A Smoothness Preserving Fictitious Domain Method for Elliptic Boundary Value Problems*

Mario S. Mommer
IGPM, RWTH-Aachen,
mommer@igpm.rwth-aachen.de

February 7, 2005

Abstract

We introduce a new fictitious domain method for the solution of second order elliptic boundary value problems with Dirichlet or Neumann boundary conditions on domains with C^2 boundary. The main advantage of this method is that it extends the solutions smoothly, which leads to better performance by achieving higher accuracy with less degrees of freedom. The method is based on a least-squares interpretation of the fundamental requirements that the solution produced by a fictitious domain method should satisfy. Careful choice of discretization techniques, together with a special solution strategy lead then to smooth solutions of the resulting underdetermined problem. Numerical experiments are provided which illustrate the performance and flexibility of the approach.

Keywords: boundary value problems, fictitious domain methods, convergence rates, wavelets.
AMS subject classification: 65N99

Introduction

Fictitious domain methods are a family of methods for the solution of boundary value problems. Their main distinguishing feature is that they reformulate the original problem by embedding the original domain into a simple, larger one (the fictitious domain) to obtain a new problem, a step that typically requires some extension of the data. The solution of this new problem then is an extension of the solution to the original problem. While this may sound wasteful, this approach has the advantage of needing little or no mesh generation, and of enabling the use of techniques which are usually reserved for very simple domains. Many methods of this type exist; see for instance [1, 6, 8].

*This research was supported in part by the EEC TMR project *Wavelets and Multiscale Methods in Numerical Analysis and Simulation*, and EEC Human Potential Programme under contract HPRN-CT-2002-00286

This type of method is used often for the solution of boundary value problems with very complex geometries, or as a subroutine in problems where the geometry changes often, as in shape optimization problems (see for instance [10]). Outside of settings where such requirements allow them to shine, fictitious domain methods have not enjoyed too much popularity.

The motivation for the construction of the fictitious domain method described here was the realization that the extended solutions produced by existing fictitious domain methods are, under normal operation, fairly immune to good approximation with standard higher order approximation tools. In [7], for example, it is proved for the fictitious domain - Lagrange multiplier (FDLM) approach that, if the extension of the right-hand side is in L_2, but was not chosen in *exactly the right way* (that is, in such a way that the resulting Lagrange multiplier is zero), then the extended solution will be in the Sobolev space $H^{3/2-\epsilon}$ only for $\epsilon > 0$, even when the solution of the original problem has higher regularity. Thus we can say that the FDLM method is not smoothness preserving. While from a theoretical point of view the choice of a better extension of the right hand side is a trivial matter, there does not seem to exist a practical way to do so, i.e. without computing the solution of the original problem *first*.

As a consequence one has that, usually, approximating the extended solution obtained through the FDLM method by smooth higher order piecewise polynomials on uniform meshes of mesh size h yields a limited convergence rate of not quite $\mathcal{O}(h^{3/2})$. Further analysis, performed in [14], showed that approximations of the extended solution with standard adaptive schemes are also subject to impoverished convergence rates in this situation. It is clear that this drawback alone puts the FDLM method, or any method with similar problems, at a disadvantage whenever mesh generation is not a major problem.

We are going to introduce here a new method for solving elliptic boundary value problems on bounded domains designed to preserve the smoothness (as manifested in the convergence rates of approximation from uniform meshes) of the solution on the original domain. A fictitious domain method with this property has the potential to compete successfully with more traditional approaches. It can obtain solutions of comparable quality but without the expensive process of creating a mesh for the domain.

To achieve this, we had to leave the trodden paths more than once. At the core, our fictitious domain formulation is based on a least-squares interpretation of the fundamental requirements that the extended solution should satisfy. Since there are many possible extensions of the solution to the original problem, we are led to a formulation which, while otherwise well posed, does accept many different solutions. Instead of adding constraints to the formulation in order to enforce smoothness, we have assigned this responsibility to a carefully engineered iterative solution process. To obtain good discrete models of our infinite-dimensional, rank deficient least-squares problem, we used a Petrov-Galerkin approach and standard B-spline wavelet bases. These discretization techniques are, in essence, those described in [4], the main difference lying in the singular nature of the operator.

While the resulting method has yet to be fully understood from a theoretical

point of view, and has its own set of limitations (in its present form it expects that the solution of the original problem lives in H^2, and that the domain has C^2 boundary), it already performs quite well in numerical experiments.

The structure of this paper is as follows. In section one, we will specify our problem scope and central assumptions, and also derive the formulation which lies at the core of the method. We will do this initially with a simple Dirichlet problem as a model, and show then how to extend the approach to cover Neumann boundary conditions. We devote section two to the construction of an actual numerical method, by settling on a discretization scheme and introducing the solution process. This includes a brief introduction to B-spline wavelet bases. In section three we present some numerical experiments to illustrate the performance of the method, before finishing with some concluding remarks.

1 The basic formulation

1.1 Problem scope and assumptions

We will concentrate on problems of the form

$$\begin{aligned} (-\triangle + \sigma I)u &= f \qquad \text{on } \Omega, \\ \mathrm{Tr}_\Gamma\, u &= g, \end{aligned} \tag{1}$$

where $\Omega \subset \mathbb{R}^2$ is a bounded domain with C^2 boundary, $\sigma \geq 0$, and $\mathrm{Tr}_\Gamma\, u$ denotes the trace on $\Gamma := \partial\Omega$ of u. We will also assume that $f \in H^0(\Omega) = L_2(\Omega)$ and $g \in H^{3/2}(\Gamma)$. Under these conditions, it is known (see [9], chapter two) that there exists a unique solution $u \in H^2(\Omega)$ of problem (1).

We will also assume that $\Omega \subset (0+\epsilon, 1-\epsilon)^2$ for some $\epsilon > 0$.

For notational convenience, we will write $A_\Omega : H^2(\Omega) \to H^0(\Omega)$ for $A = -\triangle + \sigma I$, where differentiation is always meant in the sense of distributions, and let $B : H^2(\Omega) \to H^{3/2}(\Gamma)$ be given by $Bu = \mathrm{Tr}_\Gamma\, u$. Recall that these are bounded operators with closed range, and that $P_\Omega : H^2(\Omega) \to H^0(\Omega) \times H^{3/2}(\Gamma)$, given by

$$P_\Omega := \begin{pmatrix} A_\Omega \\ B \end{pmatrix}$$

is an isomorphism. With this operator, we can write problem (1) more succinctly as follows. Find $u \in H^2(\Omega)$ such that

$$P_\Omega u = b_\Omega, \tag{2}$$

with $b_\Omega = (f, g)^T \in \mathcal{H}^r_\Omega := H^0(\Omega) \times H^{3/2}(\Gamma)$ (to obtain Hilbert spaces, we will always endow tensor product spaces with the corresponding euclidean tensor product norm).

1.2 A simple fictitious domain method

Let us construct the fictitious domain method that lies at the core of the developments that follow. To this end, we begin by embedding Ω into $\mathbb{T}^2 =$

$(\mathbb{R}/\mathbb{Z})^2$. We define $A : H^2(\mathbb{T}^2) \to H^0(\mathbb{T}^2)$ by $A = -\triangle + \sigma I$, and write again $B : H^2(\mathbb{T}^2) \to H^{3/2}(\Gamma)$ for the trace operator.

