates $\rho_1^{\nu,\nu}$ for the $V^{\nu,\nu}$-cycle and convergence rates $\rho_2^{\nu,\nu}$ for the $W^{\nu,\nu}$-cycle with $\nu = 1, 2, 3$. Covers based on composite point sets (Halton$_0^{255}(2,3,5)$ point set per layer, C_L number of layers) in three dimensions.

$\rho_1^{2,2} \simeq 0.15$ and $\rho_1^{3,3} \simeq 0.10$ for the Local-to-Local projection. Furthermore, we see no significant improvement in the convergence behavior for the W-cycle compared with the V-cycle. The increased work load on coarser levels does not pay off.

The numerical results given in Table 6.5 show a very similar convergence behavior for the three-dimensional problem. Here, we measure convergence rates $\rho_1^{1,1} \simeq 0.23$, $\rho_1^{2,2} \simeq 0.13$ and $\rho_1^{3,3} \simeq 0.10$ for the global L^2-projection. The convergence rates for the Global-to-Local projection are $\rho_1^{1,1} \simeq 0.27$, $\rho_1^{2,2} \simeq 0.16$ and $\rho_1^{3,3} \simeq 0.11$. We even measure a slight improvement in the convergence rates for the Local-to-Local projection compared with the results for the global L^2-projection. Here, we have $\rho_1^{1,1} \simeq 0.21$, $\rho_1^{2,2} \simeq 0.13$ and $\rho_1^{3,3} \simeq 0.10$. Again, the (non-optimal) W-cycle convergence rates ρ_2 are only slightly better that the V-cycle rates and certainly do not justify the increase in computational work.

The convergence behavior for higher order discretization based on the composite Halton point sets is very similar to the one in Example 1.

Example 3 (Graded Halton Point Sets). In our last example we use a grading function $\mathcal{G} : (\xi_i)_{i=1}^d \in [0,1]^d \mapsto (\xi_i^2)_{i=1}^d \in [0,1]^d$ to transform a Halton point set and use the (more irregular) transformed point set for the cover construction. Several levels of the constructed cover hierarchy for such a graded Halton$_0^{1023}(2,3)$ point set in two dimensions are depicted in Figure 6.5.

Again, we expect that the W-cycle complexity C_2 does not stay bounded due to the irregularity of the point set $\tilde{P}$ (at least in two dimensions). The results displayed in Table 6.6 indicate this behavior. For the V-cycle we still have the optimal complexity

