UNIVERSITY OF CALIFORNIA,
IRVINE

SEEM: A Kernel-Based Fictitious Domain Method

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Mathematics

by

Daniel Joseph Agress

Dissertation Committee:
Professor Patrick Guidotti, Chair
Professor Long Chen
Chancellor's Professor John Lowengrub

2020

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Shortly before I came to UCI, I received an email from Patrick Guidotti inviting me to collaborate on a project. I am very fortunate that I accepted. Over the past six years, Patrick has been a superb advisor. He has been fully accessible,wonderfully supportive, and a master pedagogue. He strikes the perfect balance of actively engaging with my work while leaving me the independence to explore on my own. Discussing issues with Patrick invariably leads to new ideas and perspectives, and exposure to his broad scope of interests - from PDEs to differential geometry to applied math - has provided me with a rich graduate experience. The core idea of *SEEM* is Patrick's and I am very lucky to have had the opportunity to work on such an exciting project with him.

During my first four years of graduate school, I was co-advised by Jeffrey Streets. Jeff was a clear and patient teacher who exposed me to the beautiful world of differential geometry and helped me publish my first paper on minimizers of the Born-Infeld energy. Although my dissertation work has shifted to applied math, I hope to remain a lifelong student of differential geometry.

Over the course of my graduate studies, I was fortunate to receive support for several academic quarters from GAANN. I also received support from a UCI Provost PhD Fellowship and received a Dissertation Fellowship this past spring.

My parents ingrained in me a love of knowledge and a commitment to lifelong learning. From my father's quirky love of space and science to my mother's strong determination that I take advantage of every available educational resource, it is only due to their loving support that I embarked on the path of obtaining a PhD in mathematics.

Since Aliza and I were married six and a half years ago, much of my time and energy has been absorbed by pursuing my PhD. Throughout these past six years, Aliza has provided me with close companionship, constant encouragement, and a warm and supporting home, allowing me to diligently focus on my graduate studies.

Graduate students face many unknowns over the course of their studies: Which university and advisor are a good fit for me? Which research projects are worth pursuing? Am I making progress or wasting my time? Should I change course and try something new? Upon finishing, I can look back and recognize the hand of G-d guiding me through each step of process, ensuring that I encountered just what I needed to get me to where I am today. Mathematics has been called the language with which G-d has created the world; thus, I feel humbled and fortunate to have had the opportunity to pursue its study in such depth these past six years.

VITA

Daniel Joseph Agress

EDUCATION

Doctor of Philosophy in Mathematics **2020**
UCI *Irvine, California*

Bachelor of Science in Mathematics **2012**
UCLA *Los Angeles, California*

RESEARCH EXPERIENCE

Graduate Research Assistant **2014–2020**
UCI *Irvine, California*

TEACHING EXPERIENCE

Teaching Assistant **2015–2019**
UCI *Irvine, California*

REFEREED JOURNAL PUBLICATIONS

Existence results for the nonlinear Hodge minimal surface energy **2019**
Calculus of Variations and Partial Differential Equations

PREPRINTS

A novel optimization approach to fictitious domain methods **2018**
arXiv Preprint

The smooth selection embedding method with Chebyshev polynomials **2019**
arXiv Preprint

ABSTRACT OF THE DISSERTATION

SEEM: A Kernel-Based Fictitious Domain Method

By

Daniel Joseph Agress

Doctor of Philosophy in Mathematics

University of California, Irvine, 2020

Professor Patrick Guidotti, Chair

This thesis presents *SEEM (Smooth Extension Embedding Method)*, a novel approach to the solution of boundary value problems within the framework of the *fictitious domain method* philosophy. The salient feature of the novel method is that it reduces the whole boundary value problem to a linear constraint for an appropriate optimization problem formulated in a larger, simpler set which contains the domain on which the boundary value problem is posed and which allows for the use of straightforward discretizations. It can also be viewed as a fully discrete *meshfree method* which uses a novel class of basis functions, thus building a bridge between fictitious domain and meshfree methods.

SEEM in essence computes a (discrete) extension of the solution to the boundary value problem by selecting it as a smooth element of the complete affine family of solutions of the original equations, which now yield an underdetermined problem for an unknown defined in the whole fictitious domain. The actual regularity of this extension is determined by that of the analytic solution and by the choice of objective functional. Numerical experiments are presented which demonstrate that the method can be stably used to efficiently solve boundary value problems on general geometries, and that it produces solutions of tunable (and high) accuracy. Divergence-free and time-dependent problems are considered as well.

Part I

Background

Chapter 1

Introduction

In this thesis, a novel optimization approach is proposed for the resolution of general boundary value problems on complex geometries. The approach is a hybrid of the *meshfree collocation* and the *fictitious domain methods* for solving boundary value problems. As will be demonstrated, this method not only establishes a direct connection between these two general approaches but also combines their strengths.

Partial differential equations are ubiquitous in mathematics, science, and engineering. Thus, the study of numerical methods for PDEs is a fundamental task of applied mathematics and many classes of methods have been developed and studied. The most established and commonly used numerical methods for solving boundary value problems are the finite element method, the finite difference method, and spectral methods. While the finite element method come with the heavy burden of generating a mesh (which becomes a serious limiting factor when dealing with some problems, like, for instance, Moving Boundary Problems or in three space dimensions), straightforward finite difference methods and spectral methods are limited by the small number of allowable shapes for the domain Ω.

Two widely used methods which seek to avoid these difficulties are known as *meshfree collo-*

cation and *fictitious domain methods*. In *meshfree collocation methods*, the solution is sought as a linear combination of radial basis functions centered at collocation points scattered throughout the domain Ω and on its boundary Γ. While these methods can achieve very high orders of convergence, in their simplest implementations, the resulting matrices are dense and poorly conditioned, see, for example, [9, Chapter 16]. This leads to difficulties when scaling to denser grids. As described in [9, 27] and as we will explain below, these methods can also be viewed as a reformulation of BVPs as constrained optimization problems on $\mathbb{R}^d$.

In *fictitious domain methods*, the problem is transplanted from the original domain Ω to an encompassing simple region $\mathbb{B}$, where straightforward discretizations and solvers can be utilized. The computed solution is an approximation of u within the domain Ω and an extension of this approximation on $\mathbb{B}\backslash\Omega$. A drawback common to many of these methods is that a lack of regularity across the boundary often leads to a lack of regularity of the global extension, and hence to a low order of convergence, see [23] for a discussion of this issue with regards to the Immersed Boundary Method. Additionally, these methods often require very different treatment of the various types of commonly occurring elliptic operators and boundary conditions, see [17].

The approach proposed here is a hybrid of these two methods. In a way similar to *meshfree methods*, it reduces the entire boundary value problem to playing the role of a linear constraint to an optimization problem for an appropriately chosen objective functional defined on a larger domain. However, borrowing from the *fictitious domain* framework, this constrained optimization procedure is carried out on a regular grid with straightforward and efficient discretizations. As we will explain below, this hybrid procedure will combine the simplicity, speed, and scalability of *fictitious domain methods* with the high order accuracy and wide ranging versatility of *meshfree collocation methods*.

The thesis is composed of four parts. In Part I, we introduce *SEEM* from the ground up and

also describe how the method can be viewed from the perspective of the fictitious domain and meshfree methods. We will then move on to Part II, where we will give a detailed discussion of how to define, discretize, and solve the *SEEM* system. In Part III, we will discuss the well posedness and convergence of the method. Finally, in Part IV, we will give numerical experiments which demonstrate the efficacy of the method in different discretization contexts.

1.1 Notation

While the ideas and the formulation readily apply to a wide variety of PDE problems, *SEEM* will be illustrated by means of second order boundary value problems of type

$$\begin{cases} \mathcal{A}u = f & \text{in } \Omega, \\ \mathcal{B}u = g & \text{on } \Gamma = \partial\Omega, \end{cases} \tag{1.1}$$

for an elliptic operator $\mathcal{A}$ such as, e.g., the Laplacian $-\Delta$, and an admissible boundary operator $\mathcal{B}$ such as, e.g., the trace γ_Γ (Dirichlet problem), the unit outer normal derivative ∂_ν (Neumann problem), or a combination thereof (Robin type problem). The full BVP will be denoted by

$$\mathcal{C}u = b, \qquad \text{where } \mathcal{C} = \begin{bmatrix} \mathcal{A} \\ \mathcal{B} \end{bmatrix} \qquad \text{and } b = \begin{bmatrix} f \\ g \end{bmatrix}.$$

In this paper, we will consider the PDE operator acting on the H^p spaces. Throughout, we will use the H^p norm given by

$$\|u\|_{H^p(\Omega)} = \left(\int_\Omega |(1-\Delta)^{p/2} u|^2 dx \right)^{1/2}.$$

We also define the negative norms

$$\|u\|_{H^{-p}(\Omega)} = \sup_{\|v\|_{H_0^p(\Omega)}=1} \langle u, v \rangle.$$

We set $s = 1/2$ for the Dirichlet problem and $s = 3/2$ for the Robin problem. Acting on these spaces, we cite the well-known result from the theory of elliptic PDEs that

$$\mathcal{C} : H^p(\Omega) \to H^{p-2}(\Omega) \times H^{p-s}(\Gamma) \tag{1.2}$$

is a bounded linear operator with bounded inverse, so long as $p > s + \epsilon$. In the Neumann case, it is necessary to consider

$$\mathcal{C} \ : H^p(\Omega)/\mathcal{K} \to \mathcal{R},$$

where $\mathcal{K}$ is the set of constant functions on Ω and

$$\mathcal{R} = \{(f, g) \in H^{p-2}(\Omega) \times H^{p-3/2}(\Gamma) \ \Big| \ \int_\Omega f dx = \int_\Gamma g dS\}.$$

We will denote the (bounded) inverse of $\mathcal{C}$ as $\mathcal{C}^{-1}$.

In the spirit of fictitious domain methods, the domain $\overline{\Omega}$ is embedded into a simple (square or rectangular) "container" domain $\mathbb{B}$, for which $\overline{\Omega} \subset \mathbb{B}$. Denote by $\mathbb{B}^m$ a regular uniform discretization of $\mathbb{B}$ consisting of N_m points, where m is the number of discretization points along each dimension. (A finite element triangulation or adaptive finite difference grid could also be used, see Section 10.2.) Ω is then discretized as

$$\Omega^m = \Omega \cap \mathbb{B}^m \qquad \text{and } N_m^\Omega = |\Omega^m|.$$

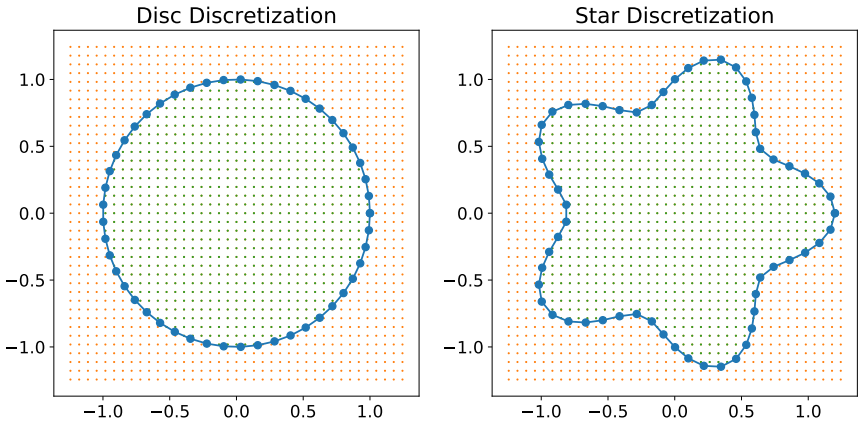

Figure 1.1: Discretizing the domain and its boundary.

The boundary is discretized by taking a set of points

$$\Gamma^m = \{y_j | j = 1, \ldots, N_m^\Gamma\} \subset \Gamma.$$

See Figure 1.1 for two examples of such discretizations.

1.2 Description of the Method

Replace the continuous differential operator by a discrete counterpart $A = A^m$, defined as a discrete evaluation of $\mathcal{A}$ at grid-points Ω^m, where A^m acts on "discrete functions" defined on $\mathbb{B}^m$. Similarly, given the discretization Γ^m, it is possible to discretize the boundary condition using any kind of interpolation and any kind of discrete differentiation (where needed) based on the grid $\mathbb{B}^m$ and to obtain a corresponding discrete equation $Bu = B^m u^m = g^m$ for the unknown vector $u^m : \mathbb{B}^m \to \mathbb{R}$ and a discretization $g^m : \Gamma^m \to \mathbb{R}$ of the boundary function g, defined on Γ^m. In this way, the continuous boundary value problem (1.1) can be replaced

by the discrete underdetermined system given by

$$Cu = C_m u^m = \begin{bmatrix} A \\ B \end{bmatrix} u = \begin{bmatrix} A^m \\ B^m \end{bmatrix} u^m = \begin{bmatrix} f^m \\ g^m \end{bmatrix} = b^m = b \tag{1.3}$$

where f^m is a discretization of f at grid points in $\mathbb{B}^m \cap \Omega$. Observe that we shall often suppress the superscripts and the subscripts in order to simplify the notation. Notice that

$$u^m \in \mathbb{R}^{N_m},\ f^m \in \mathbb{R}^{N_m^\Omega},\ \text{and}\ g^m \in \mathbb{R}^{N_m^\Gamma}.$$

In numerical experiments, the dimensions are always chosen in such a way that $N_m^\Omega + N_m^\Gamma < N_m$ is satisfied. In addition, care is taken when placing the boundary points Γ^m to make sure that all equations in the system are independent of each other, to ensure the problem remains well posed. We emphasize that the operators A^m and B^m can be constructed using any form of interpolation and discrete differentiation on the regular grid. For example, finite differences or spectral differentiation could both be used.

To deal with the fact that the system is underdetermined, a common fictitious domain approach (see Chatper 2) consists in extending the original PDE to the entire larger domain $\mathbb{B}$. Unfortunately, beyond the obvious difficulty of finding a smooth extension for the given data, such an extension of the problem will usually introduce a singularity along the boundary Γ preventing the resulting solution from attaining a high order of convergence. In contrast to these existing methods, we don't try to extend or modify the problem to or in the encompassing domain $\mathbb{B}$/grid $\mathbb{B}^m$, but rather simply try and find "the best" among the solutions of the underdetermined problem (1.3). After all, if you use high order $\mathbb{B}^m$-based discretizations of derivatives and evaluations, the equations should be sufficient to determine a solution that achieves their order of accuracy (up to what is allowed by the regularity of the data/solution themselves, of course).

A straightforward approach (which works fine when no regularity at all is expected) consists in finding a minimal norm solution of the problem, i.e. in solving the linearly constrained optimization problem

$$\operatorname{argmin}_{\{Cu=b\}} \frac{1}{2}\|u\|_2^2, \tag{1.4}$$

where $\|\cdot\|_2$ denotes the Euclidean norm on $\mathbb{R}^{N_m}$. This would lead to the so-called normal equations and to the solution

$$u = C^\top \big(CC^\top\big)^{-1} b. \tag{1.5}$$

Given that the matrix $C = C_m$ consists of differential operators including the evaluation (restriction) in the domain Ω^m and on the boundary Γ^m, its transpose then corresponds to differential operators containing trivial extensions (read extensions by 0) and this leads to oscillations generated by the lack of regularity, see Figure 1.2. As a matter of fact, the solution of the continuous optimization problem

$$\operatorname{argmin}_{\{-\Delta_\Omega u=f, \gamma_\Gamma u=g\}} \|u\|^2_{\mathrm{L}^2(\mathbb{B})}$$

is simply given by the extension by zero $\mathrm{ext}_0(u_{f,g})$ of the unique solution $u_{f,g}$ of the boundary value problem $-\Delta u = f$ with the given Dirichlet boundary condition. Even in the case $g \equiv 0$, however, the solution generated by the optimization problem takes the form of a difference of singular solutions which do not even belong to L^2. This is the origin of the oscillations that are observed in numerical implementations. In a one dimensional context a detailed explanation of this phenomenon is offered in the next remark.

Remark 1.1. *We illustrate this point more thoroughly with a one-dimensional example. We*

reformulate the boundary value problem

$$\begin{cases} -\partial_{xx} u = f \quad \text{in } (-1,1), \\ u(\pm 1) = 0, \end{cases}$$

as the optimization problem (1.4)*, which we approach by introducing a Lagrange multiplier* $\lambda = \big(\lambda_{(-1,1)}, \lambda_{-1}, \lambda_1\big)$ *and reducing it to*

$$\operatorname{argmin}_{v,\lambda} \Big\{ \frac{1}{2} \int_{-\pi}^{\pi} v^2(x)\, dx + \int_{-1}^{1} \lambda_{(-1,1)}(x) \big[\partial_{xx} v + f\big](x)\, dx + \lambda_{-1} v(-1) + \lambda_1 v(1) \Big\},$$

where $v : (-\pi, \pi) \to \mathbb{R}$ *is a periodic function. Taking a variation with respect to* v *yields the validity of*

$$\int_{-\pi}^{\pi} v\varphi\, dx = \int \lambda_{(-1,1)}(x) \partial_{xx} \varphi(x)\, dx - \lambda_{-1}\varphi(-1) + \lambda_1 \varphi(1),$$

for any and all periodic testfunctions $\varphi \in \mathrm{C}^\infty_\pi$*. Taking testfunctions satisfying* $\operatorname{supp}(\varphi) \subset (-1,1)^{\mathsf{c}}$*, one shows that* $v = 0$ *in* $(-1,1)^{\mathsf{c}}$*. Choosing testfunctions supported in* $(-1,1)$ *shows that* $\partial_{xx} \lambda_{(-1,1)} = v$ *in* $(-1,1)$*, and, finally, choosing testfunctions with* $\varphi(\pm 1) = 0$ *and others with* $\partial_x \varphi(\pm 1) = 0$*, one obtains the validity of*

$$\lambda_{(-1,1)}(\pm 1) = 0 \text{ and } \lambda_{\pm 1} = \pm \lambda'_{(-1,1)}(\pm 1).$$

One can therefore solve for $\lambda_{(-1,1)}$ *to see that* $\lambda_{(-1,1)} = S_D u$*, where* S_D *denotes the solution operator to* $-\partial_{xx}$ *with homogeneous Dirichlet conditions at the end points* ± 1*, and then determine* $\lambda_{\pm 1}$*. As it holds that*

$$\lambda = (CC^\top)^{-1} \begin{bmatrix} f \\ 0 \end{bmatrix} \text{ and } v = C^\top \lambda,$$

it follows that

$$v = \begin{bmatrix} -\partial_{xx} \circ \mathrm{ext}_0 & \delta_{-1} & \delta_1 \end{bmatrix} \begin{bmatrix} \lambda_{(-1,1)} \\ \lambda_{-1} \\ \lambda_1 \end{bmatrix}$$
$$= -\partial_{xx}\big(\mathrm{ext}_0(\lambda_{(-1,1)})\big) + \lambda'_{(-1,1)}(-1)\delta_{-1} - \lambda'_{(-1,1)}(1)\delta_1 = \mathrm{ext}_0(u),$$

where ext_0 *denotes the trivial extension (i.e. by zero) of a function defined on* $(-1,1)$ *to* $(-\pi,\pi)$*. The easily verified fact that the dual of the trace at a point is the Dirac distribution at the point, i.e. that* $\gamma'_{\pm 1} = \delta_{\pm 1}$*, was used to derive the above representation. Thus the solution* v *of the optimization problem is the trivial extension of the solution* u *of the original boundary value problem but it is obtained as the sum of singular terms with cancellation. In a discretization, the singular terms are generically not supported on grid points and thus appear in the numerical solution as oscillations. This is made apparent in Figure 1.2. Moreover, the exact analytical cancellation cannot be expected to also happen at the discrete level in general. While we used a differentiated notation for* u *and* v *in this argument for clarity of exposition, the same will not be done in the sequel, so that the same notation will be used for the solution of the original problem and that of the optimization problem.*

Reverting to the general discussion, we observe that, while the solution obtained by the normal equation (1.5) exhibits oscillations in a discrete computation, "the good" (regular and non-oscillatory) solution is, however, among those of the underdetermined problem (1.3). It can be obtained by requiring additional regularity. As already pointed out, the discretizations A^m and B^m are, after all, chosen to be of a desired accuracy and the truncations/trivial extensions destroy it. Thus enforcing an appropriate degree of regularity should allow for the recovery of the intrinsic accuracy of the chosen discretizations, again, compatibly with the expected regularity of the solution itself. We refer to the proposed method as the *Smooth Extension Embedding Method* (*SEEM*) since it implicitly selects a smooth extension of the

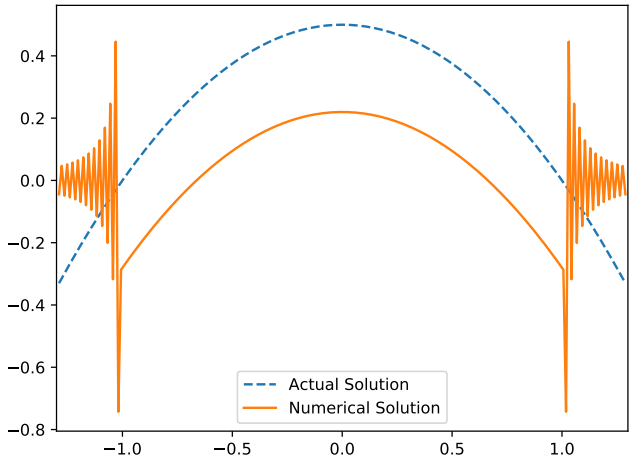

Figure 1.2: A 1D visualization of the oscillations caused by trivial extension with no regularization. The plot shows a region that is only slightly larger than Ω since the oscillations occur in a neighborhood of $\partial\Omega$.

solution. While this selection is done in a way that is natural from the point of view of optimization [7, Chapter 10], it has a nice analytic interpretation which will greatly help with the practical implementation of the method. Let $\|\cdot\|_S$ be the discretization of a high order norm such as, for instance, $\|\cdot\|_{\mathcal{S}} = \|(1-\Delta_\pi)^{p/2}\cdot\|_2$, where $-\Delta_\pi$ denotes the periodic Laplacian on $[-\pi,\pi)^d$ and $p \geq 1$. Now the problem becomes

$$\operatorname{argmin}_{\{Cu=b\}} \frac{1}{2}\|u\|_S^2, \tag{1.6}$$

where the indeces have again been dropped for ease of reading. The constrained optimization problem (1.6) can be reformulated as the unconstrained minimization

$$\operatorname{argmin}_{u\in\mathbb{R}^{N_m},\,\Lambda\in\mathbb{R}^{N_\Lambda}} \frac{1}{2}\|u\|_S^2 + \Lambda^\top\big(Cu-b\big),$$

upon introduction of Lagrange multipliers $\Lambda \in \mathbb{R}^{N_\Lambda}$, where $N_\Lambda = N_m^\Omega + N_m^\Gamma$. A direct

computation yields the saddle points system

$$\begin{bmatrix} S^*S & C^\top \\ C & 0 \end{bmatrix} \begin{bmatrix} u \\ \Lambda \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix}, \tag{1.7}$$

where S^*S is the (invertible) operator corresponding to the norm $\|\cdot\|_S$. Many techniques exist to solve such systems, see [3]. The simplest of these consists in forming the Schur complement, $C(S^*S)^{-1}C^\top$, and in then obtaining the regularized normal equation

$$u = (S^*S)^{-1}C^\top \big(C(S^*S)^{-1}C^\top\big)^{-1} b, \tag{1.8}$$

Now, recalling that C and $C^\top$ are truncated differential operators (more precisely containing differentiations, evaluations on subdomains, and extensions), we see that the effect of the norm is to replace the operator $C^\top$, which, upon being hit by C is the cause of the oscillations in the straightforward method, by the regularized operator $(S^*S)^{-1}C^\top$, which can be captured numerically to a higher degree of accuracy (no oscillations) when hit by C.

Remark 1.2. *To illustrate the effect of regularization in the one dimensional setting of Remark 1.1, consider the minimizer v corresponding to the objective functional given by the higher order expression $\frac{1}{2}\big[\|v\|_2^2 + \|\partial_x v\|_2^2\big]$. Proceeding in a similar way as in Remark 1.1, v is seen to be given by*

$$(1-\partial_{xx})_\pi^{-1}\Big\{ -\partial_{xx} \circ \mathrm{ext}_0\big[S_D(u+f)\big] + \sum_{j=-1,1} j\big[\big(\partial_x S_D(u+f)\big)(j) - u_x(j)\big]\delta_j \Big\}$$

The singular terms are now regularized. Observe that the index in the regularizer indicates inversion of the operator in the periodc sense and that S_D was defined in Remark 1.1.

Remark 1.3. *Formula* (1.8) *can be used as a starting point without any knowledge of a norm generating the operator S. One can choose any convenient smoothing kernel acting on (generalized) functions defined on the box $\mathbb{B}$ instead of $(S^*S)^{-1}$.*

Remark 1.4. *We emphasize that* SEEM *can easily be implemented in any container domain which admits:*

- *Efficient and well-developed discretizations of the differential operators and of the interpolation operators.*
- *An efficient smoothing operator S which enforces the regularity of functions on the grid.*

We will discuss various choices of discretization in Chapter 4.

Chapter 2

Fictitious Domain Perspective

As we mentioned, in a fictitious domain method, the original problem on an irregular domain Ω is embedded into a new problem on a larger regular domain $\mathbb{B}$. If the fictitious domain method is successful, the computed solution is then an extension of the actual solution on Ω. The philosophy of fictitious domain methods is to utilize simple and efficient discretizations which exist for simple domains for more complex domains as well. Example of such domains are the periodic torus, for which Fourier series can be used, the nonperiodic box with a regular grid for which finite difference methods can be used, or the sphere, for which spectral methods can be used. There are several advantages of using a larger, simpler domain.

