Comput Visual Sci 1:15–25 (1997)

Computing and
Visualization in Science

Composite finite elements for problems containing small geometric details

Part II: Implementation and numerical results

W. Hackbusch, S.A. Sauter

Lehrstuhl Praktische Mathematik, Universität Kiel, Hermann-Rodewald-Strasse 3, D-24098 Kiel, Germany (e-mail: sas@numerik.uni-kiel.de)

Received: 23 September 1996 / Accepted: 23 January 1997
Communicated by G. Wittum

Abstract. In this paper, we will present efficient strategies how composite finite elements can be realized for the discretization of PDEs on domains containing small geometric details. In contrast to standard finite elements, the minimal dimension of this new class of finite element spaces is completely independent of the number of geometric details of the physical domains. Hence, it allows coarse level discretization of PDEs which can be used, e.g., preferably for multigrid methods and homogenization of PDEs in non-periodic situations.

1 Introduction

In many engineering situations, the physical objects under consideration have an extremely complicated shape containing a huge number of geometric details. Any reasonable mesh generator will produce grids where the (minimal) number of elements is strongly related to the number of these geometric details. As a consequence, the (minimal) dimension of the corresponding finite element space is huge, too. This fact strongly reduces the efficiency of fast multi-level solvers for the arising system of linear equations since no adequate coarse discretization is available. In [9] and [10], *Composite Finite Elements* have been introduced for these kinds of problems. Here, the (minimal) dimension of the finite element space is completely independent of the geometric details while the asymptotic approximation property is valid also for the very coarse discretizations. In combination with multi-grid methods, this new class of elements provides appropriate coarse discretizations for the so-called coarse-grid corrections.

Composite finite elements can also be applied for the homogenization of partial differential operators in non-periodic situations. The finite element discretization of a differential operator can be re-interpreted as a differential operator. Studying the behaviour of these operators on coarser and coarser levels (relative to the size and number of microstructures) gives insights on the behaviour of homogenized differential operators.

This paper can be interpreted as part II of [9]. In the first part, we have set up the spaces in a theoretical way proving the basic approximation results. Here, we will focus on the implementation details and present numerical experiments.

In the literature, there exist several approaches for coarsening finite element spaces and the corresponding systems of linear equations, see [1], [2], [12], [13], [14], [3], [5]. For detailed comments on these references we refer to part I of this paper.

2 Grid generation

In this section, we will explain how a sequence of grids for composite finite elements can be generated. In contrast to standard finite element grids these grids *cannot* be regarded as an approximation of the domains. However, we will not define standard finite element spaces on these grids but include the geometric details in the definition of finite element function spaces in an appropriate way. We assume that $\Omega \subset \mathbb{R}^d$ is a domain with piecewise smooth boundary $\Gamma := \partial\Omega$. However, we have in mind that this boundary might contain a huge number of micro-structures. We begin with outlining the principal underlying ideas for the generation of composite finite element grids.

In the first phase, a hierarchy of auxiliary coverings $\{\tilde{\tau}_\ell\}_{0\le\ell\le\ell_{\max}}$ of the boundary is generated by refining only those elements of an initial covering $\tilde{\tau}_0$ of Ω which intersect the boundary. Note that the possibly very coarse auxiliary grid $\tilde{\tau}_0$ *cannot* be regarded as an approximation of the boundary but only has to satisfy $\Omega \subset \operatorname{dom}\tilde{\tau}_0$. Here, and in the following $\operatorname{dom}\tau$ denotes the interior of the domain covered by the grid: $\operatorname{dom}\tau := int\left(\overline{\bigcup_{K\subset\tau} K}\right)$. The finest "near-boundary" grid, $\tilde{\tau}_{\ell_{\max}}$, should have the property that "small distortions" of the elements and grid points results in an adapted grid $\tau_{\ell_{\max}}$ which represents a proper resolution of the boundary. Proper resolution means that Γ can be parametrized smoothly by edges and faces of elements of $\tau_{\ell_{\max}}$ lying at the boundary.

Let us assume that we want to discretize a partial differential equation on a refinement level ℓ. Then, in the second phase the full grids are generated up to level ℓ. This is done

by refining the initial grid $\tilde{\tau}_0$ by using any refinement strategy for the elements lying *inside* of Ω and, for the elements intersecting the boundary, by using the same refinement pattern as employed for assembling the near-boundary grids. This results in a *full* grid on level ℓ covering the whole domain and a sequence of near-boundary grids on finer levels $\ell' \geq \ell$ resolving the boundary. It turns out that, by using this construction of grids, the work for assembling the system of linear equations on level ℓ is essentially governed by the number of grid points on level ℓ. If all complete grids would be generated up to the finest level $\ell_{\max} \gg \ell$, then, the complexity of the method would be related to the number of grid points on the *finest* level. The definition of the composite finite element spaces on these grids will be given in the next section.

As mentioned above, for the grid generation, the decision whether an element intersects the boundary plays an important role. This information can be generated in various ways and strongly depends on how the geometric information is supplied by the user. Since the generation of this information is independent of the definition of composite finite elements, we will explain appropriate search algorithms in the appendix. In the following, we will formulate the algorithms for the grid generation in an algorithmic way using a pseudo computer language similar to PASCAL. We will need the following definitions.

Definition 1 *Let τ denote a finite element grid. The neighbourhood of an element $K \in \tau$ is defined by $\mathcal{N}(K) := \left\{K' \in \tau \mid \overline{K'} \cap \overline{K} \neq \emptyset\right\}$. The set of vertices of a finite element $K \in \tau$ is denoted by $\mathbf{V}(K)$ and the set of edges by $\mathbf{E}(K)$.*

Let $\tilde{\tau}_0$ be a possibly very coarse, user-supplied coarse grid satisfying $\Omega \subset \operatorname{dom} \tilde{\tau}_0$. We assume that, for any element K, there exists a *regular refinement pattern* defined on an affine equivalent reference element in a coordinate-free way (cf. [4, Sect. 3.4]). The *set of children* $\sigma(K)$ consists of finite elements (on the finer level) satisfying $K' \cap K'' = \emptyset$ for all non-identical $K', K'' \in \sigma(K)$ and $\operatorname{dom} K = \operatorname{dom} \sigma(K)$. For a triangle K, the set of children is given by the four triangles arising by connecting the midpoints of the edges.

Let tol denote a tolerance which is user-supplied and characterizes the required resolution of the boundary. To be more concrete we assume that the geometric details can be resolved by elements having diameter $O(tol)$ and the refinement near the boundary can be stopped for elements satisfying $\operatorname{diam} K \leq tol$. The formal definition of the near-boundary grids is given by the following recursion

$$\tilde{\tau}_{i+1/2} := \left\{K \in \tilde{\tau}_i \mid \exists K' \in \mathcal{N}(K) : \operatorname{diam} K' > tol \wedge K' \cap \Gamma \neq \emptyset\right\},$$
$$\tilde{\tau}_{i+1} := \left\{\sigma(K) : K \in \tilde{\tau}_{i+1/2}\right\}.$$

Since we will investigate the work needed for realizing composite finite elements, we formulate this definition also in an algorithmic way. The procedure **refine** is called by

$\ell_{\max} = -1$; refine$(tol, \ell_{\max})$;

and defined by

procedure refine$(tol; \ell)$;
begin
while $\tilde{\tau}_{\ell+1} \neq \emptyset$ **do begin**
$\ell := \ell + 1$;
for all $K \in \tilde{\tau}_\ell$ **do begin**
if $\{K' \in \mathcal{N}(K) : K' \cap \Gamma \neq \emptyset \wedge \operatorname{diam} K' > tol\} \neq \emptyset$ **then begin**
refine K regularly and generate set of children $\sigma(K)$;
for all $K' \in \sigma(K)$ **do begin**
$\tilde{\tau}_{\ell+1} := \tilde{\tau}_{\ell+1} \cup K'$;
$parent(K') := K$
end end end end end;