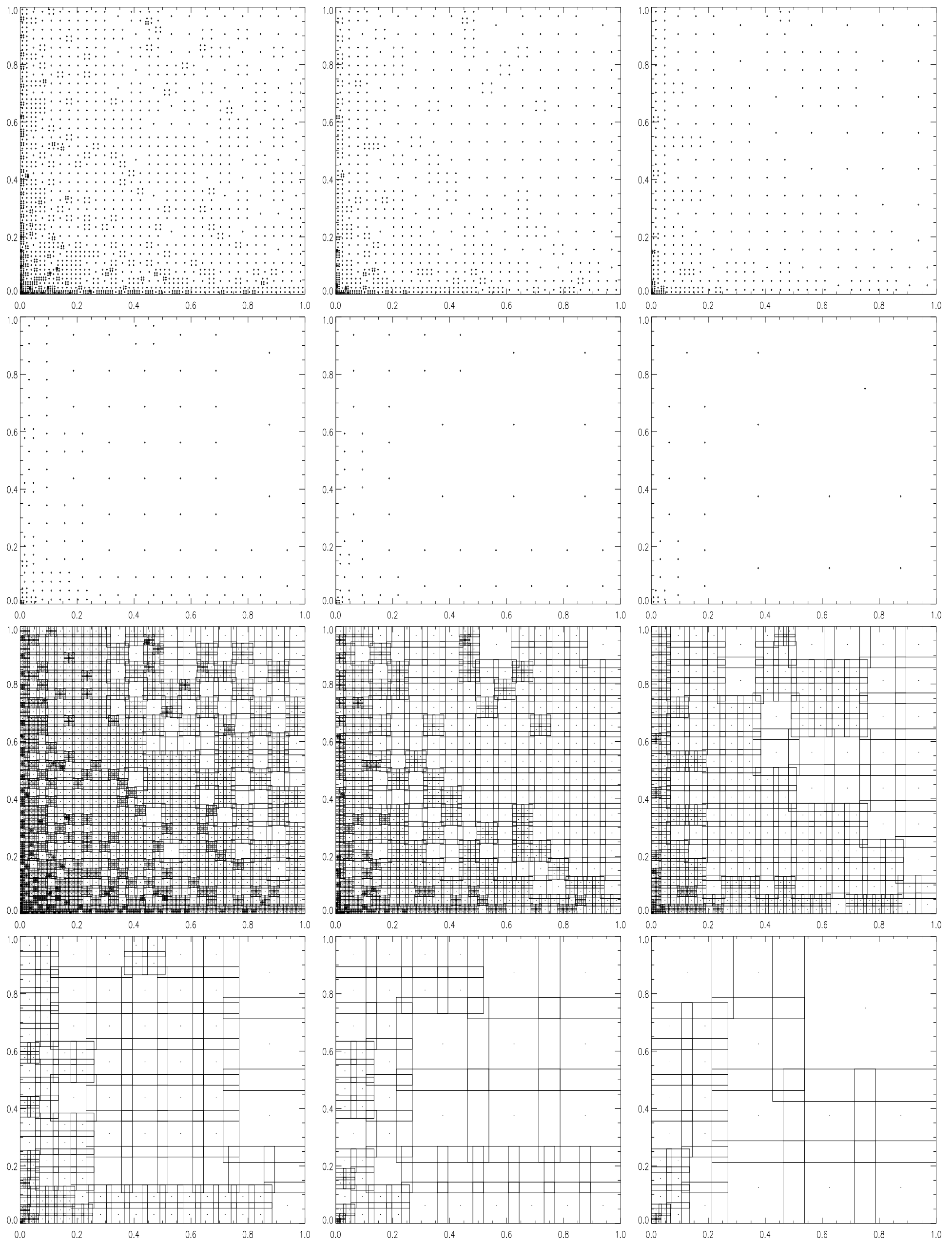

FIGURE 6.5. *Point set and cover hierarchy constructed by Algorithm 2 for a Halton*$_0^{1024}(2,3)$ *point set graded by* $\mathcal{G} : (\xi_i)_{i=1}^2 \mapsto (\xi_i^2)_{i=1}^2$.

of the iteration scheme. The measured convergence rates ρ_1 for the V-cycle and ρ_2 for the W-cycle grow slightly with the number of cover patches card(P). One reason for this could be the smoothing rate of the Gauß–Seidel smoother which is level-dependent in this example. The space filling curve ordering scheme we use is not able to cluster the indices of patches near the edges close together. The function $\mathcal{G}$ provides

Global-to-Global L^2-projection

$\tilde{N}$	card(P)	C_1	C_2	J	p	D_p	$\rho_1^{1,1}$	$\rho_1^{2,2}$	$\rho_1^{3,3}$	$\rho_2^{1,1}$	$\rho_2^{2,2}$	$\rho_2^{3,3}$
16	31	2.13	8.03	5	1	3	0.134	0.060	0.049	0.133	0.056	0.040
64	112	1.91	6.30	6	1	3	0.174	0.090	0.059	0.129	0.068	0.044
256	454	1.80	7.40	8	1	3	0.206	0.112	0.082	0.184	0.096	0.069
1024	1846	1.85	12.85	11	1	3	0.211	0.115	0.083	0.201	0.095	0.067
4096	7468	1.81	13.76	13	1	3	0.248	0.144	0.097	0.243	0.136	0.090
16384	29848	1.82	9.92	14	1	3	0.235	0.140	0.098	0.208	0.118	0.085
65536	119488	1.80	14.34	17	1	3	0.346	0.213	0.156	0.253	0.145	0.106