1. Mesh generation can be an expensive and difficult process. Avoiding it is often desirable.

2. The discretizations on the simple domains are often very simple to construct and implement. In particular, it is straightforward to construct high order discretizations on a regular grid, while it can be tricky to do so on general meshes.

3. The implementations may be very efficient, as in the case of using the FFT to implement the discrete Fourier transform.

4. The simple domains are very well understood from a theoretical perspective. For example, the eigenfunctions of the Laplace operator on the torus are just trigonometric functions. This may simplify the analysis of the problem and provide some extra tools, such as the ability to construct a basis of divergence-free vector fields (see Section 8.6).

Fictitious domain type methods were first introduced in the 1960s by [21], and a wide variety of strategies have been introduced to impose the PDE defined on the smaller domain on the full regular grid. Fictitious domain methods must solve three challenges in order to produce an accurate extension of the solution to the boundary value problem.

1. First, the fictitious domain method must find a way to impose the boundary condition. In general, the boundary does not line up with the regular grid. Thus, the boundary conditions cannot just be incorporated into the description of the problem, as they are in mesh-based methods. Instead, a method to impose them on the regular grid must be found.

2. Inherent in a fictitious domain formulation is that the problem is underdetermined on the larger fictitious domain. For a given u defined on Ω, there are many extensions of u to the fictitious domain $\mathbb{B}$. A fictitious domain method must have a way of specifying the correct solution from the set of extensions.

3. Depending on how an extension is chosen, the resulting solution will often have low regularity across the boundary. The accuracy of the discrete solution will be restricted by the regularity of the computed solution across the boundary. Thus, fictitious domain methods often have very low orders of convergence. With a poor choice of extension, the solution will fail to converge even in the L^2 sense, see Figure 1.2 for an example of the oscillations resulting from a poor choice of extension.

A wide variety of methods have been introduced to deal with these issues. Of the most direct

relevance to *SEEM* is the "Lagrange multiplier approach," which we now describe.

2.1 Lagrange Multiplier Approach

The Lagrange multiplier (LM) approach was developed by Glowinski et al. in [14, 13]. The method is also known as the immersed boundary direct forcing method in the immersed boundary community, see [23]. We consider a standard Dirichlet problem

$$\begin{aligned} -\Delta u &= f \text{ on } \Omega \\ u &= g \text{ on } \Gamma. \end{aligned}$$

In the LM formulation, the problem and its data f are extended to the entire fictitious domain $\mathbb{B}$. We note that this Dirichlet problem can also be viewed as a minimization problem on Ω.

$$\operatorname{argmin}_{\substack{u\in H^1(\Omega)\\ \gamma_\Gamma u=g}} \int_\Omega \left(\frac{1}{2}|Du|^2 - uf\right) dx.$$

A natural option for a fictitious domain method is to simply extend the same problem to the full domain $\mathbb{B}$. We let $\tilde{f}$ be any extension of f to $\mathbb{B}$. We then obtain the new minimization problem

$$\operatorname{argmin}_{\substack{u\in H^1_\pi(\mathbb{B})\\ \gamma_\Gamma u=g}} \int_{\mathbb{B}} \left(\frac{1}{2}|Du|^2 - \tilde{f}u\right) dx. \tag{2.1}$$

Remark 2.1. *Of course, for a general boundary value problem, the coefficients would also need to be extended. The extension should be chosen with C^1 smoothness to guarantee sufficient regularity of the resulting solution.*

The boundary conditions cannot be imposed directly, because the regular grid does not line up with the boundary. As a result, a Lagrange multiplier formulation is used. Equation 2.1

is equivalent to minimization problem

$$\operatorname{argmin}_{\substack{u\in H^1_{\tilde{g}}(\mathbb{B}) \\ \Lambda\in L^2(\Gamma)}}\Big(\frac{1}{2}|Du|^2 - \tilde{f}u dx + \int_\Gamma \big[\Lambda_\Gamma\ (\gamma_\Gamma u - g)\big]\, dy\Big).$$

This results in the saddle point problem

$$\begin{aligned} -\Delta u + \gamma_\Gamma^* \Lambda &= \tilde{f} \\ \gamma_\Gamma u &= g. \end{aligned} \tag{2.2}$$

As per the discussion in Chapter 6, it can be shown that the saddle point problem is well posed and that it admits an $H^{3/2-\epsilon}$ solution which satisfies $u\big|_\Omega \equiv u_S$. In fact, $u\big|_{\mathbb{B}\setminus\Omega}$ is given by the solution to the Dirichlet problem

$$\begin{aligned} -\Delta u &= \tilde{f} \text{ on } \mathbb{B}\setminus\Omega \\ u &= g \text{ on } \Gamma. \end{aligned}$$

However, in general, $\frac{\partial}{\partial\nu}\left(u\big|_\Omega\right) \neq \frac{\partial}{\partial\nu}\left(u\big|_{\mathbb{B}\setminus\Omega}\right)$, so in general u will only be $H^{3/2-\epsilon}$. (Here ν is the outward facing normal vector to Γ.) As we will discuss, this will greatly limit the order of convergence of the solution. (In general, if a function is of regularity class H^p, an interpolant will be able to achieve at most p-th order of convergence, without an interpolation scheme which takes the nature of the singularities in account.)

Remark 2.2. *Enforcing boundary conditions with Lagrange multipliers has also been considered in the finite element method, see [1].*

The saddle point problem (2.2) is then discretized, using a regular discretization on the encompassing fictitious domain $\mathbb{B}$. Many different discretization procedures exists. The chief difficulty is that the regular grid does not align with the boundary grid, so care must be taken when constructing the space $L^2(\Gamma)$. We again emphasize that even with careful choice

of discretization, the convergence order is inherently limited because of the lack of regularity of the extension of the continuous operators. However, a poor choice of discretization can destroy even this small order of convergence.

In [14], a regular finite element discretization is used on $\mathbb{B}$ while a space of piecewise constant functions is used to discretize the space of Lagrange multiplier on the boundary (whence the name "distributed Lagrange multiplier method"). In [13], it is showed that such a discretization, with three to four regular grid points between each boundary segment, satisfies a uniform inf-sup condition, and thus obtains a convergence order of $h^{1/2}$. In the direct forcing immersed boundary, the problem is discretized by placing discretized deltas at points around the boundary. Our results of Section 7.2 will imply a convergence of order 1, and, for a general direct forcing immmersed boundary method, this is the observed convergence. Other implementations exist where the Lagrange multipliers on the boundary are replaced by smoother functions located off the boundary in the fictitious domain. This is a way of preserving smoothness of the solution.

Remark 2.3. *We note that in [14], Glowinski observed a superconvergence of order* 2 *of the* ℓ^2 *error. This superconvergence is still unexplained, given that the continuous solution is globally only* $H^{3/2-\epsilon}$*. Furthermore, this superconvergence is only obtained with Glowinski's specific choices of discretization; with the simpler discretizations of the immersed boundary direct forcing method, only the predicted first order convergence is observed.*

Drawbacks

We now discuss the drawbacks of the Lagrange multiplier formulation.

- The method requires that the original elliptic operator $\mathcal{A}$ and/or right-hand-side f be extended to corresponding objects defined on the whole of $\mathbb{B}$. This is not always straighforward and simple minded extensions (like the trivial one by zero outside Ω)

introduce singularities into the problem reducing the overall accuracy of the method. See [6] regarding methods of creating smooth extensions from Ω to $\mathbb{B}$ for the purpose of implementing fictitious domain methods.

- As we have mentioned a number of times, because the extension is not smooth, only low orders of convergence can be obtained. Even the optimal second order convergece of [13] is only obtained by a careful finite element discretization of the boundary, which sacrifices some of the simplicity of the method. Furthermore, this superconvergence result is still not explained in the literature.

SEEM

SEEM builds on the fictitious domain formulation. The significant change is that rather than extending the original problem to the full domain, it instead views the full BVP as a constraint to an optimization problem on the whole domain. In other words, Equation (2.2) is replaced by

$$\begin{cases} \mathcal{S}_p u + \mathcal{A}^\top \Lambda_\Omega + \mathcal{B}^\top \Lambda_\Gamma & = 0 \\ \mathcal{A} u & = f \\ \mathcal{B} u & = g. \end{cases}$$

The optimization operator $\mathcal{S}_p$ can be chosen to be straightforward to solve and to impose the desired degree of regularity on the solution. This resolves both drawbacks of the existing fictitious domain methods. First, the BVP problem does not need to be extended; the existing problem is merely given as a constraint to a well chosen minimization problem. Second, the method can be made to achieve arbitrarily high orders of convergence by imposing the desired order of regularity on the solution. From a philosophical perspective, *SEEM* makes apparent that the real problem that any fictitious domain methods has to solve is

the selection problem among the infinitely many solutions of the original problem which are generated as the problem is viewed in a larger domain where it becomes under-determined. The direct way in which *SEEM* accomplishes this (introduction of a high order smoother) clearly shows how the order of accuracy chosen for the interior and boundary operators can be recovered in the extended problem through an affine shift obtained by a natural (both from the point of view of PDEs and of optimization) regularization.

2.1.1 Other Smooth Extension Methods

We have found two other examples in the literature which obtain high order fictitious domain methods by selecting an extension of higher regularity. We describe both of these methods below.

IBSE

In *Immersed Boundary Smooth Extension* (*IBSE*), [23], a smooth extension is generated by solving an auxilary PDE problem on Ω^c.

$$
\begin{aligned}
&(1-\Delta)^p \xi = 0 \text{ on } \Omega^c \\
&\frac{\partial^i}{\partial \nu^i}\xi = \frac{\partial^i}{\partial \nu^i} u \text{ on } \Gamma.
\end{aligned}
$$

In other words, the function ξ solves a $2p$ order PDE on Ω^c and its $1 \leq i \leq p-1$ derivatives match u on Γ. Solving for such an extension will produce a globally H^p solution. (In fact, the continuous formulation of the *IBSE* method produces the same extension as *SEEM*.) However, rather than obtaining this extension using a minimization procedure, as in *SEEM*, both u and ξ are solved for using the Lagrange multiplier fictitious domain method; a coupled system of u and ξ is then created and solved. The principal drawback of this method is that

the linear algebra of the resulting coupled system is very complex and difficult to solve. In [23], the authors resort to forming the explicit matrix for the system by performing the linear operators column by column and doing an LU factorization of the resulting explicit matrix. This procedure takes significant preparation time - approximately fifty minutes for a grid of size 2048^2 - and which is avoided in our method. However, this method is philosophically quite close to our method, and many of the issues dealt with in this thesis (e.g. numerical conditioning and rates of convergence) are discussed in [23] in a similar manner. In [24], the *IBSE* method is applied to divergence free problems as well.

Active Penalty Method

In the *Active Penalty Method*, [22], a smooth extension $\xi : \mathbb{B}\backslash\Omega \to \mathbb{R}$ for the solution u is constructed as follows. One begins by selecting linearly independent smooth functions

$$f_j(x) : \mathbb{R}_{\geq 0} \to \mathbb{R}, \quad 0 \leq j \leq p$$

which satisfy

$$\frac{d^i}{dx^i}\Big|_{x\to 0^+} f_j(x) = \delta_{ij}.$$

Given a function $u : \mathbb{R}_{\leq 0} \to \mathbb{R}$, a C^{p-1} extension to $\mathbb{R}$ can be constructed by defining

$$\tilde{u}(x) = \begin{cases} u(x) & \text{if } x \leq 0 \\ \sum_{i=0}^{p} a_i f_i(x) & \text{if } x > 0, \end{cases}$$

where $a_i = \frac{d^i}{dx^i}\Big|_{x\to 0^-} u(x)$. The a_i are chosen so that the normal derivatives up to order p match across the pont $x = 0$.

Returning to the BVP on Ω, one takes a smooth tubular neighborhood Γ_ϵ of $\Gamma = \partial\Omega$ of

radius $\epsilon > 0$, and uses the coordinates

$$\{(x, \xi) \, | x \in \Gamma \text{ and } \xi \in [-\epsilon, \epsilon]\}$$

A smooth extension of u to Ω is then obtained in the form

$$\tilde{u}(x, \xi) = \begin{cases} u(x, \xi) \text{ if } \epsilon \leq 0 \\ \sum_{i=0}^{p} a_i(x) f_i(\xi) \text{ if } \epsilon > 0, \end{cases}$$

where

$$a_i(x) = \frac{\partial^i}{\partial_\xi^i}\Big|_{y=x,\, \xi \to 0^-} u(y, \xi).$$

Using this basis, an extension of u is constructed which matches the normal derivative of u up to order p on Γ. A discrete system is formed which enforces that u has such a form and thereby imposes H^p regularity on the solution. The drawback of this method is that this extension is quite difficult to construct for an arbitrary domain Ω. Furthermore, sparsity of the matrix is lost if the support of the basis functions is held constant.

We conclude by noting that there are many other fictitious domain methods with varying strengths and weaknesses. We have only discussed those that are most relevant to *SEEM*.

Chapter 3

Kernel-Based Collocation Perspective

Meshfree methods using radial basis functions are a widely used method for scattered data interpolation and for solving PDEs. The survey books [9, 27] describe both the theory and the implementation of these methods. We give a short overview of some of the theory and application of kernel-based collocation methods and then describe how *SEEM* fits into the general framework of these methods.

3.1 Definitions

A symmetric, positive definite kernel $\mathcal{K}$ on a domain Ω is a function

$$\mathcal{K} : \Omega \times \Omega \to \mathbb{R}$$

which satisfies

(i) $\mathcal{K}(x, y) = \mathcal{K}(y, x),\ x, y \in \Omega.$

(ii) $\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j \mathcal{K}(x_i, x_j) \geq 0$ for any set of distinct points $\{x_1, \ldots, x_n\} \subset \Omega$ and for any

$c \in \mathbb{R}^n$. Equality holds if and only if $c = 0$.

To each positive definite kernel $\mathcal{K}$ there corresponds a unique reproducing kernel Hilbert space. The latter is a Hilbert space $\mathcal{N}_\mathcal{K}$ of functions defined in Ω which satisfies

(i) $\mathcal{K}(\cdot, x) \in \mathcal{N}_\mathcal{K}$ for $x \in \Omega$.

(ii) $f(x) = \langle f, \mathcal{K}(\cdot, x) \rangle$ for $f \in \mathcal{N}_\mathcal{K}$.

Two distinct forms of collocation methods exists, symmetric and nonsymmetric.

3.1.1 Nonsymmetric Collocation

In nonsymmetric collocation, two sets of points are chosen, centers $\{x_1, \ldots, x_n\}$ and collocation points $\{z_1, \ldots, z_m\}$, which are scattered on the interior and the boundary. A function is then sought of the form

$$u = \sum_{j=1}^{n} \alpha_j \mathcal{K}(\cdot, x_j),$$

which satisfies the PDE at the collocation points

$$\sum_{j=1}^{n} \alpha_j \mathcal{C}^{z_i} \mathcal{K}(z_i, x_j) = b_i \text{ for } 1 \leq i \leq m.$$

Here, b_i is the right hand side of the BVP operator $\mathcal{C}$ evaluated at the point z_i. The coefficient vector α is obtained by inverting (at least in the least squares sense) the collocation matrix given by

$$\mathcal{M}_{ij} = \mathcal{C}^{z_i} \mathcal{K}(z_i, x_j).$$

Note that in general, the problem can be overdetermined or underdetermined. Depending on the choice of kernel and the location of the centers and collocation points, the method may

or may not be well posed and the solution may or may not converge. Although the theory is not well understood, the method has achieved some success in the applied community due to its simple implementation and to successful numerical results

3.1.2 Symmetric Collocation

In symmetric kernel-based collocation, only one set of collocation points $\{z_1, \ldots, z_m\}$ are chosen. These point are scattered around the interior and the boundary. A solution is then sought in the form

$$\tilde{u}(\cdot) = \sum_{j=1}^{m} \alpha_j \mathcal{C}^{z_j} \mathcal{K}(\cdot, z_j).$$

The coefficients α_j are chosen so that $\mathcal{C}(\tilde{u})(z_i) = b_i$ for $1 \leq i \leq m$. Here b_i is the right hand side of the BVP operator $\mathcal{C}$ evaluated at the point z_i. In other words,

$$\sum_{j=1}^{m} \alpha_j \mathcal{C}^{z_i} \mathcal{C}^{z_j} \mathcal{K}(z_i, z_j) = b_i.$$

Thus, in order to obtain the unknown coefficient vector α, one solves the linear system $\mathcal{M}\alpha = b$, where the *collocation matrix* $\mathcal{M}$ is given by

$$\mathcal{M}_{ij} = \mathcal{C}^{z_i} \mathcal{C}^{z_j} \mathcal{K}(z_i, z_j). \tag{3.1}$$

As shown in [27, Chapter 13], the solution $\tilde{u}$ obtained through this process is the $\|\cdot\|_{\mathcal{N}_\mathcal{K}}$-minimizing function satisfying $\mathcal{C}(\tilde{u})(z_i) = b_i$ for $1 \leq i \leq m$.

3.2 Meshfree Implementations

In meshfree collocation methods, the kernel $\mathcal{K}$ is generally chosen to be of the form

$$\mathcal{K}(x, y) = \Phi(|x - y|) = \Phi(r),$$

where $\Phi : \mathbb{R}^+ \to \mathbb{R}$ is a positive definite function, known as the radial basis functions of the method. Many choices are available including

(i) Gaussians, where $\Phi(r) = e^{-cr^2}$ for some $c > 0$.

(ii) Multiquadrics, where $\Phi(r) = \sqrt{1 + cr^2}$ for $c > 0$.

These functions are often chosen as kernels because it is easy to perform computations on them. The corresponding collocation matrix $\mathcal{M}$ is computed explicitly by evaluating

$$\mathcal{M}_{ij} = \mathcal{C}^{z_i}\Phi(|z_i - z_j|),$$

in the case of nonsymmetric collocation and

$$\mathcal{M}_{ij} = \mathcal{C}^{z_i}\mathcal{C}^{z_j}\Phi(|z_i - z_j|),$$

in the case of symmetric collocation methods. By using smooth, globally supported kernels, high rates of convergence are observed, in line with the results in Section 7.2. One of the advantages of meshfree methods is that the collocation points can be chosen arbitrarily. This freedom is particularly useful when only scattered data is available. Furthermore, the simplicity of the formulation of the method is very attractive.

Drawbacks

There are several significant challenges which arise when one tries to use meshfree collocation on dense sets of collocation points.

- The most significant problem is that when using globally supported radial basis functions, the resulting collocation matrix is dense. For smooth problems this is not an issue, because the high order of convergence can achieve high accuracy with very few collocation points. For less smooth problems, however, where dense grids are necessary to resolve the solution's behavior, it becomes impractical to use such dense matrices.
- Additionally, when the radial basis function is smooth, the collocation matrix is very poorly conditioned, thus severely limiting the size of the set of collocation points, see [9, Chapter 16].
- Finally, for more complex differential operators, it can be difficult to form the collocation matrix. This is particularly true when a weak formulation of the problem is required due to the lack of regularity of the solution. Because of this difficulty, meshfree methods have rarely even been formulated for problems of weak regularity.

To deal with the first two of these issues, a number of successful strategies have been introduced aimed at speeding up the inversion of the collocation matrix and at reducing its condition number. Many of these techniques are described in [9]. Given the large variety of meshfree implementations, we will only briefly address those that are most pertinent to the proposed method.

a. Globally supported radial basis functions can be replaced by the compactly supported ones as introduced by Wendland in [26]. In this way, the collocation matrix becomes sparser and more collocation points can be used. Unfortunately, convergence only occurs if the width of the compactly supported functions is held constant, see [9, Chapters 41]. Consequently,

the matrix loses its good sparsity properties as the mesh becomes finer. To remedy this latter problem, multilevel schemes are used. In these, compactly supported radial basis functions with varying supports are used. Those with wide support capture the coarse details, while those with narrow support capture the fine details. The use of such multilevel methods can improve the accuracy obtained from compactly supported radial basis functions; however, convergence issues still remain, see [9, Chapter 41].

b. A number of techniques exist which seek to circumvent the issue of the ill conditioning of the collocation matrix by finding clever ways to compute the interpolant. These include the Contour-Padé Algorithm ([9, Chapter 17]) and the RBF-QR Method ([11]). We note that these methods approximate the radial basis functions using the truncation of a series expansion. *SEEM*, in contrast, is discrete from the onset.

c. Particularly relevant to us is the NFFT (nonuniform fast Fourier transform) method ([9, Chapter 28]). In the NFFT Method, the collocation matrix $\mathcal{M}$ is evaluated by using the inverse non-uniform FFT to obtain a Fourier series for the function on the torus. The kernel, a convolution operator, is then evaluated as a multiplication operator on the torus. Finally, the non-uniform FFT is applied to obtain the function values at the collocation points. By using this method, the dense collocation matrix can be evaluated with $O(N \log N)$ operations rather than with $O(N^2)$. When combined with efficient preconditioning, this method allows for the use of substantially larger grids. We will discuss the relationship of this method with *SEEM* below.

d. In RBF partition of unity methods (RBF-PUM), see [9, Chapter 29], or RBF finite difference methods (RBF-FD), see [10], the global RBF method is localized to create sparse rather than dense matrices. In RBF-PUM, the global basis functions are multiplied by cut-off functions generated by a partition of unity of the domain. The resulting basis functions are compactly supported in a small region of the domain, and therefore generate a sparse matrix. In RBF-FD, a small set of RBF basis functions is chosen at each point. The PDE operator is evaluated at each point using that point's chosen RBF functions to generate

a high order finite difference stencil at each pont. These methods are quite competitive and are rapidly growing in popularity. However, in localizing the problem, these methods lose the straightforward representation of the function as a linear combination of RBFs. Furthermore, the interpretation of the problem as a constrained optimization problem is lost.

3.3 *SEEM*

We now discuss how *SEEM* fits into the general framework of kernel-based collocation. Similar to symmetric meshfree collocation, *SEEM* starts as an optimization problem

$$\text{argmin}_{\mathcal{C}u=b}\|u\|_{\mathcal{S}}.$$

Here, as we will discuss, the smoothing operator $(\mathcal{S}^*\mathcal{S})^{-1}$ from Equation (1.8) corresponds to the kernel $\mathcal{K}$ in a symmetric collocation method and the smoothing operator $\mathcal{S}^{-1}$ correspoinds to the kernel $\mathcal{K}$ in an unsymmetric collocation method. However, the kernel $\mathcal{S}$ is chosen carefully so that it is straightforward to discretize and invert on a regular grid. For example, the Sobolev kernel $\mathcal{S}_p = (1-\Delta)^{p/2}$ can be easily discretized and inverted on a regular grid using well developed PDE techniques. Then, rather than trying to solve the continuous formulation of the problem, the norm $\|\cdot\|_{\mathcal{S}}$ and the BVP operator $\mathcal{C}$ are discretized, and produce a new discrete optimization problem

$$\text{argmin}_{Cu=b}\,\|u\|_{S}.$$

The resulting Schur complement matrix $C(S^\top S)^{-1}C^\top$ is then a discretization of the symmetric collocation matrix with coefficients $\mathcal{M}_{ij} = \mathcal{C}^{z_i}\mathcal{C}^{z_j}\mathcal{K}(z_i, z_j)$, where

$$\mathcal{K}(z_i, z_j) = \langle \delta_i, (\mathcal{S}^*\mathcal{S})^{-1}\delta_j \rangle.$$

On the other hand, if we consider the pseudoinverse formulation (Section 5.1.1), we see that the matrix CS^{-1} from Equation 5.1 is a discretization of the nonsymmetric collocation matrix with coefficients $\mathcal{M}_{ij} = \mathcal{C}^{z_i}\mathcal{K}(z_i, x_j)$, where $\{x_1, \ldots, x_{N_m}\}$ is the regular grid $\mathbb{B}^m$. Of course, this matrix is underdetermined so the problem needs to be satisfied in the least squares sense. From this perspective, *SEEM* is a nonsymmetric collocation. Namely, in this case the symmetric and nonsymmetric collocation methods coincide - one with a kernel $(\mathcal{S}^*\mathcal{S})^{-1}$ and one with a kernel $\mathcal{S}^{-1}$. Thus, *SEEM* can be viewed simultaneously as unsymmetric kernel-based collocation and symmetric kernel-based collocation.

Contrast with Meshfree Collocation Methods

In general, meshfree collocation methods evaluate the smoothing operator $\mathcal{S}^{-1}$ as well as the differential operator $\mathcal{C}$ explicitly. In *SEEM*, by contrast, these are evaluated at the discerete level. By choosing the smoothing kernel to be defined on a finite regular encompassing domain, it is possible to obtain a straightforward and efficient discretization of the operators. The obvious drawbacks are that the method is no longer meshfree and that the minimum distance between collocation points is limited by the distance of the grid points. However, the clear benefit is that the operators can be easily described using sparse matrices (either through the use of finite differences, of the FFT, or of other sparse schemes). Furthermore, the use of regular grid points in the interior allows for simple preconditioning strategies, which substantially speeds up computations. We also mention the following other benefits of *SEEM* over common global RBF methods.

(i) Since all operators are discretized, other techniques for the solution of saddle point systems can be used. In general, these alternative methods allow one to solve the system while avoiding the direct computation of the dense matrix S^{-1}. The availability of such tools should allow for the choice of denser grids. Another useful tool that becomes viable is QR-factorization, the use of which is described in Section 5.1.1. This technique makes it possible to replace the symmetric collocation method by an equivalent nonsymmetric collocation technique. This results in an improvement of the conditioning of the matrices and additionally provides a link between the methods of symmetric and unsymmetric collocation.