A fictitious domain method that aspires to be useful for solving (1) should produce solutions $u^+ \in H^2(\mathbb{T}^2)$ that satisfy, at the very least, that

$$(Au^+)_{|\Omega} \ \Big(= A_\Omega(u^+_{|\Omega})\Big) = f, \tag{3}$$

and that

$$Bu^+ \ \Big(= B(u^+_{|\Omega})\Big) = g. \tag{4}$$

Since in practice it is often much more convenient to begin with some initial extension $f^+ \in H^0(\mathbb{T}^2)$ of f, we will now reformulate these equations into an appropriate least-squares formulation that accommodates for such an extension.

Consider the operator $C_\Omega : H^0(\mathbb{T}^2) \to H^0(\mathbb{T}^2)$ which restricts each function in $H^0(\mathbb{T}^2)$ to Ω, and subsequently extends it by zero. It is easy to see that this operator is an orthogonal projector. Now, consider the least squares functional $\Phi : H^2(\mathbb{T}^2) \to \mathbb{R}^+$, given by

$$\Phi(v) = \|C_\Omega Av - f^+\|^2_{H^0(\mathbb{T}^2)} + \|Bv - g\|^2_{H^{3/2}(\Gamma)}. \tag{5}$$

Theorem 1.1. *There exists a minimizer $u^+ \in H^2(\mathbb{T}^2)$ of Φ such that*

$$\|u^+\|_{H^2(\mathbb{T}^2)} \leq C \left(\|f^+\|^2_{H^0(\mathbb{T}^2)} + \|g\|^2_{H^{3/2}(\Gamma)}\right)^{\frac{1}{2}}.$$

Furthermore, any minimizer of Φ satisfies (3) and (4).

As a preparation to the proof of theorem 1.1, we define the operator $M : H^2(\mathbb{T}^2) \to \mathcal{H}^r := H^0(\mathbb{T}^2) \times H^{3/2}(\Gamma)$ by

$$M = \begin{pmatrix} CA \\ B \end{pmatrix}, \tag{6}$$

and prove the following lemma.

Lemma 1.2. *The operator M is bounded and has closed range; thus, it also has a bounded Moore-Penrose pseudoinverse $M^\dagger$.*

Proof. The operator M is clearly bounded; let us prove that it has closed range too. To see this, use the existence of a bounded extension operator from $H^2(\Omega)$ to $H^2(\mathbb{T}^2)$ to observe that the range of M is $\mathcal{R}(M) = \{(\phi, \gamma)^T \in \mathcal{H}^r : \phi_{|\Omega^c} = 0\}$. By the continuity of the restriction operator we have that this set is closed.

An operator between two Hilbert spaces which is bounded and has closed range has a bounded Moore-Penrose pseudoinverse (see e.g. [5]). Thus the pseudoinverse $M^\dagger$ of M exists and is bounded. □

Proof of theorem 1.1. Let $v \in H^2(\mathbb{T}^2)$. Then, since C_Ω is an orthogonal projector, we have that $\Phi(v) \geq \|C_\Omega f^+ - f^+\|^2_{H^0(\mathbb{T}^2)}$. On the other hand, let $\overline{u} \in H^2(\mathbb{T}^2)$ be any extension of the solution u of problem (1). Then $C_\Omega A\overline{u} = C_\Omega f^+$, while also $B\overline{u} = g$, so that $\Phi(\overline{u}) = \|C_\Omega f^+ - f^+\|^2_{H^0(\mathbb{T}^2)}$. Clearly, $\overline{u}$ is a minimizer of Φ.

Let now u^* be *any* minimizer of Φ. We conclude from the above that $\Phi(u^*) = \|C_\Omega f^+ - f^+\|^2_{H^0(\mathbb{T}^2)}$, from where it follows then that $C_\Omega A u^* = C_\Omega f^+$, and that $Bu^* = g$.

Observe now that we can write $\Phi(v) = \|Mv - b\|^2_{\mathcal{H}^r}$, with $b = (f^+, g)^T \in \mathcal{H}^r$. Thus the minimizer u^+ of Φ with minimal norm is given by $u^+ = M^\dagger b$, and so

$$\|u^+\|_{H^2(\mathbb{T}^2)} = \|M^\dagger b\| \leq \|M^\dagger\| \left(\|f^+\|^2_{H^0(\mathbb{T}^2)} + \|g\|^2_{H^{3/2}(\Gamma)} \right)^{\frac{1}{2}}.$$

□

Remark 1.3. *Observe that theorem 1.1 would remain valid if we changed the norms on H^2 and $H^{3/2}$ to other, equivalent ones. A change of norm in $H^2(\mathbb{T}^2)$ yields a different minimizer with minimal norm that would still satisfy (3) and (4), while an equivalent norm on $H^{3/2}$ would change exactly nothing.*

In contrast, observe that a change of norm on $H^0(\mathbb{T}^2)$ is a more delicate matter. The proof of theorem 1.1 depends critically on the fact that the operator C_Ω is an orthogonal projector.

Thus, our simple fictitious domain method is as follows. To solve problem (1), find an initial extension f^+ of f and take $u^+ = M^\dagger (f^+, g)^T$. We will devote section 2 to the question of how to do this in practice with a reasonable amount of effort.

1.3 Adapting the method to more general problems

We can treat the Neumann problem in a completely analogous fashion within the above framework. The operator $B^{\mathcal{N}} : H^2(\mathbb{T}^2) \to H^{1/2}(\Gamma)$ given by

$$B^{\mathcal{N}} v \to \frac{\partial v}{\partial \mathbf{n}}$$

is bounded and surjective. The problem

$$\begin{aligned} (-\triangle + \sigma I)u &= f \qquad \text{on } \Omega, \\ B^{\mathcal{N}} u &= g, \end{aligned} \tag{7}$$

is well posed if $\sigma > 0$. It is straightforward to verify that theorem 1.1 does not depend on the type of boundary conditions. Instead of looking for the minimizer of Φ as defined in 5, we would be minimizing the functional

$$\Phi^{\mathcal{N}}(v) = \|CAv - f^+\|^2_{H^0(\mathbb{T}^2)} + \|B^{\mathcal{N}} v - g\|^2_{H^{1/2}(\Gamma)}.$$

In a similar fashion it is possible to modify the above method to deal with boundary conditions of other types. Similarly, we are not limited to the differential operator chosen for our model problem (1).

2 Construction of the numerical method

This section consists of two major parts. In the first one, we try to find suitable discrete problems to approximate the minimizer of Φ with smallest norm. Then, in subsection 2.4, we tackle the problem of finding *smooth* minimizers. For simplicity, we will continue to limit ourselves to the Dirichlet problem.

We will start by choosing finite dimensional subspaces $V_j^{\square} \subset H^2(\mathbb{T}^2)$, $V_j^{\Gamma} \subset H^{3/2}(\Gamma)$, and $V_j^0 \subset H^0(\mathbb{T}^2)$, $j \in \mathbb{N}_0$ (the parameter j refers to a chosen level of resolution; more to that in a moment). Then we construct operators $M_j : V_j^{\square} \to V_j^0 \times V_j^{\Gamma}$ that model M in an appropriate way. Of course, this requires particular care, since the operator M is singular.

We then project b onto $V_j^{\square} \times V_j^{\Gamma}$ to obtain $b_j = (f_j^+, g_j)^T$, and then find the minimizer of

$$\Phi_j(v_j) := \|M_j v_j - b_j\|^2_{\tilde{\mathcal{H}}^r}, \tag{8}$$

where the tilde on $\mathcal{H}^r$ signalizes that we plan to modify the norms, by evaluating $u_j^+ = M_j^{\dagger} b_j$ with the aid of a specialized Krylov subspace method.