Global-to-Local L^2-projection

$\tilde{N}$	card(P)	C_1	C_2	J	p	D_p	$\rho_1^{1,1}$	$\rho_1^{2,2}$	$\rho_1^{3,3}$	$\rho_2^{1,1}$	$\rho_2^{2,2}$	$\rho_2^{3,3}$
16	31	2.13	8.03	5	1	3	0.144	0.068	0.052	0.144	0.063	0.046
64	112	1.91	6.30	6	1	3	0.182	0.096	0.064	0.142	0.073	0.047
256	454	1.80	7.40	8	1	3	0.215	0.121	0.085	0.185	0.105	0.069
1024	1846	1.85	12.85	11	1	3	0.224	0.123	0.087	0.202	0.099	0.070
4096	7468	1.81	13.76	13	1	3	0.269	0.158	0.107	0.247	0.142	0.094
16384	29848	1.82	9.92	14	1	3	0.273	0.160	0.113	0.223	0.129	0.089
65536	119488	1.80	14.34	17	1	3	0.346	0.213	0.156	0.270	0.161	0.113

Local-to-Local L^2-projection

$\tilde{N}$	card(P)	C_1	C_2	J	p	D_p	$\rho_1^{1,1}$	$\rho_1^{2,2}$	$\rho_1^{3,3}$	$\rho_2^{1,1}$	$\rho_2^{2,2}$	$\rho_2^{3,3}$
16	31	2.13	8.03	5	1	3	0.129	0.065	0.049	0.126	0.063	0.036
64	112	1.91	6.30	6	1	3	0.173	0.095	0.063	0.129	0.070	0.046
256	454	1.80	7.40	8	1	3	0.214	0.112	0.083	0.189	0.094	0.069
1024	1846	1.85	12.85	11	1	3	0.218	0.121	0.088	0.200	0.099	0.069
4096	7468	1.81	13.76	13	1	3	0.243	0.141	0.097	0.227	0.130	0.086
16384	29848	1.82	9.92	14	1	3	0.234	0.144	0.105	0.210	0.115	0.083
65536	119488	1.80	14.34	17	1	3	0.310	0.190	0.139	0.248	0.136	0.095

TABLE 6.6

Convergence rates $\rho_1^{\nu,\nu}$ for the $V^{\nu,\nu}$-cycle and convergence rates $\rho_2^{\nu,\nu}$ for the $W^{\nu,\nu}$-cycle with $\nu = 1,2,3$. Covers based on Halton$(2,3)$ point sets graded by $\mathcal{G} : (\xi_i)_{i=1}^2 \mapsto (\xi_i^2)_{i=1}^2$.

a grading which is appropriate for edge singularities whereas the space filling curve we use—the Hilbert curve—is more appropriate for ordering adaptive grids for point singularities. This could be cured if we would use a generalized Hilbert curve—similar to the space filling curves developed in [30] for the load balancing problem in parallel adaptive multigrid—for our ordering scheme.

We have also applied the different multilevel iteration schemes with block-Jacobi smoothers to this model problem. Then, the smoothing rate is independent of the order of the cover patches. Here, our Hilbert ordering scheme has no effect on the measured convergence rates. The rates ρ_1 and ρ_2 are expected to drop due to the reduced smoothing quality. But if these rates are level-independent we have a good indication that the rates given in Table 6.6 are bounded independent of card(P).

The measured convergence rates (see Table 6.7) for the global L^2-projection Π_{k-1}^k and the Global-to-Local projection $\hat{\Pi}_{k-1}^k$ with a block-Jacobi smoother seem to be level independent. We measure $\rho_1^{1,1} \simeq 0.55$, $\rho_1^{2,2} \simeq 0.36$ and $\rho_1^{3,3} \simeq 0.26$ for the global L^2-projection and $\rho_1^{1,1} \simeq 0.56$, $\rho_1^{2,2} \simeq 0.37$ and $\rho_1^{3,3} \simeq 0.27$ for the Global-to-Local projection. Note that the convergence rates for the multilevel iteration based on the Local-to-Local projection $\tilde{\Pi}_{k-1}^k$ are not level-independent. Here, the iteration may even diverge.