(ii) Since all operators are described on a regular grid, their evaluation is very simple. For instance, one can use the standard finite difference stencil and cubic interpolation operators at all points. By contrast, in meshfree methods, the values $\mathcal{C}^{z_i}\mathcal{C}^{z_j}\Phi(z_i, z_j)$ can sometimes be complicated to compute, particularly for less straightforward differential operators where many terms need to be evaluated. This issue is discussed in [9, Chapter 40], for instance.

(iii) Since the kernel matrix is discretized, it can be easily modified and tailored to fit specific problems. For example, in certain singular problems it may be beneficial to use weighted Sobolev norms. While such kernels would be quite difficult to compute explicitly, they can be easily evaluated on a regular grid in the discrete sense. Another possible modification would be to use nonquadratic objective functionals. Such modifications are a subject of current investigation.

While RBF-PUM and RBF-FD are RBF type methods allowing for the use of sparse matrices, *SEEM* preserves the global nature of the pure RBF approach. This has advantages from a theoretical standpoint. Furthermore, we believe that the global formulation will have applications to a number of problems where an explicit representation of the basis functions is useful, e.g. divergence-free problems discussed in Section 8.6. Additionally, forming the

RBF-FD and RBF-PUM matrices can be a complex process. In *SEEM*, only straightforward spectral or finite difference discretizations need to be considered.

Remark 3.1. *When using a Fourier discretization,* SEEM *is implemented by imposing H^p regularity of the numerical solution on the torus. The smoothing operator is evaluated using the FFT. This allows for the evaluation of all operators in $O(N \log N)$ operations. This implementation is akin to that of the NFFT methods, see [9, Chapter 28], in that both use the FFT to evaluate the collocation matrix $\mathcal{M}$. Indeed, using NFFT methods and efficient preconditioning, it is possible to use meshfree methods on dense grids with $O(N \log N)$ operations. However, our method distinguishes itself from these existing NFFT methods in several important ways. First, because all values are first interpolated to the regular grid, the regular IFFT can be used, rather than the INFFT (inverse nonuniform fast Fourier transform), which is more computationally complex. Second, rather than evaluating RBFs as convolution operators, we use the simple Sobolev kernels, which are more natural for the torus. As pointed out above, this allows for the use of various other computational tools from the theory of saddle point problems.*

Remark 3.2. SEEM *is reminiscent of some of the techniques used in the computation of the INFFT, see [16, Chapter 5]. Given scattered data $\{u(x_i)\}_{i=1}^k$ on the torus, the INFFT seeks a Fourier series $\hat{u}$ which agrees with the data at the given points. As the problem is generally underdetermined, a Fourier series of minimal H^p-norm is computed instead. The Fourier implementation in this thesis (Section 4.1.1 and Chapter 8) can be seen as a generalization of the INFFT algorithm from the simple case of point evaluations of u to the case of general linear functionals.*

Part II

Numerical Implementation

Chapter 4

Discretization

We now give a detailed description of the implementation of *SEEM*. As described in Section 1.2 and Equation (1.4), the original BVP on Ω is rewritten as the optimization problem

$$\operatorname{argmin}_{u\in\mathbb{R}^{N_m},\,\Lambda\in\mathbb{R}^{N_\Lambda}} \frac{1}{2}\|u\|_S^2 + \Lambda^\top\big(Cu-b\big),$$

on an encompassing domain $\mathbb{B}$, where $\|\cdot\|_S$ is a regularizing norm. As discussed in Section 1.2, this leads to the saddle point system

$$\begin{bmatrix} S^\top S & C^\top \\ C & 0 \end{bmatrix} \begin{bmatrix} u \\ \Lambda \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix}.$$

We begin by describing different choices of grid and regularizing norm S. We then explain how to discretize the domain Ω and its boundary Γ. Finally, we discuss how to assemble the matrices C, $C^\top$ and S.

4.1 Choice of Discretization and Regularizing Norm S

We recall that a good choice of discretization $\mathbb{B}^m$ and discrete regularizing norm S requires several conditions to be satisfied.

- The grid $\mathbb{B}^m$ must admit straightforward and efficient discretizations of differential operators and interpolation operators.
- The smoothing operator $(S^\top S)^{-1}$ must map a function space of lower regularity to one of higher regularity. Generally, the smoother will map the Sobolev spaces $H^{-p}(\mathbb{B})$ to $H^p(\mathbb{B})$ and will be formed by inverting a (pseudo-)differential operator of order p, e.g. $(1-\Delta)^{p/2}$.
- The discretized smoother S (and $S^\top$) must be easily invertible on the regular grid. This will generally be efficiently accomplished using a multilevel algorithm, such as the FFT or multigrid methods.

We now discuss several options of discretization and norm which can be used to impose the H^p regularity of the solution. However, we emphasize that the smoothing operators considered here are by no means exhaustive; other choices exist which are the subject of ongoing work.

4.1.1 The Periodic Torus using the FFT

We let $\mathbb{B}^m$ be a uniform discretization of the d-dimensional torus $[-\pi,\pi)^d$ and introduce the regularizing norm

$$\begin{aligned}\|u\|_{\mathcal{S}_p} &= \|(1-\Delta_\pi)^{p/2}u\|_{L^2} \\ &= \|\mathcal{F}^{-1}\operatorname{diag}\big[(1+|k|^2)^{p/2}\big]\mathcal{F}(u)\|_{L_2},\end{aligned}$$

where Δ_π is the periodic Laplacian and $k \in \mathbb{Z}^d$ is the vector of frequencies on $\mathbb{B}$. Becuase $\mathcal{S}_p$ is diagonalized by the Fourier transform, this choice of smoother leads to simple and efficient computation. In particular, we let $k \in \mathbb{Z}_m^d$ be the vector of frequencies on the discrete grid $\mathbb{B}^m$. Then, the discrete norm is given by

$$\|u_m\|_{S_p} = \left\| \left(\mathcal{F}^m\right)^{-1} \operatorname{diag}\left[(1+|k|^2)^{p/2}\right] \mathcal{F}^m u_m \right\|_{\ell_2}.$$

Using the fast Fourier transform, this operator can be inverted efficiently with minimal memory requirements.

Because we are calculating a periodic extension to the solution, the interior differential operator $\mathcal{A}$ can be efficiently discretized using the FFT. In particular, $\partial_{x_i} u$ can be discretized using $(\mathcal{F}^m)^{-1} \operatorname{diag}\left[\sqrt{-1} k_{x_i}\right] \mathcal{F}^m(u)$. To discretize the boundary interpolation operators, two options can be used. Spectral interpolation can achieve high order accuracy (provided that the smoother is high enough order) but requires a large stencil which increases memory consumption and slows down computations. Alternatively, polynomial interpolation can be used, where a polynomial scheme of the desired order of convergence is chosen.

Advantages

- Implementing the smoother and differntial operators using the FFT is particularly simple and efficient.
- Using the FFT, it is straightforward to invert the pseudo-differential operator S_p for any order p. In particular, fractional orders can be used, and computational complexity is independent of the chosen smoother. Thus, the smoother can be perfectly tailored to the regularity of the solution. In fact, one can use an exponential decaying smoother

$$\mathcal{F}^{-1} \operatorname{diag}\left[e^{-|k|}\right] \mathcal{F}$$

and obtain a spectral order of convergence.

- Because our solution is expressed as a Fourier series, it is straightforward to construct divergence free or curl free bases. With other discretizations, imposing such a condition requires special choices of discretization.

Disadvantages

- Because the generated smooth extension of the BVP is periodic, only a small fraction of the periodic domain can be included in Ω. The rest is needed as a buffer in order to allow the extension to smoothly morph into a periodic function. This wastes significant computational resources, because the extended problem is solved on a far larger grid than strictly necessary for the solution of the oringal BVP. Put differently, the ratio $\frac{|\Omega^m|}{|\mathbb{B}^m|}$ is much smaller than the geometry of Ω actually requires.

- A second problem is that a 2π-periodic extension of u has far larger optimization norm than u itself. The derivatives outside Ω are required to be large to allow for the extension to become periodic. This negatively affects the accuracy of the discretized solution. Table 4.1 below shows how, in the periodic setting, higher order Sobolev norms of the extension grow even if the smoother used is chosen to control these higher order norms. This is reflected in the growth along the columns of the periodic section of the table. Clearly a norm of the extension is expected to grow if the optimization norm used is of lower order, explaining the growth along the rows of the table.

- Because we rely heavily on the use of the FFT, it is not possible to have any adaptivity; only a uniform grid can be used. This also wastes computational resources when the solution has different orders of regularity in different regions of the domain Ω.

4.1.2 The Chebyshev Grid

To mitigate for the disadvantages of using a periodic extension, the Fourier grid can be replaced with a Chebyshev one. When using a Chebyshev grid, no boundary conditions are imposed, so that a smaller buffer is required and the higher order seminorms do not get too large. In Table 4.1, we demonstrate this point by comparing the growth of the seminorms generated by a Fourier extension versus those generated by a Chebyshev extension.

The encompassing domain $\mathbb{B} = [-1, 1]^d$ is discretized by a product set of the one-dimensional Chebyshev grid $\mathfrak{C}^m$ and is given by

$$\mathbb{B}^m = \Big\{ (x^1, \ldots, x^d) \,\Big|\, x^i \in \mathfrak{C}^m \text{ for } 1 \leq i \leq d \Big\},$$

where

$$\mathfrak{C}^m = \Big\{ \cos\big(\pi \frac{2k+1}{2m}\big) \,\Big|\, 0 \leq k \leq m-1 \Big\}.$$

Remark 4.1. *The grid described above is frequently referred to as the Chebyshev roots grid or Chebyshev points of the first kind. An alternative choice is the Chebyshev extrema grid, also known as Chebyshev points of the second kind. Similar rates of convergence are observed with these points. However, in our numerical experiments, the roots grid appears to possess higher numerically stability. Furthermore, the regularizer S_p^{-1} described below is only symmetric for the Chebyshev roots grid, which makes it more convenient for the use in combination with iterative solvers. Henceforth, the Chebyshev grid will refer to the Chebyshev roots grid.*

Regularizing Norm

We begin by defining the operator

$$\mathcal{D} := \sqrt{1-x^2}\frac{\partial}{\partial x}.$$

An important property of the Chebyshev polynomials is that

$$(1-\mathcal{D}^2)^{p/2}T_m = (1+m^2)^{p/2}T_m$$

holds true for the m-th Chebyshev polynomial T_m. Exploiting this, we define the norm

$$\|\cdot\|_{\mathcal{S}_p}^2 = \|(1-\sum_{i=1}^{d}\mathcal{D}_i^2)^{p/2}\cdot\|_{L_2}^2.$$

Clearly, away from the degeneracies at -1 and 1, this norm imposes H^p regularity on the function u. In addition, due to the eigenvalue equation, the operator $\mathcal{S}_p$ is diagonalized by the Chebyshev transform. In particular, if we denote the latter by $\mathfrak{C}$ and let $(k)_{k\in\mathbb{N}^d}$ be the (Chebyshev) frequency vector, we have that

$$\mathcal{S}_p u = \mathfrak{C}^{-1}\,\mathrm{diag}\left[(1+|k|^2)^{-p/2}\right]\mathfrak{C}u.$$

Thus, $\mathcal{S}_p$ can be discretized using the discrete Chebyshev transform $\mathfrak{C}^m$,

$$S_p = (\mathfrak{C}^m)^{-1}\,\mathrm{diag}\left[(1+|k|^2)^{-p/2}\right]\mathfrak{C}^m u,$$

where $k\in\mathbb{N}_m^d$ is the vector of Chebyshev frequencies on the discrete grid $\mathbb{B}^m$. As the discrete Chebyshev transform can be implemented using the FFT, such a regularizing norm allows for efficient implementation. The numerical experiments of Chapter 9 demonstrate that, as in the Fourier case, using the $\mathcal{S}_p$ norm leads to a p rate of convergence for the error.

Remark 4.2. *In the numerical experiments, the observed rate of convergence for the $\mathcal{S}_p$ smoother is somewhat faster than the expected rate p, which was observed in the Fourier case. We suspect that this may have to do with the higher density of points near the boundary of the domain Ω, due to the non-regular spacing of the Chebyshev grid.*

Discretization of the Differential and Interpolation Operators

As with the Fourier discretization, the differential and interpolation operators may be implemented using either spectral discretizations or finite difference discretizations. For details of how to take derivatives spectrally using $\mathfrak{C}^m$, the discrete Chebyshev transform, and how to form interpolation operators using Chebyshev series, we refer to Section 9.1. As before, the advantage of using spectral interpolation is that it achieves very high orders of accuracy when coupled with a high order smoother. The disadvantage is that the large stencil consumes a lot of memory and hurts efficiency.

Advantages

- Although the implementation of the differential operators is slightly more complicated than in the Fourier case, the implementation is still straightforward and efficient using the FFT. It has the advantage over the Fourier case that only a small buffer is required outside Ω, because the chosen extension does not need to be periodic.
- As with the Fourier discretization, pseudodifferential operators of arbitrary order can be used and tailored to the regularity of the solution.

Disadvantages

- Compared to the other methods presented here, the discretizations of the derivatives are somewhat more complicated. Additionally, because of the nature of the Chebyshev points, the density of the points is highest on the boundary, which lies outside Ω.
- As with the Fourier discretization, the method does not allow for any adaptivity in the grid.

Remark 4.3. *While we used spectral discretizations based on Fourier series and Chebyshev polynomial expansions,* SEEM *can also be implemented with respect to any other spectral basis. It is enough to embed Ω into a larger domain $\mathbb{B}$ for which a full spectral resolution is known for some canonical self-adjoint and positive definite differential operator $\mathcal{D}$ with compact resolvent. If the operator admits natural discretizations $\mathbb{B}^m$ for the domain $\mathbb{B}$, $\{\psi_i\}_{i=1}^m$ for its (orthonormal) eigenfunctions, which are also orthonormal for the appropriate discrete quadrature rule, and satisfy*

$$D_m \psi_i^m = \lambda_i^2 \psi_i^m,$$

for the eigenvalues $\lambda_\bullet^2$ of $\mathcal{D}$, then a good smoothing norm given by

$$\| \cdot \|_{S_p} = \| \mathfrak{C}_m^{-1} \operatorname{diag} \left[(1 + \lambda_\bullet^2)^{p/2} \right] \mathfrak{C}_m \cdot \|_{L_2},$$

where $\mathfrak{C}_m$ is the discrete transformation which computes the coefficients of the (discrete and finite) eigenfunction expansion and $\lambda_\bullet^2$ is the corresponding vector of eigenvalues. The advantage of using a Fourier or Chebyshev basis is that they allow the use of the FFT for efficient discretization of the smoothing operator. Another area where spectral discretizations would be very useful is the implementation of SEEM *on a domain Ω which embeds in a sphere. In this case, a basis of spherical harmonics would be used.*

	Chebyshev Extension			Fourier Extension		
Smoother	$\|\nabla^2 u\|_{L_2}$	$\|\nabla^3 u\|_{L_2}$	$\|\nabla^4 u\|_{L_2}$	$\|\nabla^2 u\|_{L_2}$	$\|\nabla^3 u\|_{L_2}$	$\|\nabla^4 u\|_{L_2}$
S_2	16.0956	14.5451	319.2180	95.4606	172.2919	488.2472
S_3	16.0111	0.5137	2.3840	137.6518	177.6702	281.2872
S_4	16.0002	.0064	.0410	153.0669	184.4029	278.7028

Table 4.1: Various norms of the extensions obtained by the optimization procedure based on different smoothers. The smoothers were used to extend the function $u = x^2 - y^2$.

4.1.3 Finite Difference Grids

As a way of introducing some adaptivity, we turn to the use of finite difference discretizations. We again work in a periodic box $[-\pi, \pi)^d$ with uniform discretization and consider the regularizing norm

$$\|u\|_{\mathcal{S}_p} = \|(1 - \Delta_\pi)^{p/2} u\|_{L_2},$$

Here, p is chosen to be an even integer. In order to discretize the norm, we do not use the Fourier transform, but rather resort to finite differences. Let D^2 be any finite difference discretization of the Laplacian, e.g. the five point stencil and define the discrete norm

$$\|u\|_{\mathcal{S}_p} = \|(I - D^2)^{p/2} u\|_{\ell_2}.$$

Similarly, all differential operators are discretized using finite difference schemes and interpolation operators are implemented using polynomial interpolation. While not quite as efficient as the FFT, the matrices S_p and $S_p^\top$ can be inverted efficiently using sparse iterative solvers. Particularly efficient are multigrid solvers with computational complexity $O(N_m)$. When using a finite difference discretization we can use nested grids rather than a uniform grid. In particular, we need to define a finite difference scheme which captures the Laplacian to the desired order on a nested grid. For a concrete example of such a scheme, we refer to Section 10.2. The resulting smoothing operators, S_p^{-1} and $(S_p^\top)^{-1}$, can still be applied efficiently

using multigrid methods.

Advantages

- Allows for adaptive refinement of the mesh. Saves significant computational resources when the solution is less well behaved in some regions of Ω.
- The norm $\|(1 - D^2)^{p/2} \cdot \|_{\ell_2}$ can easily be replaced with a more complex norm. In particular, if $D^2_{x_i}$ is the second derivative in the x_i direction, one could use

  $$\|u\|_{S_p} = \|(1 - \sum_{i=1}^{d} a_i(x) D^2_{x_i})u\|_{\ell_2},$$

  for any functions $a_i(x)$ defined on the finite difference grid.

  In this way, one can construct various weighted Sobolev norms. While research in this direction is ongoing, we expect this to have advantages when the actual solution lies in a weighted Sobolev space.

Disadvantages

- The use of multigrid solvers on nested grids is less straightforward and efficient to implement than the FFT on a regular grid.
- The choice of p for the regularizer is much more constrained. First, we require $p \in 2\mathbb{N}$ in order for the norm to be defined. Furthermore, higher order norms will become more difficult to invert, both from an efficiency perspective and from a numerical stability perspective. In our numerical examples, we will only study the case $p = 2$ when using the finite difference method.

Remark 4.4. *While we used a finite difference discretization, a finite element discretization could also be used. This might allow for more flexibility in the construction of the norm.*

However, the simple structure of nested grids is replaced by the significantly more complicated process of mesh refinement, and the simplicity of the method is hence compromised.

Remark 4.5. *Although we discussed the process with periodic boundary conditions, homogenous Dirichlet or Neumann conditions could be chosen and imposed just as easily. Other boundary conditions could also be considered, with the aim of minimizing the higher order seminorms of the extension as much as possible.*

4.2 Choice of p for the Smoother S_p

We recall that the purpose of the smoother is to enforce the H^p regularity of the selected solution to the underdetermined problem. Thus, when p is chosen to be larger, a more regular approximate solution is produced and the true solution can be approximated with a higher order of convergence. In particular, if $u \in H^{p+2}$ is the solution, then, using the smoother S_p, will result in convergence of order p for the L_2-error when imposing Dirichlet conditions and order $p-1$ convergence for Neumann or Robin boundary conditions. This is demonstrated experimentally in Section 8.1. Thus, it is advantageous to use p which gives the optimal order of convergence for the regularity class of the true solution u. For example, if the true solution $u \in H^6$, one should use the S_4 smoother to obtain 4th order convergence of the L_2 error for a Dirichlet problem. Although the order of convergence increases with p, using a higher order smoother greatly increases the condition number of the resulting matrices. In particular, the matrix S_p has a condition number which grows like m^p (before preconditioning), where m is the number of grid points along each direction. Thus, using a larger p, generally requires more iterations for convergence on an equivalent grid. Furthermore, for a given grid size, the order p of the smoother cannot be pushed too high without hitting the limits of numerical precision. By way of example, we consider the Fourier smoother given by $(1-\Delta_\pi)^{-p/2}$. In Fourier space, this corresponds to a multiplication by the function $(1+|k|^2)^{-p/2}$. If k_* is the

largest mode, as soon as $|k_*|^{-p}$ drops below machine precision, which is roughly $1e-16$, some matrix entries can no longer be captured numerically and the benefits of accuracy are lost. For example, on a grid of size 64^2, the highest order smoother which can be used is $p = 10$. Thus, the smoother must be chosen to balance the greater numerical accuracy obtained with the numerical issues which arise as p is increased. The latter are discussed more extensively in Section 5.2.

Remark 4.6. *The order of convergence of the solution is constrained by the regularity of the true solution, the order of the smoother, the interpolation operators, and the differential operators. In order to avoid wasting computational resources, the order of accuracy of the various discretizations should be made to match.*

4.3 Discretization of the Domain

When discretizing the domain Ω and its boundary Γ, two choices need to be made. First, the size buffer $\mathbb{B}\backslash\Omega$ needs to be determined. Clearly, a sufficient number of points need to be left outside of Ω so that any finite difference operators on the interior of Ω or any interpolation operators on Γ can be evaluated. Additionally, the buffer needs to be sufficiently large for the extension to satisfy the boundary conditions on $\partial\mathbb{B}$ without exceedingly increasing the size of the seminorms. When Ω is a disc and periodicity is imposed on $\mathbb{B}$, we observed that the accuracy of the solution begins to decrease when the radius of the disk becomes larger than 2.75. We again note that with a Chebyshev or finite difference grid, the need for a large buffer can be avoided.

Next the distribution of the points Γ^m on the boundary needs to be addressed. In practice, it is best for these points to be uniformly distributed over the boundary. In two dimensions, this can be accomplished easily by equally spacing points along an arc length parametrization of the curve. In three dimensions, equally distributing the points around a surface is

more challenging, although straightforward algorithms exist to obtain roughly equally spaced points on a given surface, for example, see [19].

Remark 4.7. *When using the Chebyshev discretization, a slightly better boundary discretization, particularly well adaptedto the density of the Chebyshev grid, can be obtained as follows. If the boundary* Γ *is a hypersurface contained in* $\mathbb{B}$ *and parametrized by*

$$\Big(\Gamma_1(\boldsymbol{z}), \ldots, \Gamma_d(\boldsymbol{z})\Big),\ \boldsymbol{z} \in S^{d-1},$$

where S^{d-1} *is the* $d-1$*-dimensional unit sphere, we can create an even distribution of points*

$$\{\widetilde{y}^{\,i}\}_{i=1}^{N_m^\Gamma} = \big\{(\widetilde{y}^{\,i}_1, \ldots, \widetilde{y}^{\,i}_d) \,\big|\, i = 1, \ldots, N_m^\Gamma\big\}$$

along

$$\widetilde{\Gamma} := \Big(\arccos(\Gamma_1(\boldsymbol{z})), \ldots, \arccos(\Gamma_d(\boldsymbol{z}))\Big),$$

now a hypersurface of $[0,\pi)^d$*. We note that applying the* arccos *function componentwise to the points of* $\mathbb{B}^m$ *leaves one with a regular grid on* $[0,\pi)^d$*; thus, setting*

$$\Gamma^m = \big\{(y^i_1, \ldots, y^i_d)\big\}_{i=1}^{N_m^\Gamma} = \big\{(\cos(\widetilde{y}^{\,i}_1), \ldots, \cos(\widetilde{y}^{\,i}_d))\big\}_{i=1}^{N_m^\Gamma}$$

produces a boundary discretization of Γ *the density of which is proportional to the density of the Chebyshev points in* $\mathbb{B}$*. In the two dimensional numerical experiments, this method was used to discretize the boundary.*

A choice also needs to be made concerning the density of boundary points, that is, the value of N_m^Γ. When using an insufficient number of points on the boundary, the accuracy suffers, while too many points can drive up the condition number, in accordance with the theoretical discussion in Section 7.1. In practice, we found a density of $\frac{m}{4\pi}$ boundary points per unit length, where m is the number of grid points along one dimension, to be most effective in

N_m	N_Ω	N_Γ	N_m	N_Ω	N_Γ
16^2	81	17	256^2	20,865	257
32^2	325	33	512^2	83,421	513
64^2	1,305	65	1024^2	333,669	1,025
128^2	5,209	129	2048^2	1,335,029	2,049

Table 4.2: Number of collocation points used in the Fourier discretization of the disc of radius 2 used in the experiment in Section 8.3. N_m is the size of the full computational grid. N_Ω is number of interior collocation points, and N_Γ is the number of boundary collocation points.

two dimensions. This guarantees that one to two regular grid points lie between any two boundary points and thereby allows the regular grid $\mathbb{B}^m$ to easily "distinguish" the different boundary points and to keep the condition number relatively low. In Figure 1.1, we show the discretization of the unit disc and of a star shaped domain. For better visualization, we have only plotted $[-1.3, 1.3]^2$, as opposed to the entire computational domain $[-\pi, \pi)^2$. In three dimensional problems, we have found that with a grid of size m^3 points, a boundary spacing of $\frac{m^2}{2\pi^2}$ per unit area is most effective. In Table 4.2 we list the number of interior and boundary points used in the experiments in Section 8.3.

4.4 Discretization of the PDE operator

We recall that

$$C = \begin{bmatrix} A \\ B \end{bmatrix}$$

where A is a discretization of the interior differential operator $\mathcal{A}$ and B is a discretization of the boundary operator $\mathcal{B}$. As we explained in the introduction, the advantage of using a fictitious domain method is that these discretizations can be carried out in a straightforward and efficient way on a regular grid. We will discuss how to form each of these discretizations.