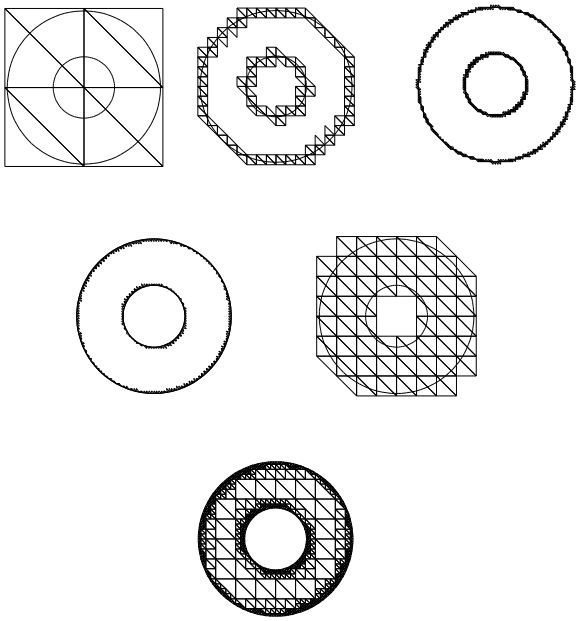

Fig. 1. The first row shows a sequence of near-boundary grids on different levels. Only those triangles are refined which intersect the boundary. In the second row the adapted near-boundary grid on level $\ell_{\max}$ is depicted which properly resolves the boundary. Then, a full grid τ_ℓ is depicted which will be used to define the degrees of freedom of the finite element space on level ℓ. Finally, in the last row, a CFE-mesh of the domain is presented which later is referred to as Ω_2^ε

A typical hierarchy of near-boundary grids is depicted in Fig. 1.

Note that this refinement algorithm can easily be generalized to the case where the tolerance tol is varying over the domain.

In the next step the finest near-boundary grid is adapted to the boundary by small distortions of the elements and grid points. Let $K \in \tau$ denote a finite element and $\mathbf{E}(K)$ the set of edges. For $e = \overline{XY} \in \mathbf{E}(K)$, we define the function $\nu(e)$ by

$$\nu(e) := \begin{cases} boundary & \text{if } \exists x_1, x_2 \in \{X, Y\} : x_1 \in \Omega \wedge x_2 \notin \overline{\Omega} \\ 0 & \text{otherwise.} \end{cases}$$

For elements K, the function is defined analogously by

$$\nu(K) := \begin{cases} boundary & \text{if } \exists e \in \mathbf{E}(K) : \nu(e) = boundary \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

This function is depicted in Fig. 2.

To adapt the finest grid to the boundary one has to investigate all edges having the attribute *boundary*. For such

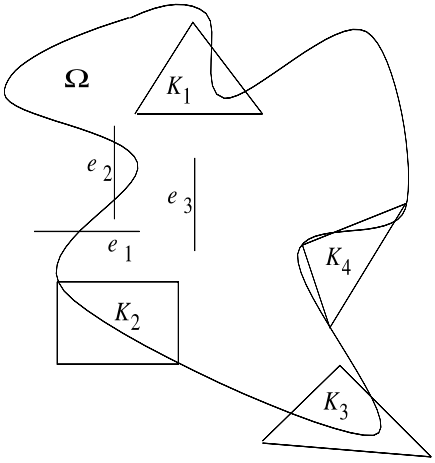

Fig. 2. Illustration of the functions γ, ν and μ. In the depicted case, there holds $\nu(e_1) = \nu(K_3) = boundary$ and $\nu(e_2) = \nu(e_3) = \nu(K_1) = \nu(K_2) = \nu(K_4) = 0$. However, all triangles intersect the boundary implying $\gamma(K_i) = boundary$ for all $1 \leq i \leq 4$. The triangle K_4 lies essentially outside the domain, $\mu(K \cap \Omega) < \delta$, and can be rejected

an edge $e = \overline{XY}$, we assume that there exists a point $Z \in \Gamma$ having the property that, if either X or Y is replaced by Z, then, the diameters of the elements touching Z are increased at most by a moderate factor and are still shape-regular. For example $Z \in e \cap \Gamma$ might be a good choice or the orthogonal projection of X onto Γ. This replacement can formally be expressed by employing a function λ mapping an edge $e = \overline{XY}$ with $\nu(e) = boundary$ onto $\mathbb{R}^d \times \mathbb{R}^d$

$$\lambda(e) = \begin{cases} (X, Z) \text{ if } Y \text{ is replaced by } Z, \\ (Z, Y) \text{ if } X \text{ is replaced by } Z. \end{cases}$$

The procedure **adapt** is called by

for all $K \in \tau_{\ell_{\max}}$ with $\nu(K) = boundary$ **do adapt**(K);

and defined by

procedure adapt(K);
begin
  for all $e = \overline{XY} \in \mathbf{E}(K)$ **do begin**
    if $\nu(e) = boundary$ **then begin**
      replace the vertices X, Y by $\lambda(e)$;
Comment: Note that this replacement changes the shape of K and *all*
other elements which have this point as a vertex on *all* levels.
    end
end end;

Remark 2 *Note that, for quadrilateral elements, the algorithm* **adapt** *could result in the following situation (see the element K_2 in Fig. 2). Let K be a quadrilateral with vertices $\{X_i\}_{1 \leq i \leq 4}$ in counter-clockwise ordering satisfying $X_1 \in \Omega$, $X_3 \notin \Omega$, and $X_2, X_4 \in \Gamma$. Then, $\nu(e) \neq boundary$ holds for all edges $e \in \mathbf{E}(K)$ and, hence, the procedure* **adapt** *does not move any point X_i. The situation can easily be remedied by subdividing K into two triangles. Similar constructions can be applied in* 3*-d as well.*

In the second row of Fig. 1, the result of the adaptation procedure is depicted applied to the near-boundary grid shown in the first row of the same figure.

It is important that, by this adaptation process, the *logical* parent/child relations are preserved. However, the physical nestedness of the grids is violated, i.e., $\operatorname{dom}\sigma(K) \subset \operatorname{dom} K$ is *not* valid in general. The computation of the set $\Gamma \cap e$ in the algorithm **adapt** strongly depends on how the boundary Γ is prescribed by the user. Appropriate search algorithms will be presented in the appendix. We assume that the given tolerance tol is small enough such that the mesh $\tau_{\ell_{\max}}$ is non-degenerate, i.e., the interior angles of the elements are not too large. Now, we will diminish the number of elements of the grid $\tau_{\ell_{\max}}$ by rejecting elements lying essentially outside of Ω. This is done by the procedure **reduce_mesh**. In the following, the function μ denotes the area measure of a set. Furthermore, we assume that a user-specified tolerance (depending on the required degree of approximation), say δ, is given to determine whether an element lying essentially outside of Ω, i.e., $\mu(K \cap \Omega) < \delta$, can be rejected from a grid without reducing the approximation order.

procedure reduce_mesh;
begin
  for all $K \in \tau_{\ell_{\max}}$ **do begin**
    if $\mathbf{V}(K) \not\subset \Omega$ **or** $\mu(K \cap \Omega) < \delta$ **then**
      $\tau_{\ell_{\max}} := \tau_{\ell_{\max}} \setminus K$;
  end;
  for $\ell = \ell_{\max} - 1$ **downto** 0 **do begin**
    $\tau_\ell := \{K \in \tau_\ell \mid \sigma(K) \cap \tau_{\ell+1} \neq \emptyset\}$
    $\cup \{K \in \tau_\ell \mid$ there exists a vertex x in $\tau_{\ell+1}$
      with $x \in \overset{\circ}{K}\}$
end end;

The element K_4 in Fig. 2 is an example of a triangle which satisfies $\mathbf{V}(K_4) \subset \Omega$ but $\mu(K_4 \cap \Omega) < \delta$. In the appendix, we will discuss strategies for the computation of $\mu(K \cap \Gamma)$.