This is due to that fact that the Local-to-Local projection produces “rougher” approximations $\tilde{u}_k$ than the other two projections since no neighbor relations are taken into account in the transfer. Therefore, a small number of steps of the block-Jacobi smoother—which pays less attention to the neighboring patches than the block-Gauß–

Global-to-Global L^2-projection

$\tilde{N}$	card(P)	C_1	C_2	J	p	D_p	$\rho_1^{1,1}$	$\rho_1^{2,2}$	$\rho_1^{3,3}$	$\rho_2^{1,1}$	$\rho_2^{2,2}$	$\rho_2^{3,3}$
16	31	2.13	8.03	5	1	3	0.523	0.309	0.184	0.519	0.307	0.182
64	112	1.91	6.30	6	1	3	0.511	0.276	0.160	0.502	0.270	0.151
256	454	1.80	7.40	8	1	3	0.542	0.343	0.221	0.539	0.340	0.212
1024	1846	1.85	12.85	11	1	3	0.533	0.323	0.205	0.529	0.315	0.201
4096	7468	1.81	13.76	13	1	3	0.532	0.328	0.205	0.530	0.324	0.199
16384	29848	1.82	9.92	14	1	3	0.534	0.334	0.228	0.531	0.328	0.206
65536	119488	1.80	14.34	17	1	3	0.546	0.358	0.254	0.544	0.356	0.222

Global-to-Local L^2-projection

$\tilde{N}$	card(P)	C_1	C_2	J	p	D_p	$\rho_1^{1,1}$	$\rho_1^{2,2}$	$\rho_1^{3,3}$	$\rho_2^{1,1}$	$\rho_2^{2,2}$	$\rho_2^{3,3}$
16	31	2.13	8.03	5	1	3	0.560	0.342	0.209	0.555	0.339	0.206
64	112	1.91	6.30	6	1	3	0.542	0.316	0.185	0.533	0.310	0.182
256	454	1.80	7.40	8	1	3	0.568	0.365	0.228	0.566	0.364	0.227
1024	1846	1.85	12.85	11	1	3	0.561	0.356	0.221	0.558	0.350	0.215
4096	7468	1.81	13.76	13	1	3	0.558	0.350	0.215	0.556	0.348	0.212
16384	29848	1.82	9.92	14	1	3	0.570	0.376	0.242	0.567	0.374	0.233
65536	119488	1.80	14.34	17	1	3	0.560	0.367	0.268	0.559	0.366	0.227

Local-to-Local L^2-projection

$\tilde{N}$	card(P)	C_1	C_2	J	p	D_p	$\rho_1^{1,1}$	$\rho_1^{2,2}$	$\rho_1^{3,3}$	$\rho_2^{1,1}$	$\rho_2^{2,2}$	$\rho_2^{3,3}$
16	31	2.13	8.03	5	1	3	0.519	0.301	0.179	0.515	0.300	0.176
64	112	1.91	6.30	6	1	3	0.508	0.272	0.160	0.501	0.267	0.145
256	454	1.80	7.40	8	1	3	0.533	0.330	0.223	0.530	0.326	0.209
1024	1846	1.85	12.85	11	1	3	0.551	0.313	0.203	0.518	0.304	0.192
4096	7468	1.81	13.76	13	1	3	0.608	0.322	0.204	0.524	0.313	0.192
16384	29848	1.82	9.92	14	1	3	0.618	0.329	0.230	0.524	0.320	0.203
65536	119488	1.80	14.34	17	1	3	1.096	0.613	0.306	1.032	0.582	0.308

TABLE 6.7

Convergence rates $\rho_1^{\nu,\nu}$ for the $V^{\nu,\nu}$-cycle and convergence rates $\rho_2^{\nu,\nu}$ for the $W^{\nu,\nu}$-cycle with $\nu = 1,2,3$ and Jacobi smoothing. Covers based on Halton$(2,3)$ point sets graded by $\mathcal{G} : (\xi_i)_{i=1}^2 \mapsto (\xi_i^2)_{i=1}^2$.

Seidel smoother—is not sufficient to achieve a level-independent convergence. In fact the iteration diverges for the $V^{1,1}$- and $W^{1,1}$-cycle for the graded Halton point set on the finest level with 358464 unknowns. The rates $\rho^{2,2}$ and $\rho^{3,3}$ for the cycles with a larger number of smoothing steps are close to the level-independent rates we measure for the other two projections but it is not clear if these rates will stay bounded. The jump in $\rho_1^{2,2}$ to 0.61 in the last refinement step may indicate that they will not be bounded away from one. But this jump could also be due to the substantial change in the local resolution we have in this refinement step. Here, the finest patchsize is of order $O(2^{-17})$ whereas the previous resolution was of order $O(2^{-14})$ only. Furthermore, we observe a slight increase in the convergence rates also for the global L^2-projection (with block-Jacobi or block-Gauß–Seidel smoothing) on this refinement level.