4.4.1 Discretization of Interior PDE $\mathcal{A}$

Recall that $\mathcal{A}$ is a second order differential operator of the form

$$\mathcal{A}u(x) = \sum_{i,j=1}^{d} a_{ij}(x)\partial_i\partial_j u + \sum_{i=1}^{d} b_i(x)\partial_i u + c(x)u.$$

To discretize $\mathcal{A}$, we will need to introduce the restriction operator

$$R_\Omega : \mathbb{R}^{N_m} \to \mathbb{R}^{N_m^\Omega},$$

mapping a function defined on $\mathbb{B}^m$ to its values at the points of Ω^m. Denoting the discretization of the derivative ∂_i on $\mathbb{B}^m$ by D_i, we let $\bar{u}$ be the vector containing the values of the function u on $\mathbb{B}^m$ and $\bar{a}_{ij}$, $\bar{b}_i$ and $\bar{c}$ be the vectors containing the values of the functions a_{ij}, b_i, and c, respectively, at the points of Ω^m. The discretization of $\mathcal{A}$ is then given by

$$A\bar{u} = \sum_{i,j=1}^{d} \bar{a}_{ij} R_\Omega D_i D_j \bar{u} + \bar{b} R_\Omega D_i \bar{u} + \bar{c} R_\Omega \bar{u}.$$

Similarly,

$$A^\top \bar{w} = D_j^\top D_i^\top R_\Omega^\top \bar{a}_{ij} \bar{w} + D_i^\top R_\Omega^\top \bar{b} \bar{w} + R_\Omega^\top \bar{c} \bar{w}.$$

Here, $R_\Omega^\top$ is the transpose of R_Ω and amounts to an extension by 0 from Ω^m to $\mathbb{B}^m$. $\bar{w}$ is a vector containing values on Ω^m.

As discussed at length in Section 4.1, the derivative matrices D_i can be obtained using any straightforward method on the background grid $\mathbb{B}^m$. In our implementations, we used spectral differentiation for the Fourier and Chebyshev discretization, and finite difference discretizations for the nested grids of Section 10.2.

4.4.2 Discretization of the Boundary Operator $\mathcal{B}$

Recall that $\mathcal{B}$ is a boundary operator on Γ given by

$$\mathcal{B}u = a(y)\gamma_\Gamma u + b(y)\partial_\nu u.$$

B is a discretization of $\mathcal{B}$ on the set $\Gamma^m \subset \Gamma$. Since the boundary points Γ^m do not lie on the regular grid, interpolation operators are needed when imposing (discrete) boundary conditions. We let

$$I_{y_i} : \mathbb{B}^m \to \mathbb{R}$$

be an interpolant which estimates a grid function at y_i. As before, we denote by $\bar{a}$ and by $\bar{b}$ be the vector of values of the coefficients a and b at the points of Γ^m. Finally, we let $\bar{\nu}$ be the vector of values of ν at the points Γ^m. The i-th row of B is then given by

$$B_{i\bullet} = \bar{a}_i I_{y_i} + \bar{b}_i \sum_{j=1}^{d} (\bar{\nu}_i)_j I_{y_i} D_j.$$

As mentioned in Section 4.1, we have the option of using spectral or polynomial interopolation operators. In practice, spectral interpolants can only be used on sparse grids, when the number of boundary points is small because of the added memory and computational costs. Thus, in our experiments we have used spectral interpolants when using high order regularizers on sparse grids, and cubic interpolation when using lower order regularizers on dense grids. We note that the computational complexity of the operator B is $O(N_m^\Gamma)$, when using polynomial interpolation.

Chapter 5

Linear Algebra

We now take a closer look at the saddle point system

$$\begin{bmatrix} S^\top S & C^\top \\ C & 0 \end{bmatrix} \begin{bmatrix} u \\ \Lambda \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix},$$

at the core of *SEEM*. This kind of system arise in many areas of computational mathematics, particularly in optimization. Many techniques exist for its solution, see [3]; we investigate several of them below. As the upper left block of the matrix factors in terms of S, additional more specific techniques are also available. Notice that the solution u satisfies two different equations.

- **The Pseudoinverse Formulation.**

$$u = S^{-1}(CS^{-1})^{+}b, \tag{5.1}$$

where $(CS^{-1})^{+}$ is the Moore-Penrose pseudoinverse of CS^{-1}.

- **The Schur Complement Formulation.**

$$u = (S^\top S)^{-1} C^\top (C(S^\top S)^{-1} C^\top)^{-1} b,$$

where $C(S^\top S)^{-1} C^\top$ is the Schur complement matrix.

The problem can be solved directly or iteratively using either formulation. As described in Chapter 3, the pseudoinverse formulation originates from the treatment of the problem as an unsymmetric collocation method, while the Schur complement formulation appears when treating the problem as a symmetric collocation method. As a general rule, methods which avoid implementing the full Schur complement matrix are more stable because they avoid the numerical instability stemming from the calculation of $(S^\top S)^{-1}$ and only require the indipendent computation of S^{-1} and $(S^\top)^{-1}$. The condition number of S^{-1} is only the square root of that of $(S^\top S)^{-1}$.

Occasionally, one may wish to replace the $(S^\top S)^{-1}$ matrix of the Schur complement with another smoothing matrix which cannot be factored. For example, one might want to use $(1-\Delta)^3$ in place of $S^\top S$. If one is using a finite difference discretization, it is not possible to exploit factorization and one would need to invert the full $(1-\Delta)^3$. In such a case, the pseudoinverse formulation would not be available.

5.1 Methods for Solving the System

5.1.1 Direct Methods

We first describe a method to invert the matrices directly, rather than using iterative methods. In this case, it is clearly advantageous to use the pseudoinverse method because of its

numerical stability. We seek to solve

$$u = S^{-1}(CS^{-1})^{+}b.$$

Consider the QR decomposition

$$(S^{\top})^{-1}C^{\top} = QR,$$

where, $Q \in \mathbb{R}^{N_m \times (N_\Omega + N_\Gamma)}$ is orthogonal and satisfyies $Q^T Q = I$, while $R \in \mathbb{R}^{(N_\Omega + N_\Gamma)\times(N_\Omega + N_\Gamma)}$ is upper triangular. Recalling that the pseudoinverse of CS^{-1} is given by $Q(R^{\top})^{-1}$, we see that

$$u = S^{-1}Q(R^{\top})^{-1}.$$

Owing to the fact that the QR decomposition only involves the matrix CS^{-1}, this method is the most numerically stable of the available methods. Thus, on sparse grids and with high order regularizers, where numerical stability becomes an issue, it is the method of choice. Unfortunately, however, it cannot be used on dense grids where iterative methods must be used. We note that an SVD decomposition could also be used to calculate the pseudoinverse. In the numerical experiments of Sections 8.1, 8.2, 8.6, and 9.3, the pseudoinverse method is used to demonstrate how the order of convergence relates to the order of the smoother. Using the QR method, we are able to use the $p = 10$ smoother on sparse grids.

5.1.2 Iterative Methods

Next we address the iterative inversion problem on dense grids. A straightforward method consists of inverting the Schur complement matrix directly. Because it is a symmetric positive definite matrix, it can be inverted by means of the conjugate gradient algorithm. As mentioned previously, all operators can be efficiently evaluated using either the FFT or mutligrid methods. The Schur complement matrix can therefore be applied implicitly very

efficiently. Although the matrix S_p is of order $2p$ and is therefore ill-conditioned for large grids, natural preconditioning exists which allows the Preconditioned Conjugate Gradient method (PCG) to converge quickly; see Section 5.2. It follows that the PCG-Schur complement method can be used to efficiently compute the solution on very dense grids. Although this method converges quickly on dense grids, it is not the most numerically stable of the iterative algorithms. Iterative algorithms exist which leverage the fact that the regularizer is of the form $(S^\top S)^{-1}$, to which the pseudoinverse formulation applies. Although these algorithms ultimately are variants of the CG method, they attain somewhat better stability properties by avoiding the direct computation of $(S^\top S)^{-1}$. Two such methods are the LSQR algorithm (see [18]) and the CGNE (Craig's) algorithm (see [20, Chapter 8]). These methods find the solution of the pseudoinverse equation and achieve the same efficiency as the PCG method, while limiting the numerical instability. On the other hand, when using the straight PCG algorithm, one can use the symmetric positive definite matrix described in Section 5.2 as a split preconditioner without actually factoring it. In contrast, when using the LSQR or CGNE algorithms, one would need to factor the matrix to obtain a split preconditioner or settle for a less effective right or left preconditioning. Work is ongoing to determine the optimal solver for various grid sizes and smoothers S_p.

5.1.3 Other Iterative Methods

The methods described above assume that we are able to easily invert S and $S^\top$. However, when using a finite difference discretization, a multigrid method will be required to invert these. In this case, it may be more efficient to use an inexact Uzawa method, described in [8]. In the inexact Uzawa method, each iteration only requires that we invert $S^\top S$ approximately. In practice, this would refer to performing several mutigrid iterations, but not fully inverting $S^\top S$. Work is ongoing to find the optimal strategy to solve the system on dense grids with large p.

5.2 Preconditioning for Iterative Methods

5.2.1 The PCG Method

We now discuss the appropriate preconditioner for the PCG algorithm in the Schur complement method. As observed in 5.1.2, the normal matrix $C(S_p^\top S_p)^{-1}C^\top$ is ill-conditioned and requires a good preconditioner to be inverted using the conjugate gradient method. The ill-conditioning stems from the use of the high order operator S_p^{-1} and from the difference in order between the boundary and the interior operators. We think of the operator $C(S_p^\top S_p)^{-1}C^\top$ as a block matrix

$$C(S_p^\top S_p^{-1})C^\top = \begin{bmatrix} A^m(S_p^\top S_p)^{-1}(A^m)^\top & A^m(S_p^\top S_p)^{-1}(B^m)^\top \\ B^m(S_p^\top S_p)^{-1}(A^m)^\top & B^m(S_p^\top S_p)^{-1}(B^m)^\top \end{bmatrix} = \begin{bmatrix} C_1 & C_2 \\ C_2^T & C_3 \end{bmatrix},$$

and notice that $(S_p^\top S_p)^{-1}$ represents an operator of order $-2p$, A^m one of order 2, and C_1 corresponds to an operator of order $4-2p$, whereas C_2 to one of order $2-2p$ and C_3 to one of order $-2p$ (when choosing a discrete boundary operator B^m discretizing a continuous one of order 0). In general, if an operator is of order $-2p$, the condition number of its matrix approximation will grow like a polynomial of degree $2p$ as the grid size increases (for example, on a grid of size m, the largest eigenvalue of the Laplace operator is of size m^2). Thus, the large order combined with the mismatch in scaling causes a very large condition number. We describe a simple preconditioner which works effectively for S_2, S_3, and S_4. The preconditioning consists of finding approximate inverses to the C_1 and C_3 blocks independently. Specifically, we use the preconditioner

$$\widetilde{C}^{-1} = \begin{bmatrix} \widetilde{C}_1^{-1} & 0 \\ 0 & \widetilde{C}_3^{-1} \end{bmatrix},$$

where $\widetilde{C}_1^{-1}$ and $\widetilde{C}_3^{-1}$ are approximate inverses to C_1 and C_3, respectively. The general philosophy consists in preconditioning each of these two operators so that they become order 0.

We begin by describing $\widetilde{C}_1^{-1}$. The matrix C_1 depends on the order of the smoother we have chosen. For S_2, the matrix C_1 is of order 0. Thus, no preconditioning is necessary, and $\widetilde{C}_1^{-1}$ can be taken to be the identity. For S_3 and S_4, the operator C_1 is the discretization of a differential operator of order $4-2p$. We wish to precondition in such a way as to reduce the order of the operator to 0. Thus, we define the preconditioner

$$\widetilde{C}_1^{-1} u = \big(1 - \Delta_\Omega\big)^{\frac{2p-4}{2}} u,$$

where Δ_Ω is the Laplace operator on Ω (applied with Neumann boundary conditions). In order to implement it, we use the domain discretization $\Omega^m = \mathbb{B}^m \cap \Omega$ and a finite difference scheme to discretize the Laplacian on Ω^m. In examples, the five points stencil (seven points in three dimensions) was chosen.

As for an approximate inverse of the C_3 block, we will consider a Dirichlet problem, where the boundary operator $\mathcal{B}$ consists of evaluations on the boundary. The discrete boundary points belonging to Γ^m are denoted by y_i for $1 \leq i \leq N_m^\Gamma$. Recall that

$$B^m : \mathbb{R}^{\mathbb{B}^m} \to \mathbb{R}^{\Gamma^m} \text{ and } S_p : \mathbb{R}^{\mathbb{B}^m} \to \mathbb{R}^{\mathbb{B}^m}.$$

Here the rows of B^m are discretizations of the delta distribution supported at the various boundary points. Thus, $C_3 = B^m (S_p^\top S_p)^{-1} (B^m)^\top$ is a discretization of the collocation matrix

$$\mathcal{M}_{ij} = \mathcal{K}_p(y_i, y_j) = \langle \delta_i, (\mathcal{S}_p^* \mathcal{S}_p)^{-1} \delta_j \rangle.$$

It follows that an excellent preconditioner for the C_3 block is obtained by inverting the

explicit collocation matrix. Although this involves solving a collocation problem, seemingly defeating the whole purpose of discretizing the collocation method as we do in this paper, several crucial points must be recognized. First, the new collocation problem only resides on the boundary. Thus, the number of points is only N_m^Γ, as opposed to the $N_m^\Omega + N_m^\Gamma$ points of the full BVP. This dimensional reduction allows working with far denser grids. Second, because the matrix $\widetilde{C}_3^{-1}$ is only used as a preconditioner, we do not need to actually calculate the inverse of the collocation matrix: a crude approximation or even any pseudodifferential operator of the proper order will do. Finally, as we will discuss shortly with regards to the Robin and Neumann problems, the collocation matrix that is inverted does not depend on the actual differential and boundary operators arising from the problem so that a standard one can be used for all such operators.

In all numerical experiments performed later, we make use of few enough boundary points that it is possible to directly invert the collocation matrix. To build the latter, observe that $(S^\top S)^{-1}$ is given by convolution with a kernel. Thus, it suffices to calculate it at one point and simply shift its "center" to the different collocation points. A discretization of the fundamental solutions of the continuous smoother $(\mathcal{S}_p^* \mathcal{S}_p)^{-1}$, given by

$$h(y) = ((\mathcal{S}_p^* \mathcal{S}_p)^{-1} \delta)(y),$$

is computed on a dense grid. In our experiments we use a grid of size 4096^2 in two dimensions and 512^3 in three dimensions. This solution can be stored in a lookup table. The collocation matrix is then given by

$$\mathcal{M}_{ij} = h(y_i - y_j),$$

where the values are interpolated from those in the lookup table.

For the Neumann and Robin problems, we note that the order of the matrix C_3 is decreased by 2, because C and $C^\top$ both evaluate one derivative on the boundary. Thus, rather than

using the function $(\mathcal{S}_p^*\mathcal{S}_p)^{-1}\delta$ in the collocation matrix, we instead resort to the function $(\mathcal{S}_{p-1}^*\mathcal{S}_{p-1})^{-1}\delta$.

Using the described preconditioning technique, the PCG method converges fast enough for an efficient evaluation on very dense grids. Numerically, we show in Section 8.3 that preconditioning is most effective for the S_2 smoother. As mentioned above, for a second order BVP, the matrix corresponding to the interior equations "is" order 0 with no preconditioning. As p increases, the preconditioning is somewhat less effective. In general, the ill-conditioning can be somewhat improved by increasing the distance between the boundary points. Another approach to improving the condition number consists in replacing the smoother with the parameter dependent $(1 - \epsilon\Delta_\pi)^{-p/2}$, where $0 < \epsilon < 1$. (In fact, ϵ should be taken small enough so that the smallest eigenvalue of S_p^{-1} is greater than machine precision to avoid the issues described in Section 4.2.) This did not prove necessary for the problems studied here. We refer to Table 8.6 for actual condition numbers of the preconditioned matrices, as well as to the numerical experiments for the CPU times of the corresponding computations.

Part III

Theory

Chapter 6

Continuous Theory

Our theoretical discussion will have two components. We first study the continuous minimization problem on which *SEEM* is based, and prove several properties about the continuous minimizer. We then move to discussing the discretized problem and prove that the problem is well posed and show some convergence properties.

6.1 The Continuous Problem

We recall that

$$\mathcal{C} = \begin{pmatrix} \mathcal{A} \\ \mathcal{B} \end{pmatrix}, \qquad \mathcal{C} : H^p(\Omega) \to H^{p-2}(\Omega) \times H^{p-s}(\Gamma)$$

is a second order differential operator. Here $s = \frac{1}{2}$ in the case of the Dirichlet problem, and $s = \frac{3}{2}$ in the case of the Neumann or Robin problem. (In the Neumann problem, we need to take the space of mean 0 functions as the domain and add a compatibility condition to the

codomain.) The original BVP is now expressed as

$$\mathcal{C}\, u = b, \tag{6.1}$$

where $b = [f; g]$. In the *SEEM* formulation, we replace the original boundary value problem with the minimization problem on $\mathbb{B} \supset \bar{\Omega}$ given by

$$\operatorname{argmin}_{\substack{u \in H^p(\mathbb{B}) \\ \widetilde{\mathcal{C}} u = f}} \|u\|^2_{\mathcal{S}_p}.$$

Here $\|\cdot\|_{\mathcal{S}_p} = \|\mathcal{S}_p \cdot\|_{L_2(\mathbb{B})}$ is a norm equivalent to the H^p norm on $\mathbb{B}$ and $\widetilde{\mathcal{C}} = \mathcal{C} \circ \mathcal{R}_\Omega$, where $\mathcal{R}_\Omega$ is the restriction operator to Ω. This optimization problem can be reformulated as an unconstrained problem by introducing a Lagrange multiplier

$$\operatorname{argmin}_{u \in H^p(\mathbb{B})} \|u\|^2_{\mathcal{S}^p} + \langle \Lambda, \widetilde{\mathcal{C}} u - b \rangle,$$

where $\Lambda \in H^{p-2}(\Omega)' \times H^{p-s}(\Gamma)' = H^{p-2}(\Omega)' \times H^{s-p}(\Gamma)$. Upon taking variations in u and Λ, we obtain the saddle point problem

$$\begin{cases} \mathcal{S}_p^* \mathcal{S}_p u + \widetilde{\mathcal{C}}^\top \Lambda & = 0 \\ \widetilde{\mathcal{C}} u & = b. \end{cases}$$

We reformulate the problem by evaluating against a test function $v \in H^p(\Omega)$ and $p \in H^{p-2}(\Omega)' \times H^{s-p}(\Gamma)$. We get the weak formulation

$$\begin{cases} (u, v)_{\mathcal{S}_p} + \langle \widetilde{\mathcal{C}} v, \Lambda \rangle & = 0 \\ \langle \widetilde{\mathcal{C}} u, p \rangle & = \langle b, p \rangle. \end{cases} \tag{6.2}$$

We will assume that the operator $\mathcal{S}_p$ is chosen so that $\mathcal{S}_p^* \mathcal{S}_p$ is a differential operator of order $2p$ (as opposed to only being a pseudodifferential operator). We now introduce a boundary

value problem of order $2p$ on $\mathbb{B}\backslash\Omega$. We let u_Ω be the unique solution of the original BVP within Ω. For $0 \leq i < p$, we let $g_i = \partial_\nu^i u_\Omega$, the i-th normal derivative of u_Ω, where ν is the outward facing unit normal. Then we define the BVP

$$\begin{cases} \mathcal{S}_p^* \mathcal{S}_p u = 0 & \text{on } \mathbb{B}\backslash\Omega \\ \partial_\nu^i u = g_i & \text{on } \Gamma. \end{cases} \tag{6.3}$$

Theorem 6.1. *If Ω is a smooth domain, then the saddle point problem* (6.2) *is well posed. Furthermore, the solution u satisfies the Equation* (6.1) *on Ω and Equation* (6.3) *on $\mathbb{B}\backslash\Omega$.*

Proof. We recall that a saddle point problem is well posed if it satisfies the LBB conditions. We verify each of these conditions here.

1. We let Z be the null space of $\mathcal{C}$. The first condition, ellipticity, is that

   $$\inf_{u\in Z} \sup_{v\in Z} (u, v)_{\mathcal{S}_p} \geq \alpha \|u\|_{H^p(\Omega)} \|v\|_{H^p(\Omega)}.$$

   However, in our construction, we took the $\|\cdot\|_{\mathcal{S}_p}$ norm to be equivalent to the H^p norm, so this inequality is straightforward.

2. The second condition, continuity, is that

   $$\begin{aligned} \forall u, v \in H^p, \quad & (u, v)_{\mathcal{S}_p} \leq c\|u\|\|v\|_{H^p}, \\ \forall v \in H^p, q \in (H^{p-2} \times H^{p-s})', \quad & \langle \widetilde{\mathcal{C}} v, q \rangle \leq c\|v\|_{H^p} \|q\|_{(H^p \times H^{p-s})'}. \end{aligned}$$

   This is true because by assumption $\mathcal{S}_p$ and $\mathcal{C}$ are continuous operators.

3. The inf-sup inequality,

   $$\inf_{q\in(H^{p-2}\times H^{p-s})'} \sup_{v\in H^p(\mathbb{B})} \frac{\langle \widetilde{\mathcal{C}} v, q \rangle}{\|v\|\|q\|} = \beta > 0.$$

We fix $q \in (H^{p-2} \times H^{p-s})'$ and we produce the desired v as follows. We let

$$\mathcal{R} : (H^{p-2} \times H^{p-s})' \to H^{p-2} \times H^{p-s}$$

be the Riesz dual. Because Ω is smooth, there exists

$$\mathcal{E}_{\mathbb{B}} : H^p(\Omega) \to H^p(\mathbb{B}),$$

a bounded operator. We then let

$$v = \mathcal{E}_{\mathbb{B}} \mathcal{C}^{-1} \mathcal{R} q.$$

As each of these operators is bounded, we find that $\|v\|_{H^p} \leq c\|q\|_{(H^{p-2} \times H^{p-s})'}$. We then calculate

$$\begin{aligned}
\langle \widetilde{\mathcal{C}} v, q \rangle &= \langle \mathcal{C} \circ \mathcal{R}_\Omega \circ \mathcal{E}_{\mathbb{B}} \circ \mathcal{C}^{-1} \circ \mathcal{R} q, q \rangle \\
&= \langle \mathcal{R} q, q \rangle \\
&= \|q\|^2 \\
&\leq c^{-1} \|v\| \|q\|.
\end{aligned}$$

The first equality follows from the definition of $\widetilde{\mathcal{C}}$ and v. The second follows from the fact that $\mathcal{R}_\Omega \circ \mathcal{E}_{\mathbb{B}} = \mathrm{Id}$.

We turn to showing the seond part of the theorem. Clearly, given data b, the solution u satisfies Equation (6.1) on Ω and Γ - this is the constraint equation $\mathcal{C}u = b$. We now show that the solution u satisfies Equation (6.3) on $\mathbb{B} \backslash \Omega$. We first evaluate Equation (6.2) against a test function v with $\mathrm{supp}(v) \subset \mathbb{B} \backslash \Omega$. We satisfy

$$(u, v)_{\mathcal{S}_p} + \langle \widetilde{\mathcal{C}} v, q \rangle = 0$$

However, because $\text{supp}(v) \cap \bar{\Omega} = \emptyset$, the second term is 0. As choice of such v is arbitrary, we find that

$$\mathcal{S}_p^* \mathcal{S}_p u = 0 \text{ on } \mathbb{B} \backslash \Omega.$$

Next, we note that because our solution u lies in $H^p(\mathbb{B})$, for $0 \leq i \leq p-1$, $\partial_\nu^i u$ (the normal derivative along Γ) is well defined (in the trace sense). This implies (again in the trace sense) that the normal derivative taken from Ω is equal to that taken from $\mathbb{B}\backslash\Omega$. Thus, $u\big|_{\mathbb{B}\backslash\Omega}$ satisfies Equation (6.3). □

Corollary 6.1. *If $0 < \epsilon < 1/2$ and the data $b \in H^{p+1/2-2-\epsilon}(\Omega) \times H^{p+1/2-\epsilon-s}(\Gamma)$, then the solution u of Equation (6.2) is in $H^{p+1/2-\epsilon}(\mathbb{B})$.*

Remark 6.1. *Corollary 6.1 tells us that although we are only minimizing the H^p norm of the function, if the data is good enough, the solution actually lies in a better space, up to $H^{p+1/2-\epsilon}$. We note that it is impossible to obtain higher regularity than $H^{p+1/2-\epsilon}$ in general because the trace of the p-th normal derivative will be different on Γ from Ω and $\mathbb{B}\backslash\Omega$. However, if $u \in H^{p+1/2}$, the p-th derivative would be well defined on Γ, and this is a contradiction.*

Proof. We note that by elliptic theory, if $b \in H^{p+1/2-\epsilon-2}(\Omega) \times H^{p+1/2-\epsilon-s}(\Gamma)$, then $u\big|_\Omega \in H^{p+1/2-\epsilon}(\Omega)$. Therefore, for $0 \leq i \leq p-1$, $\partial_\nu^i \left(u\big|_\Omega\right) \in H^{p+1/2-i-\epsilon}(\Gamma)$. Because $u\big|_{\mathbb{B}\backslash\Omega}$ satisfies Equation (6.3), again by elliptic theory, we know that $u\big|_{\mathbb{B}\backslash\Omega} \in H^{2p+1/2-\epsilon}(\mathbb{B}\backslash\Omega)$. We set $f = (\mathcal{S}_p u)\big|_\Omega$. We note that $f \in H^{1/2-\epsilon}(\Omega)$ by assumption. We let $\tilde{f} = \mathcal{E}_{\mathbb{B}}(f)$, an $H^{1/2-\epsilon}$ extension of f to $\mathbb{B}$. Similarly, we set $g = (\mathcal{S}_p u)\big|_{\mathbb{B}\backslash\Omega}$, and we let $\tilde{g} = \mathcal{E}_{\mathbb{B}}(g) \in H^{1/2-\epsilon}(\mathbb{B})$ (it will in fact have higher regularity than this). We now let

$$h = \tilde{f}\chi_\Omega + \tilde{g}\chi_{\mathbb{B}\backslash\Omega}.$$

Then u satisfies

$$\mathcal{S}_p u = h.$$

We cite Theorem 5.1 from [2].