This completes the generation of the hierarchy of near-boundary meshes. We come now to the second phase of the algorithm where the full grid on level ℓ is generated. The full grids are generated by refining sequentially the initial grid $\tilde{\tau}_0$ using, for the elements which intersects the boundary, the same refinement pattern as for the generation of the "near-boundary" grids. Let $\ell \leq \ell_{\max}$. The procedure for generating τ_ℓ is defined by

procedure generate_full_grid(ℓ);
begin
  for $m = 0$ **to** $\ell - 1$ **do begin**
    for all $K \in \tilde{\tau}_m$ **do begin**
      if $\sigma(K) = \emptyset$ **do begin**
        refine K and generate the set of children $\sigma(K)$;
        for all $K' \subset \sigma(K)$ **do** $parent\,(K') = K$;
      end;
      $\tilde{\tau}_{m+1} := \tilde{\tau}_{m+1} \cup \sigma(K)$
end end end;

The final grids τ_ℓ are given by replacing those grid points of $\tilde{\tau}_\ell$ which were adapted to the boundary in **procedure adapt** by the modified grid points. This step can conveniently be included in the **procedure generate_full_grid**

such that τ_ℓ is obtained in one stroke. We omit here the algorithmic details.

The result of these algorithms is a hierarchy of near-boundary grids where the finest grid, $\tau_{\ell_{\max}}$ represents a proper resolution of the boundary. Furthermore, full grids up to a level ℓ are generated by refining an initial coarse grid τ_0. We emphasize that, for $\ell < \ell_{\max}$, the grids τ_ℓ *cannot* be regarded as approximations of the domain but simply satisfy

$$\begin{array}{ll} \Omega \subset \operatorname{dom} \tau_m, & 0 \le m \le \ell, \\ \operatorname{dom} \tau_\ell \subset \operatorname{dom} \tau_{\ell'} & 0 \le \ell' \le \ell \le \ell_{\max}. \end{array}$$

However, we assume that the grid $\tau_{\ell_{\max}}$ properly resolves the boundary which, in order to obtain higher accuracy, can be refined and, furthermore, can be adapted to the boundary by standard techniques (cf. [4, Sect. 8.2]). A full mesh on a relatively coarse level is depicted in the second row of Fig. 1. Composite finite element spaces will be defined on composite finite element grids. To be more concrete we have to choose a covering of Ω which will be a subset of $\tau_\ell \cup \bigcup_{m=\ell+1}^{\ell_{\max}} \tau_m$. The details are in the following

Definition 3 *Let $\{\tau_m\}_{\ell \le m \le \ell_{\max}}$ be generated as explained above. The subset of these grids lying inside of Ω are denoted by $\overset{\circ}{\tau}_m$ and defined by*

$$\overset{\circ}{\tau}_m := \{K \in \tau_m \mid \sigma(K) = \emptyset\}.$$

Finally, the composite finite element grid on level ℓ is given by

$$\tau_\ell^{CFE} := \bigcup_{m=\ell}^{\ell_{\max}} \bigcup_{K \in \overset{\circ}{\tau}_m} K.$$

A typical composite finite element grid is depicted in the last row of Fig. 1.

Remark 4 *The grid τ_ℓ^{CFE} contains those triangles of $\bigcup_{m=\ell}^{\ell_{\max}} \tau_m$ which are not refined furthermore, i.e., lie (essentially) inside of Ω. The grid τ_ℓ^{CFE} represents a proper resolution of the domain Ω, while, in general, it cannot be regarded as a standard finite element mesh since hanging nodes occur and τ_ℓ^{CFE} is highly non-uniform. We emphasize that the higher resolution near the boundary is needed to describe the support of the coarse level basis functions. The near-boundary meshes are not used, e.g., to add more degrees of freedom to the coarse level space.*

In the next section we will define the composite finite element space on these grids.

3 Composite finite elements

In the previous section we have generated a sequence of near-boundary grids $\{\tau_m\}_{\ell+1 \le m \le \ell_{\max}}$ and full grids $\{\tau_m\}_{0 \le m \le \ell}$ which covers the domain Ω. On these finite element grids τ (either full grids or a near-boundary grid), we can define standard finite element spaces S_τ by

$$S_\tau := \{v \in C^r(\operatorname{dom}\tau) \mid \forall K \in \tau : v\,|_K \text{ is a polynomial of degree } p_K\}.$$

In this paper, we will focus on the efficient realization of composite finite elements. For this reason, we will formulate the method in terms of grid functions and nodal values instead of finite element functions itself. In this light, let $\{\varphi_x^\tau\}_{x \in \Theta_\tau}$ denote the usual Lagrange basis of S_τ with Θ_τ denoting the set of corresponding nodal points, i.e., for all $x, y \in \Theta_\tau$, we have

$$\varphi_x^\tau(y) = \begin{cases} 1 & x = y, \\ 0 & \text{otherwise.} \end{cases}$$

A grid function is a mapping $\beta : \Theta_\tau \to \mathbb{C}$ while the space of grid functions is denoted by $\mathbb{C}^{\Theta_\tau}$. Each grid function $\beta \in \mathbb{C}^{\Theta_\tau}$ is linked to a finite element function by the global finite element interpolation operator via

$$I_\tau^{int}[\beta](x) = \sum_{y \in \Theta_\tau} \beta(y)\, \varphi_y^\tau(x), \quad x \in \operatorname{dom}\tau.$$

The local version of I_τ^{int} is defined on an element K and given by

$$I_K^{int}[\beta](x) = \sum_{y \in \Theta_K} \beta(y)\, \varphi_y^\tau(x), \quad x \in K$$

with $\Theta_K := \Theta_\tau \cap \bar{K}$. In the following, we will replace the index τ_ℓ frequently by ℓ. For example, we set

$$\Theta_\ell := \Theta_{\tau_\ell}, \quad I_\ell^{int} := I_{\tau_\ell}^{int}, \quad \varphi_x^\ell := \varphi_x^{\tau_\ell},$$

and analogously for other quantities. We are now able to define intergrid operators by using the finite element interpolation. Let $\beta_\ell \in \mathbb{C}^{\Theta_\ell}$ denote a grid function. We interpolate β_ℓ recursively on the near-boundary grids by

$$\beta_{m+1}(x) := I_m^{int}[\beta_m](x) \quad \forall x \in \Theta_{m+1}, \; m = \ell, \ell+1, \ldots, \ell_{\max} - 1.$$

The intergrid operator $\beta_m \to \beta_{m+1}$ is denoted by

$$P_m^{m+1} : \mathbb{C}^{\Theta_m} \to \mathbb{C}^{\Theta_{m+1}},$$
$$\beta_{m+1} =: P_m^{m+1} \beta_m.$$

We recall the definition of the interior grids $\overset{\circ}{\tau}_\ell$ (see Definition 3). A grid function $\beta_\ell \in \mathbb{C}^{\Theta_\ell}$ is linked to the corresponding *composite finite element function* by, recursively, computing nodal values on the near-boundary grids by

$$\beta_m := P_{m-1}^m \beta_{m-1}, \quad \ell < m \le \ell_{\max}$$

and then interpolating these values on elements $K \in \tau_\ell^{CFE}$:

$$u_\ell(x) := I_{\overset{\circ}{\tau}_m}^{int}[\beta_m](x), \quad \forall x \in \operatorname{dom} \overset{\circ}{\tau}_m .$$