To find *smooth* minimizers of Φ, and thus smooth extensions of the solution u of problem (1), we will build upon the above framework. We will use the fact that the minimizers u_j^+ are smooth themselves, and try to rescue that smoothness to the infinite dimensional problem via a lifting process.

2.1 B-splines and B-spline wavelet bases

A reasonably complete introduction of B-spline wavelet bases would take too many pages. Instead, we provide here a mostly anecdotal account of the theory in order to provide a basic insight into the concepts, and to introduce the necessary notation. We will also comment briefly on the role each tool will play in the realization of the plan outlined above. Roughly speaking, we obtain with B-spline wavelets a powerful and convenient approximation tool, together with a simple way of identifying Sobolev spaces with copies of ℓ_2. These abilities of wavelets let us obtain good discrete models of M.

For a complete introduction to B-spline wavelet bases we refer the interested reader to [4].

Given $m \geq 2$ (this restriction can be weakened, but in its current form serves to give a streamlined account), let

$$\phi^m(x) = \left[*_{i=1}^m \chi_{[0,1)}\right](x + \left\lfloor \frac{m}{2} \right\rfloor),$$

where $*$ denotes the convolution operator, and $\lfloor x \rfloor$ denotes the largest integer which is smaller or equal to x. The function ϕ^2, for example, is just the standard hat function,

$$\phi^2(x) = \begin{cases} x+1 & \text{if } x \in [-1,0), \\ 1-x & \text{if } x \in [0,1), \\ 0 & \text{otherwise.} \end{cases}$$

In general, ϕ^m is a member of C^{m-2}, is compactly supported, and is a polynomial of degree $m-1$ when restricted to an interval of the form $[k, k+1]$, $k \in \mathbb{Z}$.

For $j \geq 0$, write $\mathcal{Z}_j = \mathbb{Z}/2^j\mathbb{Z}$,

$$\phi_{jk}(x) := 2^{j/2} \sum_{z \in \mathbb{Z}} \phi(2^j(x-z)-k), \tag{9}$$

and set $\mathcal{B}_j = \{\phi_{jk} : k \in \mathcal{Z}_j\}$. Then $\mathcal{B}_j$ is a basis for the space $V_j = \operatorname{span} \mathcal{B}_j$ of all periodic B-splines of order m and mesh size $h = 2^{-j}$. We will call this basis the *scaling function basis* of V_j. The factor $2^{j/2}$ ensures that $\|\phi_{jk}\| \sim 1$.

This type of space is convenient for a variety of reasons. It has good approximation properties; Given a function $h \in H^s(\mathbb{T})$, one has that if $m \geq s$, then

$$\inf_{v \in V_j} \|h - v\|_{H^0(\mathbb{T})} \lesssim 2^{-js} \|h\|_{H^s(\mathbb{T})}. \tag{10}$$

(Here, $a \lesssim b$ means $a \leq Cb$ for some generic constant C independent of any parameters on which a and b may depend. The notation $a \gtrsim b$ has an analogous meaning, and $a \sim b$ means that both $a \lesssim b$ and $a \gtrsim b$ hold.)

Furthermore, increasing the order m of these B-spline spaces does not create any complications, just as it is straightforward to extend this construction to higher dimensions by taking tensor products.

The next step in the construction of B-spline wavelet bases is the choice of appropriate *dual* spaces $\tilde{V}_j$, spanned by the a dual scaling function basis $\tilde{\mathcal{B}}_j = \{\tilde{\phi}_{jk} : k \in \mathcal{Z}_j\}$, with the $\tilde{\phi}_{jk}$ defined from a single $\tilde{\phi}$, also compactly supported, through an expresion analogous to (9). The bases $\mathcal{B}_j$, $\tilde{\mathcal{B}}_j$, must be *biorthogonal*, that is, their elements must satisfy

$$\langle \phi_{jk}, \tilde{\phi}_{jl} \rangle = \delta_{kl}, \tag{11}$$

where we have denoted dual pairing[1] by $\langle \cdot, \cdot \rangle$. Here we choose the dual spaces constructed in [3], which are parametrized not only by m, but also by another integer $\tilde{m} \geq m$, with $m + \tilde{m} \in 2\mathbb{N}$, which gives the order of local polynomial approximation power of the spaces $\tilde{V}_j$. There are some additional restrictions on $\tilde{m}$ that ensure that $\tilde{\phi} \in L_2(\mathbb{R})$; see again [3]. The choices we will make in this article satisfy these constraints.

Another requirement is that the (oblique) projectors $Q_j : H^0(\mathbb{T}) \to V_j$, $\tilde{Q}_j : H^0(\mathbb{T}) \to \tilde{V}_j$, given by

$$Q_j f = \sum_{k \in \mathcal{Z}_{jk}} \langle \tilde{\phi}_{jk}, f \rangle \phi_{jk}, \tag{12}$$

$$\tilde{Q}_j f = \sum_{k \in \mathcal{Z}_{jk}} \langle \phi_{jk}, f \rangle \tilde{\phi}_{jk}, \tag{13}$$

[1] Perhaps artificially in this case, as $H^0(\mathbb{T}) = [H^0(\mathbb{T})]'$. This definition of biorthogonality is intended to be more general

are uniformly bounded.

With these elements, one can construct compactly supported functions $\psi, \tilde{\psi} : \mathbb{R} \to \mathbb{R}$ from which we obtain the wavelet bases. The functions ψ_{jk}, $\tilde{\psi}_{jk}$, defined from $\psi, \tilde{\psi}$ as in (9), together with the scaling functions for $j = 0$ form biorthogonal sets

$$\Psi = \{\phi_{00}\} \cup \{\psi_{jk} : k \in \mathcal{Z}_j\}, \qquad \tilde{\Psi} = \{\tilde{\phi}_{00}\} \cup \{\tilde{\psi}_{jk} : k \in \mathcal{Z}_j\}, \tag{14}$$

called *biorthogonal B-spline wavelet bases.*

The fact that there is one function in each of these sets that is not like any of the others causes some unfortunate notational tension which we must address before going on. We define an index $\lambda = (j, k, e)$, with $j \geq 0$, $k \in \mathcal{Z}_j$, and $e \in \{0, 1\}$, and will write

$$\psi_\lambda = \psi_{(j,k,e)} = \begin{cases} \psi_{kl} & \text{if } e = 1, \\ \phi_{kl} & \text{if } e = 0. \end{cases}$$

For convenience, we write $j(\lambda)$, $k(\lambda)$, $e(\lambda)$ for each of the fields in the indices. Of course, if $e(\lambda) = 0$, and $\psi_\lambda \in \Psi$, then it must hold that $j(\lambda) = k(\lambda) = 0$. We will collect all these valid indices λ in the set ∇. Furthermore, we define for later use $\nabla_j := \{\lambda \in \nabla : j(\lambda) < j\}$.