Lemma 6.1. *If $u \in H^{1/2-\epsilon}(\mathbb{B})$ and $v \in H^{d/2-\epsilon}(\mathbb{B})$ then $uv \in H^{1/2-\epsilon}(\mathbb{B})$.*

As Ω is a smooth domain by assumption, we know that $\chi_\Omega, \chi_{\mathbb{B}\setminus\Omega} \in H^{d/2-\epsilon}$. Thus, by the lemma, $\mathcal{S}_p u = \tilde{f}\chi_\Omega + \tilde{g}\chi_{\mathbb{B}\setminus\Omega} \in H^{1/2-\epsilon}(\mathbb{B})$. This implies that $u \in H^{p+1/2-\epsilon}(\mathbb{B})$. □

Remark 6.2. *We now give a heuristic for the optimal level of convergence one obtains when using the $\mathcal{S}_p$ regularizing norm. We note that for a general H^p function, the optimal order an approximation technique could expect is order p. In our case, we assume we have an $H^{p+1/2-\epsilon}$ solution of the continuous problem(the optimal case) and that we are using regularizing norm $\mathcal{S}^p$. In this case, the data is $H^{p+1/2-\epsilon-2}(\Omega) \times H^{p+1/2-\epsilon-s}(\Gamma)$. We can try to approximate the data in the weakest regularity for which the BVP is well defined - in $H^{s+\epsilon-2}(\Omega) \times H^{\epsilon}(\Gamma)$. (In the Neumann case this can be further weakened by imposing the boundary condition weakly, but we have not yet implemented such a strategy using* SEEM*.) Doing this, we can hope to approximate the data to order $p + 1/2 - s - 2\epsilon$. Thus, order p convergence for the Dirichlet problem and order $p - 1$ for the Neumann problem are the optimal rates of convergence we could expect. In fact, this is the observed rate of convergence for the method.*

Chapter 7

Discrete Theory

The theory of the discretized solution involves two parts:

1. Is the matrix well posed - is there a unique solution for each right hand side f?
2. Does the solution converge to the true solution as the grid is taken finer and finer?

We address each of these questions with regards to the Fourier implementation of *SEEM* in the next two sections. We will assume that all interpolation operators are evaluted spectrally, meaning that the Fourier series representation of the discrete solution satisfies the BVP exactly at the collocation points.

7.1 Well Posedness

We discuss the question of whether the discrete problem is well posed. We first note that because S_p is chosen to be a positive definite matrix, the existence of a solution of the full saddle point system depends on the surjectivity of C, which we now show. The surjectivity

of C is equivalent to the question of whether the rows of C are linearly independent. We recall that each row of C corresponds to a collocation point, where either the interior or the boundary conditions of the BVP are imposed. Intutitively, we would assume that as the collocation points are brought closer together relative to the background regular grid, the rows of the matrix will become increasingly more dependent, until the matrix becomes singular at some point. Conversely, we would assume that as the points become further apart, the rows of the matrix become increasingly independent. We formalize this result by proving that if the points are sufficiently distant relative to the regular grid, then the rows will be linearly independent. The important question of how large a gap is necessary will be addressed numerically in Remark 7.2.

We will denote the grid distance on the regular grid as $h_{\mathbb{B}}$. We assume that the collocation points $\{y_i\}_{i=1}^{N_\Gamma}$ on Γ have a minimum distance $2h_\Gamma$ and the collocation points $\{x_i\}_{i=1}^{N_\Omega}$ in the interior Ω have a minimum distance $2h_\Omega$. For simplicity, we will assume that $h_\Omega \leq h_\Gamma$. We also require that the interior collocation points are a minimum distance of h_Ω from Γ.

Theorem 7.1. *Given $\alpha < 1$, $\exists \kappa \in \mathbb{R}^+$ such that if $h_\Gamma \geq h_\Omega > \kappa h_{\mathbb{B}}^\alpha$, then the matrix C is onto.*

Remark 7.1. *The theorem leaves open what happens when $\alpha = 1$: can the collocation points be taken to have constant density relative to the regular grid. It also leaves open what the optimal constant κ might be. Our numerical experiments suggest that the theorem holds for $\alpha = 1$ and that the value of κ can be taken to be 1. We will provide some numerical evidence in Remark 7.2.*

Proof. The idea of the proof is to place an approximate delta at each collocation point. The radius of the approximate delta is chosen so that the approximate deltas do not overlap. If the points on the regular grid are spaced close enough together, the grid will accurately approximate each of these approximate deltas. Thus, for any vector $v \in \mathbb{R}^{N_\Gamma + N_\Omega}$, we can use

a linear combination of these deltas to find a grid function u_m such that $Cu_m \sim v$. We now formalize this discussion. We will first need to define some new constants and operators.

- **A Choice of Sobolev Space.** Fix any $\epsilon > 0$ and choose $p \in \mathbb{N}$ such that

$$\alpha < \frac{p - 3 - d/2}{p + d/2}. \tag{7.1}$$

  We recall from Equation (1.2) that

$$\mathcal{C} : H^p(\Omega) \to H^{p-2}(\Omega) \times H^{p-s}(\Gamma).$$

  Here $s = 1/2$ for the Dirichlet problem, and $3/2$ for the Neumann and Robin problems.

- **A Projection Operator.** We define the projection from $H^p(\mathbb{B})$ onto the space of trigonomietric polynomials of degree less than m. We will denote this operator by $\mathcal{P}_m$, where

$$\mathcal{P}_m u = \sum_{|k|<m} a_k \exp^{ikx}.$$

  We will use the approximation result

$$\|(I - \mathcal{P}_m)u\|_{H^k} \leq c\|u\|_{H^p} h^{p-k}.$$

- **A Restriction Operator** We define the restriction operator

$$\mathcal{R}_\Omega : H^p(\mathbb{B}) \to H^p(\Omega)$$

  given by restriction to Ω.

- **An Approximate δ Distribution.** For a given dimension d, we select a function φ_d

satisfying $0 \leq \varphi_d \in \mathrm{C}^\infty(\mathbb{R}^d)$ with $\mathrm{supp}(\varphi_d) \in \mathbb{B}(0,1)$, $\int \varphi_d = 1$, and $\varphi_d(0) = 1$.

- **An Interpolation Operator.** We define the interpolation operators

$$\mathcal{I}_1 : \mathbb{R}^{N_\Omega} \to C_c^\infty(\Omega),$$
$$\mathcal{I}_2 : \mathbb{R}^{N_\Gamma} \to C^\infty(\Gamma)$$

by

$$\mathcal{I}_1(v_1) = \sum_{i=1}^{N_\Omega} (v_1)_i \varphi_d \left(\frac{x - x_i}{h_\Omega} \right),$$
$$\mathcal{I}_2(v_2) = \sum_{i=1}^{N_\Gamma} (v_2)_i \varphi_{d-1} \left(\frac{y - y_i}{h_\Gamma} \right).$$

(We assume that h_Γ is small enough so that $B(y_i, h_\Gamma)$ is homeomorphic to the Euclidean ball.) We then define the interpolation operator

$$\mathcal{I} : \mathbb{R}^{N_\Omega} \oplus \mathbb{R}^{N_\Gamma} \to H^{p-2}(\Omega) \times H^{p-s}(\Gamma),$$

with

$$\mathcal{I}(v_1 \oplus v_2) = (\mathcal{I}(v_1),\ \mathcal{I}(v_2)).$$

We note that by construction, the supports of the approximate δ distributions do not overlap. Straightforward computations allow to verify that for $s \in \mathbb{R}^+$,

$$\|\mathcal{I}_1 v_1\|_{H^s} \leq c h_\Omega^{d/2-s} \|v_1\|_{\ell_2},$$
$$\|\mathcal{I}_2 v_2\|_{H^s} \leq c h_\Gamma^{(d-1)/2-s} \|v_2\|_{\ell_2}.$$

- **An Inner Product.** We introduce a norm on $\mathbb{R}^{N_\Omega}$ given by

$$\|v_1\|_{K_1} = \|\mathcal{I}\, v_1\|_{H^{p-2}}.$$

Similarly, we introduce a norm on $\mathbb{R}^{N_\Gamma}$ given by

$$\|v_2\|_{K_2} = \|\mathcal{I}\, v_2\|_{H^{p-s}}.$$

Finally, we introduce a norm on $\mathbb{R}^{N_\Omega} \oplus \mathbb{R}^{N_\Gamma}$ given by

$$\|v_1 \oplus v_2\|_K = \left(\|v_1\|_{K_1}^2 + \|v_2\|_{K_2}^2\right)^{1/2}.$$

This norm defines the magnitude of a vector by using the $\mathcal{I}$ operator to generate a function in $H^{p-2}(\Omega) \times H^{p-s}(\Gamma)$. We note that this is guaranteed to be a norm because the supports of the approximate δ distributions do not overlap. We now note that

$$\begin{aligned}
&\|v_1\|_{K_1} \leq c h_\Omega^{d/2+2-p} \|v_1\|_{l_2}, \\
&\|v_2\|_{K_2} \leq c h_\Gamma^{(d-1)/2+s-p} \|v_2\|_{l_2}, \\
&\|v\|_K \leq c \left(h_\Omega^{d/2+2-p} + h_\Gamma^{(d-1)/2+s-p}\right) \|v\|_{l_2}.
\end{aligned}$$

We now begin the proof of the surjectivity of C. We consider the matrix

$$H = C\mathcal{P}_m\, \mathcal{E}_{\mathbb{B}}\, \mathcal{C}^{-1}\, \mathcal{I}\,.$$

We will use the fact that the surjectivity of H implies that of C. To study the surjectivity

of H, we note that $H = H_1 - H_2$, where

$$H_1 = C\mathcal{E}_{\mathbb{B}}\,\mathcal{C}^{-1}\,\mathcal{I},$$
$$H_2 = C(I - \mathcal{P}_m)\mathcal{E}_{\mathbb{B}}\,\mathcal{C}^{-1}\,\mathcal{I}.$$

By construction, H_1 is the identity matrix. Thus, it is sufficient to show that $\exists 0 < \beta \leq 1$ such that $\|H_2\|_K \leq \beta$. We will show that

$$\forall v \in \mathbb{R}^{N_\Omega} \oplus \mathbb{R}^{N_\Gamma}, \quad \langle H_2 v, v\rangle_K \leq \frac{1}{2}\|v\|_K^2.$$

We will use the notation $v = v_1 \oplus v_2$ and $Hv = w_1 \oplus w_2$. We note that by the definition of $\mathcal{I}$, we are guaranteed that $\|\mathcal{I}\,v\|_{H^{p-2}(\Omega)\times H^{p-s}(\Gamma)} = \|v\|_K$. (This is the definition of the K-norm.) Because $\mathcal{C}^{-1}$ and $\mathcal{E}_{\mathbb{B}}$ are bounded operators, we find that

$$\|\mathcal{E}_{\mathbb{B}}\,\mathcal{C}^{-1}\,\mathcal{I}\,v\|_{H^p} \leq c\|v\|_K,$$

where the constant c is determined by the continuous operator and by the choice of extension operator. We let $u = \mathcal{E}_{\mathbb{B}}\,\mathcal{C}^{-1}\,\mathcal{I}\,v$. We recall that the number of collocation points Ω^m grows like h_Ω^{-d}. We now study $w_1 = A(I - \mathcal{P}_m)u$. We find

$$\begin{aligned}
\|A(I - \mathcal{P}_m)u\|_{\ell_2} &\leq h_\Omega^{-d/2}\|A(I - \mathcal{P}_m)u\|_{\ell_\infty} \\
&\leq ch_\Omega^{-d/2}\|(I - \mathcal{P}_m)u\|_{C^2} \\
&\leq ch_\Omega^{-d/2}\|(I - \mathcal{P}_m)u\|_{H^{2+d/2+\epsilon}} \\
&\leq ch_\Omega^{-d/2}h_{\mathbb{B}}^{p-2-d/2-\epsilon}\|u\|_{H^p} \\
&\leq ch_\Omega^{-d/2}h_{\mathbb{B}}^{p-2-d/2-\epsilon}\|v\|_K.
\end{aligned}$$

We also recall that the number of collocation points Γ^m grows like h_Γ^{-d+1}. We study $w_2 = B(I - \mathcal{P}_m)u$. With $j = 0$ for the Dirichlet problem and $j = 1$ for the Neumann or Robin

problems (equivalently $j = s - \frac{1}{2}$), we calculate

$$\begin{aligned}
\|B(I-\mathcal{P}_m)u\|_{\ell_2} &\leq h_\Gamma^{(-d+1)/2}\|B(I-\mathcal{P}_m)u\|_{\ell_\infty} \\
&\leq ch_\Gamma^{(-d+1)/2}\|(I-\mathcal{P}_m)u\|_{C^j} \\
&\leq ch_\Gamma^{(-d+1)/2}\|(I-\mathcal{P}_m)u\|_{H^{j+d/2+\epsilon}} \\
&\leq ch_\Gamma^{(-d+1)/2}h_{\mathbb{B}}^{p-j-d/2-\epsilon}\|u\|_{H^p} \\
&\leq ch_\Gamma^{(-d+1)/2}h_{\mathbb{B}}^{p-j-d/2-\epsilon}\|v\|_K.
\end{aligned}$$

Altogether, we obtain the estimate

$$\begin{aligned}
\|C(I-\mathcal{P}_m)\mathcal{E}_{\mathbb{B}}\,\mathcal{C}^{-1}\,v\|_{l_2} &\leq c\left(h_\Omega^{-d/2}h_{\mathbb{B}}^{p-2-d/2-\epsilon} + h_\Gamma^{(-d+1)/2}h_{\mathbb{B}}^{p-j-d/2-\epsilon}\right)\|v\|_K \\
&\leq c\left(h_\Omega^{-d/2}h_{\mathbb{B}}^{p-2-d/2-\epsilon} + h_\Gamma^{(-d+1)/2}h_{\mathbb{B}}^{p-j-d/2-\epsilon}\right) \\
&\qquad \left(h_\Omega^{d/2+2-p} + h_\Gamma^{(d-1)/2+s-p}\right)\|v\|_{\ell_2} \\
&\leq c\left((h_\Omega^{-d/2} + h_\Gamma^{-d/2})h_{\mathbb{B}}^{p-2-d/2-\epsilon}\right)(h_\Omega^{-p} + h_\Gamma^{-p})\|v\|_{\ell_2}.
\end{aligned}$$

We note that in the last inequality, we have sacrificed on the optimal constants by dropping several terms with a positive exponents. This was done to simplify the calculation. We next recall that $h_\Gamma \geq h_\Omega$. Therefore,

$$\|C(I-\mathcal{P}_m)\mathcal{E}_{\mathbb{B}}\,\mathcal{C}^{-1}\,v\|_{\ell_2} \leq ch_\Omega^{-p-d/2}h_{\mathbb{B}}^{p-2-d/2-\epsilon}\|v\|_{\ell_2}.$$

The theorem assumed that $h_\Omega \geq \kappa h_{\mathbb{B}}^\alpha$. We therefore find that

$$\|C(I-\mathcal{P}_m)\mathcal{E}_{\mathbb{B}}\,\mathcal{C}^{-1}\,v\|_{\ell_2} \leq c\kappa^{-p-d/2}h_{\mathbb{B}}^{\alpha(-p-d/2)}h_{\mathbb{B}}^{p-2-d/2-\epsilon}\|v\|_{\ell_2}.$$

We next note that in (7.1), p was chosen sufficiently large to satisfy

$$\alpha(-p-d/2) + (p-2-d/2-\epsilon) > 0,$$

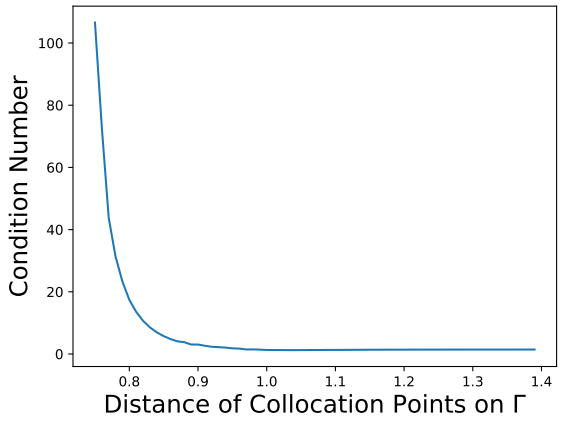

Figure 7.1: Growth of condition number as the boundary collocation points become closer spaced relative to the regular grid distance. The x-axis measures $h_\Gamma/h_{\mathbb{B}}$. The domain considered is the unit disc.

so $h_{\mathbb{B}}^{\alpha(-p-d/2)} h_{\mathbb{B}}^{p-2-d/2-\epsilon} < 1$. Thus, by choosing κ so that $c\kappa^{-p-d/2} < \frac{1}{2}$, we obtain the desired inequality,

$$\|C(I - \mathcal{P}_m)\mathcal{E}_{\mathbb{B}}\,\mathcal{C}^{-1}\,v\|_{\ell_2} \leq \frac{1}{2}\|v\|_{\ell_2}.$$

□

Remark 7.2. *Our proof leaves open the question of the optimal constants for how close the collocation points can be placed together. However, we offer the following numerical result. We create an interpolation matrix for points equally distributed on the unit disc with distance h_Γ between the points. In Figure 7.1, we plot how the condition number of the resulting matrix depends on the ratio $h_\Gamma/h_{\mathbb{B}}$, where $h_{\mathbb{B}}$ is the grid distance on the background regular grid. We note that it appears that as long as the distance between the collocation points on the boundary is greater than the grid distance $h_{\mathbb{B}}$, the condition number remains under control.*

7.2 Convergence Analysis

7.2.1 A Sampling Inequality

For the convergence proofs, we will use a sampling inequality developed for the theory of meshfree methods. The inequality is taken from [27]. We will assume that a Ω is a bounded domain and X is a finite discrete set of Ω. We define

$$h_{X,\Omega} := \sup_{x\in\Omega} \min_{x_j\in X} |x - x_j|,$$

the fill distance of X on Ω.

Theorem 7.2. *Let $u \in H^p(\Omega)$, with $p > |\alpha| + \frac{d}{2}$. Assume that*

1. $u\big|_X \equiv 0$.
2. *Assume that $\exists \theta \in [0, \pi/2), r > 0$ such that $\forall x \in \Omega$, the cone*

$$C(x, \xi(x), \theta, r) := \{x + \lambda y \mid y \in \mathbb{R}^d, \|y\|_{\ell_2} = 1, y^\top \xi(x) > \cos(\theta), \lambda \in [0, r]\} \subset \Omega.$$

   This is known as an interior cone condition, see [27, Chapter 3].

3. *The following relationship holds between the fill distance and the interior cone condition.*

$$h_{X,\Omega} \leq \frac{r \sin(\theta)}{4(1 + \sin(\theta))p^2}.$$

Then

$$\|D^\alpha u\|_{L_2(\Omega)} \leq c h^{p-|\alpha|} \|u\|_{H^p(\Omega)}.$$

Remark 7.3. *We note that although the theorem in [27] is cited with regards to subset of* $\mathbb{R}^d$*, using partitions of unity it is straightforward to extend the theorem to the compact boundary manifold* Γ*, provided that* Γ *is smooth.*

7.2.2 Convergence Proof

We note that our discrete solution u_h is an H^p bounded function which satisfies

$$\mathcal{C}(u_h)(z_i) = b_i.$$

For any such function, we can apply Theorem 7.2 to both the interior and the boundary right hand sides. The boundedness of $\mathcal{C}^{-1}$ will then imply the convergence of the discrete solution. For the purposes of this discussion, we will denote $h = \max\{h_\Omega, h_\Gamma\}$.

Theorem 7.3. *If* $p > 2 + \frac{d}{2}$ *and* $\exists c$ *s.t.* $\|u\|_{H^p(\Omega)} < c$ *and* u_h *is the* $H^p(\Omega)$ *minimizing function which satisfies*

$$\forall z_i \in Z, \mathcal{C}(u_h)(z_i) = b_i.$$

Then we have the bound

$$\|u - u_h\|_{H^2(\Omega)} \leq ch^{p-2}(\|f\|_{H^{p-2}(\Omega)} + \|g\|_{H^{p-s}(\Gamma)}).$$

Proof. We first note that u_h is the H^p minimizing function which satisfies $\mathcal{C}(u)(z_i) = b_i$.

Thus,

$$\|\mathcal{A}u_h\|_{H^{p-2}(\Omega)} + \|\mathcal{B}u_h\|_{H^{p-s}(\Gamma)} \leq c\|u_h\|_{H^p(\Omega)} \tag{7.2}$$

$$\leq c\|u\|_{H^p(\Omega)} \tag{7.3}$$

$$\leq c(\|f\|_{H^{p-2}(\Omega)} + \|g\|_{H^{p-s}(\Gamma)}). \tag{7.4}$$

The first and third inequalities are the boundedness of $\mathcal{C}$ and $\mathcal{C}^{-1}$. The second relies on the minimality of u_h. We now note that because $\mathcal{C}(u - u_h)(z_i) = 0$, the functions $\mathcal{A}(u - u_h)$ and $\mathcal{B}(u - u_h)$ are functions with scattered zeros to which we can apply Theorem 7.2. We calculate

$$\begin{aligned}
\|u - u_h\|_{H^2} &\leq c(\|\mathcal{A}(u - u_h)\|_{L^2(\Omega)} + \|\mathcal{B}(u - u_h)\|_{H^{p-s}(\Gamma)}) \\
&\leq ch^{p-2}\|\mathcal{A}(u - u_h)\|_{H^{p-2}} + ch^{p-s}\|\mathcal{B}(u - u_h)\|_{H^{p-s}} \\
&\leq ch^{p-2}(\|f\|_{H^{p-2}} + \|\mathcal{A}u_h\|_{H^{p-2}}) + ch^{p-k}(\|g\|_{H^{p-s}} + \|\mathcal{B}u_h\|_{H^{p-s}}) \\
&\leq ch^{p-2}(\|f\|_{H^{p-2}} + \|g\|_{H^{p-s}}).
\end{aligned}$$

The first inequality come the from the boundedness of $\mathcal{C}$, while the second is an application of Theorem 7.2. The third is just the triangle inequality and the definitions of f and g. The fourth is an application of Equation (7.2). □

Remark 7.4. *We note again that Theorem* (7.3) *only guarantees* $(p-2)$ *order convergence when using the pth order Sobolev kernel. In numerical experiments, we observe pth order convergence instead, provided that the solution* $u \in H^{p+2}$*. This has been demonstrated in Sections 8.1, 8.3, and 8.4. We refer to Remark 6.2 for a discussion of this point.*

Part IV

Numerical Experiments

Chapter 8

Fourier Torus

We present several numerical experiments to demonstrate the Fourier implementation of *SEEM*. As we discussed in Section 4.1.1, in the Fourier discretization, the operator S_p is given by

$$S_p u = \mathcal{F}^{-1} \operatorname{diag}[(1 + |k|^2)^{p/2}] \mathcal{F}(u),$$

where $\mathcal{F}$ is the discrete Fourier transform and $k \in \mathbb{Z}_m^d$ is the vector of frequency vectors on the grid $\mathbb{B}$. Derivatives in the matrix C are taken spectrally, and interpolation is done spectrally on sparse grids and with cubic interpolation on dense grids.

We present experiments with the goal of demonstrating the rate of convergence and the numerical efficiency of the method. We first present a Dirichlet problem with analytic solution. This will allow us to highlight how the rate of convergence depends on p and that the S_p regularizer attains a p-th order of convergence for the Dirichlet problem. Next, we present a Robin BVP, also with analytic solution. This will allow us to demonstrate the $p-1$-order convergence which is attained for the S_p regularizer for the Neumann and Robin problems. Next, we will study two problems of lower global regularity. In these examples, the problem

will be solved on denser grids using iterative methods, as described in Section 5.1.2. We will demonstrate how the method remains efficient on dense grids when using the preconditioning described in Section 5.2. We will also consider a three dimensional problem, where the number of grid points is substantially larger. Finally, we will show how our method can be used to solve the Stokes problem, which seeks a solution in a divergence-free space. In our method, a basis for such a space can be easily constructed and we discuss how to implement such a solver.

8.1 A Dirichlet Problem

In our first experiment, we study the function

$$u = x^2 - y^2.$$

We will study a Dirichlet problem on the unit disc, given by

$$\begin{cases} -\Delta u = 0 & \text{in } D, \\ u = x^2 - y^2 & \text{on } \partial D. \end{cases} \tag{8.1}$$

Since u is an analytic function, any algebraic order of convergence could be attained with the use of an appropriate order smoother and discretizations. We solve the problem on sparse grids (up to 128^2) using the QR-method of Section 5.1.1. We note that using the QR formulation allows us to use smoothers with order $2 \leq p \leq 10$. Beyond $p = 10$, the diagonal entries of the S_p^{-1} matrix (in Fourier space) are smaller than machine precision. We show the errors both in Figure 8.1 and in Tables 8.1 and 8.2. The order is calculated by comparing the errors on the densest grid to those on the sparsest grids. We observe the p-th order convergence of both the L_2 and L_∞ errors.