In three dimensions not only the V-cycle but also the W-cycle complexity is bounded, see Table 6.8. The convergence behavior is very similar to that of the two-dimensional example. The rates are $\rho_1^{1,1} \simeq 0.23$, $\rho_1^{2,2} \simeq 0.12$ and $\rho_1^{3,3} \simeq 0.10$ on the finest level. We also observe a slight level-dependence of these rates ρ_1 and ρ_2. Again, this increase can be explained with the reduced smoothing rate of the order-dependent Gauß–Seidel smoother. The convergence rates ρ_2 for the W-cycle are essentially the same as the respective V-cycle rates ρ_1.

The convergence behavior for higher order discretization based on the graded Halton point sets is very similar to the one in Example 1.

Global-to-Global L^2-projection

$\tilde{N}$	card(P)	C_1	C_2	J	p	D_p	$\rho_1^{1,1}$	$\rho_1^{2,2}$	$\rho_1^{3,3}$	$\rho_2^{1,1}$	$\rho_2^{2,2}$	$\rho_2^{3,3}$
16	50	1.92	4.68	4	1	4	0.077	0.041	0.029	0.076	0.028	0.023
128	358	1.56	3.01	5	1	4	0.158	0.089	0.062	0.152	0.077	0.050
1024	3193	1.53	3.06	7	1	4	0.193	0.105	0.075	0.164	0.086	0.056
8192	24977	1.53	3.12	9	1	4	0.222	0.111	0.077	0.214	0.099	0.052

Global-to-Local L^2-projection

$\tilde{N}$	card(P)	C_1	C_2	J	p	D_p	$\rho_1^{1,1}$	$\rho_1^{2,2}$	$\rho_1^{3,3}$	$\rho_2^{1,1}$	$\rho_2^{2,2}$	$\rho_2^{3,3}$
16	50	1.92	4.68	4	1	4	0.074	0.048	0.040	0.076	0.033	0.030
128	358	1.56	3.01	5	1	4	0.199	0.103	0.070	0.182	0.090	0.055
1024	3193	1.53	3.06	7	1	4	0.233	0.129	0.087	0.189	0.107	0.069
8192	24977	1.53	3.12	9	1	4	0.247	0.130	0.091	0.227	0.112	0.063

Local-to-Local L^2-projection

$\tilde{N}$	card(P)	C_1	C_2	J	p	D_p	$\rho_1^{1,1}$	$\rho_1^{2,2}$	$\rho_1^{3,3}$	$\rho_2^{1,1}$	$\rho_2^{2,2}$	$\rho_2^{3,3}$
16	50	1.92	4.68	4	1	4	0.090	0.040	0.030	0.088	0.025	0.021
128	358	1.56	3.01	5	1	4	0.153	0.088	0.063	0.145	0.075	0.050
1024	3193	1.53	3.06	7	1	4	0.192	0.105	0.076	0.162	0.083	0.055
8192	24977	1.53	3.12	9	1	4	0.226	0.115	0.083	0.221	0.099	0.054

TABLE 6.8

Convergence rates $\rho_1^{\nu,\nu}$ for the $V^{\nu,\nu}$-cycle and convergence rates $\rho_2^{\nu,\nu}$ for the $W^{\nu,\nu}$-cycle with $\nu = 1, 2, 3$. Covers based on Halton$(2, 3, 5)$ point sets graded by $\mathcal{G} : (\xi_i)_{i=1}^3 \mapsto (\xi_i^2)_{i=1}^3$.

7. Concluding Remarks.

We presented a multilevel solver for linear systems arising from a partition of unity discretization of an elliptic PDE of second order. The main ingredients of the solver are the use of (localized) L^2-projections for the interlevel transfer between the nonnested function spaces and the employment of block-smoothers to treat all local degrees of freedom on a PUM patch simultaneously. The results of our numerical examples clearly showed that the convergence behavior of the presented iteration schemes are independent of the number of discretization points card(P) and the distribution of these points; yet the convergence rates are slightly dependent on the order of the approximation.