The sets Ψ, $\tilde{\Psi}$ are both *Riesz bases* for $H^0(\mathbb{T})$. We have for each $f \in H^0(\mathbb{T})$ that

$$f = \sum_{\lambda \in \nabla} \langle f, \tilde{\psi}_\lambda \rangle \psi_\lambda = \sum_{\lambda \in \nabla} \langle \psi_\lambda, f \rangle \tilde{\psi}_\lambda, \tag{15}$$

and also that

$$\|f\|_{H^0(\mathbb{T})} \sim \|\{\langle f, \tilde{\psi}_\lambda \rangle\}_{\lambda \in \nabla}\|_{\ell_2} \sim \|\{\langle \psi_\lambda, f \rangle\}_{\lambda \in \nabla}\|_{\ell_2}. \tag{16}$$

For $j \geq 1$, the sets $\Psi_j = \{\psi_\lambda : \lambda \in \nabla_j\}$, $\tilde{\Psi}_j = \{\tilde{\psi}_\lambda : \lambda \in \nabla_j\}$ are bases of the spaces V_j, $\tilde{V}_j$, respectively.

If $v_j \in V_j$, we have two possible representations for it. One is $v_j = \sum_{\lambda \in \nabla_j} d_\lambda \psi_\lambda$, called the *wavelet representation* and the other is $v_j = \sum_{k \in \mathcal{Z}_j} c_{jk} \phi_{jk}$, called the *scaling function representation.* The map $T_j : \ell_2(\nabla) \to \ell_2(\mathcal{Z}_j)$ that gives us $c_j = T_j d$ is linear and uniformly bounded. We will write $\tilde{T}_j$ for the analogous map on the dual side, and point out the remarkable fact that $T_j^{-1} = \tilde{T}_j^T$ and $\tilde{T}_j^{-1} = T_j^T$.

The maps $T_j^{-1}, \tilde{T}_j^{-1}$ are called *fast wavelet transforms.* The adjective "fast" comes from the fact that evaluating the transformations from scaling function representation to wavelet representation can be done at a cost of $\mathcal{O}(N)$ operations, where N is the dimension of V_j (in the one-dimensional case we have been considering for sketching the construction, $N = 2^j$).

If one has the scaling function representation of v_j and wants to compute its wavelet representation, how does one proceed? It is simple. One has $T_j^{-1} = \tilde{T}_j^T$ and $\tilde{T}_j^{-1} = T_j^T$.

We obtain, for a certain range of s depending on m and $\tilde{m}$ (see [4]), Riesz bases for the spaces $H^s(\mathbb{T})$, $H^{-s}(\mathbb{T})$ simply by rescaling the bases Ψ, $\tilde{\Psi}$. This is accomplished by multiplying each ψ_λ, and each $\tilde{\psi}_\lambda$, by $2^{-j(\lambda)s}$ and $2^{j(\lambda)s}$, respectively. In an abuse of notation we will also write Ψ, $\tilde{\Psi}$ for these rescaled bases. To avoid confusion we always will say of which space they are a basis. We will also assume that the wavelet transforms are written with respect to these scaled bases.

It is important to note that the projectors Q_j and $\tilde{Q}_j$ are not affected by the rescaling. What is more, they stay uniformly bounded with respect to the norms of $H^s(\mathbb{T})$ and $H^{-s}(\mathbb{T})$, respectively.

If Ψ, $\tilde{\Psi}$ are biorthogonal wavelet bases for $H^s(\mathbb{T})$, $H^{-s}(\mathbb{T})$, we have that

$$\langle \psi_\lambda, \tilde{\psi}_\mu \rangle \sim \delta_{\lambda\mu}, \tag{17}$$

and that for $f \in H^s(\mathbb{T})$, $h \in H^{-s}(\mathbb{T})$,

$$f = \sum_{\lambda \in \nabla} \langle f, \tilde{\psi}_\lambda \rangle \psi_\lambda, \qquad\qquad h = \sum_{\lambda \in \nabla} \langle \psi_\lambda, h \rangle \tilde{\psi}_\lambda,$$

while also

$$\|f\|_{H^s} \sim \|\{\langle f, \tilde{\psi}_\lambda \rangle\}_{\lambda \in \nabla}\|_{\ell_2}, \qquad \|h\|_{H^{-s}} \sim \|\{\langle \psi_\lambda, h \rangle\}_{\lambda \in \nabla}\|_{\ell_2}.$$

In other words, the bases Ψ and $\tilde{\Psi}$ induce isomorphisms between ℓ_2 and $H^s(\mathbb{T})$, and between ℓ_2 and $H^{-s}(\mathbb{T})$, respectively.

This construction can be repeated on $\mathbb{T}^2$ simply by carefully taking tensor products. We will omit the details here and refer again to the literature, e.g. [4].

2.2 The discrete operators

Before we begin to discretize the operator M, we will simplify that task a bit by changing the norms on the spaces $H^2(\mathbb{T}^2)$ and $H^{3/2}(\Gamma)$ for equivalent norms given by wavelet bases. The justification for this was given in remark 1.3. For the space $H^2(\mathbb{T}^2)$ we will choose for instance $m = 3$ and $\tilde{m} = 5$, and let Ψ, $\tilde{\Psi}$ be the corresponding B-spline wavelet bases of $H^2(\mathbb{T}^d)$, $H^{-2}(\mathbb{T}^d)$, respectively. We will not need the space $H^{-2}(\mathbb{T}^d)$. We will refer to the norm induced by Ψ through the notation $\|\cdot\|_{H^2,\Psi}$, and write $V_j^{\square} = \operatorname{span} \Psi_j$.

For $H^{3/2}(\Gamma)$, we begin by identifying Γ with $\mathbb{T}$ through a suitable parametrization $\gamma : \mathbb{T} \to \Gamma$, and choose for instance $m^\Gamma = 2$, $\tilde{m}^\Gamma = 6$. We then rescale the corresponding bases in such a way that Ψ^Γ (originally the primal basis) becomes a basis of $H^{-3/2}(\Gamma)$, and $\tilde{\Psi}^\Gamma$ (originally the dual basis) becomes a basis for the space $H^{3/2}(\Gamma)$. This means that the boundary values g will have to be approximated from the *dual* spaces $\tilde{V}_j$. The reason for this choice is that given a trace $h \in H^{3/2}(\Gamma)$ of a $B-$spline on $\mathbb{T}$, computing the projection $\tilde{Q}_j^\Gamma h$ does not involve scalar products with the functions $\tilde{\phi}_{jk}$, which cannot easily be evaluated at arbitrary non-dyadic points.

Again basing ourselves on remark 1.3, we will also multiply the norm on $H^{3/2}(\Gamma)$ by a constant weight ω_B. This has as a consequence that the boundary conditions are enforced better at a lower level (in practice, a value of $\omega_B = 70$ seemed to work well). We will refer to the norm of the space $H^{3/2}(\Gamma)$ induced by $\tilde{\Psi}^\Gamma$ through the notation $\|\cdot\|_{H^{3/2},\tilde{\Psi}^\Gamma}$, and write $V_j^\Gamma = \operatorname{span} \tilde{\Psi}_j^\Gamma$.

Since by remark 1.3 we are not allowed to touch the norm on $H^0(\mathbb{T})$, we have no alternative but to choose a space for which it is easy to construct an orthonormal basis.