The corresponding operator $\beta_\ell \to u_\ell$ is denoted by $u_\ell := I_\ell^{CFE}[\beta_\ell]$. We illustrate this definition by characterizing the basis functions of the corresponding composite finite element space. Let $e_x^\ell \in \mathbb{C}^{\Theta_\ell}$ denote the unit grid function on the grid τ_ℓ characterized by

$$e_x^\ell(y) = \begin{cases} 1 \text{ for } x = y, \\ 0 \text{ otherwise,} \end{cases}$$

and the corresponding basis function $\varphi_x^{CFE}(x) := I_\ell^{CFE}[e_x^\ell]$. On an element $K \in \overset{\circ}{\tau}_\ell$ which is not refined furthermore, the basis function φ_x^{CFE} is given by the standard finite element interpolation $\varphi_x^{CFE}\,|_K := I_K^{int}[e_x^\ell]$. Otherwise, there exists children $\sigma(K) \neq \emptyset$ on the finer level

Fig. 3. Two typical basis functions of S_ℓ^{CFE}. The shape of the basis functions is very similar to the standard hat functions where the support is restricted to the domain Ω

$\ell+1$. Let $K' \in \sigma(K)$ with nodal points $\Theta_{K'} := \overline{K'} \cap \Theta_{\ell+1}$. In these nodal points, the prolonged grid function $e_x^{\ell+1}$ is defined by evaluating the standard finite element interpolation $I_\ell^{int}\left[e_x^\ell\right]$. This interpolation process is well-defined since the near-boundary grids satisfy $\Theta_{\ell+1} \subset \operatorname{dom}\tau_\ell$. If $K' \in \mathring{\tau}_{\ell+1}$, then $\varphi_x^{CFE}|_{K'}$ is given by interpolating the generated nodal values on K'. Otherwise, if $\sigma\left(K'\right) \neq \emptyset$, the interpolation process has to be iterated on finer levels. In Fig. 2, a typical basis function is depicted.

The definition of the composite finite element space on level ℓ is given by the span of the basis functions:

$$S_\ell^{CFE} := \operatorname{span}\left\{\varphi_x^{CFE} := I_\ell^{CFE}\left[e_x^\ell\right] \middle| x \in \Theta_\ell\right\}.$$

In the following we will explain how the system matrix corresponding to a Galerkin discretization of a boundary value problem can be assembled via local Galerkin products.

4 Assembling of the linear system

We have in mind that, on a domain $\Omega \subset \mathbb{R}^d$, a differential equation has to be solved by using the Galerkin method. As a model problem we consider the Helmholtz problem in a variational setting. Let $V := H^1(\Omega)$ denote the usual Sobolev space and V' the dual space of V. For given $F \in V'$, we are seeking $u \in V$ such that

$$a(u,v) = F(v), \quad \forall v \in V \tag{2}$$

with the bilinear form $a : V \times V \to \mathbb{R}$ defined by

$$a(u,v) := \int_\Omega \langle \nabla u, \nabla v\rangle + k^2 uv dx \tag{3}$$

and a positive constant k. We assume that the linear form F is given by

$$F(v) := \int_\Omega f v dx$$

with suitable $f \in L^2(\Omega)$. We state that more complicated problems can be treated with composite finite elements as well but, in order to explain the main application, namely, the discretization of complicated domains, we restrict to this simple model equation.

The Galerkin discretization to problem (2) is given by seeking $u_\ell \in S_\ell^{CFE}$ such that

$$a(u_\ell, v_\ell) = F(v_\ell), \quad \forall v_\ell \in S_\ell^{CFE} \tag{4}$$

holds. Introducing the basis representation

$$u_\ell(x) = \sum_{y\in\Theta_\ell} \mathbf{u}_\ell(y)\,\varphi_y^{CFE}(x), \tag{5}$$

problem (4) is equivalent to solving the system of linear equation

$$\mathbf{A}_\ell^{CFE}\mathbf{u}_\ell = \mathbf{f}_\ell$$

with the grid operator $\mathbf{A}_\ell^{CFE}$ defined by

$$\mathbf{A}_\ell^{CFE}(x,y) = a\left(\varphi_y^{CFE}, \varphi_x^{CFE}\right), \quad \forall x,y \in \Theta_\ell$$

and the right hand side $\mathbf{f}_\ell(x) := F\left(\varphi_x^{CFE}\right)$. In the following, we will explain how the grid operator $\mathbf{A}_\ell^{CFE}$ and the right hand side can be generated efficiently by using local Galerkin products. We use the fact that the images of the unit grid functions e_x^ℓ form the columns of the grid operator $\mathbf{A}_\ell^{CFE}$. We first compute an auxiliary fine grid operator $\mathbf{A}_{\ell_{\max}}$ corresponding to the near-boundary grid $\tau_{\ell_{\max}}$. To be more concrete, we define local versions a_τ of the bilinear form a by replacing the integration domain Ω in (3) by $\operatorname{dom}\tau$. Let τ denote a finite element grid and Θ the corresponding set of nodal points. Then, the corresponding grid operator is given by

$$\mathbf{A}_\tau(x,y) := a_\tau\left(\varphi_y^\tau, \varphi_x^\tau\right), x,y \in \Theta.$$

In the first step, we compute $\mathbf{A}_{\tau_{\ell_{\max}}}$. This grid operator is recursively coarsened by employing the *Galerkin product*. For $\ell \le m < \ell_{\max}$, we set

$$\mathbf{A}_m(x,y) := \mathbf{A}_{\mathring{\tau}_m}(x,y) + \left\langle P_m^{m+1} e_y^m, \mathbf{A}_{m+1} P_m^{m+1} e_x^m\right\rangle_{m+1}, \quad x,y \in \Theta_m$$

where, for $\eta, \eta' \in \mathbb{C}^{\Theta_m}$, the scalar product $\langle\cdot,\cdot\rangle_m$ is defined by

$$\langle \eta, \eta'\rangle_m := \sum_{x\in\Theta_m} \eta(x)\overline{\eta'}(x).$$

More formally, we define the adjoint operator $R_{m+1}^m : \mathbb{C}^{\Theta_{m+1}} \to \mathbb{C}^{\Theta_m}$ of P_m^{m+1} with respect to the $\langle\cdot,\cdot\rangle_m$ scalar product by

$$\left\langle P_m^{m+1}\eta, \omega\right\rangle_{m+1} = \langle \eta, R_{m+1}^m \omega\rangle_m, \quad \forall \eta \in \mathbb{C}^{\Theta_m}, \omega \in \mathbb{C}^{\Theta_{m+1}}.$$

Using this notation the system matrix $\mathbf{A}_\ell^{CFE}$ can be computed recursively as follows

$$\mathbf{A}_{\ell_{\max}} := \mathbf{A}_{\tau_{\ell_{\max}}}, \tag{6}$$

$$\mathbf{A}_m := \mathbf{A}_{\mathring{\tau}_m} + R_{m+1}^m \mathbf{A}_{m+1} P_m^{m+1}, \quad m = \ell_{\max}-1, \ell_{\max}-2, \ldots, \ell. \tag{7}$$

It is easy to see that $\mathbf{A}_\ell^{CFE} := \mathbf{A}_\ell$ holds. The algorithmic formulation of the recursion is given below. The procedure **coarsen** is called by

for $m = \ell_{\max}-1$ **downto** $\ell+1$ **do coarsen**$(m, \tau_m, \mathbf{A}_m)$;
coarsen$\left(\ell, \tau_\ell, \mathbf{A}_\ell^{CFE}\right)$

and defined by

procedure coarsen$(m, \tau, \mathbf{A})$;
begin
  for all $x \in \Theta_m$ **do**
    compute $\mathbf{A}(\cdot, x) := R^m_{m+1}\mathbf{A}_{m+1}P^{m+1}_m e^m_x$;
  for all $K \in \overset{\circ}{\tau}_m$ **do**
    for all $x, y \in \Theta_K := \Theta_m \cap \overline{K}$ **do**
      $\mathbf{A}(x, y) := \mathbf{A}(x, y) + a_K\left(\varphi^m_y, \varphi^m_x\right)$;
end;

The generation of the right-hand sides on level ℓ can be done analogously. The result of these procedure is a system of linear equations

$$\mathbf{A}^{CFE}_\ell \mathbf{u}_\ell = \mathbf{f}_\ell \tag{8}$$

where the grid function $\mathbf{u}_\ell$ is linked to the continuous solution of (4) by (5). The algorithm is efficient in the sense that the computational cost of assembling (8) is only moderately higher as the cost for generating the standard finite element matrix on level ℓ. The complexity of the presented algorithm is discussed in detail in the following section.