The global L^2-projection is certainly too expensive to be used in applications but the convergence rates of a multilevel iteration based on this projection give a good idea of the convergence rate we can achieve with the projection approach.

The L^2-projections could be localized due to the general PUM construction. This reduced the iteration times significantly but the setup for such interlevel transfer operators was still substantial. Note that this first localized multilevel solver based on the Global-to-Local L^2-projections though can be used for any partition of unity and any coarsening or refinement strategy. Due to our hierarchical cover construction we were able to localize the approximation problem even further. Now the setup cost for the Local-to-Local projection operators was negligible compared with the cost of the operator assembly. Yet the convergence rates of a multilevel iteration with these completely localized projections were (almost) identical to those for the global L^2-projection. This observation holds even for higher order discretizations.

Altogether, the computational effort involved in the construction of the interlevel transfer operators was substantially reduced by the presented localizations of the projection approach. Yet the results of our numerical examples in two and three dimensions showed that the quality of the interlevel transfer with the Local-to-Local L^2-projection $\tilde{\Pi}_{k-1}^k$ is comparable to that of the global L^2-projection Π_{k-1}^k although $\tilde{\Pi}_{k-1}^k$ produces “rougher” approximations u_k. The resulting convergence rates ρ_1 and ρ_2 are similar if we use a block-Gauß–Seidel smoother with a space filling curve ordering induced by the Hilbert curve of the cover patches. If we employ only a block-Jacobi smoother in the multilevel iteration the impact of the roughing effect of $\tilde{\Pi}_{k-1}^k$ can be significant for highly irregular point sets. Here, iterations with the global projection Π_{k-1}^k or with the Global-to-Local projection $\hat{\Pi}_{k-1}^k$ still converge independent of the number of discretization points $\mathrm{card}(P)$ and their distribution (yet they are still dependent on the local approximation orders $p_{i,k}$), whereas a multilevel iteration with the Local-to-Local projection $\tilde{\Pi}_{k-1}^k$ may even diverge if the number of block-Jacobi-smoothing steps is too small.

In all our experiments we found that the use of a W-cycle does not pay off. The convergence rates were essentially the same as the V-cycle rates. In summary, our numerical experiments indicate that a multilevel V-cycle based on the Local-to-Local projection and a block Gauß–Seidel smoother with an appropriate space filling curve ordering of the cover patches is a cheap and efficient solver for the linear systems arising from a PUM discretization of an elliptic PDE. The convergence rate is independent of the number and the distribution of the discretization points, yet it is slightly dependent on the local approximation orders $p_{i,k}$.

REFERENCES

[1] N. R. Aluru, *A Point Collocation Method Based on Reproducing Kernel Approximations*, Int. J. Numer. Meth. Engrg., 47 (2000), pp. 1083–1121.

[2] I. Babuška and J. M. Melenk, *The Partition of Unity Finite Element Method: Basic Theory and Applications*, Comput. Meth. Appl. Mech. Engrg, 139 (1996), pp. 289–314. Special Issue on Meshless Methods.

[3] ———, *The Partition of Unity Method*, Int. J. Numer. Meth. Engrg., 40 (1997), pp. 727–758.

[4] T. Belytschko, Y. Y. Lu, and L. Gu, *Element-free Galerkin methods*, Int. J. Numer. Meth. Engrg., 37 (1994), pp. 229–256.

[5] D. Braess and W. Hackbusch, *A New Convergence Proof for the Multigrid Method Including the V Cycle*, SIAM J. Numer. Anal., 20 (1983), pp. 967–975.

[6] D. Braess and R. VerfühRT, *Multi-Grid Methods for Nonconforming Finite Element Methods*, SIAM J. Num. Anal., 27 (1990), pp. 979–986.

[7] J. H. Bramble, J. E. Pasciak, and J. Xu, *The Analysis of Multigrid Algorithms with Nonnested Spaces or Noninherited Quadratic Forms*, Math. Comp., 56 (1991), pp. 1–34.

[8] J. H. Bramble, J. E. Pasicak, and J. Xu, *Parallel Multilevel Preconditioners*, Math. Comp., 55 (1990), pp. 1–22.