Given $k = (k_1, k_2) \in \mathcal{Z}_j^2 = (\mathbb{Z}/2^j\mathbb{Z})^2$, write $\square_{jk} := 2^{-j}[k_1, k_1+1) \times [k_2, k_2+1)$, and let

$$V_j^0 = \{f \in L_2(\mathbb{T}^d) : f_{|\square_{jk}} \in \Pi_{m-1},\ k \in \mathcal{Z}_j^2\}$$

define the spaces of discontinuous piecewise polynomials of degree $m-1$ on a dyadic grid with mesh size $h = 2^{-j}$. To construct an orthonormal basis for V_j^0, we proceed as follows. We apply first Gram-Schmidt orthonormalization in $L_2([0,1]^2)$ to the monomials $x^i y^j$ with $i + j \leq m-1$, $i, j \geq 0$. We write $\{\phi^0, \phi^1, \ldots, \phi^n\}$ for the functions we thus obtain (here, $n = (m+1)m/2$), and note that it also is a basis for V_0^0. We write $\phi_{jk}^i(x) = 2^j \phi^i(2^j x - k)$, and observe that the set $\mathcal{B}_j^0 := \{\phi_{jk}^i : i = 1, \ldots, n,\ k \in \mathcal{Z}_j^2\}$ is an orthonormal basis for V_j^0. We use the canonical norm on $L_2(\mathbb{T}^d)$ and no additional notation for it.

We will model the operator M defined in (6) by discrete operators of the form

$$M_j = \begin{pmatrix} C_j A_j \\ B_j \end{pmatrix}. \tag{18}$$

The operators A_j and B_j will be obtained simply by projections (corresponding to a Galerkin-like discretization), while the operator C_j will be a modified restriction operator engineered to keep in bounds the effects of the large kernel introduced by C_Ω.

Given $v_j \in V_j^\square$, we define $A_j v_j := P_{V_j^0} A v_j$, where $P_{V_j^0}$ is the orthogonal projection onto V_j^0. The operator B_j is given by $B_j v_j = \tilde{Q}_j^\Gamma B v_j$.

If we write T_j for the inverse wavelet transform for $V_j^\square$, we observe that we can factorize the matrix $\underline{A}_j$ of A_j with respect to the bases Ψ_j and $\mathcal{B}_j^0$ as

$$\underline{A}_j = \underline{A}_j^0 T_j, \tag{19}$$

where $\underline{A}_j^0$ is the matrix of A_j with respect to the bases $\mathcal{B}_j^0$ and the scaling function basis $\mathcal{B}_j^\square$ of $V_j^\square$. This factorization has the advantage that a matrix-vector multiplication with any of $\underline{A}_j^0$ and T_j can be evaluated at a cost of $\mathcal{O}(N)$, with $N = \dim\ V_j^\square$, and thus we can evaluate the product with $\underline{A}_j$ in $\mathcal{O}(N)$ operations using this factorization. The direct multiplication with $\underline{A}_j$ would take $\mathcal{O}(N \log N)$ operations (see [4] for further discussion).

The entries in the matrix $\underline{A}_j^0$ can be computed exactly using straightforward quadrature techniques.

We factorize the matrix $\underline{B}_j$ of B_j with respect to the bases Ψ_j and $\tilde{\Psi}_j^\Gamma$ in a similar way, and for similar reasons. We obtain

$$\underline{B}_j = (\tilde{T}_j^\Gamma)^{-1} \underline{B}_j^0 T_j = (T_j^\Gamma)^T \underline{B}_j T_j. \tag{20}$$

The entries in the matrix $\underline{B}_j^0$ can again be approximated to high accuracy using standard quadrature techniques.

Given $f_j \in V_j^0$, we could of course define $\overline{C}_j f_j := P_{V_j^0} C_\Omega f_j$. The drawback of this approach is that the matrix of $\overline{C}_j$ with respect to the basis $\mathcal{B}_j^0$ can contain arbitrarily small entries. Numerical errors that occur during the computation of these entries have the potential of changing the rank of M_j, a situation which could lead to unpredictable changes in the solution (see [15], pages 335-338, for a thorough discussion). As if that were not enough, we also have that the entries in this matrix are expensive to compute.

To eliminate both problems, we define the discrete operator C_j as follows. Write $f_j = \sum_{i,k} c_k^i \phi_{jk}^i$, and then let

$$C_j f_j := \sum_{ik} (\underline{C}_j c)_k^i \phi_{jk}^i,$$

with

$$(\underline{C}_j c)_k^i = \begin{cases} c_k^i & \text{if supp } \phi_{jk}^i \cap \Omega \neq \emptyset \\ 0 & \text{otherwise.} \end{cases} \tag{21}$$

The entries in the matrix $\underline{C}_j$ are either one or zero. They are easy to compute, and they are far more robust against numerical error (they are not completely immune, as one still has to find out whether supp $\phi_{jk}^i \cap \Omega \neq \emptyset$).

2.3 The discrete least-squares problems

What remains now is to formulate our discrete least-squares problems, and decide how to solve them. Given an element $v_j \in V_j^\square$, we will write $\underline{v}_j$ for its wavelet representation, and $\overline{v}_j$ for its scaling function representation. We will also write

$$\underline{M}_j = \begin{pmatrix} \underline{C}_j \underline{A}_j \\ \underline{B}_j \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & (T_j^\Gamma)^T \end{pmatrix} \begin{pmatrix} \underline{C}_j \underline{A}_j^0 \\ \underline{B}_j^0 \end{pmatrix} (T_j)\,. \tag{22}$$

What we have achieved with this setup is that

$$\Phi_j(v_j) = \|M_j v_j - b_j\|_{\mathcal{H}^r}^2 = \|\underline{M}_j \underline{v}_j - \underline{b}_j\|_2^2, \tag{23}$$

where $\mathcal{H}^r = H^2(\mathbb{T}^d) \times H^{3/2}(\Gamma)$ but with the norms induced by the wavelet bases, and $b_j = (P_{V_j^0} f^+, \tilde{Q}_j^\Gamma g)$. That is, we have converted the problem on the (finite-dimensional) space $V_j^\square$ into a fully equivalent problem on $\mathbb{R}^N$ with $N = \dim V_j^\square = 2^{2j}$.

To actually find the minimizer of

$$\|\underline{M}_j \underline{v}_j - \underline{b}_j\|_2^2 \tag{24}$$

we will use an iterative Krylov subspace solver specialized for solving least-squares problems to evaluate (approximatively) the pseudoinverse of $\underline{M}_j$ on the right-hand side $\underline{b}_j$. While not the best in terms of performance, we choose CGLS [11]. Mathematically, CGLS is equivalent to CG applied to the normal equations, but has better numerical properties. We have made this choice for reasons that will become apparent at the end of the next subsection.

2.4 Recovering smoothness

The idea behind our strategy to recover smoothness is as follows. The minimizer $u_j^+ = M_j^\dagger b_j$ of Φ_j will have the same smoothness as any other element in $V_j^\square$, and, under the right circumstances, we will have that u_j is a good approximation of *some* smooth minimizer of Φ.

While we may expect u_j^+ to converge to a minimizer of Φ in (5), we cannot expect this limit to be smooth. Looking at the kernel of M, we see that it consists of functions $\kappa \in H^2(\mathbb{T}^d)$ which are zero on Ω, and which satisfy $B\kappa = 0$. We cannot expect extensions of u to $\mathbb{T}^d$ with higher Sobolev smoothness than that found in H^2 to be orthogonal to this kernel.