5 Complexity analysis for composite finite elements

The work for generating the system of linear equations (8) strongly depends on the complexity of the boundary Γ. Let N_m denote the number of elements of τ_m:

$$N_m := \#\{\tau_m\}, \quad m = \ell, \ell+1, \ldots, \ell_{\max}.$$

We assume that the number of nodal points Θ_m is of the same order as N_m, i.e.,

$$\#\Theta_m \le CN_m$$

with a constant C depending only on the degree of approximation p but not on the level m. The complexity of the algorithm will depend on the sum $\sum_\ell := \sum_{m=\ell+1}^{\ell_{\max}} N_m$ and the number of elements of the full grid $N_\ell := \#\tau_\ell$. We assume that the information whether an element K intersects the boundary Γ is available and also (approximations to) the pairs

$$(A, B) := \arg\min\{\|x - y\| : (X, Y) \in \mathbf{V}(K) \times \Gamma_K\}$$

(cf. procedure **adapt)** either are already computed or require $O(1)$ operations per element K. In the appendix we will describe algorithms which meet these requirements. Then, it follows directly from the definition of the procedures **refine**, **adapt**, **reduce_mesh**, and **generate_full_grid** that the generation of the sequence of near-boundary meshes $\{\tau_m\}_{0\le\ell\le\ell_{\max}}$, the full grid τ_ℓ and the composite finite element grid τ^{CFE}_ℓ needs $O\left(N_\ell + \sum^\Gamma_\ell\right)$ arithmetic operations. The storage requirements are of the same order.

In order to control the work for assembling $\mathbf{A}^{CFE}_\ell$ it is necessary to make appropriate assumptions on the localness of these operators. To formulate these conditions we will need the following notation. The support of a grid function $\beta \in \mathbb{C}^{\Theta_\ell}$ is defined by

$$\operatorname{supp}\beta := \{x \in \Theta_\ell \mid \beta(x) \ne 0\}.$$

For subsets $\omega \subset \Omega$, we define recursively layers of finite elements about ω.

$$L^1_m(\omega) := \operatorname{dom}\left\{K \in \tau_m \mid \overline{K} \cap \overline{\omega} \ne \emptyset\right\},$$
$$L^{i+1}_m(\omega) := \operatorname{dom}\left\{K \in \tau_m \mid \overline{K} \cap \overline{L^i_m(\omega)} \ne \emptyset\right\}.$$

The corresponding spaces of grid functions with local support are defined by

$$G^i_m(\omega) := \left\{\beta \in \mathbb{C}^{\Theta_m} \mid \operatorname{supp}\beta \subset \overline{L^i_m(\omega)}\right\}.$$

Now, we can express the localness of the grid operators. We have to impose the following

Assumption 5 *We assume that there are constants c_P, c_R, and c_A such that*

$$\forall\omega \subset \Omega, \forall i \in \mathbb{N}_0:$$
$$P^{m+1}_m G^i_m(\omega) \subset G^{1+c_p}_{m+1}\left(L^i_m(\omega)\right), \ \forall \ell \le m \le \ell_{\max} - 1,$$
$$R^m_{m+1} G^i_{m+1}(\omega) \subset G^{1+\lfloor i/2\rfloor + c_R}_m(\omega), \ \forall \ell \le m \le \ell_{\max} - 1,$$
$$A_{\overset{\circ}{\tau}_m} G^i_m(\omega) \subset G^{i+c_A}_m(\omega). \qquad \forall \ell \le m \le \ell_{\max}.$$

Under these assumption the operators $\mathbf{A}^{CFE}_\ell$ is local, independent of the refinement levels ℓ and $\ell_{\max}$. The details are in the following

Theorem 6 *Assume that the Assumption 5 is fulfilled with constants independent of the refinement levels $\ell, \ell_{\max}$. Then, the number of non-zero entries per line and column of $\mathbf{A}^{CFE}_\ell$ is bounded by a constant independent of ℓ and $\ell_{\max}$.*

Proof. The assertion is proved inductively. We employ the recursion (6, 8). For $m = \ell_{\max}$, we know that

$$\mathbf{A}_m = \mathbf{A}_{\overset{\circ}{\tau}_m}$$

holds and hence, for all $\omega \subset \Omega$, we have

$$\mathbf{A}_{\ell_{\max}} G^i_m(\omega) \subset G^{i+c_A}_m(\omega).$$

Now, assume that, for fixed m, $\ell \le m < \ell_{\max}$, there exists a constant ρ_{m+1} with the property that, for all $\omega \subset \Omega$, $i \in \mathbb{N}_0$

$$\mathbf{A}_{m+1} G^i_{m+1}(\omega) \subset G^{i+\rho_{m+1}}_{m+1}(\omega). \tag{9}$$

In the following, we will investigate the sparsity of the operator $\mathbf{A_m}$ assuming condition (9). For $\omega \subset \Omega$, let $\beta \in G^i_m(\omega)$ be an arbitrary grid function. From Assumption 5, we know that $\mathbf{A}_{\overset{\circ}{\tau}_m}\beta \in G^{i+c_A}_m(\omega)$. Furthermore,

$$P^{m+1}_m \beta \in G^{1+c_p}_{m+1}\left(L^i_m(\omega)\right)$$

and by (9) it follows that

$$\mathbf{A}_{m+1}P^{m+1}_m\beta \in G^{1+c_p+\rho_{m+1}}_{m+1}\left(L^i_m(\omega)\right)$$

and, finally again by using Assumption 5, we obtain

$$R^m_{m+1}\mathbf{A}_{m+1}P^{m+1}_m\beta \in G^{1+c_R+\left\lfloor(1+c_p+\rho_{m+1})/2\right\rfloor}_m\left(L^i_m(\omega)\right)$$

which is equivalent to

$$R^m_{m+1}\mathbf{A}_{m+1}P^{m+1}_m\beta \in G^{1+c_R+\left\lfloor(1+c_p+\rho_{m+1})/2\right\rfloor+i}_m(\omega).$$

It follows that (9) implies that

$$\mathbf{A}_m G^i_m(\omega) \subset G^{i+\rho_m}_m(\omega)$$

with $\rho_m = \max\left(c_A, 1 + c_R + \left\lfloor \left(1 + c_p + \rho_{m+1}\right)/2 \right\rfloor\right)$. This leads to the estimate

$$\rho_m \leq \max\left(c_A, 1 + c_R + \frac{1 + c_p}{2} + \frac{\rho_{m+1}}{2}\right).$$

Clearly, this recursion implies that ρ_m is bounded independent of $\ell_{\max}$ and ℓ. A more detailed investigation shows that $\rho_m \leq 3 + 2c_R + c_p + c_A$. This directly implies that the number of non-zero entries per row and column is bounded by a constant depending only on the degree of approximation but is independent on the refinement parameters ℓ and $\ell_{\max}$. □

From this theorem it follows directly that the work needed for assembling the system matrix for composite finite elements is bounded by $O\left(N_\ell + \sum_\ell^\Gamma\right)$. Obviously, the cost of generating the system matrix for standard finite elements on a grid with N_ℓ elements is of order N_ℓ implying that the additional amount of work is of order $\sum_\ell^\Gamma$. In the following, we will give estimates of $\sum_\ell^\Gamma$ for two typical situations. We assume that the grids are quasi-uniform and and express the estimates in terms of the step size $h_m := \max\{\operatorname{diam} K : K \in \tau_m\}$. Furthermore, we assume that the step sizes satisfy $h_{\ell+1} \leq c_{ref} h_\ell$ with $c_{ref} < 1$.