[9] S. C. Brenner, *Convergence of Nonconforming Multigrid Methods without Full Elliptic Regularity*, Math. Comp., 68 (1999), pp. 25–53.

[10] A. Caglar, M. Griebel, M. A. Schweitzer, and G. Zumbusch, *Dynamic Load-Balancing of Hierarchical Tree Algorithms on a Cluster of Multiprocessor PCs and on the Cray T3E*, in Proceedings 14th Supercomputer Conference, Mannheim, H. W. Meuer, ed., Mannheim, Germany, 1999, Mateo.

[11] C. A. M. Duarte and J. T. Oden, *hp Clouds – A Meshless Method to Solve Boundary Value Problems*, Num. Meth. for PDE, 12 (1996), pp. 673–705.

[12] G. Fasshauer, *Solving Differential Equations with Radial Basis Functions: Multilevel Methods and Smoothing*, Adv. Comp. Math., 11 (1999), pp. 139–159.

[13] C. Franke and R. Schaback, *Convergence Orders of Meshless Collocation Methods using Radial Basis Functions*, Adv. in Comput. Math., 8 (1998), pp. 381–399.

[14] ———, *Solving Partial Differential Equations by Collocation using Radial Basis Functions*, Appl. Math. and Comput., 93 (1998), pp. 73–82.

[15] M. Griebel and M. A. Schweitzer, *A Particle-Partition of Unity Method for the Solution of Elliptic, Parabolic and Hyperbolic PDE*, SIAM J. Sci. Comp., 22 (2000), pp. 853–890.

[16] ———, *A Particle-Partition of Unity Method—Part II: Efficient Cover Construction and Reliable Integration*, SFB Preprint, Sonderforschungsbereich 256, Institut für Angewandte Mathematik, Universität Bonn, 2001. submitted.

[17] ———, *A Particle-Partition of Unity Method—Part IV: Parallelization*, SFB Preprint, Sonderforschungsbereich 256, Institut für Angewandte Mathematik, Universität Bonn, 2001. in preparation.

[18] F. C. Günther and W. K. Liu, *Implementation of Boundary Conditions for Meshless Methods*, Comput. Meth. Appl. Mech. Engrg., 163 (1998), pp. 205–230.

[19] W. Hackbusch, *Multi-Grid Methods and Applications*, vol. 4 of Springer Series in Computational Mathematics, Springer, Berlin, Heidelberg, New York, 1985.

[20] T. J. Liszka, C. A. M. Duarte, and W. W. Tworzydlo, *hp-Meshless Cloud Method*, Comp. Meth. Appl. Mech. Engrg., 139 (1996), pp. 263–288.

[21] J. J. Monaghan, *Why Particle Methods Work*, SIAM J. Sci. Stat. Comput., 3 (1982), pp. 422–433.

[22] ———, *An Introduction to SPH*, Comput. Phys. Comm., 48 (1988), pp. 89–96.

[23] H. Neunzert, A. Klar, and J. Struckmeier, *Particle Methods: Theory and Applications*, Tech. Rep. 95-153, Arbeitsgruppe Technomathematik, Universität Kaiserslautern, 1995.

[24] P. Oswald, *Intergrid Transfer Operators and Multilevel Preconditioners for Nonconforming Discretizations*, Appl. Numer. Math., 23 (1997).

[25] L. F. Pavarino and O. B. Widlund, *A Polylogarithmic Bound for an Iterative Substructuring Method for Spectral Elements in Three Dimensions*, SIAM J. Numer. Anal., (1996).

[26] H. Sagan, *Space-Filling Curves*, Springer, New York, 1994.

[27] M. A. Schweitzer, *Ein Partikel–Galerkin–Verfahren mit Ansatzfunktionen der Partition of Unity Method*, Diplomarbeit, Institut für Angewandte Mathematik, Universität Bonn, 1997.

[28] J. Xu, *Theory of Multilevel Methods for Elliptic Problems*, PhD thesis, Cornell University, 1989.

[29] ———, *Iterative Methods by Space Decomposition and Subspace Correction*, SIAM Review, 34 (1992), pp. 581–613.

[30] G. Zumbusch, *Adaptive Parallel Multilevel Methods for Partial Differential Equations*, Habilitation, Insitut für Angewandte Mathematik, Universität Bonn, 2001.