So to obtain such a smooth extension of u using the solutions u_j^+ of the discrete problems we may have to endow it with some component in this kernel. Our plan is to "lift" the smoothness of the finite dimensional spaces $\{V_j^\square\}$ by collecting the components of the solutions u_j^+ in the kernels of the operators M_{j+1}. Thus, the definition of our solution operator starts with a standard solution for some initial j (for simplicity we take $j = 0$),

$$S_0 b := M_0^\dagger b_0, \tag{25}$$

and then define

$$S_{j+1} b := P_{\mathcal{N}(M_{j+1})} S_j b + M_{j+1}^\dagger b_{j+1}. \tag{26}$$

To realize this iteration, we will need a way to evaluate the projections onto the kernels of the operators M_j. We will engage this task in a moment. (For further discussion and analysis of the sequence defined by (25) and (26), see [14])

2.4.1 Realizing the projections onto the kernel

The obtain the minimizer $\underline{u}_j$ of

$$\Phi_j(\underline{v}_j) = \left\|\underline{M}_j \underline{v}_j - \underline{b}_j\right\|_2^2$$

we can use, for example, the conjugate gradients (CG) algorithm[12] to solve the normal equations,

$$\underline{M}_j^T \underline{M}_j \underline{u}_j = \underline{M}_j^T \underline{b}_j. \tag{27}$$

While this has well known disadvantages, it also has an important advantage, which is that it can give us the projection of $\underline{v}_{j-1}$ onto $\mathcal{N}(\underline{M}_j)$ essentially for free.

The key to that insight is obtained by taking a look at what the CG algorithm does. To find an approximate solution of the linear equation $Ax = d$, the CG method produces iterates x^i, each of which is the minimizer in $W_i = x^{(0)} + \text{span}\{r^0, r^{(1)}, \ldots, r^{(i-1)}\}$ of the functional $\gamma_i(y) = (y - x^*)^T A(y - x^*)$, where x^* is the exact solution of $Ax = d$, $x^{(0)}$ is some initial guess, and $r^{(k)} = A^k(d - Ax^{(0)})$. The minimizer of γ_i in W_i exists, and is unique, only if A is symmetric positive definite on W_i. One has that $x^{(i)} = x^*$ when $W_i = W_{i+1}$ (if the algorithm is performed with exact arithmetic), but if the condition number of A is reasonable, then the $x^{(i)}$ will be a good approximation of x^* far earlier.

Suppose now that A is symmetric and positive semidefinite. If $d \perp \mathcal{N}(A)$, then $r^k \perp \mathcal{N}(A)$ for all k, and thus A is symmetric positive definite on

$$\begin{aligned} W_i &= x^{(0)} + \text{span}\{r^0, r^{(1)}, \ldots, r^{(i-1)}\} \\ &= P_{\mathcal{N}(A)} x^{(0)} + P_{\mathcal{N}(A)^\perp} x^{(0)} + \text{span}\{r^0, r^{(1)}, \ldots, r^{(i-1)}\} \end{aligned}$$

for all i [11]. Given an initial guess $x^{(0)}$, we will obtain at the i-th step an approximation of x^* which satisfies $P_{\mathcal{N}(A)} x^{(i)} = P_{\mathcal{N}(A)} x^{(0)}$. Since $\underline{M}_j^T \underline{b}_j \perp \mathcal{N}(\underline{M}_j^T \underline{M}_j)$, and since $\mathcal{N}(\underline{M}_j^T \underline{M}_j) = \mathcal{N}(\underline{M}_j)$, we can compute expressions of the form

$$P_{\mathcal{N}(\underline{M}_{j+1})} \underline{u}_j + \underline{M}_{j+1}^\dagger \underline{b}_{j+1},$$

as needed in (26), by solving (27) with the conjugate gradient method using $\underline{u}_j$ as an initial guess.

2.5 Realization

Now write

$$\text{CG}(A, d, x_0, \epsilon)$$

for the approximate solution of $Ax = d$, with $x^{(0)}$ as an initial guess, obtained by iterating until the norm of the residual is smaller than ϵ. Then the numerical realization of (25), (26), which we will call the SPFD ("Smoothness Preserving Fictitious Domain") method, is given by

$$\begin{aligned} &\text{SPFD}(j_0, j, \{b_k\}, \{\epsilon_k\}) := \\ &\begin{cases} 0 & \text{if } j < j_0 \\ \text{CG}(\underline{M}_j^T \underline{M}_j,\ \underline{M}_j^T \underline{b}_j,\ \text{SPFD}(j_0, j-1, \{b_k\}, \{\epsilon_k\},),\ \epsilon_j) & \text{otherwise.} \end{cases} \end{aligned} \tag{28}$$

Computing an approximation to $S_J b$ amounts to evaluate SPFD$(j_0, J, \{b_j\}, \{\epsilon_k\})$. The sequence $\{\epsilon_k\}$ gives us additional flexibility by allowing us to use a smaller ϵ on a lower level (where iterations are cheap), than on a higher level.

The question arises as to what effect the inexact evaluation of $\underline{M}_j^\dagger \underline{b}_j$ has on the sequence $\{S_j b\}$. In practice, it does not seem to play an important role; further research is needed to throw light on this issue.

Instead of using standard CG with the normal equations, one should use the mathematically equivalent but numerically somewhat superior CGLS, developed in [11]. The direct application of other Krylov subspace least-squares solvers is a delicate matter. In the case of LSQR[16], a very robust least squares solver, the problem is to implement the projections onto the kernel. Still other methods, like RRGMRES [2], assume that the system is given through a square matrix. Again, we see in further research an opportunity for improvements in performance of the method described in this chapter.

3 Numerical experiments

To test our method, we formulated a few different boundary value problems on the domain

$$\Omega = \{(x, y) : \|(x, y) - (1/2, 1/2)\|_2 < 0.3\},$$

a disc of radius $r = 0.3$ and center $(1/2, 1/2)$. For the experiments, we chose the orders mentioned in 2.2, namely, $m = 3$, $\tilde{m} = 5$ and $m^\Gamma = 2$, $\tilde{m}^\Gamma = 6$.

Example E1: We formulated problem (1) with homogeneous Dirichlet boundary conditions, $\sigma = 0$ and $f = 1$. We chose the obvious extension of the right-hand side to $\mathbb{T}^2$, $f^+ = 1$, and evaluated (28) for $J = 8$, $j_0 = 3$. The iteration history is summarized in Table 1, where we list the tolerance parameter passed to the CGLS solver, together with the number of iterations needed at each level, and the norm of the residual before the first iteration. A plot of the solution, along with the boundary values can be found in figures 1(a) and 1(b).

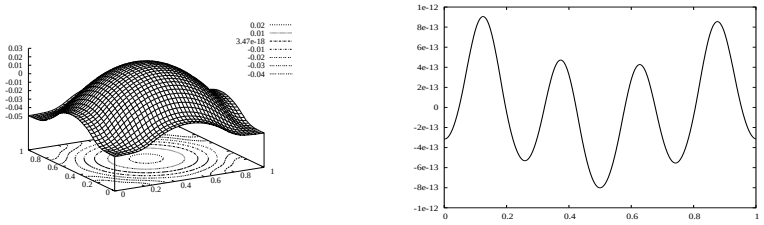
j	Tolerance	Iterations	Initial residual
3	1.0000e-05	11	7.0711e-01
4	2.5119e-05	0	6.7104e-11
5	6.3096e-05	0	6.6714e-11
6	1.5849e-04	0	6.4922e-11
7	3.9811e-04	0	6.4058e-11
8	1.0000e-03	0	6.3563e-11

Table 1: *Iteration history (example E1)*

We believe that this experiment is quite remarkable. It shows that the SPFD can indeed find very smooth solutions if that is possible. In this case, the solution on the domain is polynomial; one easily checks that the solution of the original problem is

$$u = 0.25 \left(r^2 - (x - 0.5)^2 - (y - 0.5)^2 \right).$$

The SPFD method is able to exploit this and finds in V_3 an extension of u to $\mathbb{T}^2$!