First, let us assume that the full fine grid has to be generated due to the required accuracy, i.e., $\ell = \ell_{\max}$ holds. Then, the amount of work for generating $\mathbf{A}_{\ell_{\max}}^{CFE}$ is of order $h_{\ell_{\max}}^{-d}$. In order to apply a multi-grid solver to the system of linear equations, one has to generate the whole sequence of matrices $\left\{\mathbf{A}_m^{CFE}\right\}_{0 \leq m \leq \ell_{\max}}$ by means of the Galerkin product. From the complexity analysis, it follows that the work needed for assembling the hierarchy of matrices $\left\{\mathbf{A}_m^{CFE}\right\}_{0 \leq m \leq \ell_{\max}}$ is of order

$$\sum_{m=0}^{\ell_{\max}} h_m^{-d} \leq \sum_{m=0}^{\ell_{\max}} \left(h_{\ell_{\max}} \left(c_{ref}\right)^{-m}\right)^{-d} = h_{\ell_{\max}}^{-d} \frac{1}{1 - c_{ref}^d}.$$

This implies that the work needed for assembling the full hierarchy of matrices is of the same order as the generation of only the fine grid matrix.

Now, we will consider the following situation. Assume that the aim is to generate directly the matrix $\mathbf{A}_\ell^{CFE}$ while the finer matrices $\mathbf{A}_m^{CFE}$ are not required. Then, the amount of work is determined by the number of elements of $\tau_m, m > \ell$ which intersects the boundary. Let $c_\Omega := \int_\Omega dx$ denote the area (volume) of the domain and $c_\Gamma := \int_\Gamma d\Gamma_x$ the length (area) of the surface Γ. From the quasi-uniformity of the meshes, it follows that $N_\ell = O\left(c_\Omega h_m^{-d}\right)$ and, for $m > \ell$, we have $N_m = O\left(c_\Gamma h_m^{1-d}\right)$ with the space dimension $d = 2, 3$. Therefore, the quantity $\sum_\ell^\Gamma$ can be estimated by

$$\begin{aligned}\sum_\ell^\Gamma &= \sum_{m=\ell+1}^{\ell_{\max}} N_m \leq C \sum_{m=\ell+1}^{\ell_{\max}} c_\Gamma h_m^{1-d} \\ &\leq C \sum_{m=\ell+1}^{\ell_{\max}} c_\Gamma \left(h_{\ell_{\max}} \left(c_{ref}\right)^{-m}\right)^{1-d} \leq C \frac{c_\Gamma}{1 - c_{ref}^{d-1}} h_{\ell_{\max}}^{1-d}.\end{aligned}$$

This estimate can be interpreted, e.g., in two dimensions as follows. The additional work for generating $\mathbf{A}_\ell^{CFE}$ compared to standard finite elements is $h_{\ell_{\max}}^{-1}$ where $h_{\ell_{\max}}^{-2}$ is the amount of work which would be necessary to generate the whole fine grid matrix. Note that the estimate above is too pessimistic in cases where the micro-structures consist of a fixed number of very small holes intersecting only $O(1)$ coarse grid elements. The estimate is too optimistic, e.g., in cases of porous media where the number of micro-structures is of order h_m^{-d}.

6 Numerical results

In this section, we present results of numerical experiments performed with composite finite elements. We have considered CFE-spaces based on linear interpolation on triangulations. We have investigated

1. the approximation quality of this CFE-space for H^2-functions,
2. the performance of CFE coarse-level discretizations for multi-grid methods,
3. the locally homogenized partial differential operators.

6.1 Approximation property

For composite finite elements based on linear interpolation, it was proved in [9, Theorems 8, 10] that, for quasi-uniform, shape regular triangulation and additional weak but technical assumptions, the space S_ℓ^{CFE} has the approximation property for H^2-functions. For all $u \in H^2(\Omega)$, there exists $u_\ell \in S_\ell^{CFE}$ such that

$$\|u - u_\ell\|_0 \leq C h_\ell^{2-m} \|u\|_2, \quad m = 0, 1, \tag{10}$$

where the constant C is independent of u, ℓ and $\ell_{\max}$. Furthermore, it was proved in [15] that the constant C depends only on the *shape* of the micro-structures but not on the size of them. In order to check this estimate, we have chosen u_ℓ as the Galerkin approximation to u by means of problem (2, 3) with appropriate chosen right-hand side and setting $k^2 = 1$. The domain Ω_1 was the unit sphere where 576 holes have been removed in the interior. The domain together with the corresponding CFE-grid is depicted in Fig. 4.

The right-hand side of (2) was chosen such that $u(x) = x^2 + y^2$ is the exact solution. The following table reports the observed convergence rates.

Level	dof	$\|u-u_\ell\|_1$	$\frac{\|u-u_{\ell-1}\|_1}{\|u-u_\ell\|_1}$	$\|u-u_\ell\|_0$	$\frac{\|u-u_{\ell-1}\|_0}{\|u-u_\ell\|_0}$
1	4	4.88e-1		5.35e-2	
2	9	2.91e-1	1.67	2.34e-2	2.28
3	24	1.57e-1	1.85	6.93e-3	3.38
4	74	8.28e-2	1.90	1.73e-3	4.00
5	245	4.27e-2	1.94	4.24e-4	4.08
6	895	2.22e-2	1.93	1.15e-4	3.68
7	3323	1.17e-2	1.89	3.29e-5	3.50
8	12586	5.98e-3	1.96	8.09e-6	4.07

We see that, starting from the very coarse refinement levels, the observed convergence rates are very close to the expected asymptotic rates. We point out that, for the considered domains, the size of the holes are of the order of the

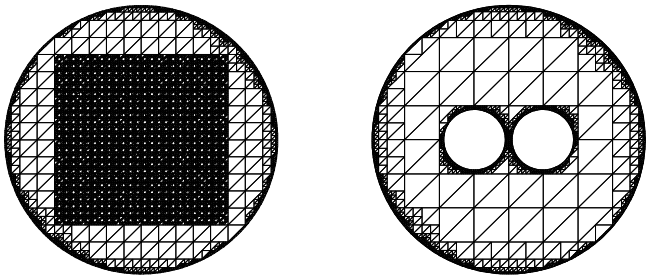

Fig. 4. On the left-hand side, the domain Ω_1 containing 576 holes is depicted while on the right-hand side the anisotropic situation of the domain Ω_3 is shown.

step size of the fine grid $h_{\ell_{\max}}$ confirming the assertion that the constant of the estimate (10) is independent of the *size* of the micro-scales. In the next subsection, we will study the performance of CFE-discretization in combination with the multi-grid method.

6.2 Composite finite elements for multi-grid methods

We have generated the coarse-level discretizations by composite finite elements for the multi-grid method to check how the number of iteration depends on the complexity of the geometry of Ω. The approximation property for multi-grid methods is strongly related to the approximation property of the finite element space. The approximation property for multi-grid methods follows from the approximation quality of the finite element spaces and assumptions on the differential equations on the continuous but **not** on the discrete level (see [8, Sect. 6.3.1]). As a smoother we have employed the symmetric Gauß-Seidel smoother with one pre- and one post- smoothing steps. Due to the scaling, the diagonal entries of the stiffness matrix are very small if the support of a basis function is overlapping the domain only on a very short region. The situation is very similar to the Shortley-Weller finite difference scheme considered in [6], [7]. The detailed proof of the smoothing property of composite finite elements will be presented in a forthcoming paper. The following table reports the performance of the multi-grid method (V-cycle) for the same problem as described for the previous section. The stopping criterion for all the following calculations was given by the condition that the l^2-norm of the residual is smaller than 1.0e$-$8. Alternatively, we have considered the same equation on the parameter-dependent domain Ω_2^ε and the domain Ω_3 defined as follows. Ω_2^ε is the disc of radius 0.4 centred at $M_0 := (0.5, 0.5)$ with a circular hole with radius ε centred at M_0, too. The domain Ω_3 is given by the disc of radius 0.4 about M_0 containing two circular holes of radius 0.09 centred at $(0.4, 0.5)$ and $(0.6, 0.5)$. The domains Ω_2^ε and Ω_3 are depicted in Figs. 1 and 4.