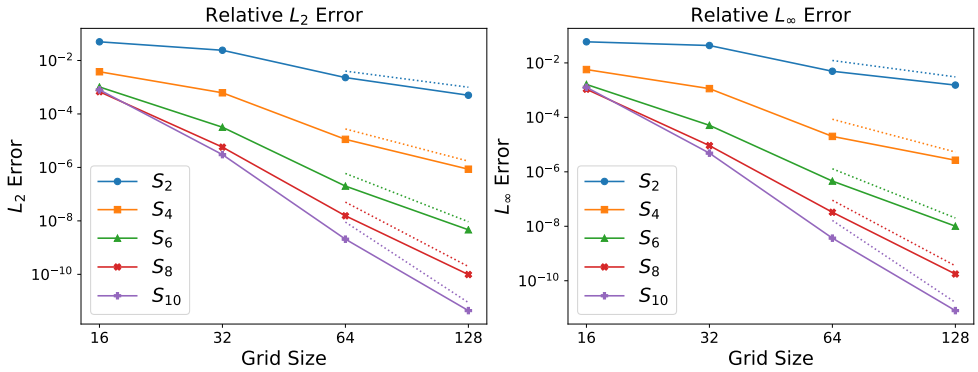

Figure 8.1: Convergence of the L_2 and L_∞ relative errors for different order smoothers solving Equation (8.1). The dotted reference lines have slope $-p$.

Grid Size	Relative L_2 Error				
	S_2	S_4	S_6	S_8	S_{10}
16^2	$5.00E-02$	$3.79E-03$	$1.02E-03$	$6.78E-04$	$7.96E-04$
32^2	$2.41E-02$	$6.19E-04$	$3.19E-05$	$5.74E-06$	$3.01E-06$
64^2	$2.29E-03$	$1.12E-05$	$2.00E-07$	$1.55E-08$	$2.07E-09$
128^2	$5.00E-04$	$8.58E-07$	$4.64E-09$	$9.82E-11$	$4.40E-12$
Order:	2.21	4.04	5.92	7.57	9.14

Table 8.1: Relative L_2 error for Equation (8.1).

Grid Size	Relative L_∞ Error				
	S_2	S_4	S_6	S_8	S_{10}
16^2	$5.96E-02$	$5.70E-03$	$1.62E-03$	$1.08E-03$	$1.27E-03$
32^2	$4.36E-02$	$1.14E-03$	$5.11E-05$	$9.18E-06$	$4.85E-06$
64^2	$4.94E-03$	$2.00E-05$	$4.57E-07$	$3.27E-08$	$3.67E-09$
128^2	$1.53E-03$	$2.67E-06$	$1.01E-08$	$1.77E-10$	$8.00E-12$
Order:	1.76	3.69	5.76	7.52	9.08

Table 8.2: Relative L_∞ error for Equation (8.1).

8.2 A Robin Problem

In this experiment, we again study the analytic function

$$u = x^2 - y^2.$$

In this experiment, in addition to showing the order of convergence of the Robin BVP, we seek to demonstrate the that our method is robust with regards to nonconstant coefficients, a variety of boundary conditions, and irregular geometries. Therefore, we let

$$\Omega = \left\{ \big[1 + .2\cos(5\theta)\big]\big(\cos(\theta), \sin(\theta)\big) \;\middle|\; \theta \in [0, 2\pi) \right\}.$$

This is the star-shaped domain shown in Figure 1.1. We study the Robin BVP given by

$$\begin{cases} -\left((2+y)\partial_x^2 + (2-x)\partial_y^2\right) u = -2x - 2y & \text{in } \Omega, \\ u + \frac{\partial u}{\partial \nu} = g & \text{on } \partial\Omega, \end{cases} \tag{8.2}$$

where $\frac{\partial}{\partial \nu}$ is the normal derivative on $\partial\Omega$ and $g = (\gamma_\Gamma + \partial_\nu)(x^2 - y^2)$ is the Robin boundary operator on $\partial\Omega$ applied to u. As in the previous experiment, we solve the problem using the QR formulation. We show the errors both in Figure 8.2 and in Tables 8.3 and 8.4. We observe the $p-1$-th order convergence of both the L_2 and L_∞ errors. In particular, comparing Tables 8.1 and 8.3, we see that the Dirichlet problem converges an order faster than the Robin problem. As explained in Remark 6.2, this is also true of the Neumann problem.

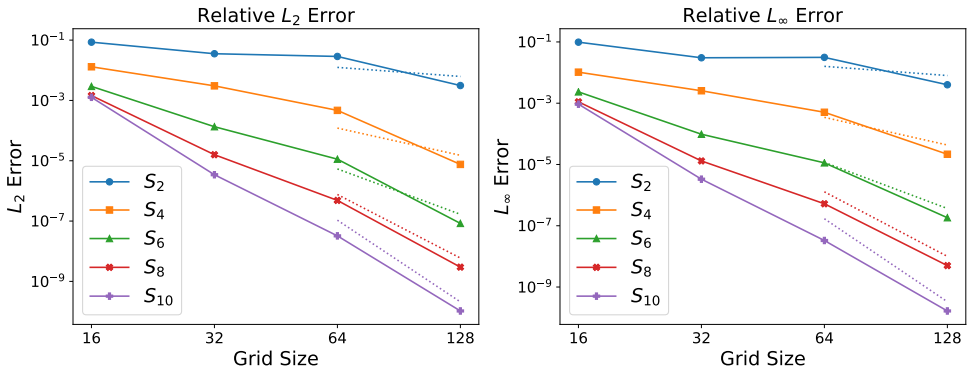

Figure 8.2: Convergence of the L_2 and L_∞ relative errors for different order smoothers solving Equation (8.2). The dotted reference lines have slope $-p+1$.

Grid Size	S_2	S_4	S_6	S_8	S_{10}
16^2	$8.55E-02$	$1.31E-02$	$2.92E-03$	$1.45E-03$	$1.27E-03$
32^2	$3.54E-02$	$3.05E-03$	$1.35E-04$	$1.59E-05$	$3.44E-06$
64^2	$2.88E-02$	$4.68E-04$	$1.13E-05$	$4.85E-07$	$3.23E-08$
128^2	$3.12E-03$	$7.57E-06$	$8.37E-08$	$2.97E-09$	$1.04E-10$
Convergence Rate:	1.59	3.58	5.03	6.3	7.85

Table 8.3: Relative L_2 error for Equation (8.2).

Grid Size	S_2	S_4	S_6	S_8	S_{10}
16^2	$9.83E-02$	$1.03E-02$	$2.34E-03$	$1.11E-03$	$9.31E-04$
32^2	$3.04E-02$	$2.55E-03$	$9.72E-05$	$1.31E-05$	$3.31E-06$
64^2	$3.13E-02$	$5.06E-04$	$1.13E-05$	$5.22E-07$	$3.30E-08$
128^2	$4.00E-03$	$2.16E-05$	$1.83E-07$	$5.00E-09$	$1.65E-10$
Convergence Rate:	1.54	2.97	4.55	5.92	7.48

Table 8.4: Relative L_∞ error for Equation (8.2).

8.3 H^6 Regularity

We now study a Dirichlet problem of global H^6 regularity. To construct such a function, we take the inverse Fourier transform of a random sequence of coefficients which decay like $(1 + |k|^2)^{-3}$ and by restricting the resulting function to the domain Ω. This is done on a dense grid of sizes 8192^2. The right-hand-side is obtained by evaluating the interior and boundary differential operators spectrally at the collocation points. A contour plot of the (unrestricted) function ca nbe seen in Figure 8.3. A Dirichlet problem is then solved on the disc of radius 2.

Because this function is of lower global regularity, the fastest order convergence which can be achieved with a strong formulation is 4-th order. (The interior data is $H^4(\Omega)$ regularity.) We therefore use the S_2, S_3, and S_4 smoothers, and achieve 2, 3, and 4-th order convergence respectively. Because the order of convegence is limited to 4-th order, we will need to solve the problem on dense grids to get an accurate solution. We therefore use iterative methods to solve on dense grids. To save memory and computational complexity, we use cubic interpolation on the boundary. We use this problem to demonstrate the effectiveness of our preconditioning procedure. We have solved this problem using the PCG method as described in Section 5.1.2. In Table 8.6, we show the condition number of the preconditioned matrix, as well as the number of iterations and CPU time to convergence. The computations were performed using the standard Python packages on an Intel-i7-7700, and we used a stopping criteria of $1e-8$ for the PCG iterations.

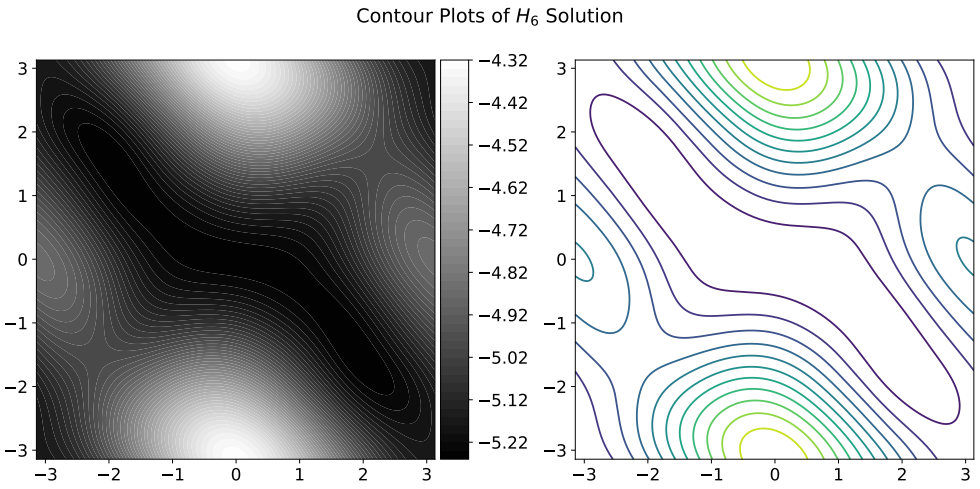

Figure 8.3: Contour plots of H^6 Solution

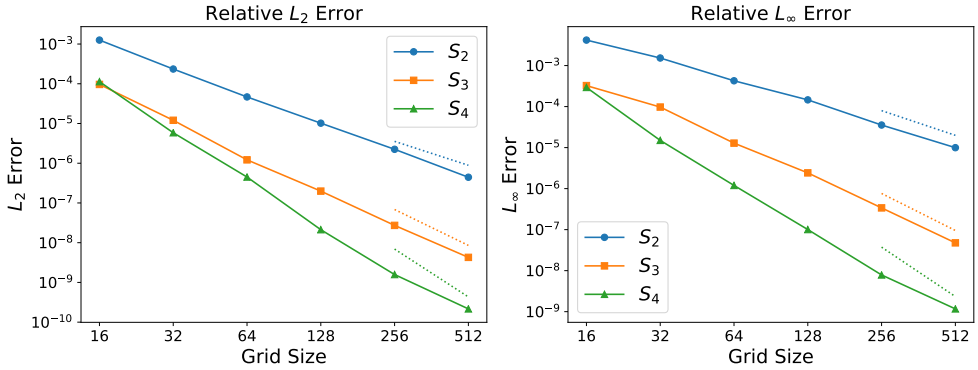

Figure 8.4: Relative L_2 and L_∞ errors for Dirichlet problem with H^6 Solution

Grid Size	S_2	S_3	S_4
16^2	$1.26E-03$	$9.62E-05$	$1.13E-04$
32^2	$2.35E-04$	$1.21E-05$	$5.85E-06$
64^2	$4.66E-05$	$1.22E-06$	$4.49E-07$
128^2	$1.02E-05$	$1.99E-07$	$2.12E-08$
256^2	$2.24E-06$	$2.73E-08$	$1.58E-09$
512^2	$4.44E-07$	$4.27E-09$	$2.16E-10$

Table 8.5: Relative L_2 error for the Dirichlet problem with H^6 solution

Grid Size	PCG Iterations			CPU Time			Condition Number		
	S_2	S_3	S_4	S_2	S_3	S_4	S_2	S_3	S_4
16^2	24	40	66	0.01	0.02	0.04	8	30	214
32^2	25	39	73	0.01	0.03	0.06	10	37	310
64^2	28	44	107	0.03	0.05	0.13	12	40	545
128^2	29	45	131	0.06	0.11	0.42	14	40	1087
256^2	33	48	191	0.48	0.75	2.7	-	-	-
512^2	33	68	368	2.24	5.47	36.27	-	-	-

Table 8.6: CPU times, number of iterations, and condition number for the preconditioned Schur complement matrix $\widetilde{C}^{-1/2}(CS_p^{-1}C^\top)\widetilde{C}^{-1/2}$ when solving Dirichlet problem with H^6 solution on the disc with radius 2.

8.4 H^3 Regularity

Next consider $u \in H^3$ generated using the same technique used in the previous section. A contour plot of the (unrestricted) function can be seen in Figure 8.5. In this case, $-\Delta u \in H^1$, and thus cannot be defined pointwise. We therefore need to turn to a weak formulation of the problem. We proceed as follows. Rather than imposing the pointwise conditions

$$-\Delta u(x_i) = f(x_i), \quad x_i \in \Omega^m,$$

(because f cannot be evaluated at a point), we instead resort to a weak formulation on the interior. Letting $\{\phi_i\}$ be the standard finite element basis of piecewise linear functions on the regular grid $\mathbb{B}^m$, we seek a solution of the form

$$u = \sum_{i=1}^{N_m} c_i \phi_i.$$

We then take the subset of those ϕ_i's the support of which is entirely within the domain Ω. We denote this set Ω^m and set $|\Omega^m| = N_m^\Omega$. The interior conditions

$$\int_\Omega \nabla u \nabla \phi_i = \int_\Omega f \phi_i,$$

are imposed for $\phi_i \in \Omega^m$. Here, the integral $\int_\Omega f\phi$ is calculated using Fourier series on a dense grid. The boundary constraint is unchanged from the strong formulation. In line with our proposed method, we then seek a solution which minimizes the discrete H^2 norm, computed using the discrete Fourier transform of c. Notice that, for a uniform grid, the differentiation matrix for the basis ϕ_i coincides with the standard five point stencil finite difference discretization. We solve the Dirichlet BVP on the unit disc. The resulting L_2 and L_∞ errors, as well as the CPU times are shown in Figure 8.6 and Table 8.7. In Table 8.7, we also show a comparison with a global RBF method and the RBF-FD method. We note that in [12], the claim is made that the usage of meshfree methods for second order elliptic problems is restricted to $u \in H^p$ with $p > 2 + \frac{d}{2}$, leading to a $p-2$ order of convergence of the L^2 error. Here, using a discrete collocation procedure, we are able to treat weaker solutions as well, leading to second order convergence for an H^3 solution as opposed to first order convergence for the meshfree implementation of [12] for $H^{3+\epsilon}$

Remark 8.1. *Analytically, only nodal basis functions which are compactly supported inside Ω should be used. Numerically it is, however, better to include any basis functions with support intersecting Ω. For such basis functions ϕ, we impose the condition*

$$c\int_{\mathbb{B}} \nabla u \nabla \phi = \int_\Omega f\phi, \text{ where } c = \frac{\int_\Omega \phi}{\int_{\mathbb{B}} \phi}.$$

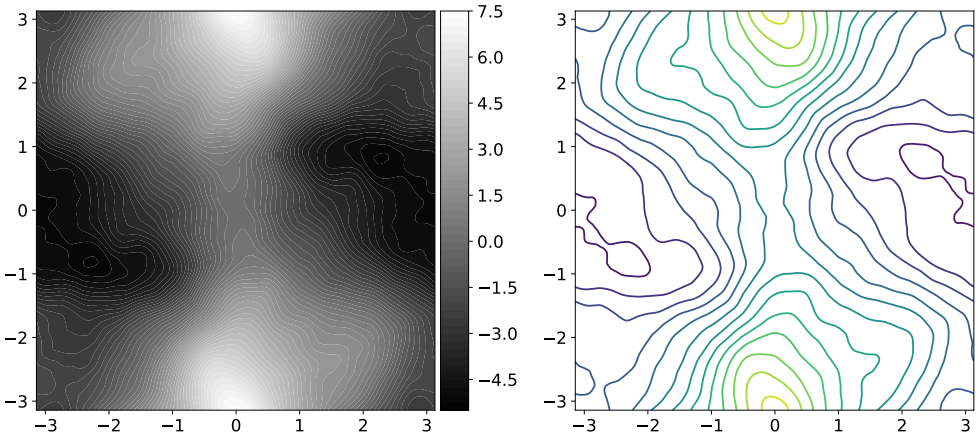

Figure 8.5: Contour plots of H^3 Solution

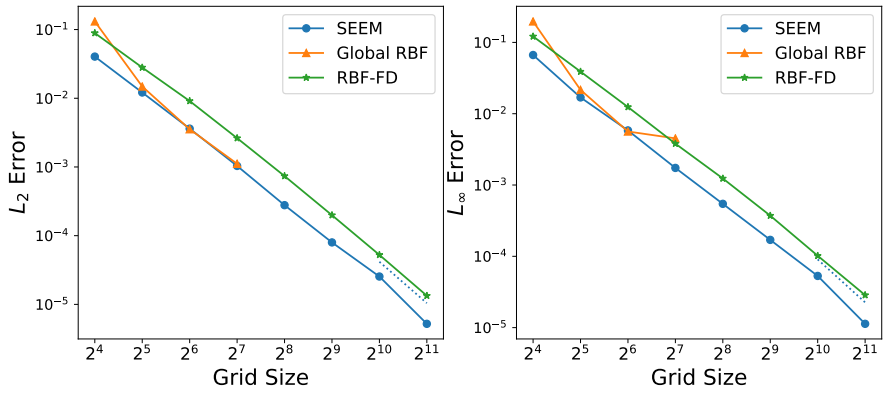

Figure 8.6: Relative L_2 and L_∞ errors for Dirichlet problem with H^3 solution. The dotted reference lines have slope -2.

Grid Size	L_2 Error	L_2 Order	L_∞ Error	PCG Iterations	CPU Time
16^2	$4.04E-02$	-	$6.66E-02$	25	0.24
32^2	$1.21E-02$	1.74	$1.69E-02$	27	0.18
64^2	$3.61E-03$	1.75	$5.82E-03$	28	1.36
128^2	$1.03E-03$	1.81	$1.73E-03$	29	0.54
256^2	$2.79E-04$	1.89	$5.49E-04$	31	0.85
512^2	$7.45E-05$	1.91	$1.60E-04$	31	2.2
1024^2	$1.92E-05$	1.95	$4.18E-05$	34	11.4
2048^2	$4.12E-06$	2.22	$9.56E-06$	45	59.84

Table 8.7: Relative L_2 and L_∞ errors, as well as PCG iterations and CPU times for the Dirichlet problem with H^3 solution

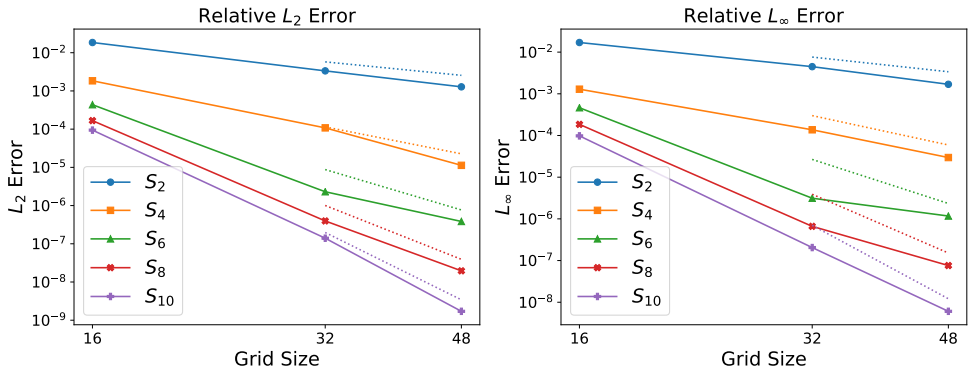

Figure 8.7: Convergence of the L_2 and L_∞ errors for different order smoothers solving Equation (8.3). The dotted reference lines have slope $-p$.

8.5 A Three Dimensional Problem

We present a three dimensional Dirichlet problem. Again, we will use an analytic solution to demonstrate the different rates of convergence of the different smoothers. We set our domain Ω to be the unit sphere embedded in the three dimensional torus. The boundary is discretized by the well known Fibonacci lattice which comes close to distributing points uniformly. With a box discretization consisting of m^3 points, we use $\frac{m^2}{2\pi^2}$ points for the discretization of the boundary. The problem considered is

$$\begin{cases} -\Delta u = 1 & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega. \end{cases} \tag{8.3}$$

The exact solution is $\frac{1-r^2}{6}$. We see in Figure 8.7 that the rate of convergence achieved is similar to that of the two dimensional problem. We note that because we are calculating explicit matrices for the pseudoinverse method, RAM limitations prevented us from using a grid larger than 48^3. We also compute solutions on denser grids using the Schur complement method, and record the number of iterations and CPU times. We note that when using the Schur complement method, we have decreased the density of boundary points to $\frac{m^2}{64\pi^2}$ points per unit area to improve the conditioning of the matrices.

Grid Size	S_2	S_4	S_6	S_8	S_{10}
16^3	$1.85E-02$	$1.84E-03$	$4.36E-04$	$1.67E-04$	$9.43E-05$
32^3	$3.35E-03$	$1.08E-04$	$2.30E-06$	$3.96E-07$	$1.40E-07$
48^3	$1.28E-03$	$1.13E-05$	$3.84E-07$	$1.96E-08$	$1.71E-09$
Convergence Rate:	2.43	4.63	6.4	8.23	9.94

Table 8.8: Relative L_2 error solving Equation (8.3).

Grid Size	S_2	S_4	S_6	S_8	S_{10}
16^3	$1.70E-02$	$1.29E-03$	$4.63E-04$	$1.85E-04$	$9.79E-05$
32^3	$4.48E-03$	$1.37E-04$	$3.13E-06$	$6.62E-07$	$2.05E-07$
48^3	$1.69E-03$	$2.96E-05$	$1.17E-06$	$7.59E-08$	$6.06E-09$
Convergence Rate:	2.1	3.44	5.44	7.1	8.82

Table 8.9: Relative L_∞ error solving Equation (8.3).

Grid Size	PCG Iterations			CPU Time		
	S_2	S_3	S_4	S_2	S_3	S_4
16^2	13	21	27	0.03	0.03	0.08
32^2	17	27	53	0.25	0.18	0.63
64^2	21	30	55	2.6	3.37	7.27
128^2	22	32	196	13.52	25.75	185.51

Table 8.10: CPU times and number of iterations for the PCG method solving Equation (8.3).

8.6 The Stokes Equation

One of the advantages of fictitious domain methods is that the solution can be represensted as a linear combination of simple and well understood basis functions, e.g. as a Fourier series. A potential use of this is in imposing the divergence-free condition to solve the Stokes problem. The Stokes problem is given by

$$\begin{aligned} -\Delta \mathbf{u} + \nabla p &= \mathbf{f} \text{ on } \Omega \\ \mathbf{u} &= g \text{ on } \Gamma \\ -\nabla \cdot \mathbf{u} &= 0 \text{ on } \Omega. \end{aligned} \tag{8.4}$$

In general, it is difficult to create a basis of divergence-free vector fields for solving the Stokes equation. See [28] for examples of solving the Stokes equation using RBF-type divergence-free basis functions. Furthermore, with most discretization schemes, it is necessary to have different discretizations of the velocity space and the pressure space, a so called "stable-pair," see [5]. If one is not careful in the discretization, the discrete gradient will have a kernel, and the discrete problem will become ill-posed. Many numerical schemes exist to address this issue, including the MAC scheme [15], which places the finite difference points in different locations for each component of the velocity as well as the pressure.

Using *SEEM*, we are able to discretize the Stokes equation on an arbitrary domain using Fourier series. As the gradient operator can be diagonalized in Fourier space, it is straightforward to generate a basis for the divergence-free vector fields on the torus. This method relies on the fact that divergence-free vector fields on Ω can be extended to divergence-free vector fields on the entire torus. In particular, in two dimensions, for example, we take

vector fields of the form

$$\left(a_{0,0} + \sum_{k_x=0}^{m} \sum_{k_y=0}^{m} a_{k_x,k_y} k_y e^{ik_x x} e^{ik_y y}, b_{0,0} - \sum_{k_x=0}^{m} \sum_{k_y=0}^{m} b_{k_x,k_y} k_x e^{ik_x x} e^{ik_y y} \right).$$

It is straightforward to see that such vector fields are divergence-free, and, using the Helmholtz decomposition, that these are a basis for the divergence-free vector fields.