(a) Extended solution

(b) Boundary values (note the order of magnitude on the y-axis)

Figure 1: *Solution and boundary values of the solution at $\partial\Omega$ (example E1).*

Example E2: We formulated again problem (1), with $\sigma = 0$ and Ω as above, but this time with less trivial data. We took $f = f^+_{|\Omega}$, with

$$f^+(x,y) = 1 + 0.5(\cos(4\pi x) + \sin(2\pi y)) \tag{29}$$

and

$$g(s) = 0.1\cos(2\pi s). \tag{30}$$

j	Tolerance	Iterations	Initial residual
3	1.0000e-05	109	7.3961e-01
4	2.5119e-05	152	4.8627e-02
5	6.3096e-05	144	2.1562e-02
6	1.5849e-04	194	1.0467e-02
7	3.9811e-04	187	5.1617e-03
8	1.0000e-03	8	2.6071e-03

Table 2: *Iteration history (example E2)*

We evaluated (28) for $J = 8$, $j_0 = 3$. We show the iteration history in Table 2, and a picture of the solution, along a plot of the boundary values, in figures 2(a) and 2(b).

In figure 3, we plotted the decay of the approximation error for the approximate solution u_8^+ to example E2

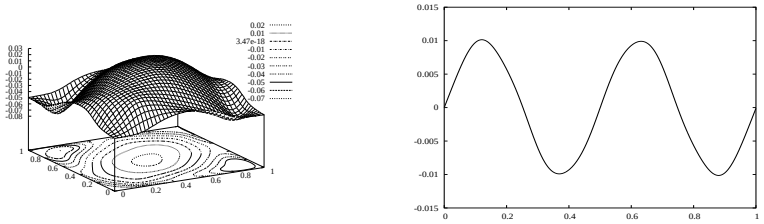
$$E_j^Q(u_8^+) = \|Q_j u_8^+ - u_8^+\|_{H^0(\mathbb{T}^2)},$$

which is uniformly equivalent to the real approximation error

$$E_j(u_8^+) = \inf_{v \in V_j} \|v - u_8^+\|_{H^0(\mathbb{T}^2)},$$

but easier to obtain through the use of the wavelet representation of u_8^+.

To measure the actual order of the method in this case, we did a linear fit to the errors (in logarithmic coordinates) of the function $\kappa(j) = 2^{js}C$ (plotted along the errors in figure 3). This gave us an observed convergence order of $s = -3.427$, which amounts to a convergence rate of $\mathcal{O}(h^{3.427})$. This is higher than the expected order, which is $\mathcal{O}(h^3)$.



(a) Extended solution

(b) Boundary values

Figure 2: *Solution and boundary values of the solution at $\partial\Omega$ (example E2).*

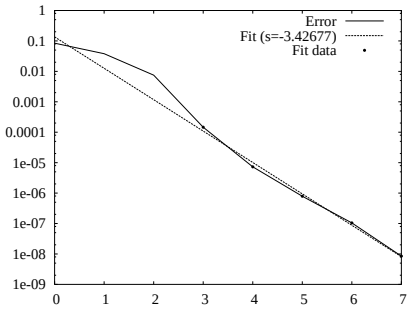
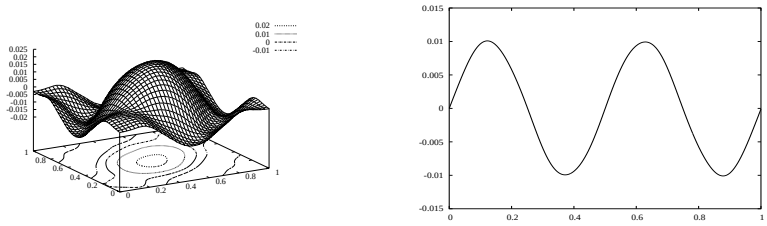


Figure 3: *Approximation errors and fitted idealized convergence rate for u_J^+ (example 2).*

Example E3: We repeat example E2, but this time without the nested iteration scheme, letting CGLS iterate for $j = 8$ until the residual was smaller than $\epsilon = 10^{-3}$, which took 646 iterations. In figures 4(a) and 4(b) it is possible to observe the solution and the boundary values. In figure 5 we see the approximation error, which shows a much slower error decay than when using the nested iteration scheme. We conclude that the nested iteration is essential to recover additional smoothness.

Example E4: Finally, we tried our method with a Neumann problem. We used for this the same data as in example E2. Find an iteration history in Table 3, plots of the solution and the values of the normal derivative at the boundary



(a) Extended solution

(b) Boundary values

Figure 4: *Solution and boundary values of the solution at $\partial\Omega$ (example 3).*

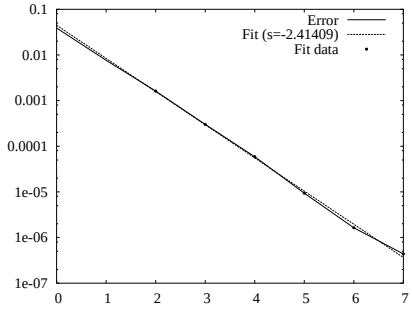


Figure 5: *Approximation errors and fitted idealized convergence rate for u_J^+ (example 3).*

in figure 6(a) and 6(b), and a plot of the decay of the error (together with a fitted, idealized convergence history) in figure 7.

3.1 Comparision with the FDLM method

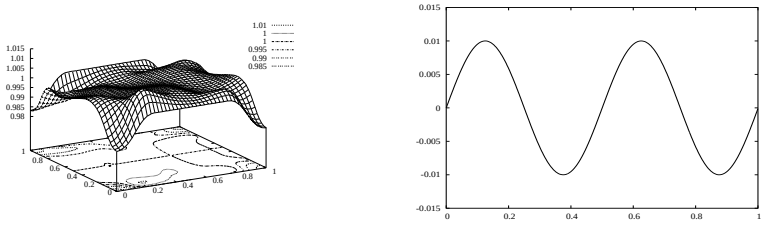
The fictitious domain - Lagrange multiplier method (we will be very brief here; see for instance [8] for a more detailed account) is derived by appending the boundary conditions to the energy functional on the fictitious domain using a Lagrange multiplier. From the optimality conditions for the resulting saddle-point problem, and with $\Omega \subset \mathbb{T}^2$ and (1) as above, one obtains the following problem. Given $(f^+, g) \in H^{-1}(\mathbb{T}^2) \times H^{1/2}(\partial\Omega)$, find $(\tilde{u}^+, p) \in H^1(\mathbb{T}^2) \times H^{-1/2}(\partial\Omega)$ such that

$$\begin{pmatrix} A & B^* \\ B & 0 \end{pmatrix} \begin{pmatrix} \tilde{u}^+ \\ p \end{pmatrix} = \begin{pmatrix} f^+ \\ g \end{pmatrix}. \tag{31}$$

In our case, the operator $A : H^1(\mathbb{T}^2) \to H^{-1}(\mathbb{T}^2)$ is, as above, given by $A = -\triangle + \sigma I$. The operator $B : H^1(\mathbb{T}^2) \to H^{1/2}(\partial\Omega)$ is the trace operator, and the

j	Tolerance	Iterations	Initial residual
3	1.0000e-05	157	8.3501e-01
4	2.5119e-05	152	7.0388e-02
5	6.3096e-05	236	2.6126e-02
6	1.5849e-04	182	1.1880e-02
7	3.9811e-04	146	5.7156e-03
8	1.0000e-03	38	2.8311e-03