Level	# of Iterations for Ω_1	# of Iterations for Ω_2^ε	# of Iterations for Ω_3
1	direct solver	direct solver	direct solver
2	5	5	5
3	7	9	8
4	8	11	10
5	8	11	13
6	8	10	12
7	8	9	11
8	9	8	10

We observe very high convergence rates being independent of the refinement level and the number and size of the micro structures.

We come now to the investigation of the homogenized operators.

7 Composite finite elements for discrete homogenization

We have computed homogenized operators on different refinement levels in order to get insights how these operators depends on the local geometry. To explain the details, let $\mathbf{A}_\ell^{CFE}$ denote the system matrix on the grid τ_ℓ and $x \in \Theta_\ell$ a nodal point. Then, we define the application of $\mathbf{A}_\ell^{CFE}$ to a smooth function $u \in C^\infty\left(\mathbb{R}^d\right)$ by

$$\mathbf{A}_\ell^{CFE}[u](x) := \sum_{y\in\Theta} \mathbf{A}_\ell^{CFE}(x,y)\, u(y).$$

Using Taylor expansion of u about x we obtain

$$\begin{aligned}\mathbf{A}_\ell^{CFE}[u](x) &\approx \sum_{y\in\Theta} \mathbf{A}_\ell^{CFE}(x,y) \sum_{|\nu|\le 2} \frac{1}{\nu!} D^\nu u(x)(y-x)^\nu \\ &= \sum_{|\nu|\le 2} \frac{1}{\nu!} \left\{ \sum_{y\in\Theta} \mathbf{A}_\ell^{CFE}(x,y)(y-x)^\nu \right\} D^\nu u(x).\end{aligned}$$

Using the coefficients $c_\nu(x) := \frac{1}{\nu!}\left\{\sum_{y\in\Theta} \mathbf{A}_\ell^{CFE}(x,y)(y-x)^\nu\right\}$ we define the homogenized differential operator at the nodal point x corresponding to the scale ℓ by

$$\mathbf{A}_\ell^{\text{hom}}(x) := \sum_{|\nu|\le 2} c_\nu(x)\, D^\nu.$$

We have studied the dependence of the coefficients $c_\nu(x)$ on the geometry and the refinement scales. First, we have considered the Laplace operator discretized on the domain Ω_2^ε and chosen $x = M_0 = (0.5, 0.5)^T$. Therefore, we expect a rotational symmetric behaviour expressed by

$$\mathbf{A}_\ell^{\text{hom}}(M_0) = c(\varepsilon)\,\Delta.$$

We have sampled the function $c(\varepsilon)$ for different values of ε. It was interesting that, for the considered range of $\varepsilon \in [0.1, 3]$, the function $c(\varepsilon)$ can perfectly be fitted by a quadratic polynomial as shown in Fig. 5.

Next, we have studied the anisotropic situation of domain Ω_3 on different refinement levels. Again, we have chosen x as the midpoint of the disc. It is clear that, for much larger step sizes h_ℓ compared to the holes, the homogenized

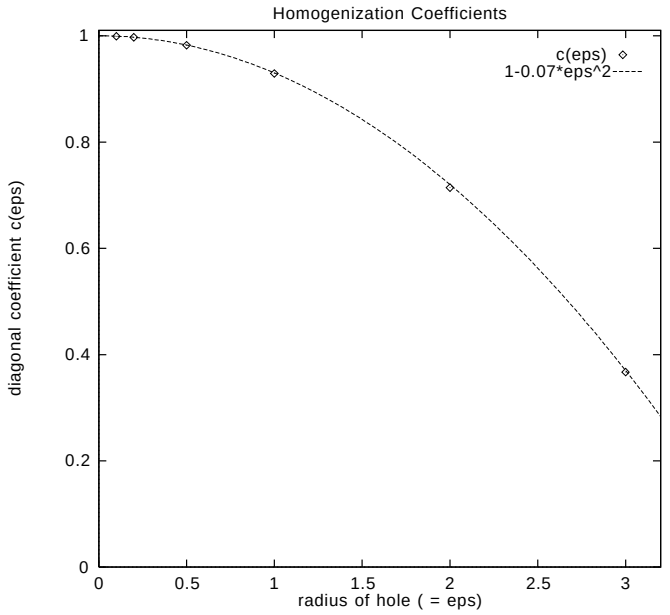

Fig. 5.

operator looks relatively isotropic. With decreasing step size h_ℓ the situation becomes anisotropic while, for very small step sizes the local operator does not "see" the holes and the situation is more isotropic again. It turns out that the computed homogenized operator can be written in the form

$$\mathbf{A}_\ell^{\text{hom}}(M_0) = c_1(h_\ell)\,\partial_{xx} + c_2(h_\ell)\,\partial_{yy}.$$

In Fig. 7 we have plotted these coefficients $c_{1,2}(h_\ell)$ which clearly confirm the considerations above.

8 Conclusions

In this paper, we have presented Composite Finite Elements for the discretization of PDEs on complicated domains. We have presented numerical experiments which show the efficiency of the approach. The asymptotic approximation property also holds for the very coarse discretizations. We emphasize that the method is flexible in the sense that the realization in three dimensions does not differ in principle from the two-dimensional version. Also generalizations to higher order approximations and iso-parametric modelling of the boundary is straightforward by using the canonical finite element interpolation operators for the definition of the intergrid operators.

In this paper we have considered a simple model problem with Neumann boundary conditions. If essential boundary conditions are imposed the prolongation operators P_m^{m+1} have to be slightly modified such that the CFE-spaces satisfy the boundary condition, too. It turns out that the modified prolongation for this situations is very easy to realize and cheap to evaluate. This modification will be presented along with an analysis of the approximation property in a forthcoming paper. A similar situation arises if interfaces are present. Also in this case the intergrid operators have to be modified.

A Generation of hierarchical boundary information

In this section, we will describe how the information required in the algorithms **refine**, **adapt** and **reduce_mesh** can be generated efficiently. As mentioned earlier, the computation of this information strongly depends on how the boundary is prescribed. If the boundary Γ of the domain is available by an explicit parametrization it is clear that the information whether, e.g. an element $K \in \tau$ intersects the boundary or not can be assembled in $O(1)$ arithmetic operations per element. This was realized and described in [11]. In our paper, we will present search strategies which requires less structured boundary information (at the cost of increased computing time). Here and in the following, we require that, for any point $x \in \mathbb{R}^d$, the information whether $x \in \Omega$ holds can be computed in $O(1)$ arithmetic operations. Such a situation typically arise if the domain Ω is defined via the characteristic function χ_Ω. The information whether a finite element $K \in \tau$ has non-zero cut with the boundary or not is needed in procedure **refine**. We define the function γ by

$$\gamma(K) = \begin{cases} boundary \text{ if } K \cap \Gamma \neq \emptyset, \\ 0 \qquad\quad \text{otherwise} \end{cases}$$

(see Fig. 2). Since the boundary Γ is not available as a parametrization, we have to approximate the function γ by a function $\tilde{\gamma}$ which is defined recursively as follows. We recall the definition of the function $\nu(K)$ (see (1)).

$$\tilde{\gamma}(K) := \begin{cases} boundary \text{ if } K \in \tau_{\ell_{\max}} \text{ and } \nu(K) = boundary \\ boundary \text{ if } K \in \tau_{m-1} \wedge \exists K' \in \sigma(K) \\ \qquad\qquad \text{with } \tilde{\gamma}(K') = boundary \\ 0 \qquad\quad \text{otherwise} \end{cases}$$