We introduce the projection operator $\mathcal{P}_{\text{div}}$ that maps a vector field onto its divergence-free component. In two dimensions, if

$$\mathbf{u} = \left(\sum_{k_x=0}^{m} \sum_{k_y=0}^{m} a_{k_x,k_y} e^{ik_x x} e^{ik_y y}, \sum_{k_x=0}^{m} \sum_{k_y=0}^{m} b_{k_x,k_y} e^{ik_x x} e^{ik_y y} \right).$$

then

$$\mathcal{P}_{\text{div}} \mathbf{u} = \left(a_{0,0} + \frac{k_y (k_y a_{k_x,k_y} - k_x b_{k_x,k_y})}{k_x^2 + k_y^2} e^{ik_x x} e^{ik_y y}, b_{0,0} + \frac{k_x (k_x b_{k_x,k_y} - k_y a_{k_x,k_Y})}{k_x^2 + k_y^2} e^{ik_x x} e^{ik_y y} \right)$$

Proposition 8.1. *The Stokes equation* (8.4) *is equivalent to the problem*

$$-\Delta \boldsymbol{v} = f \ \text{ on } \Omega$$

$$\mathcal{P}_{\text{div}} \boldsymbol{v} = g \ \text{ on } \Gamma,$$

where

$$\boldsymbol{u} = \mathcal{P}_{\text{div}} \boldsymbol{v}.$$

Proof. We choose ϕ such that $\Delta \phi = p$. We will seek a solution $\mathbf{u}$ which is divergence-free on all of $\mathbb{B}$ rather than just on Ω. This can be done because divergence-free vector fields can be

smoothly extended from Ω to $\mathbb{B}$. The Stokes equation is then given by

$$\begin{aligned} -\Delta\mathbf{u} + \nabla\Delta\phi &= \mathbf{f} \text{ on } \Omega \\ -\nabla\cdot\mathbf{u} &= 0 \text{ on } \Omega \\ \mathbf{u} &= g \text{ on } \Gamma. \end{aligned}$$

As ∇ and Δ commute, we obtain the formulation

$$\begin{aligned} -\Delta(\mathbf{u} + \nabla\phi) &= \mathbf{f} \text{ on } \Omega \\ -\nabla\cdot\mathbf{u} &= 0 \text{ on } \Omega \\ \mathbf{u} &= g \text{ on } \Gamma. \end{aligned}$$

We set $\mathbf{v} = \mathbf{u} + \nabla\phi$ and using the Helmholtz decomposition, we recognize that $\mathbf{u} = \mathcal{P}_{\text{div}}\mathbf{v}$. Thus, the two formulations are equivalent. □

Using this formulation, we are able to very easily use *SEEM* to solve the Stokes problem. We introduce the notation $\mathcal{P}_x$ and $\mathcal{P}_y$ to represent projection onto the x and y components of the velocity. We define the *SEEM* PDE and smoother matrices given by

$$C = \begin{pmatrix} -\Delta_{\Omega^m}\mathcal{P}_x \\ -\Delta_{\Omega^m}\mathcal{P}_y \\ \gamma_{\Gamma^m}\mathcal{P}_x\mathcal{P}_{\text{div}} \\ \gamma_{\Gamma^m}\mathcal{P}_y\mathcal{P}_{\text{div}} \end{pmatrix} \text{ and } S_p = \begin{pmatrix} (1-\Delta)^{p/2} & 0 \\ 0 & (1-\Delta)^{p/2}, \end{pmatrix}$$

where Δ_{Ω^m} is the discretized Laplacian evaluated at the interior grid points and γ_{Γ^m} is the trace operator evaluated on the discrete boundary, as described in Section 4.4. We then find

that

$$u = (S_p^\top S_p)^{-1} C^\top (C(S_p^\top S_p)^{-1} C^\top)^{-1} b, \tag{8.5}$$

where b is the vector of evaluatoins of $\mathbf{f}$ and $\mathbf{g}$ on the discrete interior and boundary, respectively. We consider a Dirichlet problem posed on Ω, the unit disc, given by

$$\begin{cases} -\Delta \mathbf{u} + \nabla p = 0 & \text{on } \Omega \\ -\nabla \cdot \mathbf{u} = 0 & \text{on } \Omega \\ u_1 = 20xy^3 & \text{on } \Gamma \\ u_2 = 5x^4 - 5y^4 & \text{on } \Gamma. \end{cases} \tag{8.6}$$

The solution is given by

$$\begin{cases} u_1 &= 20xy^3 \\ u_2 &= 5x^4 - 5y^4 \\ p &= 60x^2y - 20y^3 + c. \end{cases}$$

In light of Proposition 8.1, we solve

$$\begin{pmatrix} u \\ \lambda \end{pmatrix} = \begin{pmatrix} S_p & C^\top \\ C & 0 \end{pmatrix} \begin{pmatrix} 0 \\ b \end{pmatrix},$$

where C, S_p, and b are the matrices and vector described in Equation (8.5). Note that there is no need for us to explicitly calculate the pressure. We solve the system using both the QR methods and the PCG methods. For the PCG method, we use the preconditioner described in Section 5.2 applied to the x and y components independently. In Table 8.11 and Figure 8.8, we record the L_2 error of the u_1 and u_2 components of the velocity. In Table 8.12, we

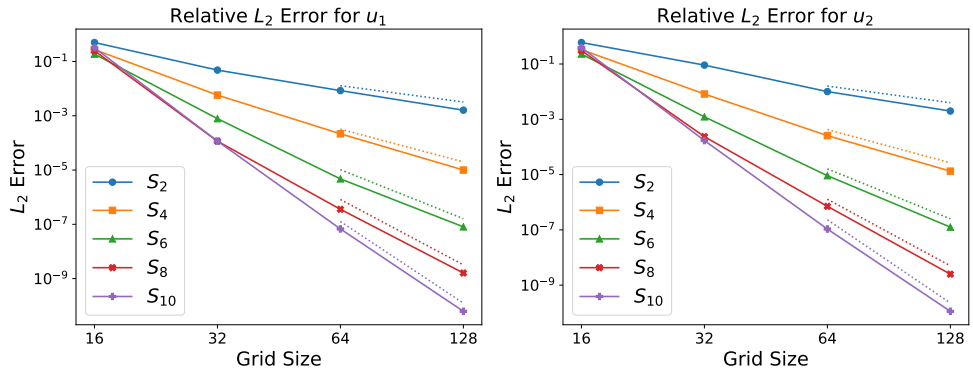

Figure 8.8: Plots of relative L_2 errors of u_1 and u_2 for the Stokes problem, Equation (8.6). The dotted reference lines have slope $-p$.

Grid Size	S_2	S_4	S_6	S_8	S_{10}
16^2	$5.00E-01$	$2.75E-01$	$1.87E-01$	$2.48E-01$	$3.20E-01$
32^2	$4.82E-02$	$5.78E-03$	$7.83E-04$	$1.16E-04$	$1.17E-04$
64^2	$8.45E-03$	$2.16E-04$	$4.72E-06$	$3.58E-07$	$6.76E-08$
128^2	$1.61E-03$	$1.00E-05$	$8.06E-08$	$1.60E-09$	$6.23E-11$
Convergence Rate:	2.4	4.44	5.87	7.8	10.1

Table 8.11: Relative L_2 error of u_1 for the Stokes problem, Equation (8.6).

record the number of iterations and CPU time to convergence of the PCG method.

Remark 8.2. *A more effective preconditioner could potentially be constructed by taking the form of the $\mathcal{P}_{\text{div}}$ operator into account while building the part of the preconditioner relating to the boundary. Work is currently being done in this direction.*

Grid Size	PCG Iterations			CPU Time		
	S_2	S_3	S_4	S_2	S_3	S_4
16^2	24	44	85	0.03	0.08	0.21
32^2	36	70	157	0.07	0.18	0.59
64^2	44	50	124	0.16	0.27	0.81
128^2	50	59	184	0.52	0.7	3.46
256^2	58	63	264	9.52	5.31	38.75
512^2	63	70	423	21.56	40.81	283.43

Table 8.12: CPU times and number of iterations for the PCG method solving the Stokes problem, Equation (8.6).

Chapter 9

Chebyshev Box

Before we describe some numerical implementations, we give a brief description of some of the properties of Chebyshev polynomials. We refer to [25] for a more detailed discussion.

9.1 The Chebyshev Polynomials

Setting $\mathbb{B} = [-1, 1]$, the Chebyshev polynomials of the first kind are given by

$$T_m(x) = \cos\big(m \arccos(x)\big),\ x \in \mathbb{B},\ m \in \mathbb{N}.$$

The Chebyshev roots

For fixed $m \in \mathbb{N}$, the m roots of $T_m(x)$ are given by

$$x_k = \cos\Big(\pi \frac{2k-1}{2m}\Big),\ 0 \leq k \leq m-1.$$

The Chebyshev grid comprising all roots of T_m, given by $\{x_k \,|\, k = 0, \ldots, m-1\}$, is well adapted to the spectral calculation of derivatives. In higher dimensions, a tensor product of one-dimensional Chebyshev grids can be used. Such a tensor product (of the appropriate dimension) will be denoted as $\mathbb{B}^m$.

Orthogonality relations

The sequence $(T_m)_{m\in\mathbb{N}}$ forms an orthogonal basis for $L_2(\mathbb{B})$ with respect to the measure $\frac{dx}{\sqrt{1-x^2}}$. More specifically, for $i, j \in \mathbb{N}$,

$$\int_{-1}^{1} T_i(x)T_j(x)\frac{dx}{\sqrt{1-x^2}} = \begin{cases} 0, & \text{if } i \neq j, \\ \pi, & \text{if } i = j = 0, \\ \pi/2, & \text{if } i = j \neq 0. \end{cases}$$

The Chebyshev functions restricted to $\mathbb{B}^m$ also satisfy a discrete orthogonality relation. Indeed, for $0 \leq i, j \leq m-1$, one has that

$$\sum_{k=0}^{m-1} T_i(x_k)T_j(x_k) = \begin{cases} 0, & \text{if } i \neq j, \\ m, & \text{if } i = j = 0, \\ \frac{m}{2}, & \text{if } i = j \neq 0. \end{cases}$$

The Chebyshev transform

Because $(T_m)_{m\in\mathbb{N}}$ forms an orthogonal basis of $\mathrm{L}_2(\mathbb{B})$, any function $u \in \mathrm{L}_2(\mathbb{B})$ can be developed in a "Chebyshev series". We set

$$c_k = \frac{p_k}{\pi}\int_{-1}^{1} u(x)T_k(x)\frac{dx}{\sqrt{1-x^2}}, \text{ where } p_k = \begin{cases} 1, & \text{if } k = 0, \\ 2, & \text{if } k \neq 0. \end{cases}$$

so that

$$u(x) = \sum_{m=0}^{\infty} c_k T_k(x).$$

It is also possible to define the Chebyshev transform, denoted by $\mathfrak{C}$, which maps a function to the sequence of its Chebyshev coefficients.

$$\mathfrak{C}(u) = (c_k)_{k\in\mathbb{N}}.$$

The discrete orthogonality relation also yields a discrete version of a Chebyshev expansion. Given $u : \mathbb{B}^m \to \mathbb{R}$, let

$$c_k = \frac{p_k}{m}\sum_{i=0}^{m-1} u_i T_k(x_i) \text{ for } p_k = \begin{cases} 1, & \text{if } k = 0, \\ 2, & \text{if } k \neq 0. \end{cases}$$

Then u can be written as a discrete Chebyshev series

$$u(x) = \sum_{k=0}^{m-1} c_k T_k(x).$$

As in the continuous case, we can define the discrete Chebyshev transform $\mathfrak{C}_m$, which maps u to its discrete Chebyshev series, i.e., we set

$$\mathfrak{C}_m(u) = (c_k)_{k=0,\ldots,m-1} =: c.$$

The discrete Chebyshev transform, $\mathfrak{C}_m$, as well as its inverse $\mathfrak{C}_m^{-1}$, can be implemented efficiently using an FFT algorithm, in the form of the discrete cosine transform. More specifically,

$$\mathfrak{C}_{\mathfrak{m}}(u)_k = a_k \mathsf{DCT}(u)_k \text{ for } a_k = \begin{cases} \frac{1}{2m}, & \text{if } k = 0, \\ \frac{1}{m}, & \text{if } k > 0, \end{cases}$$

and

$$\mathfrak{C}_{\mathfrak{m}}{}^{-1}(c) = \mathsf{IDCT}(\widetilde{c}) \text{ for } \widetilde{c}_k = \begin{cases} c_k, & \text{if } k = 0, \\ c_k/2, & \text{if } k > 0. \end{cases}$$

In dimension larger than one, $\mathfrak{C}$ and $\mathfrak{C}_m$ will denote the continuous and discrete one-variable Chebyshev transforms applied successively in each direction. Numerically, this can be accomplished with the use of DCTN, where the factors a_k and b_k are raised to the power of the dimension.

Derivative formulæ

Discrete derivatives can be efficiently evaluated on the Chebyshev grid using the DCT and DST. We denote by $\bullet$ the discrete frequency vector $(k)_{k\in\{0,\dots,m-1\}}$ or the continuous variable x depending on the context, and let $M[f]$ represent multiplication by the discrete function f. We also define a shifting operator R with

$$\mathsf{R}_{ij} = \delta_{i+1,j},$$

so that R is the matrix with ones on the superdiagonal. Then, given a function $u = (u_i)_{i\in\{0,\dots,m-1\}}$ defined on the Chebyshev grid $\mathbb{B}^m$, a spectrally accurate discrete derivative Du can be calculated using the matrix given by

$$D = M\Big[\frac{1}{\sqrt{1-\bullet^2}}\Big] \circ \mathsf{IDST} \circ \mathsf{R} \circ M\Big[\frac{\bullet}{2m}\Big] \circ \mathsf{DCT}.$$

Similarly, we can compute

$$D^2 = M\Big[\frac{-1}{1-\bullet^2}\Big] \circ \mathsf{IDCT} \circ M\Big[-\frac{\bullet^2}{2m}\Big] \circ \mathsf{DCT} + M\Big[\frac{\bullet}{(1-\bullet^2)^{3/2}}\Big] \circ \mathsf{IDST} \circ \mathsf{R} \circ M\Big[-\frac{\bullet}{2m}\Big] \circ \mathsf{DCT}.$$

Of course, a corresponding operator can be formed in higher dimensions, where the DCT, DST as well as the frequency vector $\bullet$ are taken along the desired direction of differentiation. In the body of the paper, the derivative operator in the x_i direction is denoted as D_i.

Eigenvalue equation

Setting $\mathcal{D} = M\big[\sqrt{1-x_\bullet^2}\big] \circ \frac{\partial}{\partial x}$, the Chebyshev polynomials satisfy the eigenvalue equation

$$-\mathcal{D}^2 T_m = m^2 T_m. \tag{9.1}$$

Similarly, given the Chebyshev grid $\mathbb{B}^m$, the discrete Chebyshev functions $T_j(x_\bullet)$ satisfy a discrete eigenvalue equation. Defining $\mathcal{D}_m = M\big[\sqrt{1-x_\bullet^2}\big] \circ D$, $T_j(x_\bullet)$ satisfies

$$-\mathcal{D}_m^2 T_j(x_\bullet) = j^2 T_j(x_\bullet),\ j \in \{0, \ldots, m-1\}.$$

This implies that

$$(1-\mathcal{D}_m^2)^{-p/2} = \mathfrak{C}_m^{-1} \circ M\Big[(1+|\bullet|^2)^{-p/2}\Big] \circ \mathfrak{C}_m.$$

Interpolation operators

Functions defined on the Chebyshev grid can be interpolated at arbitrary points in $\mathbb{B}$. Such interpolation can be stably computed by means of the barycentric interpolation formulæ

described in [4]. Define first the vector w by

$$w_k = (-1)^k \sin\Big(\frac{2k-1}{2m}\Big),\ 0 \le k \le m-1.$$

If $y \in \mathbb{B}$ and $(x_i)_{i\in\{0,\dots,m-1\}}$ is the vector of points in $\mathbb{B}^m$, a spectrally accurate interpolation of a discrete function u defined on the Chebyshev grid $\mathbb{B}^m$ can be obtained by

$$u(y) = \delta_y \cdot u \text{ where } (\delta_y)_i = \frac{1}{\sum_{k=0}^{m-1} \frac{w_k}{y-x_k}} \frac{w_i}{y-x_i}.$$

To calculate the interpolation of the first derivative, which will be used in the Neumann problem, we use the derivative of the above formula,

$$Du(y) = \delta_y \circ D \text{ where } (\delta_y \circ D)_i = -\frac{1}{\sum_{k=0}^{m-1} \frac{w_k}{y-x_k}} \frac{w_i}{(y-x_i)^2} + \frac{\sum_{k=0}^{m-1} \frac{w_k}{(y-x_k)^2}}{\Big(\sum_{k=0}^{m-1} \frac{w_k}{y-x_k}\Big)^2} \frac{w_i}{y-x_i}.$$

To interpolate in several dimensions, we use a tensor product of the given interpolants, which are denoted by δ_y and $(\delta_y \cdot \nabla)$. To calculate a directional derivative of the grid function u in the direction ν at the point y, we use $(\delta_y \circ \nabla u) \cdot \nu_y$.

9.2 Discretizing the matrices C and S_p

We now briefly descirbe how to use the described Chebyshev discretizations to form the matrices C and S_p, which form the basis of *SEEM*.

Construction of C

We note that in Section 4.4, C is formed using derivative discretizations D_i and interpolatin operators δ_{y_i}. We can therefore discretize C using the derivative discretizations and

interpolation operators used in the previous section. As usual, we can use spectral or finite difference discretizations. We will use spectral discretizations when using high order smoothers to obtain high orders of convergence, and finite difference discretization when using low lorder smoothers on dense grids, allowing the interpolation matrices to be sparse.

Construction of S^{-1}

Recall from Section 4.1.2 that we utilize the smoothers

$$\mathcal{S}_p^{-1} = \mathfrak{C}^{-1} \operatorname{diag}\left[(1+|k|^2)^{-p/2}\right] \mathfrak{C}$$

Here $k = (k)_{k\in\mathbb{N}^d}$ is the Chebyshev frequency vector on the d dimensional box $\mathbb{B}$. This operator can be discretized simply and efficiently using the discrete Chebyshev transform $\mathfrak{C}_m$. When $k = (k)_{k\in\{0,\dots,m-1\}^d}$ is the discrete Chebyshev frequency vector on the d dimensional grid $\mathbb{B}^m$, we define the discrete smoothers as

$$S_p^{-1} = \mathfrak{C}_m^{-1} \operatorname{diag}\left[(1+|k|^2)^{-p/2}\right] \mathfrak{C}_m.$$

9.3 Experiments

We now present two experiments using the Chebyshev discretization. We first present a Dirichlet problem on the unit disc, and then present a parabolic problem in spacetime.

$\lvert B^m \rvert$	N_Ω^m	N_Γ^m	L_2 Error					
			S_2	S_4	S_6	S_8	S_{10}	S_{12}
8^2	24	10	3.42E-02	2.31E-02	1.13E-02	3.83E-03	1.04E-03	2.38E-04
16^2	104	19	3.81E-03	1.04E-04	1.05E-05	4.93E-07	1.59E-08	4.51E-10
24^2	240	29	1.37E-03	9.70E-06	7.75E-08	1.75E-09	2.71E-11	8.83E-12
32^2	408	38	1.13E-03	9.55E-06	3.20E-08	7.54E-11	1.62E-12	5.24E-12
36^2	520	43	4.67E-04	4.07E-06	1.75E-08	3.69E-11	7.58E-13	1.77E-11
Rate of Convergence:			2.85	5.75	8.89	12.27	13.99	10.91

Table 9.1: Relative L_2 and L_∞ errors for Equation (9.2).

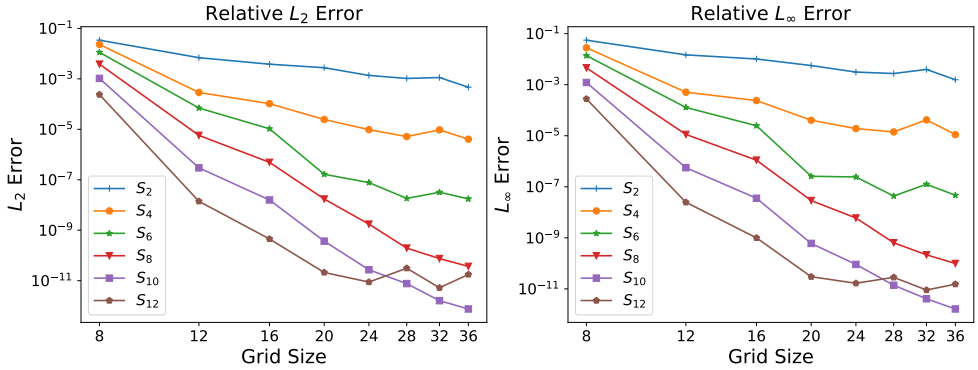

Figure 9.1: Convergence of the relative L_2 error and L_∞ errors for Equation (9.2).

9.3.1 A Dirichlet Problem

On $\Omega = \{(x, y) \,|\, \sqrt{x^2 + y^2} \leq .95\}$, we consider the Dirichlet problem

$$\begin{cases} -\Delta u = -6x - 6y & \text{in } \Omega_1, \\ u = x^3 + y^3 & \text{on } \partial\Omega_1. \end{cases} \tag{9.2}$$

The exact solution is $x^3 + y^3$. We solve the problem using the pseudoinverse method, with explicit matrices. The sizes of the different discretizations considered, together with the L_2 and L_∞ errors, are listed in Table 9.1. A graph of the L_2 and L_∞ errors are show in Figure 9.1. In Table 9.2, we demonstrate number of iterations and time to convergence for the PCG algorithm using Schur complement method.

$\lvert B^m \rvert$	PCG Iterations			CPU Times		
	S_2	S_3	S_4	S_2	S_3	S_4
16^2	36	26	37	0.05	0.03	0.05
32^2	60	41	60	0.1	0.06	0.11
64^2	94	67	106	0.37	0.2	0.33
128^2	122	97	173	1.11	0.94	1.64
256^2	149	118	328	7.26	5.41	14.9

Table 9.2: PCG Iterations and CPU times for the Chebyshev Discretization.

9.3.2 A Parabolic Problem

We next describe a procedure to solve a time dependent problem using *SEEM*. This example is considered to show both that *SEEM* is effective for a wide variety of PDEs and to show that it is effective in three dimensions. Since we are calculating the solution across a space-time cylinder (and not marching in time, although this could also be done), the problem is effectively a three dimensional problem. We consider the parabolic cylinder $\Omega \times [0, 2]$, where Ω is the star-shaped domain shown in Figure 1.1. Define $j_0(r)$ to be the 0-th Bessel function of the first kind. Letting r denote the Euclidean distance from 0, consider the radial function

$$u(r, t) = e^{-t} j_0(r) - e^{-\frac{t}{4}} j_0\Big(\frac{r}{2}\Big).$$

The function u then satisfies the parabolic BVP

$$\begin{cases} u_t - \Delta u = 0 & \text{in } \Omega \times (0, 2], \\ u = j_0(r) - j_0(\frac{r}{2}) & \text{in } \Omega \times \{0\}, \\ u = e^{-t} j_0(r) - e^{-\frac{t}{4}} j_0(\frac{r}{2}) & \text{on } \Gamma \times (0, 2]. \end{cases} \tag{9.3}$$

To discretize the domain, we again use the Chebyshev grid $\mathbb{B}^m$ described in Section 9.1. As for the time interval $[0, 2]$, we use a (shifted) Chebyshev extrema grid,

$$\mathbb{B}_E^n = \{t_j\}_{j=0}^n, \text{ where } t_j = -\cos\Big(\frac{\pi j}{n}\Big) + 1.$$

In this section $n = 10$ is chosen in all of the experiments. The full discretization of the parabolic cylinder $[-1,1]^2 \times [0,2]$ is then given by $\mathbb{B}^m \times \mathbb{B}_E^n$. The choice to use the extrema grid rather than the standard Chebyshev (roots) grid in the time variable was made because imposing the boundary condition at $t = 0$ is slightly more straightforward, since the boundary point $t = 0$ lies on the grid. For a description of how to construct the time differentiation matrix, D_t, we refer to [25]. With the discretization $\mathbb{B}^m \times \mathbb{B}_E^n$, the interior of the parabolic cylinder is given by $\Omega^m \times \widetilde{\mathbb{B}}_E^n$, where

$$\widetilde{\mathbb{B}}_E^n = \big\{ t_i \in \mathbb{B}_E^n \,\big|\, t_i > 0 \big\}.$$

The discretized "bottom" boundary of the cylinder is given by $\Omega^m \times \{0\}$, whereas the discretization of the lateral boundary $\Gamma \times (0,2]$ is simply given by $\Gamma^m \times \widetilde{\mathbb{B}}_E^n$. Letting R_{K^m} denote the evaluation operator on the discrete set K^m, we can define the matrices

$$\begin{aligned} A &= R_{\Omega^m \times \widetilde{\mathbb{B}}_E^n} \circ (D_t - D_{x_1}^2 - D_{x_2}^2), \\ B_1 &= R_{\Omega^m \times \{0\}}, \\ B_2 &= R_{\Gamma^m \times \widetilde{\mathbb{B}}_E^n}. \end{aligned}$$

Notice that evaluation of a function on the above sets simply amounts to their restriction to the sets since $\Omega^m \times \widetilde{\mathbb{B}}_E^n$ and $\Omega^m \times \{0\}$ are sets of regular grid points. However, because Γ^m does not contain regular grid points in general, the evaluation matrix $R_{\Gamma^m \times \widetilde{\mathbb{B}}_E^n}$ will require the use of the interpolation operators described in Sections 4.4 and 9.1. If b_1 and b_2 represent the evaluations of the function $e^{-t} j_0(r) - e^{-\frac{t}{4}} j_0\left(\frac{r}{2}\right)$ at the points of $\Omega^m \times \{0\}$ and $\Gamma^m \times \widetilde{\mathbb{B}}_E^n$, respectively, the BVP is fully discretized by the matrix equation $Cu = b$ where

$$C = \begin{bmatrix} A & B_1 & B_2 \end{bmatrix}^\top \text{ and } b = \begin{bmatrix} 0 & b_1 & b_2 \end{bmatrix}^\top$$

As we do for elliptic problems, the problem is converted to a constrained optimization problem

$$\text{argmin}_{\{Cu=b\}} \frac{1}{2}\|u\|_S^2,$$

where $\|\cdot\|_S$ is a smoothing norm. In the parabolic case, $\|\cdot\|_S$ needs to be a space-time norm over $\mathbb{B}^m \times \mathbb{B}_E^n$. We recall from Sections 4.1.2 and 9.1 the operator

$$(\mathcal{D}^m)^2 = \mathfrak{C}_m^{-1} \,\text{diag}\left[|k|^2\right] \mathfrak{C}_m.$$

The operators $\mathcal{D}_i^m$ and $\mathcal{D}_t^m$ represent applying the operator in the x_i and t directions, respectively. Motivated by our choice of smoothing norm used in the elliptic case and described in Section 4.1.2, the norm given by

$$\|u\|_{S_p} = \left\| \left(1 - \sum_{i=1}^{2} (\mathcal{D}_i^m)^2 - (\mathcal{D}_t^m)^2\right)^{p/2} (u) \right\|,$$

is used in order to enforce space-time regularity of the numerical solution. As with the norms described in the elliptic case, this norm has the benefit of simple implementation using the discrete Chebyshev transform. We remark that while this norm is clearly effective, as demonstrated by our numerical experiments, it is not natural from the point of view of parabolic PDEs and may not be the optimal one to use; we are continuing to investigate the best choice of smoother in the parabolic case.