Table 3: *Iteration history (example E4)*



(a) Extended solution

(b) Boundary values

Figure 6: *Solution and boundary values of the solution at $\partial\Omega$, example E4*

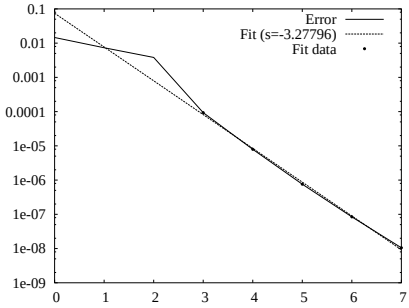


Figure 7: *Approximation errors and fitted idealized convergence rate for u_J^+ (example 4).*

additional variabe p is the Lagrange multiplier. The operator

$$\begin{pmatrix} A & B^* \\ B & 0 \end{pmatrix} : H^1(\mathbb{T}^2) \times H^{-1/2}(\partial\Omega) \to H^{-1}(\mathbb{T}^2) \times H^{1/2}(\partial\Omega) \tag{32}$$

is an isomorphism, and thus problem (31) always has a unique solution. The $\tilde{u}^+$ thus obtained yields a solution to our original boundary value problem when restricted to Ω, while the Lagrange multiplier p is the jump in the normal derivatives of $\tilde{u}^+$ at the boundary of Ω whenever $f^+ \in H^0(\mathbb{T}^2)$.

Using the same data (and the same domain) as in example E2 above, we constructed, using the techniques in [13], a good approximation $\tilde{u}_8^+ \in V_8^\square$, $p_5 \in V_5^\Gamma$ of the corresponding solution of problem (31). We have plotted it, together with the values at the boundary, in figure 8. In figure 9 we show the decay of the approximation errors together with the fitted idealized convergence rate. The observed rate of $\mathcal{O}(h^{1.497})$ corresponds to that predicted by the theory. Finally, we plot in figure 10 side by side the approximation errors for the solutions obtained for this data with the SPFD and the FDLM methods. Clearly, the SPFD method can achieve significantly higher levels of accuracy with fewer degrees of freedom.

Concluding remarks

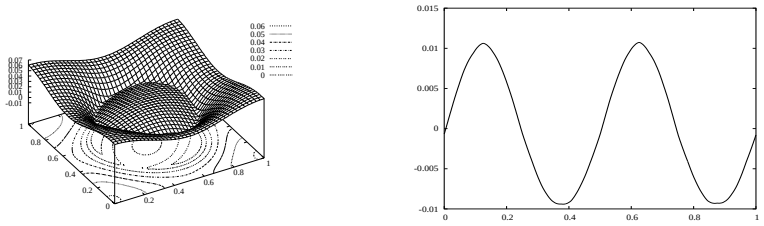
We have presented a fictitious domain method which is able to produce an extension of the solution of (1) with seemingly the same smoothness properties as those of the original solution. This enables it to achieve better accuracy with fewer degrees of freedom. While this is encouraging, we should rate the present method as experimental. A lot can and should be improved, both in the theoretical understanding of the method and in the actual implementation.

Improvements can be expected, for instance, from a better choice of iterative solver. It would be necessary to find an alternative way of realizing the projections onto the kernels in (26).

Other important directions of research include the construction of an adaptive version of this method, and a modified formulation that allows boundaries with less regularity. Breakthroughs in those directions have the potential of making the method a practical universal solver for elliptic boundary value problems.

Acknowledgments

The inspiration for the material in this article came from interesting discussions with Wolfgang Dahmen, Silvia Bertoluzza, Peter Oswald and Angela Kunoth. I am indebted to Henning Esser and Nicolas Neuss for their careful proofreading and their suggestions. All remaining mistakes are the authors sole responsibility.



(a) Extended solution

(b) Boundary values

Figure 8: *Solution obtained using the FDLM method, and its boundary values of the solution at* $\partial\Omega$.

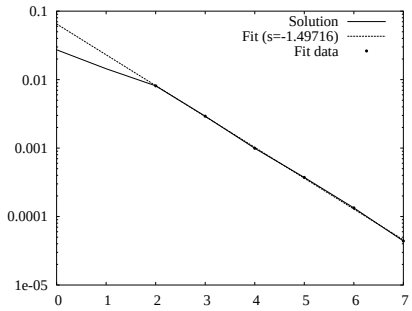


Figure 9: *Approximation errors and fitted idealized convergence rate for* $\tilde{u}_8^+$ *(FDLM method).*

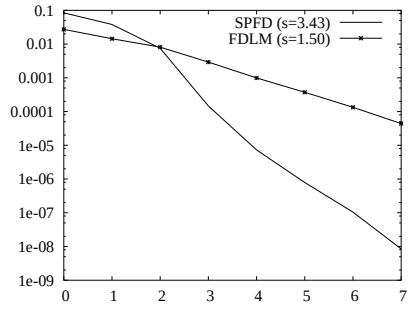


Figure 10: *Comparision of linear approximation errors.*

References

[1] C. Börgers and O. B. Widlund. On finite element domain imbedding methods. *SIAM Journal of Numerical Analysis*, 27(4):963–978, 1990.

[2] D. Calvetti, B. Lewis, and L. Reichel. GMRES-type methods for inconsistent systems. *Linear Algebra and its Applications*, 316(1–3):157–169, 2000.

[3] A. Cohen, I. Daubechies, and J. C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 45:485–560, 1992.

[4] W. Dahmen. Wavelet and multiscale methods for operator equations. *Acta Numerica*, 6:55–228, 1997.

[5] F. Deutsch. *Best approximation in inner product spaces.* Springer-Verlag, Berlin, Germany / Heidelberg, Germany / London, UK / etc., 2001.

[6] H. Fujita, H. Kawahara, and H. Kawarada. Distribution theoretic approach to fictitious domain method for neumann problems. *East-West Journal for Numerical Analysis*, 3(2):111–126, 1995.

[7] V. Girault and R. Glowinski. Error analysis of a fictitious domain method applied to a dirichlet problem. *Japan Journal of Industrial and Applied Mathematics*, 12(3):487–514, 1995.

[8] R. Glowinski, P. Tsorng-Whay, and J. Periaux. A fictitious domain method for dirichlet problem and applications. *Comput. Methods Appl. Mech. Eng.*, 111(3-4):283–303, 1994.

[9] P. Grisvard. *Elliptic problems in nonsmooth domains.* Monographs and studies in mathematics. Pitman, Boston, 1985.

[10] J. Haslinger and R. A. E. Mäkinen. *Introduction to Shape Optimization: Theory, Approximation, and Computation.* Number 7 in Advances in Design and Control. SIAM, Philadelphia, 2001.

[11] M. R. Hestenes. Pseudoinverses and conjugate gradients. *Comm. of the ACM*, 18(1):40–43, 1975.

[12] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Stand.*, 49:409–436, 1952.

[13] A. Kunoth. Wavelet techniques for the fictitious domains - lagrange multiplier approach. *Preprint, University of Bonn*, 2000.

[14] Mario S. Mommer. *Towards a fictitious domain method with optimally smooth solutions.* PhD thesis, in preparation. 2005.

[15] M. Z. Nashed. Perturbations and approximations for generalized inverse and linear operator equations. In M. Z. Nashed, editor, *Generalized Inverses and Applications*, pages 325–396, London, October 1973. Academic Press.

[16] C. C. Paige and M. A. Saunders. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM Transactions in Mathematical Software*, 8(1):43–71, 1982.