For the computation of $\tilde{\gamma}$, we propose the following algorithm. Formally, we put $parent(K) = nil$ for all $K \in \tilde{\tau}_0$ and initialize the function $\tilde{\gamma}$ as the zero-function.

for all $K \in \tilde{\tau}_0$ with $\operatorname{diam} K > tol$
  do $check\_boundary(K)$;

procedure check_boundary(K);
begin
  if $\operatorname{diam} K > tol$ **do begin**
    refine K regularly and generate the set of children $\sigma(K)$;
    for all $K' \in \sigma(K)$ **do begin**
      $parent(K') := K$;
      check_boundary(K)
    end
  end else begin
    if $\nu(K) = boundary$ **then begin**
      while $K \neq$ **nil do begin** $\tilde{\gamma}(K) := boundary$;
        $K := parent(K)$
end end end end;

We emphasize that the procedure above realizes only the principal idea of the search algorithm. Some more details are concerned in the following

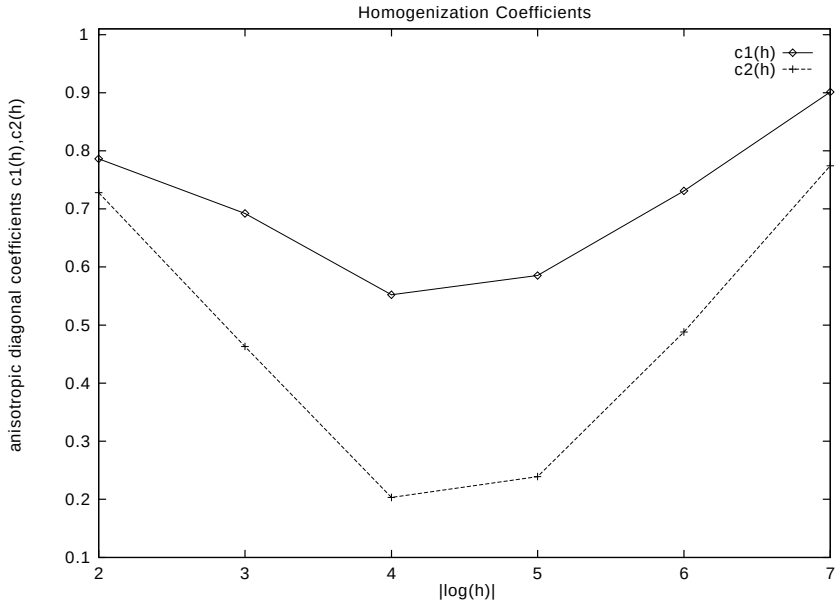

Fig. 6.

Remark 7 *Obviously, the procedure* **check_boundary** *can be parallelized straightforwardly since all elements K are treated independently.*

From the definition of the near-boundary grids it is clear that the elements K satisfying $\tilde{\gamma}(K) = boundary$ directly can be incorporated in the near-boundary grid of the corresponding level.

The elements K not having the attribute $\tilde{\gamma}(K) = boundary$ are only needed temporarily and can be rejected afterwards.

In the algorithm **adapt,** grid points lying very close to the boundary are projected onto the boundary. Let $e := \overline{XY}$ be an edge having the attribute $\nu(e) = boundary$. Then, in many cases, a good candidate for replacing either X or Y is given by the minimizer

$$(x_1, x_2) := \operatorname{argmin} \{\|A - B\|, (A, B) \in \{X, Y\} \times \{e \cap \Gamma\}\}.$$

This can be computed easily by a standard bisection algorithm. If the required precision is $0 < \delta \leq \|x - y\|$, then, it is obvious that the complexity of the bisection algorithm is of order $\log \frac{\delta}{|X-Y|}$. Other choices as, e.g. the othogonal projection of either X or Y onto Γ, can be computed in a similar fashion.

Finally, we come to the computation of the intersection of elements $K \in \tau_{\ell_{\max}}$ with the domain Ω required in the procedure **reduce_mesh**. In view of this procedure, we require that $\mathbf{V}(K) \cap \Omega = \mathbf{V}(K)$ holds. The quantity $\mu(K \cap \Omega)$ is approximated by the following algorithm. We subdivide K into sub-elements K' of size $\operatorname{diam} K' < \delta$ with a prescribed tolerance δ. We put

$$\mathscr{A}(K') = \begin{cases} 0 & \text{if } \mathbf{V}(K') \notin \bar{\Omega}, \\ \mu(K') & \text{otherwise.} \end{cases}$$

As an approximation to $\mu(K \cap \Omega)$ we use $\tilde{\mu} := \sum_{K'} \mathscr{A}(K')$. Clearly, the accuracy of $\tilde{\mu}$ depends on the regularity of the boundary or, alternatively, on the step size $h_{\ell_{\max}}$ of the finest near-boundary mesh. The complexity of this algorithm is of order $\left(h_{\ell_{\max}}/\delta\right)^d$ where d denotes the space dimension. In cases where the boundary is given explicitly by a parametrization the quantity $\mu(K \cap \Omega)$ can be approximated with significantly reduced amount of work.

Acknowledgements. Thanks are due to L. Grasedyck for the implementation of the algorithms.

References

1. R. Bank, J. Xu: An Algorithm for Coarsening Unstructured Meshes. Numer. Math. **73**(1):1–36 (1996)
2. R. E. Bank, J. Xu: A Hierarchical Basis Multi-Grid Method for Unstructured Grids. In W. Hackbusch, G. Wittum, editors, Fast Solvers for Flow Problems, Proceedings of the Tenth GAMM-Seminar, Kiel. Wiesbaden: Vieweg, 1995
3. D. Braess: Towards Algebraic Multigrid for Elliptic Problems of Second Order. Computing **55**: 379–393 (1995)
4. S. Brenner, L. Scott: The Mathematical Theory of Finite Element Methods. New York: Springer, 1994
5. T. F. Chan, B. F. Smith: Domain Decomposition and Multigrid Algorithms for Elliptic Problems on Unstructured Meshes. ETNA **2**:171–182 (1994)
6. W. Hackbusch: On the Multi-Grid Method Applied to Difference Equations. Computing **20**:291–306 (1978)
7. W. Hackbusch: Convergence of Multi-Grid Iterations Applied to Difference Equations. Math. Comp. **34**(150):425–440 (1980)
8. W. Hackbusch: Multi-Grid Method and Applications. New York: Springer, 1985
9. W. Hackbusch, S. Sauter: Composite Finite Elements for the Approximation of PDEs on Domains with Complicated Micro-Structures. Technical Report 95-05, Inst. f. Prakt. Math., Universität Kiel, Germany, 1995. Numer Math **75**:447–472 (1997)
10. W. Hackbusch, S. Sauter: Adaptive Composite Finite Elements for the Solution of PDEs on domains containing non-uniformly distributed micro-scales. Matematicheskoe modelirovanie **8**(N9):31–43 (1996)
11. F. Kickinger: Automatic Mesh Generation for 3D Objects. Technical Report 96-1, University of Linz, Austria, Dept. of Mathematics, February 1996
12. R. Kornhuber, H. Yserentant: Multilevel Methods for Elliptic Problems on Domains not Resolved by the Coarse Grid. Contemporay Mathematics **180**:49–60 (1994)

13. V. Mikulinsky: Multigrid Treatment of Boundary and Free-Boundary Conditions. PhD thesis, The Weizmann Institute of Science, Rehovot 76100, Israel, 1992
14. J. Ruge, K. Stüben: Algebraic Multigrid. In S. McCormick (ed.) Multigrid Methods, chapter 4, pages 73–130. SIAM, Philadelphia, 1987
15. S.A. Sauter: A Remark on Extension Theorems for Domains Containing Small Geometric Details. Technical Report 96-03, Lehrstuhl Praktische Mathematik, University Kiel, Germany, 1996