As in the elliptic case, the problem then reduces to finding

$$u = S_p^{-1}(CS_p^{-1})^+ f.$$

The solution is obtained using a QR decomposition of $S_p^{-1}C^\top$, as described in Section 5.1.1. (A PCG method could also be used to obtain the solution on denser grids.) The numerical results for the initial boundary value problem are summarized in Table 9.4 and Figure 9.2.

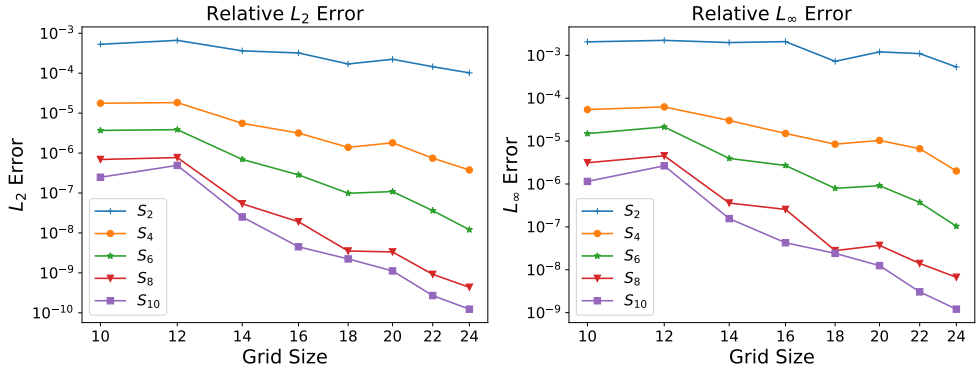

Figure 9.2: Convergence of the relative L_2 error and L_∞ errors for Equation (9.3).

$\lvert\mathbb{B}^m \times \mathbb{B}_E^n\rvert$	$\lvert\Omega^m \times \tilde{\mathbb{B}}_E^n\rvert$	$\Omega^m \times \{0\}\rvert$	$\lvert\Gamma^m \times \tilde{\mathbb{B}}_E^n\rvert$
$10^2 \times 11 = 1100$	280	28	130
$14^2 \times 11 = 2156$	500	50	180
$18^2 \times 11 = 3564$	860	86	230
$24^2 \times 11 = 6336$	1540	154	300

Table 9.3: Grid sizes Equation (9.3).

$\lvert\mathbb{B}^m \times \mathbb{B}_E^n\rvert$	L_2 Error				
	S_2	S_4	S_6	S_8	S_{10}
$10^2 \times 11$	5.30E-04	1.77E-05	3.69E-06	6.91E-07	2.48E-07
$14^2 \times 11$	3.63E-04	5.56E-06	6.95E-07	5.36E-08	2.51E-08
$18^2 \times 11$	1.70E-04	1.39E-06	9.84E-08	3.53E-09	2.23E-09
$24^2 \times 11$	1.02E-04	3.78E-07	1.20E-08	4.36E-10	1.23E-10
Rate of Convergence:	1.88	4.39	6.54	8.42	8.69

Table 9.4: Relative L_2 error for Equation (9.3).

Chapter 10

Finite Difference Box

As mentioned in Section 4.1.3, using a finite difference discretization is somewhat less efficient than the use of spectral discretization. However, its advantage is that it allows for adaptive grids, does not require periodic boundary conditions on $\mathbb{B}$, and makes it easier to use smoothers which have nonconstant coefficients.

10.1 A Dirichlet Problem on a Regular Grid

When discretizing

$$\mathcal{S}_p = (1 - \Delta)^{p/2},$$

we require that $p \in 2\mathbb{N}$. In the following experiments, we restrict ourselves to the case $p = 2$, which will allow for second order convergence. Although $p = 4$ could also be considered, we have not yet successfully implemented this smoother due to numerical stability issues. We let our fictitious domain $\mathbb{B}$ be given by $[0, 1]^2$ and take a regular discretization $\mathbb{B}^m$ on $\mathbb{B}$. In this case, we will discretize the operator $S_2 = 1 - \Delta$ using the standard five

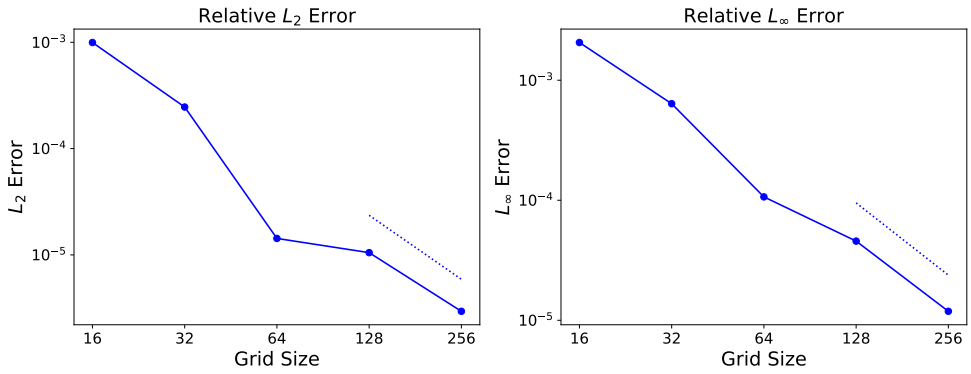

Figure 10.1: Relative L_2 and L_∞ errors for Equation (10.1). The dotted reference lines have slope -2.

point stencil. We impose periodic boundary conditions for simplicity, although Dirichlet or Neumann conditions on the boundary of the box could also be used. We discretize the BVP C using the five point stencil and cubic interpolation. We consider the domain

$$D = \{(x, y) | \sqrt{(x - .5)^2 + (y - .5)^2} < .3\}.$$

We study the Dirchlet problem

$$\begin{cases} -\Delta u = 0 & \text{in } D, \\ u = \frac{(x-.5)^2-(y-.5)^2}{4} & \text{on } \partial D. \end{cases} \tag{10.1}$$

We solve the problem using the Schur complement formulation, using PCG iterations on the Schur complement matrix to obtain the solution. Although the matrix S_2 would typically be inverted using a fast multigrid solver to allow for computation on dense grids, for simplicity we have used a sparse LU factorization solver. We show the L_2 and L_∞ errors in Figure 10.1 and in Table 10.1. Additionally, show CPU time in Table 10.1.

Grids	L_2 Error	L_∞ Error	CPU Time	PCG Iterations
16^2	0.000995001	0.0020617	0.0789979	25
32^2	0.000245965	0.000637346	0.230005	29
64^2	1.42839e-05	0.00010667	1.604	34
128^2	1.05004e-05	4.56741e-05	15.051	34
256^2	2.94656e-06	1.18771e-05	101.894	36

Table 10.1: Relative L_2 and L_∞ errors as well as PCG iterations and CPU times for Equation (10.1).

10.2 A Dirichlet Problem on Nested Grids

We now give a short discussion of the potential of using nested grids to obtain the benefits of adaptivity in *SEEM*. We present one experiment, although much work is still needed before the method can be widely used. For simplicity, we will restrict ourselves to the case where two grids only intersect if one is twice as dense as the other (this restricts the number of special boundary stencils needed). By way of example, we consider Figure 10.2. Here, we have placed a regular grid on $[0,1)^2$ with grid distance $h_s = \frac{1}{16}$. In the region $[\frac{7}{16}, \frac{9}{16}]^2$, we have placed a denser grid with grid distance $h_d = \frac{1}{32}$.

On points which are interior to either the sparse or the dense grid, we can simply use the five points stencil. On the interface between the dense and sparse grids, we need to calculate a stencil which accurately captures the Laplacian. The way we have structured the nested grids, we only need to consider the stencil at two types of points. We refer to Figure 10.4. We see there are two kinds of boundary points in a two grid discretization: those like A which lie in both the sparse and dense grid, and those like B which lie only in the dense grid. While multiple choices of finite difference stencil are available, for simplicity we have used the following. For points of type A, we simply use the five point stencil on the sparse grid, shown in green. For the points of type B, we use a seven point stencil which include the point B and the six points nearest to it on the sparse grid, again shown in green. The weights are $\frac{-2}{h_d^2}$ at the point B, $\frac{1}{8h_d^2}$ at each of the corner points, and $\frac{3}{h_d^2}$ at each of the points

directly above and below B, where h_d is the grid distance on the dense grid.

While several methods could be used for the boundary interpolation operator, we use a straightforward cubic interpolation with a sixteen point stencil. At any point where cubic interpolation is possible using the fine grid, that is, if the point is at least h_d from the dense grid boundary, we apply cubic interpolation on the dense grid. Otherwise, we use cubic interpolation on the sparse grid.

Once we have a discretization of the Laplacian and of the interpolation operators, we are able to construct the S_2 and C matrices used in *SEEM*. The S_2 matrix is formed by simply taking $1 - \Delta$ using the discretization just described. The C matrix is formed by applying the Laplace discretization to the interior and the interpolation operators to the boundary.

We now demonstrate how such nested grids can be used to solve a problem with a singularity more efficiently. We consider the function $u = \sqrt{(x-.5)^2 + (y-.5)^2}$, which is smooth away from the singularity at $(.5, .5)$. We let

$$\Omega = \{(x, y) | \sqrt{(x-.5)^2 + (y-.5)^2} < .3\}$$

and consider the Dirichlet problem

$$\begin{cases} -\Delta u = -\frac{1}{\sqrt{(x-.5)^2+(y-.5)^2}} & \text{on } \Omega \\ u = \sqrt{(x-.5)^2 + (y-.5)^2} & \text{on } \Gamma. \end{cases} \tag{10.2}$$

We note that the Laplacian is not defined at $(.5, .5)$ so we do not impose the interior condition at that point. Incidentally, an advantage of the *SEEM* formulation is that we do not need to impose the condition at any points where it is inconvenient to do so. Due to the nature of the solution, the error is concentrated at $(.5, .5)$. We demonstrate this in Figure 10.3, where we plot the error of the discrete *SEEM* solution calculated on the grid of size 128^2. Thus, it

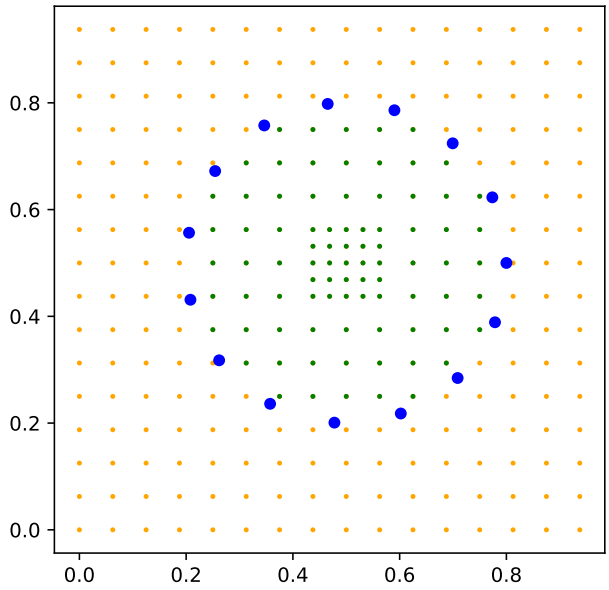

Figure 10.2: Example of a two level nested grid with a discretization of the disc of radius .3 centered at $(.5, .5)$.

is logical that refining the grid at $(.5, .5)$ will improve the accuracy of the solution. We solve the problem on grids with the following four levels of refinement.

1. Grid 1: Uniform grid with mesh size h.
2. Grid 2: Add refinement with mesh size $\frac{h}{2}$ on $[\frac{3}{8}, \frac{5}{8}]^2$.
3. Grid 3: Add refinement with mesh size $\frac{h}{4}$ on $[\frac{7}{16}, \frac{9}{16}]^2$.
4. Grid 4: Add refinement with mesh size $\frac{h}{8}$ on $[\frac{15}{32}, \frac{17}{32}]^2$.

In Tables 10.2 and 10.3 and Figure 10.5, we plot the L_2 and L_∞ errors, comparing the unrefined grids with grids of several levels of refinement. In Table 10.4, we show how using nested grids is computationally more efficient than using a denser regular grid.

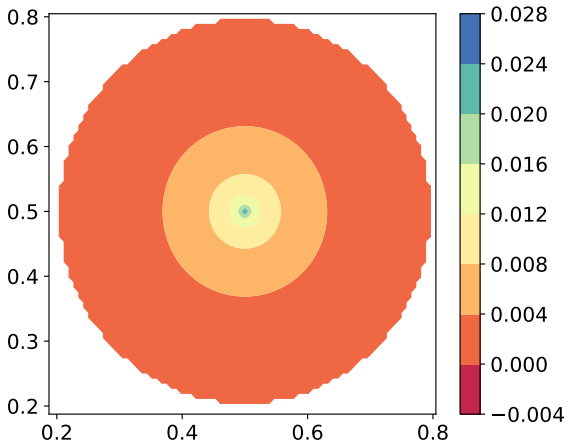

Figure 10.3: Error in the computed solution on a regular grid for Equation (10.2). We note that the error is concentrated at the singularity at $(.5, .5)$, making the problem suitable for the use of an adaptive grid.

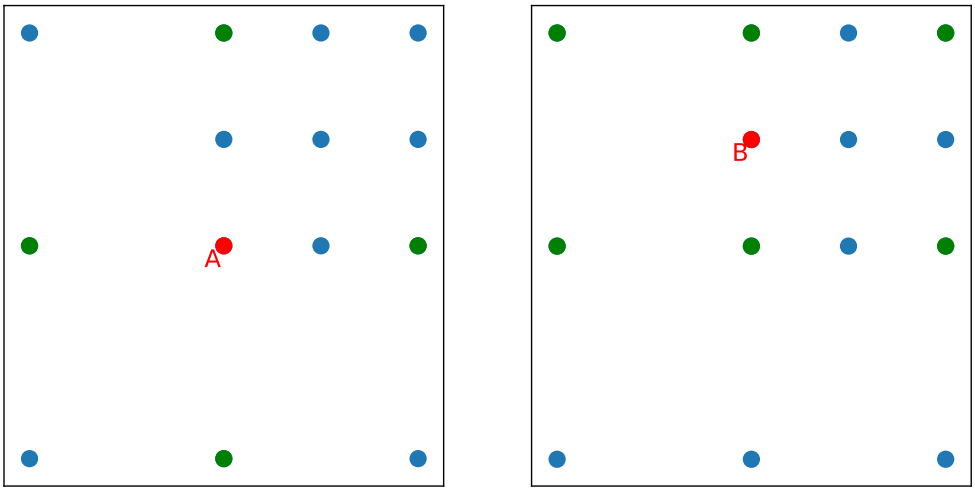

Figure 10.4: Finite difference stencils at the interface between grids.

Base Grid	Grid 1	Grid 2	Grid 3	Grid 4
16^2	$1.38E-01$	$3.82E-02$	$1.49E-02$	$7.22E-03$
32^2	$6.40E-02$	$1.89E-02$	$7.85E-03$	$3.73E-03$
64^2	$3.26E-02$	$1.02E-02$	$4.48E-03$	$2.24E-03$
128^2	$1.61E-02$	$5.10E-03$	$2.24E-03$	$1.09E-03$
256^2	$8.08E-03$	$2.58E-03$	$1.14E-03$	$5.60E-04$

Table 10.2: Relative L_2 error for Equation (10.2).

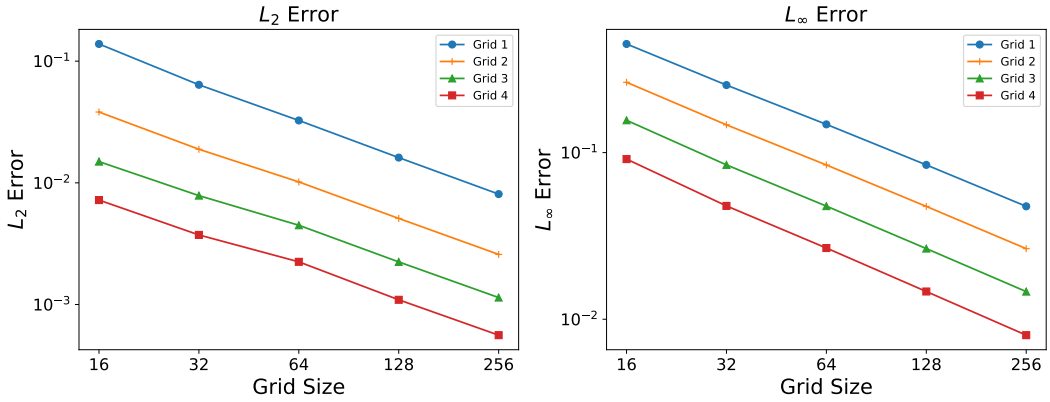

Figure 10.5: Relative L_2 and L_∞ errors for Equation (10.2) on grids with increasing levels of refinement. The x-axis represents the number of points in the base uniform grid. Each subsequent grid has an additional layer of refinement.

Base Grid	Grid 1	Grid 2	Grid 3	Grid 4
16^2	$4.49E-01$	$2.64E-01$	$1.56E-01$	$9.14E-02$
32^2	$2.54E-01$	$1.47E-01$	$8.44E-02$	$4.79E-02$
64^2	$1.48E-01$	$8.43E-02$	$4.77E-02$	$2.68E-02$
128^2	$8.44E-02$	$4.75E-02$	$2.65E-02$	$1.47E-02$
256^2	$4.76E-02$	$2.65E-02$	$1.47E-02$	$8.03E-03$

Table 10.3: Relative L_∞ error for Equation (10.2).

Base Grid	PCG Iterations				CPU Times			
	Grid 1	Grid 2	Grid 3	Grid 4	Grid 1	Grid 2	Grid 3	Grid 4
16^2	25	25	25	24	0.07	0.09	0.1	0.11
32^2	25	28	27	28	0.4	0.34	0.38	0.46
64^2	29	28	29	29	1.95	2.23	2.46	2.86
128^2	30	25	21	26	18.8	11.97	12.68	17.1
256^2	32	31	35	30	93.63	119.77	146.51	147.22

Table 10.4: PCG iterations and CPU time to convergence for Equation (10.2).

Chapter 11

Conclusion

In this thesis, we have described the *smooth selection embedding method* (*SEEM*) and how it fits within the general framework of fictitious domain and meshfree methods. We have detailed how it can be implemented using Fourier series, Chebyshev series, and finite difference discretizations. We also showed that the *SEEM* system generated with a Fourier discretization is well-posed and that the numerical solution converges to the actual solution. We have offered numerical experiments which demonstrate the effectiveness of the method, both for smooth problems and problems of lower global regularity. Divergence free and time dependent problems have also been studied. We now give our perspective on the overall advantages and disadvantages of *SEEM*.

Advantages of *SEEM*

- *SEEM* allows for the use of straightforward and efficient discretizations for PDEs on complex domains. For example, one can implement *SEEM* using only the five point stencil and polynomial interpolation, with no need to create a mesh for the domain, form a stiffness matrix, or create complicated finite difference stencils near the bound-

ary.

- The method's performance is quite competitive. In particular, it achieves high orders of convergence and can be efficiently implemented on dense grids.

- The *SEEM* formulation is natural and straightforward from both a PDE and optimization perspective. This is particularly true when compared with other fictitious domain formulations. In addition, *SEEM* falls under the well studied and robust theoretical framework of kernel methods.

- *SEEM* is flexible in that it can be used with many choices of discretization and of kernel. For example, spectral discretizations can be used on arbitrary shaped domains, providing advantages in divergence free problems.

Disadvantages of SEEM

The advantages of *SEEM* come with some significant computational costs.

- The method trades an elliptic problem with a saddle point problem. From a linear algebra perpsective, a saddle point problem is inherently more difficult to solve than an elliptic problem. In particular, when solving saddle point problems iteratively, an elliptic problem will need to be solved at each step of the iteration. Thus, we expect our method to be somewhat less efficient than other methods, such as the finite element method, which do not require a saddle point formulation.

- *SEEM* exchanges a second order BVP for a problem of order $-2p$. Thus, the method introduces significant numerical stability issues and implementing the method requires significant preconditioning. The preconditioning implemented in this thesis is limited to cases where a matrix of size $N_\Gamma \times N_\Gamma$ can be LU-factored. For grids denser than this, alternative preconditioning needs to be developed. Such preconditioning will likely be

somewhat complex and less effective than the method we have implemented so far. Moreover, even with preconditioning, hard limits on grid depth exist due to the limits of machine precision on ill-conditioned problems.

- Inherent in a fictitious domain formulation is that extra computation will be needed on the encompassing domain $\mathbb{B}$. In addition, fictitious domain formulations assume that the solution can be smoothly extended to the domain $\mathbb{B}$, which is not the case for all domains Ω.

Despite these difficulties, our work in this thesis demonstrates that *SEEM* is a useful and effective method for solving boundary value problems.

Bibliography

[1] I. Babuška. The finite element method with Lagrangian multipliers. *Numerische Mathematik*, 20(3):179–192, 1973.

[2] A. Behzadan and M. Holst. Multiplication in Sobolev spaces, revisited. *arXiv preprint arXiv:1512.07379*, 2015.

[3] M. Benzi, G. H. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 2005.

[4] J. Berrut and L. Trefethen. Barycentric Lagrange Interpolation. *SIAM Review*, 46(3):501–517, 2004.

[5] J. Boland and R. A. Nicolaides. Stability of finite elements under divergence constraints. *SIAM Journal on Numerical Analysis*, 20(4):722–731, 1983.

[6] J. P. Boyd. Fourier embedded domain methods: extending a function defined on an irregular region to a rectangle so that the extension is spatially periodic and C^∞. *Applied Mathematics and Computation*, 161(2):591–597, 2005.

[7] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[8] H. C. Elman and G. H. Golub. Inexact and preconditioned Uzawa algorithms for saddle point problems. *SIAM Journal on Numerical Analysis*, 31(6):1645–1661, 1994.

[9] G. E. Fasshauer. *Meshfree Approximation Methods with MATLAB*. World Scientific, 2007.

[10] B. Fornberg, E. Lehto, and C. Powell. Stable calculation of Gaussian-based RBF-FD stencils. *Computers & Mathematics with Applications*, 65(4):627–637, 2013.

[11] B. Fornberg and C. Piret. A stable algorithm for flat radial basis functions on a sphere. *SIAM Journal on Scientific Computing*, 30(1):60–80, 2007.

[12] C. Franke and R. Schaback. Convergence order estimates of meshless collocation methods using radial basis functions. *Advances in Computational Mathematics*, 8(4):381–399, 1998.

[13] 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, 1995.

[14] R. Glowinski, T. W. Pan, and J. Périaux. A fictitious domain method for Dirichlet problem and applications. *Computer Methods in Applied Mechanics and Engineering*, 111(3-4):283–303, 1994.

[15] F. H. Harlow and J. E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The physics of fluids*, 8(12):2182–2189, 1965.

[16] S. Kunis. *Nonequispaced FFT: Generalisation and Inversion*. Berichte aus der Mathematik. Shaker Verlag, 2007.

[17] X. Li, J. Lowengrub, A. Rätz, and A. Voigt. Solving PDEs in complex geometries: a diffuse domain approach. *Communications in Mathematical Sciences*, 7(1):81–107, 2009.

[18] C. C. Paige and M. A. Saunders. LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares. *ACM Transactions on Mathematical Software (TOMS)*, 8(1):43–71, 1982.

[19] R. Palais, B. Palais, and H. Karcher. PointClouds: Distributing Points Uniformly on a Surface. *arXiv preprint arXiv:1611.04690*, 2016.

[20] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2003.

[21] V. Sauliev. Solution of certain boundary value problems on high-speed computers by the fictitious domain method [Russian]. *Sibirsk. Mat. Zh.*, 4:912–925, 1963.

[22] D. Shirokoff and J.-C. Nave. A Sharp-Interface Active Penalty Method for the Incompressible Navier–Stokes Equations. *Journal of Scientific Computing*, 62(1):53–77, 2015.

[23] D. B. Stein, R. D. Guy, and B. Thomases. Immersed boundary smooth extension: A high-order method for solving PDE on arbitrary smooth domains using Fourier spectral methods. *Journal of Computational Physics*, 304:252–274, 2016.

[24] D. B. Stein, R. D. Guy, and B. Thomases. Immersed Boundary Smooth Extension (IBSE): A high-order method for solving incompressible flows in arbitrary smooth domains. *Journal of Computational Physics*, 335:155–178, 2017.

[25] L. Trefethen. *Spectral Methods in MATLAB*. SIAM, 2000.

[26] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4(1):389–396, 1995.

[27] H. Wendland. *Scattered Data Dpproximation*. Cambridge University Press, 2004.

[28] H. Wendland. Divergence-free kernel methods for approximating the Stokes problem. *SIAM Journal on Numerical Analysis*, 47(4):3158–3179, 2009.